Evolving Complexity and Similarity in an Artificial Life Framework based on Formal Language Theory

Gema M. Martín Reyes

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática

Universidad de Málaga, Spain

A thesis submitted for the degree of

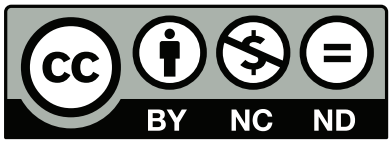
Doctor

2010 July

UNIVERSIDAD
DE MÁLAGA

AUTOR: Gema María Martín Reyes

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



UNIVERSIDAD
DE MÁLAGA

Ph.D. Francisco J. Vico Vela, Associate Professor for Computer Science, Department of Lenguajes y Ciencias de la Computación, Universidad de Málaga (Spain), and

Ph.D. Jürgen Dassow, Professor for Theoretical Computer Science, Faculty of Computer Science, Otto-von-Guericke-Universität of Magdeburg (Germany).

hereby CERTIFY:

That **Gema María Martín Reyes**, Mathematics Graduate, has carried out her doctoral thesis:

Evolving Complexity and Similarity in an Artificial Life Framework based on Formal Language Theory,

in the Department of Lenguajes y Ciencias de la Computación in Universidad de Málaga, under our supervision. We have revised this thesis and allow it can be presented to the committee in Universidad de Málaga.

Málaga, May 2010.

Ph.D. Francisco J. Vico
Associate Professor
Universidad de Málaga
Spain

Ph.D. Jürgen Dassow
Professor
Otto-von-Guericke-Universität
Germany

UNIVERSIDAD
DE MÁLAGA

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

The thesis work was conducted under the supervision of Ph.D. Francisco J. Vico Vela (Universidad de Málaga, Spain), and Ph.D. Jürgen Dassow (Otto-von-Guericke-Universität Magdeburg, Germany).

Málaga, May 2010.

Gema M. Martín Reyes

UNIVERSIDAD
DE MÁLAGA

To my parents and Jorge.

UNIVERSIDAD
DE MÁLAGA

Acknowledgements

The acknowledgment section is a section that is compulsory in any thesis, this is due purely for politeness. However, in my case, writing this section is without doubt almost as important as writing the rest of the contents. The main reason for this is that, from my point of view, a thesis is not only a collection of research results that you have obtained (by working really hard) along the years, but also a source that has allowed you to acquire a really big personal enrichment; if you have kept all your senses about you. Thus, I feel the necessity to thank some of the people who have contributed into making this thesis possible because there is a part of these pages, which belongs to them.

My first and foremost debt of gratitude is to my supervisors, Francisco and Jürgen, for dedicating their valuable time and expertise to me. I would like to thank Francisco for giving me the opportunity to start this journey almost four years ago, and leading me into a very interesting research area that has let me learn computer science and biology and mainly, that has allowed me to apply my mathematical knowledge in both of these fields. Of course I cannot let this opportunity go by without thanking you for providing the funds for my thesis. Thanks for your extraordinary support and inspiration, and for all those interesting discussions that have helped me look at things, both scientifically and personally, in a new way. At the same time, I would like to thank Jürgen, one of the most excellent theoretical computer scientist, for sharing his extensive knowledge and allowing me to consolidate my mathematical bases under his supervision. I would like to show my appreciation for giving me the chance to spend those four months in Otto-von-Guericke-University in Magdeburg (Germany). It was one of the most rewarding experiences of my life, in which I matured not only as

a researcher, but as a person, and I would like to express how much the maintained and heightened collaboration that you have given us means to me. Without both of you, this thesis would not have been possible at all! From the bottom of my heart, thank you!

I must thank Michael as well for giving me the opportunity to spend one month in Harvey Mudd College in Claremont (Los Angeles) at the beginning of this thesis. It was a great experience that allowed me to understand that science has no frontiers. Thanks for letting me stay with your lovely family.

I would like to thank all my Lab mates (even those who left and those who have just arrived) whom I have spent part of my life with. To my colleagues Dani and Jose David for creating a good work environment and helping me with some computational problems. To Ellen for being disposed to help me with English at any time and giving me her friendship. To Vicente for sharing his creative ideas and making me laugh. To Curro for facilitating all the bureaucratic staff and inducing interesting conversations. To Julio for providing technical support and having that ironic sense of humor. There are also many others who I feel I would really like to thank for all that they have given me in the long hours we have worked together in the Lab: Salva, Miguel, Ale, Maria José, Carlos, David, Jose, Samuel, Antonio, Juan...

I do not have enough room to adequately thank my parents, Maria and Pedro, for everything they have done to enable me to be in the position of submitting this thesis. Thanks for being proud of me even without knowing very well in what I have been working on during these years. In my case, the main reason why I am proud of them is clear; without their love and support, I would not be the person I am. At the same time, I have to thank Jorge for his constant love, affection, friendship and support and for respecting that this is the work that I love and tolerating the long periods of being neglected because of my thesis.

Finally, I would like to thank those who have kept me away from the thesis and helped to make my life so enjoyable. Thus, many thanks to Cristina Pa, Cristina Pe, Isa, Margui, Corpas, Jose, and Salva. They are the sisters and the brothers that I have never had. Moreover, I want to thank them

and many others: Jesus, Diego, Yasmin, Berta, Jony, Belen, Anne, Rocio, Mª Angeles, Jorge, Borja..., for having made me think up different ways to explain my thesis to them that accidentally had its own contribution to my own better understanding.

UNIVERSIDAD
DE MÁLAGA

Abstract

Although the process of natural selection described by Charles Darwin in *The Origin of Species* does not guarantee that organisms will increase in complexity as they evolve, since for certain lineages have been in this way, a big part of the scientific community defend the fact that the tendency of the complexity has been increasing during the evolution. For that reason, many researchers have modeled evolving artificial ecosystems in order to make a case for or against a trend in the evolution of complexity and study the factors that cause it (in the case there exists such a trend).

The main problem of these proposed artificial ecosystems is that their results can be questionable since they do not use a rigorous complexity measure. This problem comes from the fact that complexity is a complex concept in itself and presents so many variations that it is only valid in specific situations.

In this thesis, a formal framework where the evolution of biological complexity can be studied in an objective way is defined. That objectivity is due to state complexity for regular languages is used and it is a well-known and rigorous complexity measure. Such a framework is composed of a population of cyclic unary regular languages (individuals) that try to adapt to a given environment (that also consists of cyclic unary regular languages) by means of evolutionary computation. The genotypes of the individuals are defined as the cyclic unary deterministic finite automata that recognize them and it is shown how they can be represented as binary words. A similarity measure for cyclic unary regular languages is proposed and it is used as fitness function (i.e., the more similar an individual is to the environment, the more adapted to the environment the individual is), to define the species concept and to analyze the disruptive effect that the usual genetic

operations produce when they are applied over the genotypes represented as binary words.

Many properties and characterizations of such a framework are presented. A relation between the cyclic unary deterministic finite automata and the primitive words is presented, and it provides a characterization of the minimality of such automata. By using it, two more appropriate representations for the genotypes are introduced. The first one provides a set of operations that preserve primitivity of words and the second one provides a relation between the primitive words and number theory.

In this framework, the evolution shows a tendency of the complexity of the individuals to increase. Moreover, results show a strong correlation between the complexity of the population of automata and the complexity of the environment, and that the predatory behavior promotes the emergence of more complex individuals. By using the framework proposed in this thesis, a wide variety of ecological experiments could be done in a rigorous way.

Contents

Chapter 1

Introduction

1.1 The Whole Picture

The aim of this thesis is mainly concerned with defining a formal framework where the evolution of biological complexity can be studied. Along this research direction, many properties and characterizations of such a framework have been found and are presented in this thesis.

Although, in the course of history, many open-ended evolutionary systems have been defined to establish connections between artificial and natural life, to our knowledge, very few of them have contributed to a better understanding of the conditions in which complexity could have evolved on Earth. This is mainly due to the fact that complexity is a complex concept in itself, what makes non-trivial to define a formalism in which the complexity can be analyzed in an objective way. For that reason, the first research steps were given in finding the mathematical basis of a framework in which objective studies of complexity could be performed. Thus, several formalisms, like grammatical systems or cellular automata, were analyzed.

State complexity, as defined for regular languages, is a well-known and an objective complexity measure. The Minimization Theorem allows to define it as the number of states of the minimal deterministic finite automaton that represents a given regular language. Therefore, considering a regular language as an individual in an evolving population, the set of regular languages might be an appropriate substrate to study complexity dynamics. Thus, the regular languages will be the phenotypes and the automata that represent them will be the genotypes. We need to have a description of

the genotypes which is simple and evolvable, that is to say, that allows the application of genetics operations. Thus, words would be a good candidate. For that reason, we choose cyclic unary regular languages[1] (CURL, for short) as the automata that represent them, cyclic unary deterministic finite automata (CUDFAs, for short), can be described by binary words. In (70) it is proved that the set of minimal CUDFAs can be represented as binary primitive words[2]. Thus, if complexity issues are involved, then we have to restrict to primitive words (in order to have a unique minimal description for the CURL).

Another common problem in evolutionary systems is the lack of studies concerning the disruption that genetic operations produce when they are applied over the genotype. The most common operations that are used in evolutionary systems are the edit operations of substitution, deletion, and insertion of a symbol over a word. In (25; 71), the edit operations are extended by introducing two new operations (partial copy and partial elimination). A disruption measure for an operation over a word is defined and proved that whereas the traditional edit operations were disruptive, partial copy and partial elimination were non-disruptive. Moreover, it is showed that the application of just edit operations does not generate (with low disruption) all the words over a binary alphabet, but this can be done by combining partial copy and partial elimination with the substitution operation.

In spite of such an extension reduces the disruption that the edit operations produce in the genotype (the binary word), we do not use them as they do not preserve the minimality of automata, that is to say, they do not preserve the primitivity of words. If primitivity is not preserved, then individuals with the same complexity can be represented by automata (genotypes) with very different number of states and this seems not to be very logical from a biological point of view.

Thus, in order to define a representation of the genotype of the individuals in which we have not the previous problems, in (26) some operations that preserve the primitivity of words are defined. These operations, mostly inspired by biological gene duplication, essentially add a modified copy of a given primitive word at any given place. A large subset of binary primitive words can be obtained by using sequences of these operations as genotypes. Genetic operations will be applied over these sequences. Moreover, other

[1] CURLs are regular languages over unary alphabets that are represented by cyclic automata.
[2] A word is primitive if it is not a proper power of a shorter word.

methods to generate the set of primitive words (68) in a form that might be used as genotypes have been explored.

In the first studies dealing with the dynamics of the complexity, a population of CURLs (individuals) adapt to a given environment (represented by another CURL). In order to calculate how well an individual adapts to the environment, a measure of the similarity between both languages is needed. Therefore, in (24) a similarity measure for CURLs by modifying the Jaccard similarity coefficient and the Sørensen coefficient is defined to measure the overlap level between such languages. This measure computes the proportion of strings that are shared by two or more CURLs. By using this similarity measure, a dissimilarity measure for CURLs that is a semimetric distance is also defined.

In (70), two different approaches to study the behavior of the complexity during the evolutionary process in which a set of CURLs adapts to a given environment (another CURL) are proposed. The first one uses a greedy algorithm and edit operations over binary words. The second one uses a genetic algorithm and the edit operations over sequences of operations that preserve the primitivity (defined in (26)) of the CUDFAs of the population during the evolution. In both cases, it is concluded that the complexity increases. Moreover, in the second approach, the correlation between the adaptation level to the environment and the complexity of the individual is stronger than in the first approach.

Finally, an artificial ecosystem of regular languages has been defined (69). It merges most of the knowledge gained through out the last almost 4 years, which is summarized above. In this artificial ecosystem, a population of CURLs with minimal complexity coexists in a world where a number of sources (defined also as CURLs, usually with high complexity) are located over a finite surface. These sources define niches or subenvironments. The individuals compete to gain energy from inert resources (generated as strings from the subenvironments) or from non-inert resources (depredation between individuals). The fitness function encourages the acceptance of strings generated at the subenvironments. By using this model, we have found: (1) that there is a dependency between the complexity of the population and the complexity of the environment, in the sense that the more complex the environment where the individuals evolve, the more complex they do develop, and (2) that predatory behavior promotes a higher complexification of the individuals.

1.2 Organization of the Thesis

In this section, an overview of the thesis is given. The content of each chapter and the connection between them is briefly described. Moreover the more relevant addressed problems and contributions in each chapter are highlighted.

In Chapter 2, the notation used throughout the thesis is presented. Some concepts of formal language theory are recalled, paying particular attention to deterministic finite automata. The representations for UDFAs and URLs that are used in this thesis are introduced. In the particular case of CUDFAs, we show that they can be described by words over $\{0,1\}$, where the zeros represent the non-accepting states of the automaton, and the ones represent the accepting states of the automaton. Moreover, we describe a CURL represented by a CUDFA $w \in \{0,1\}^+$ by the union of natural successions with period $|w|$. Thus, a CUDFA is considered as a genotype (over which the genetic operations will be applied), and its accepted CURL as the corresponding phenotype. Finally, a theorem that characterizes the case in which a UDFA is minimal is presented. In the particular case of CUDFAs, we conclude that a CUDFA $w \in \{0,1\}^+$ is minimal if and only if w is a primitive word (these characterizations are part of (70) in which I am first author[1]).

The results presented in Chapter 3 are part of (24) in which I am first author[2] (it is published in the international journal *Fundamenta Informaticae*). Since, we want to study the dynamics of the complexity during the evolution of individuals that trying to adapt to a given environment, a similarity measure that calculates how well an individual is adapted is needed. Since, in this thesis, individuals are CURLs, a similarity measure between CURLs is needed. It is shown that there exist some measures of the overlap between either URLS or CURLs but they do not satisfy the principle: sets are more similar if they have more elements in common. For that reason, we can not use them and a similarity measure between CURLs that satisfied that principle is presented in this chapter. This measure computes the portion of words that are shared by two CURLs. It is also proved that it is an upper bound of the well-known Jaccard coefficient and the Sørensen coefficient. By using such similarity measure, a dissimilarity measure for CURLs is defined that is a semimetric distance. Moreover, we show that both of

[1] the rest of coauthors are my supervisors and gave hints, suggestions, and comments

[2] authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments

them can be used also for URLs, although in this case the dissimilarity measure is not a semimetric distance, but a symmetric distance.

The results presented in Chapter 4 are part of (25; 71) in which I am first author((25) is published[1] in *Proceedings of NCMA09*, it was selected by the editors for an extended version (71) that is to appear [2] in the international journal *Fundamenta Informaticae*). Since in order to study the evolution of the complexity we need to use an evolutionary system, genetic operations have to be applied over the genotypes. The most common operations that are used in evolutionary systems are the edit operations of substitution, deletion and insertion of a symbol. Usually, in the literature, edit operations are used without worries about the disruption that they produce in the genotypes. In this chapter, studies on the disruption that such operations produce when they are applied over the genotypes proposed in this thesis (represented as binary words) are done. Thus, a disruption measure for an operation over a word is defined by using the similarity measure defined in Chapter 3. Intuitively, the disruption of an operation O with respect to a word w is a pair (a, b) with $a, b \in \mathbb{R}$, where a is the portion of words that are accepted by w and are not accepted by $O(w)$, and b is the portion of words that are accepted by $O(w)$ and are not accepted by w. It is proved that not all words can be obtained by iterated applications of edit operations where each application is accompanied by low disruption. To solve this problem, the edit operations are extended by introducing two new operations (partial copy and partial elimination) inspired by biological gene duplication. These new operations have no disruption and by iteratively applying them combine with the edit operations all words can be obtained by low disruption.

The results presented in Chapter 5 are part of (26; 68) in which I am first author ((26) is published[3] in the international journal *Theoretical Computer Science*, and (68) is in preparation[4]). As we mentioned before, in Chapter 4, two new operations have been proposed in order to reduce the disruption produced by the most common operations used in evolutionary systems: the edit operations, when they are applied over the

[1] authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments

[2] the rest of coauthors gave hints, suggestions, and comments

[3] authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments

[4] the other coauthor is my supervisor and gave hints, suggestions, and comments

representation of the genotypes as binary words. Despite reducing such a disruption by using the new operations, the minimality of automata is not preserved by them, thus individuals with the same complexity can be represented by automata with very different number of states and this seems not to be very logical from a biological point of view. For that reason, a representation of the genotypes over which genetic operations preserve the minimality of automata, that is to say, preserve the primitivity of words (since our genotypes can be represented as binary words), are required. In this Chapter, two different ways of generating primitive words are presented. For the first one, a set of operations inspired by biological gene duplication that preserve primitivity of words is proposed. A large subset of binary primitive words can be obtained by using sequences of these operations as genotypes. For the second one, a characterization of the non-primitive words that provides a relation between primitive words and number theory is proposed. This gives a non-grammatical method to generate the set of all the primitive words. While genetic operations can be directly applied over the sequences of the operations preserving primitivity, the application of the genetic operations in the second approach is not as trivial. For that reason, the sequences of the operations preserving primitivity will be the representation of the genotypes used to study the complexity during the evolution (now, the minimality of the automata is preserved).

The results presented in Chapter 6 are part of (69; 70) in which I am first author[1] (both of them are in preparation). At this point, we have defined: (1) a characterization of the minimality of CUDFAs (based on primitive words) that allows us define a set of operations that preserve the minimality of CUDFAs that can be used as representation of the genotypes of the CURLs over which the genetic operations will be applied, (2) a similarity measure for CURLs that allows us to calculate how well an individual is adapted to a given environment, and (3) an objective complexity measure for CURLs: the state complexity, that allows us to study the evolution of the complexity of the individuals in a rigorous way. Thus, it is clear that a framework based on CURLs is a perfect framework to accomplish the main aim of this thesis: defining a formal framework where the evolution of biological complexity can be studied. In this chapter, studies on the evolution of the complexity during the evolution of CURLs that try to adapt to a given environment are done. First of all, some preliminary studies on the behavior of the complexity by using a simple framework are shown. Two different

[1] the rest of coauthors are my supervisors and gave hints, suggestions, and comments

approaches are proposed. The first one uses a greedy algorithm and the edit operations over binary words. The second one uses a genetic algorithm and the edit operations over sequences of the operations that preserve the minimality that were defined in the previous chapter. We show that the behavior of the complexity is increasing in both of them, but in the second approach, the correlation between the similarity to the environment and the complexity of the individual is stronger. This makes the determination of representing the genotypes as sequences of the operations preserving primitivity stronger. In the last section, different components (such that the complexity of the environment, the predatory behavior of the individuals,...) that affect such an increasing behavior of the complexity are studied in an evolving artificial ecology consisting of CURLs.

A summary of the major contributions of the thesis is presented in Chapter 7.

1.3 Spanish Summary

Charles Darwin describió el proceso de la evolución de las especies por la selección natural hace más de 150 años. A pesar de que este proceso no garantiza que la complejidad de los organismos crezca durante la evolución, que así ha sido es un hecho aceptado por muchos investigadores (12; 47; 75) debido a que se pueden encontrar evidencias de ello en la evolución biológica. Por este motivo, el problema de medir la tendencia de la complejidad durante la evolución ha sido y es de gran interés para la comunidad científica desde hace años. Muchas investigaciones en vida artificial se han centrado en el modelado de ecosistemas artificiales (entre ellos modelos basados en individuos) en los que puedan realizarse estudios sobre la tendencia de la complejidad.

El gran problema de la mayoría de estas investigaciones es que sus resultados son cuestionables debido a que no cuentan con una medida que cuantifique de forma rigurosa la complejidad de los organismos de los ecosistemas que proponen (38; 52; 72; 82; 111; 112). El principal motivo de esto es que el término complejidad es un concepto complejo en sí mismo. Es decir, medir la complejidad es una estimación abstracta que depende del contexto en el cual se esté trabajando.

El principal objetivo de esta tesis es definir un marco de trabajo formal en el que podamos estudiar de manera rigurosa la tendencia de la complejidad durante la evolución. A lo largo de esta línea de investigación, se han encontrado otras muchas propiedades

y caracterizaciones derivadas de la definición de tal marco de trabajo y serán también presentadas en esta memoria.

Como se explicaba previamente, definir un formalismo en el cual la complejidad pueda analizarse de manera objetiva no es trivial. Por esta razón, los primeros pasos de investigación de esta tesis estuvieron dirigidos a la búsqueda de las bases matemáticas necesarias para este propósito. De esta manera, se analizaron sin éxito varios formalismos tales como sistemas gramaticales o autómatas celulares.

La complejidad de estados (115) definida para lenguajes regulares, es una medida de complejidad objetiva y aceptada. Asímismo, la complejidad de un lenguaje regular es definida como el número de estados del autómata finito determinista mínimo (DFA mínimo, para abreviar) que lo reconoce. Claramente, esta medida de complejidad es objetiva y computable debido a que el teorema de Myhill-Nerode (83) prueba que dado un lenguaje regular, existe un único DFA mínimo que lo reconoce (mínimo con respecto al número de estados y único salvo isomorfismo) y además, mediante su uso, pueden definirse algoritmos de minimización. Por tanto, los lenguajes regulares constituyen un buen substrato en el que estudiar la dinámica de la complejidad, considerándolos como los individuos de una población en evolución. De esta manera, los lenguajes regulares serán los fenotipos de nuestro modelo, mientras que los autómatas que los reconocen serán los genotipos.

Debido a que para el estudio de la evolución de la complejidad es necesario introducir una dinámica en la población de lenguajes y que la computación evolutiva puede ser usada con este propósito, necesitamos encontrar una representación de los genotipos usados en nuestro modelo sobre la cual se puedan aplicar los operadores genéticos. Dado que un autómata finito determinista unario cíclico (CUDFA, para abreviar) puede representarse como una palabra w sobre el alfabeto $\{0, 1\}$, donde los ceros representan los estados de no aceptación del autómata, y los unos representan los estados de aceptación del autómata, y esta es una representación sobre la que pueden aplicarse los operadores genéticos, restringimos nuestro marco de trabajo al conjunto de los CUDFAs. El lenguaje regular reconocido por un CUDFA es un lenguaje regular unario cíclico (CURL, para abreviar), esto es, un lenguaje regular sobre un alfabeto unario que es reconocido por un autómata cíclico. Debido al carácter unario e infinito de los CURLs, describiremos a un CURL como la unión finita de sucesiones infinitas de números naturales con el mismo periodo (el número de estados del CUDFA mínimo que

lo reconoce). Cada número natural representa a la cadena de esa longitud perteneciente al alfabeto unario. Esto es, un CURL no es más que un conjunto infinito de números naturales.

Además, en la Sección 2.2, damos una caracterización de los UDFAs mínimos y como corolario obtenemos una caracterización de los CUDFAs que proporciona una relación entre el campo de los autómatas cíclicos y el de las palabras primitivas (una palabra es primitiva si no es potencia de una palabra de longitud menor, es decir, dado un alfabeto V, $w \in V^+$ es una palabra primitiva si no existe otra palabra $v \in V^+$ tal que $w = v^n$ con $|v| < |w|$ y $n > 1$). Esta caracterización es la siguiente: un CUDFA $w \in V^+$ es mínimo si y sólo si w es una palabra primitiva. Esta relación encontrada, provoca que diversas propiedades y caracterizaciones del conjunto de las palabras primitivas sean analizadas en esta tesis. Estos resultados son interesantes para una gran parte de la comunidad que investiga los lenguajes formales (incluso independientemente del componente evolutivo introducido en esta tesis), debido a que el lenguaje de las palabras primitivas ha sido ampliamente estudiado en la literatura.

Los operadores genéticos más comúnmente usados en computación evolutiva son los operadores *edit* de substitución, eliminación e inserción de un símbolo, también conocidos como operadores de mutación puntual (por estar biológicamente inspirados). A pesar de estar ampliamente aceptada su utilización y de haber sido considerablemente estudiados (20; 23; 28; 57; 86; 87; 88; 113), en la mayoría de los trabajos que los usan no se analiza (y por tanto, no se intenta evitar) el efecto disruptivo que estos operadores pueden provocar al ser aplicados sobre los genotipos. Esto es, no se analiza cuánto difiere del fenotipo inicial el fenotipo resultante tras aplicar un operador sobre un genotipo. Desde un punto de vista biológico, los operadores genéticos que provocan cambios muy drásticos en los fenotipos no tienen sentido. Por otro lado, desde un punto de vista computacional tampoco es razonable, ya que los algoritmos de búsqueda no aleatoria se benefician de una baja disrupción en la aplicación de los operadores para refinar las soluciones.

Para nosotros estos dos puntos de vista (normalmente olvidados en la mayoría de los estudios que usan los operadores *edit*) son importantes y deben ser tenidos en cuenta. Por tanto nos preocupa la disrupción que los operadores *edit* puedan provocar al aplicarlos sobre los genotipos que son objeto de estudio en esta tesis. Para llevar a cabo este estudio, es necesario contar con medidas de disrupción apropiadas que

cuantifiquen en qué medida cambia el fenotipo resultante respecto al inicial tras la aplicación de un operador *edit*. Definir esta medida no es trivial, y éste es uno de los principales inconvenientes por los que estos estudios de disrupción no se realizan de forma habitual. En el caso que nos compete en esta tesis, y debido a que los individuos han sido definidos como CURLs, para definir una medida de disrupción apropiada necesitamos una medida que calcule cómo de similares son dos CURLs dados (y en consecuencia, cómo de diferentes son).

En la literatura, las medidas de disimilitud propuestas entre conjuntos regulares son escasas, no ocurriendo lo mismo para otros conjuntos de lenguajes (17; 18; 79; 81). Por otro lado, de entre las existentes para conjuntos regulares (tales como la distancia de Bodnarchuk, de Baire, de Hamming o relativa a la teoría de información) ninguna son de nuestro interés, debido a que no satisfacen el siguiente principio de similitud: dos conjuntos son más similares, si tienen más elementos en común. Asímismo, en el Capitulo 3, se propone una medida de similitud para CURLs que calcula la porción de palabras que comparten dos CURLs dados y por tanto cumple el principio anterior. El calculo de esta porción es fácilmente realizable gracias a la representación para CURLs propuesta en esta tesis (y comentada previamente), esto es, un CURL es representado por la unión finita de sucesiones de números naturales. Esta medida de similitud se obtiene básicamente a partir de una modificación de los conocidos coeficientes de Jaccard y de Sørensen (que no pueden ser directamente usados para conjuntos infinitos) que ha sido propuesta para que puedan usarse en el caso de los CURLs y constituye una cota superior de los mismos. Es decir, la medida de similitud propuesta en esta tesis, funciona como un indicador de convergencia mejor que los coeficientes de Jaccard y de Sørensen. Además, la distancia (medida de disimilitud) que puede definirse usando la medida de similitud propuesta es una distancia semimétrica. Por otro lado, se muestra cómo estas medidas (tanto la de similitud como la distancia) pueden usarse en el caso más general de los URLs, aunque en este caso la distancia es una distancia simétrica y no semimétrica ya que no verifica la identidad de los indiscernibles. A parte de todas las aplicaciones que esta medida de similitud tiene en esta tesis, es posible señalar la aplicabilidad de esta medida en otros campos que no son objeto de estudio en esta tesis como son la inferencia gramatical y la teoría de la recuperación de información.

Por tanto, tras definir dicha medida de similitud para CURLs, estamos en condiciones de definir la medida de disrupción de un operador al aplicarlo sobre la repre-

sentación de los genotipos (CUDFAs) como palabras binarias y de esta manera, realizar estudios sobre la disrupción que los operadores *edit* producen al aplicarlos en los genotipos que son objeto de estudio en esta tesis, para intentar encontrar soluciones en el caso en el que la disrupción sea alta. Asímismo, en el Capitulo 4, se define la disrupción de un operador O sobre una palabra w como un par (a, b) con $a, b \in \mathbb{R}$, donde a es la porción de palabras que son aceptadas por w y no por $O(w)$ y b es la porción de palabras que son aceptadas por $O(w)$ y no por w, donde $O(w)$ es la palabra resultante tras la aplicación del operador O sobre w. Por los motivos explicados previamente, bajo nuestro punto de vista, lo natural es que, durante la evolución, la aplicación de cada uno de los operadores genéticos lleve consigo una disrupción baja, es decir, que el fenotipo resultante no difiera demasiado del original.

Los resultados presentados en esta tesis muestran que aplicando los operadores *edit* iterativamente no pueden obtenerse todas las palabras sobre el alfabeto $\{0, 1\}$ si añadimos el requerimiento adicional de que la disrupción sea baja en cada paso. Esto es, en general, los operadores *edit* producen una disrupción alta al aplicarlos sobre los genotipos representados como palabras binarias. Para intentar solucionar este problema, se han extendido los operadores *edit* mediante la inserción de dos nuevos operadores inspirados en la duplicación biológica de genes: copia y eliminación parcial. Los resultados muestran que mediante la aplicación iterativa de estos operadores conjuntamente con los operadores *edit* sí que pueden obtenerse todas las palabras sobre el alfabeto $\{0, 1\}$ con el requerimiento adicional de que la disrupción sea baja en cada paso. Este resultado puede ser interpretado como que la duplicación biológica de genes reduce la disrupción causada por las mutaciones durante la evolución. Esto se debe a que la duplicación de genes es un tipo de mutación silenciosa, en el sentido de que es una mutación neutral que no produce disrupción, ya que no aporta nuevas funciones. Sin embargo tal duplicación proporciona el substrato necesario para la producción de nuevas proteínas y funciones.

A pesar de que con la introducción de estos dos nuevos operadores parece haber disminuido la disrupción causada por los operadores *edit* cuando se aplican junto con ellos sobre los genotipos representados como palabras binarias, tras la aplicación de un operador O sobre una palabra primitiva w (esto es, sobre un CUDFA mínimo), no tenemos asegurado que $O(w)$ siga siendo una palabra primitiva. De esta manera, podría darse el caso de que un individuo (un CURL) estuviese representado por dos

genotipos (dos CUDFA) con un número de estados muy diferente entre sí. Por ejemplo, los autómatas 1 y $1^{1.000.000}$ representan al CURL que acepta todas las cadenas sobre un alfabeto unario y claramente tienen un número de estados muy diferente (uno frente a un millón). Desde un punto de vista biológico, esto no es lógico y podría ser criticado. Por esta razón, necesitamos una representación de los genotipos sobre la que no sólo se puedan aplicar los operadores genéticos, si no que además, al aplicarlos se preserve el carácter mínimo del autómata.

Asímismo, en el intento de buscar mejores (en el sentido explicado en el párrafo anterior) representaciones de los genotipos, en esta tesis se introducen dos maneras diferentes de generar palabras primitivas. Estos resultados tienen relevancia por si solos, es decir, no necesitan de la componente evolutiva introducida en esta tesis, para ser de interés para una gran parte de la comunidad científica que estudia la teoría de los lenguajes formales.

El primer método generativo de palabras primitivas está basado en la definición de ciertos operadores para los que el lenguaje de las palabras primitivas (Q, para abreviar) es cerrado y es presentado en la Seccion 5.2. Antes de explicar un poco más en profundidad en que consisten estos operadores, merece la pena mencionar que debido al gran interés que causan las palabras primitivas, en la literatura, existen algunos resultados relativos al cierre de este lenguaje bajo ciertos operadores (50; 58; 76; 77; 89; 90; 99). Los operadores propuestos en esta tesis están inspirados por la duplicación biológica de genes y preservan la primitividad de las palabras. En resumen, estos operadores están basados en el siguiente mecanismo: para una palabra primitiva w dada, se construye la palabra ww' donde w' es una copia de w modificada o una copia espejo de w modificada, donde si $w = x_1 \dots x_n$, su copia espejo es $w = x_n \dots x_1$.

Los resultados muestran que la palabra ww' sigue siendo primitiva. Computacionalmente, se ha demostrado que aplicando el conjunto propuesto de operadores de manera iterativa, y a partir de una sola letra, pueden generarse todas las palabras primitivas de longitud ≤ 11 , en el caso de alfabetos de dos letras (de hecho, son generadas casi todas hasta la longitud 20). Es decir, puede obtenerse un subconjunto grande de palabras primitivas usando secuencias de estos operadores. Debido a que el principal interés de esta tesis se centra en el estudio de la evolución de la complejidad, y que para esto no es necesario generar el conjunto total de palabras primitivas, este conjunto de operadores

nos sirve para tal fin, ya que los operadores genéticos pueden aplicarse sobre secuencias de estos operadores de manera que se preserva la primitividad.

Por otro lado, y a pesar de que, como dijimos antes, el lenguaje de las palabras primitivas despierta un gran interés, hoy en día aún se desconoce si tal lenguaje es o no independiente del contexto. Por tanto, el problema de la clasificación del lenguaje de las palabras primitivas en la jerarquía de Chomsky está sin resolver, habiéndose probado únicamente que no es un lenguaje regular (32). Por este motivo, y con el objetivo de encontrar tal clasificación, hay diversos estudios que relacionan subconjuntos del lenguaje de las palabras primitivas con otras familias de lenguajes (7; 16; 45; 53). También han sido propuestos diversos métodos generativos de las palabras primitivas, los cuales son principalmente métodos gramaticales (35; 61). En la Sección 5.3 de esta tesis, se propone un método generativo de Q que no es gramatical y por tanto, es totalmente diferente a los métodos presentados hasta ahora en la literatura. Para ello, el concepto de número primitivo es definido: un entero es un número primitivo en base q y de longitud m si su representación en base q es una palabra primitiva de longitud m sobre un alfabeto V con $|V| = q$.

Los resultados muestran que un entero es primitivo si y sólo si no es congruente con cero módulo ciertos números obtenidos tras el análisis. Por tanto, además de tener un método generativo de Q, hemos encontrado una relación entre las palabras primitivas y la teoría de números. Debido a que el módulo no es más que el resultado de una serie de operaciones aritméticas básicas y que estas pueden formularse de manera gramatical, podríamos (no lo hacemos porque el objetivo de esta tesis no es la clasificación de Q) convertir este método a su forma gramatical y tal gramática podría contribuir a esclarecer la clasificación del lenguaje Q en la jerarquía de Chomsky. Aunque, a diferencia del método anterior, con este método es posible generar el conjunto de todas las palabras primitivas, no lo usaremos como método de representación de los genomas debido a que la aplicación de los operadores genéticos sobre tal representación no es trivial. También se ha encontrado una propiedad curiosa que presentan las palabras primitivas: la distribución de las distancias entre dos números primitivos consecutivos es simétrica.

Por tanto, llegados a este punto, parece que tenemos todos los ingredientes necesarios para que, introduciendo una dinámica en una población formada por CURLs, podamos estudiar el comportamiento de la complejidad durante la evolución. Entre

estos ingredientes, parece que la representación de los genotipos que mejor se acerca a nuestras necesidades en la formada por secuencias de operadores que preservan la primitividad. Aunque todos los caminos parecen llevarnos hacia tal representación (y no tanto hacia la representación que consiste en palabras binarias), en la Sección 6.1 hacemos un último estudio en el que comparamos los resultados obtenidos usando ambas representaciones. A la vez, de estos estudios se concluye una tendencia hacia una complejidad cada vez mayor durante la evolución. En estos estudios preliminares sobre el comportamiento de la complejidad se usa un marco de trabajo muy simple en el cual una población de CURLs (individuos) intentan ser similares a un CURL dado (entorno).

En el primer estudio, los genotipos son representados como palabras binarias sobre las cuales se aplican los operadores *edit*. Además, se usa un algoritmo voraz de manera que de entre todas las posibles mutaciones en cada paso, la elegida es aquella que proporciona el individuo de mayor similitud con respecto al entorno (usando la medida de similitud propuesta en el Capitulo 3). En el segundo estudio, los genotipos son representados como secuencias de los operadores que preservan primitividad definidos en esta tesis, sobre las cuales se aplican los operadores *edit*. Además, se usa un algoritmo genético donde el cruzamiento entre individuos no es utilizado ya que es muy disruptivo y la medida de similitud (propuesta en el Capitulo 3) es usada como función de bondad.

En ambos estudios se observa una tendencia hacia una complejidad cada vez mayor, es decir, los individuos tienden a ser más complejos durante la evolución. En particular, si nos centramos en la relación existente durante la evolución entre la similitud de los individuos con el entorno y la complejidad de estos, vemos que en el caso en el que los operadores que preservan primitividad son usados como representación del genotipo, estas dos magnitudes correlan en mayor grado, es decir, la complejidad de los individuos incrementa a medida que estos están mejor adaptados al entorno (son más similares a él). Por tanto, esto nos da un argumento más para que la representación elegida para los genotipos sea la dada por la secuencia de operadores que preservan la primitividad.

Asímismo, tenemos evidencias de que una tendencia hacia una complejidad cada vez mayor ha tenido lugar durante la evolución. Aunque esto es de interés por si solo, ya que solapa con la idea bastante extendida entre la comunidad científica de que esto es así, conocer que factores influyen en que esto sea así es crucial. Para ello, es necesaria

la definición de un marco de trabajo en el que se vean involucrados un mayor número de elementos que en el caso anterior.

En la Sección 6.2, se ha propuesto un modelo basado en individuos donde una población de CURLs (individuos) con complejidad baja es colocada en un ecosistema artificial que está compuesto de un conjunto de CURLs (subentornos) aleatoriamente posicionados en un toro. Los individuos compiten por recursos inertes (números naturales obtenidos de los subentornos) y por recursos no inertes (otros CURLs del entorno, esto es, depredación entre individuos). Además, los recursos no inertes pueden reaccionar o no frente al ataque de un individuo (cuando un individuo intenta procesarlo). La dinámica en este sistema ha sido introducida mediante un algoritmo genético implementado en el lenguaje de programación *Matlab* (las simulaciones han sido realizadas en un cluster de computación de 32 CPUs (2 GHz)). Por las razones explicadas previamente, los genotipos están representados por secuencias de los operadores que preservan primitividad presentados en esta tesis, siendo además parte del genotipo la tasa de depredación y de reacción del individuo (en el caso de modelos con depredación). Estas tasas podrán ser o no mutadas por los operadores genéticos. Teniendo en cuenta la descripción dada anteriormente, donde un CURL es una sucesión de infinitos números naturales, decimos que un individuo (un lenguaje) procesa un recurso inerte (y por tanto, gana energía), si tal natural pertenece al lenguaje. Por otro lado, decimos que un individuo L procesa un recurso no inerte L' que no ha reaccionado, si dado un conjunto de números naturales S' pertenecientes a L', la energía obtenida por L tras intentar procesar a todos los elementos de S' es positiva. Si L' reacciona ante el ataque, decimos que el individuo L procesa a L', si dados dos conjuntos de números naturales S y S' pertenecientes a L y a L', respectivamente, L obtiene mayor energía al intentar procesar los elementos de S', que la energía obtenida por L' cuando intenta procesar los elementos de S. Un individuo sólo podrá dejar descendencia si ha adquirido la suficiente energía para ello. Un individuo que intenta procesar un recurso pierde energía en el desplazamiento que realiza para alcanzar la posición de tal recurso, por tanto, mientras mayor sea esta distancia, mayor energía perderá el individuo. También pierde energía si intenta procesar un recurso y no lo consigue. Si un individuo se queda sin energía, entonces muere (esto es, desaparece de la población).

Usando este modelo se pueden analizar los efectos derivados de las interacciones de los individuos con el entorno y también de las interacciones entre individuos y estudiar,

entre otras cosas, si estos afectan de alguna manera a la tendencia hacia una complejidad de los individuos cada vez mayor. Se han analizado independientemente modelos con y sin depredación. Además, en el caso de los experimentos con depredación, se han analizado de forma separada aquellos en los que las tasas de depredación y reacción son fijas y aquellos en los que no son fijas.

Independientemente del número de subentornos que componen el entorno, los resultados muestran que en el caso en el que no hay depredación en el modelo, los individuos tienden a agruparse alrededor de los subentornos. También de manera independiente del número de subentornos, cuando la depredación es introducida en el modelo, se ha obtenido que cuanto mayor es la tasa de depredación de los individuos, menor es la acumulación de los individuos alrededor de los subentornos, estando totalmente repartidos en el entorno cuando la tasa de depredación es uno y fija. En ambos casos (con o sin depredación), el tamaño de la población se reduce drásticamente en las primeras generaciones y después su comportamiento es creciente, llegando a estabilizarse al final (exceptuando cuando ambas tasas están próximas a uno y fijas, en este caso el tamaño de la población no llega a estabilizarse). Además, en el caso de los modelos con depredación donde las tasas de depredación y reacción no son fijas, se ha observado que la población evoluciona hacia un atractor que se mueve a la izquierda en el espacio de fases (donde los grados de libertad son la tasa media de depredación y de reacción de los individuos de la población) cuando crece el número de subentornos.

Con respecto a la complejidad, los resultados muestran que existe una fuerte correlación entre la complejidad del entorno y la complejidad de los individuos. Para ello, se proponen dos definiciones diferentes para la complejidad del entorno. Por un lado, podemos definir la complejidad del entorno como la media de la complejidad de los subentornos que lo componen. En este caso, se obtiene que mientras más complejo sea el entorno, más complejos son los individuos. Por otro lado, podemos definir la complejidad del entorno como el número de subentornos que lo componen. En este caso, se obtiene que mientras mayor sea el número de subentornos, menor es la velocidad de crecimiento de la complejidad de los individuos. Estos resultados pueden interpretarse como que la complejidad de los individuos se ve afectada por la complejidad del subentorno que habitan más que por la complejidad del entorno, lo que tiene sentido desde un punto de vista biológico. Por otro lado, los resultados muestran que la componente de depredación también afecta a la complejidad de los individuos, obteniendo individuos

más complejos cuando existe depredación en el modelo. Es decir, cuando los individuos compiten entre sí necesitan ser más complejos para poder sobrevivir. Sin embargo, en el caso particular donde las tasas de depredación y reacción están cercanas a uno y son fijas se ha obtenido que el crecimiento de la complejidad de los individuos de la población es mucho menor.

Finalmente, se muestra que mediante el uso de la medida de similitud definida en el Capitulo 3, se puede definir el concepto de especie en el modelo presentado en esta tesis: dos individuos son de la misma especie si y sólo si son al menos 90% similares. Esto puede usarse para realizar estudios sobre la dinámica de las especies. Estudios preliminares muestran que en general existe una fuerte relación entre el tamaño de la población y el número de especies, creciendo y estabilizandose en general en el mismo intervalo.

En conclusión, usando el marco de trabajo que ha sido propuesto en esta tesis, pueden realizarse (además de los estudios presentados en esta tesis) una gran cantidad de experimentos de ecología teórica de forma rigurosa. Es decir, este modelo constituye una fuente fiable y objetiva, desde un punto de vista matemático y computacional, en la que sacar conclusiones que, hasta cierto punto, pueden encontrar paralelismos con la biología.

Chapter 2

Some Notation, Definitions and Preliminary Results on CUDFAs and Primitive Words

The reader is assumed to be familiar with the basic concepts of formal language theory. In this chapter, only some notation used throughout the thesis will be recalled and defined. For further information the reader is referred to (102).

In the sequel, we will consider that $0 \in \mathbb{N}$. For the cases in which zero is not included, we will write $\mathbb{N}^+$. The greatest common divisor of two natural numbers n and m is denoted by $gcd(n, m)$. The cardinality of a finite set X is designated by $|X|$.

For a given alphabet V, we denote by V^* and V^+ the set of all words and all non-empty words over V, respectively. The empty word is designated by λ. For $w \in V^*$ and $x \in V$, we denote the length of w and the number of occurrences of x in w by $|w|$ and $|w|_x$, respectively. Let $w \in V^+$ be a word such that $|w| = n$ for some $n \in \mathbb{N}^+$. Then, for $1 \leq i \leq n$, $w(i)$ is the i-th letter of w, i.e., if $w = x_1 \dots x_n$, then $w(i) = x_i$. For $1 \leq j \leq n-1$ and $1 \leq i \leq n-j$, a subword $w(i)w(i+1)\dots w(i+j)$ of w is notated as $w(i : i+j)$.

A periodic sequence of numbers with period y, i.e., x, $x+y$, $x+2y$, $x+3y, \dots$, will be called a (natural) succession and will be represented as $\{x + yn \mid n \in \mathbb{N}\}$. If A is the union of the successions $A_1, A_2, \dots, A_m$, with $A_i = \{x_i + yn \mid n \in \mathbb{N}\}$ for $1 \leq i \leq m$, then we say that $A_i \sqsubseteq M$ (instead of the usual inclusion $\subseteq$ we use $\sqsubseteq$ to point ot the requirements that A_i has to be a succession and cannot be an arbitrary

subset and that the periods of A_i in all successions in M are equal) and we write $A = \{\{x_i + yn \mid n \in \mathbb{N}\}\}_{i=1,\ldots,m}$. If $m = 1$, then we omit $i = 1, \ldots, m$ in the index. The number m is called the number of successions of M.

2.1 Deterministic Finite Automata

A deterministic finite automaton (DFA, for short) is a finite state machine where for each pair of state and input symbol there is one and only one transition to a next state. DFAs recognize the set of regular languages.

A DFA will take in a string of input symbols. For each input symbol it will then transition to a state given by following a transition function. When the last input symbol has been received it will either accept or reject the string depending on whether the DFA is in an accepting state or a non-accepting state.

DFAs have many interesting properties. One of the most important is showed in the Myhill-Nerode theorem, (83). It proves that there exists a unique minimal DFA that recognizes a given regular language (minimal with regard to the number of states and unique up to an isomorphism). There are many different algorithms accomplishing this task and are described in standard textbooks on automata theory.

Such a Minimization Theorem allows to define the state complexity of a regular language, (115), as the number of states of the minimal DFA that represents it (since for any regular language there exists a unique minimal DFA that recognizes it and can be calculated).

2.1.1 Unary Deterministic Finite Automata

In this thesis we work with languages over a unary alphabet. Let A be a deterministic finite automaton over a unary alphabet (UDFA, for short) that represents a regular language. As the alphabet is unary, each UDFA will have the structure that is shown in Figure 2.1. Its states are divided into two groups, the first one, that we call initial phase, contains the states from the first state to the $i-1$-st state, the second one, that we call loop, contains the remaining states. The initial word can be empty in those automata that transits from its last state to its initial state.

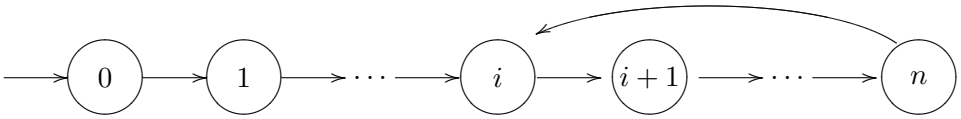


Figure 2.1: Structure of a UDFA

A UDFA will be represented as a vector (v, w) where $v \in \{0,1\}^*$ describes the initial phase and $w \in \{0,1\}^+$ describes the loop. The zeros represent the non-accepting states of the automaton, and the ones represent the accepting states of the automaton.

For example, the representation of the automaton in Figure 2.2 is $(011, 110)$.

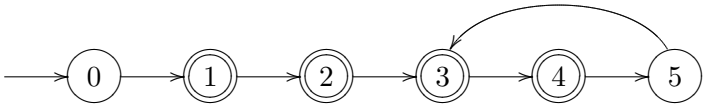


Figure 2.2: An example of a UDFA where the states with two circles are the accepting states

For a UDFA (v, w), where the states are numbered starting from zero, let

$$A(v,w) = \{a - 1 \mid 1 \leq a \leq |v|,\ v(a) = 1\}$$

and

$$B(v,w) = \{b - 1 + |v| \mid 1 \leq b \leq |w|,\ w(b) = 1\}.$$

Let $n, m \in \mathbb{N}^+$. If $|A(v,w)| = n$ and $|B(v,w)| = m$ is assumed, then

$$A(v,w) = \{a_1, \ldots, a_n\} \text{ and } B(v,w) = \{b_1, \ldots, b_m\}.$$

Since the strings accepted by UDFAs are sequences of the same symbol, we can identify a string with its length. Then, the set of strings accepted by a given UDFA will be represented by a subset of the natural numbers. Any natural number k that belongs to such a subset represents the string of length k. Thus, we say that (v, w) represents the unary regular language (URL, for short)

$$L(v,w) = \{a_1, a_2, \ldots, a_n\} \cup \{b_1 + |w|k, b_2 + |w|k, \ldots, b_m + |w|k \mid k \in \mathbb{N}\} \tag{2.1}$$

For example, the URL that is given by the automaton in Figure 2.2 is represented by $\{1, 2\} \cup \{3 + 3k, 4 + 3k \mid k \in \mathbb{N}\}$. In the sequel we use a shorter notation where the set

of (2.1) is given by

$$L(v,w) = \{\{a_i, b_j + |w|k\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,n,\ j=1,2,\ldots,m} \tag{2.2}$$

Without loss of generality, we will use the minimal UDFA (for short, MUDFA) that represents a given URL to obtain the previous notation for the URLs, in this way, we will have the minimal n and m, that will provide a unique representation for each URL (since there is a unique MUDFA for any given URL).

2.1.2 Cyclic Unary Deterministic Finite Automata

A UDFA is cyclic if its initial phase is empty (see Figure 2.3). Then, instead of (λ, w), we represent the CUDFA as a word $w \in \{0,1\}^+$. A language accepted by some CUDFA is a CURL.

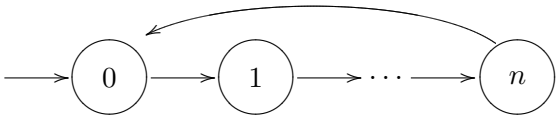


Figure 2.3: Structure of a CUDFA

Moreover, in this case, instead of $B(\lambda, w)$, we represent the set of the accepting states of the automaton as $B(w) = \{i-1 \mid w(i) = 1\}$. Thus, it is clear that if $B(w) = \{b_1, \ldots, b_m\}$ for some $m \in \mathbb{N}^+$, then $0 \leq b_i < |w|$ for all $1 \leq i \leq m$.

Therefore, the notation for a CURL $L(w)$, that is represented by the CUDFA w, will be

$$L(w) = \{b_1 + |w|k, b_2 + |w|k, \ldots, b_m + |w|k \mid k \in \mathbb{N}\}.$$

Thus a CURL is given by an infinite set of natural numbers, more precisely, by the union of finitely many disjoint successions of natural numbers. In the sequel we use the notation

$$L(w) = \{\{b_i + |w|k\}_{k\in\mathbb{N}}\}_{i=1,\ldots,m}. \tag{2.3}$$

In this thesis, we consider a CUDFA w as a genotype, and its accepted CURL $L(w)$ as the corresponding phenotype.

2.2 Primitive Words and a Characterization of Minimal Unary Deterministic Finite Automata

The results presented in this section are part of (70) in which I am first author[1].

For a given alphabet V, a word $w \in V^+$ is said to be a primitive word if and only if there does not exist a word $u \in V^+$ with $|u| < |w|$ such that $w = u^n$ with $n \in \mathbb{N},\ n > 1$. The set of all primitive words over V is denoted by Q_V. If V is understood from the context we omit the index V and write simply Q.

The next theorem characterizes the case in which a UDFA is minimal. In this case, $V = \{0, 1\}$.

Theorem 1. *Let $v \in V^*$ and $w \in V^+$. Then (v, w) is a minimal UDFA if and only if $w \in Q$ and $v(|v|) \neq w(|w|)$.*

Proof. Let (v, w) be a minimal UDFA, let us see that then $w \in Q$ and $v(|v|) \neq w(|w|)$. First of all, it is shown $w \in Q$, i.e., there does not exist n, u such that $w = u^n$ with $n > 1$ and $u \in \{0, 1\}^+$:

Let us suppose that $w = u^n$ with $n > 1$ and $u \in \{0, 1\}^+$. Let $|u| = m$, $|w| = q$ and $|u|_1 \leq m$.

Since $w = u^n$, we have $w(j + mr) = u(j)$ for any $1 \leq j \leq m$ and $0 \leq r \leq n - 1$.

Let $A(v, u) = A(v, w) = \{a_1, \ldots, a_r\}$ and $B(v, u) = \{b_1, \ldots, b_t\}$. Then

$$B(v, w) = \{b_1, \ldots, b_t, b_1 + m, \ldots, b_t + m,$$

$$b_1 + 2m, \ldots, b_t + 2m, \ldots, b_1 + (n - 1)m, \ldots, b_t + (n - 1)m\}$$

Thus,

$$L(v, u) = \{\{a_j, b_i + km\}_{k \in \mathbb{N}}\}_{j=1,\ldots,r,\ i=1,\ldots,t}$$

and

$$L(v, w) = \{\{a_j, b_i + rm + kq\}_{k \in \mathbb{N}} \mid j = 1, \ldots, r, i = 1, \ldots, t, r = 0, \ldots, n - 1\}.$$

Since

$$(b_i + rm) + kq = b_i + rm + knm = b_i + (r + kn)m$$

for any $i = 1, \ldots, t$, we have $L(w) \subseteq L(u)$.

Since $k = k'n + r$ for some $0 \leq k'$ and $r \leq n - 1$, we have

$$b_i + km = b_i + (k'n + r)m = (b_i + rm) + k'nm = (b_i + rm) + k'q$$

[1] the rest of coauthors are my supervisors and gave hints, suggestions, and comments

for any $i = 1, \ldots, t$, which means $L(u) \subseteq L(w)$.

Therefore $L(w) = L(u)$. Because u represents the same language as w, we have $L(v, u) = L(v, w)$. Since $|u| < |w|$, the UDFA (v, w) is not minimal. Thus, if (v, w) is a minimal UDFA, then $w \in Q$.

Now, let us see that if (v, w) is a minimal UDFA, then $v(|v|) \neq w(|w|)$.

Let us suppose that $v(|v|) = w(|w|)$. There are two possibilities: $v(|v|) = w(|w|) = 1$ or $v(|v|) = w(|w|) = 0$.

Let us suppose $v(|v|) = w(|w|) = 1$. Considering $|v| = p + 1$ and $|w| = q$ such that $|v| + |w| = n + 1$, if the last state of w is removed and the automaton

$$(v', w') = (v(1 : p), v(p + 1)w(1 : q - 1))$$

is considered, then the language represented by such an automaton, $L(v', w')$, is:

$$\{a_1, \ldots, a_{r-1}, a_r + qk, b_1 + qk, \ldots, b_{t-1} + qk \mid k \in \mathbb{N}\}$$

with $|v'| + |w'| = n$.

Let us observe that $L(v, w) = L(v', w')$. Since the only difference between both languages are the sublanguages $\{a_r, b_t + qk\}$ in $L(v, w)$ and $\{a_r + qk\}$ in $L(v', w')$, $L(v, w) = L(v', w')$ if and only if $F = \{a_r, b_t + qk \mid k \in \mathbb{N}\} = \{a_r + qk \mid k \in \mathbb{N}\} = S$.

Firstly, $F \subseteq S$ because $a_r = a_r + q0$ and

$$b_t + qk = n + 1 + qk = n + 1 + q(k' - 1) =$$

$$= n - q + 1 + qk' = p + 1 + qk' = a_r + qk'$$

with $k' \in \mathbb{N}$ and $k = k' - 1$.

Furthermore, $S \subseteq F$ because

$$a_r + qk' = p + 1 + qk' = p + 1 + q(k + 1) =$$

$$= p + q + 1 + qk = n + 1 + qk = b_s + qk$$

with $k \in \mathbb{N}$ and $k' = k + 1$.

Therefore $L(v, w) = L(v', w')$. Then the automaton (v, w) is not minimal, because its last state can be removed and the automaton $(v(1 : p), v(p + 1)w(1 : q - 1))$ still represents the same language.

Now let us consider the case where $v(|v|) = w(|w|) = 0$. If the last state of w is removed and the automaton $(v(1 : p), v(p + 1)w(1 : q - 1))$ is considered, the language represented by such an automaton is the same language as (v, w), it can be proved analogously as before. Then the automaton (v, w) is not minimal because its last state

can be removed and the automaton $(v(1:p), v(p+1)w(1:q-1))$ still represents the same language.

Thus, if (v, w) is a MDFA, necessarily $v(|v|) \neq w(|w|)$.

Let $v \in V^*$, $w \in Q$ and $v(|v|) \neq w(|w|)$, let us see that then (v, w) is a minimal UDFA.

Let us assume that (v, w) is not minimal. Then there exists a minimal UDFA (q, p) accepting the same language as (v, w). Since (q, p) is minimal, we have that $|q|+|p| < |v|+|w|$ and moreover, by the proof of the previous implication, we also have $p \in Q$ and $q(|q|) \neq p(|p|)$.

Let us suppose that $|v| = |q|$. Then, we have $v = q$. Then, $vw^{|p|} = vp^{|w|}$. Therefore, $w^{|p|} = p^{|w|}$. By Lemma 1.9 in (64), w and p are powers of some word. Since w and p are primitive words, we have $w = p$. Therefore, $(q, p) = (v, w)$ in contrast to our assumption that (v, w) is not minimal.

Let us suppose that $|v| < |q|$. Let $r = |q| - |v|$. Let $r = k|w| + z$ for some $k \geq 0$ and some z with $0 \leq z \leq |w| - 1$. Since we can write $w = uu'$ for some words $u, u' \in V^+$ with $|u| = z$, we have $|q| = |v| + k|w| + |u|$. Therefore, $qp^{|w|} = vw^k uw'^{|p|}$ where $w' = u'u$. Thus, we have $qp^{|w|} = qw'^{|p|}$. As above, we have $p = w'$. Since $|q| + |p| < |v| + |w| = |v| + |w'| = |v| + |p|$, we have $|q| < |v|$ in contrast to our assumption that $|v| < |q|$.

Let us suppose that $|q| < |v|$. Let $r = |v| - |q|$. Let $r = k|p| + z$ for some $k \geq 0$ and some z with $0 \leq z \leq |p| - 1$. Since we can write $p = uu'$ for some words $u, u' \in V^+$ with $|u| = z$, we have $|v| = |q| + k|p| + |u|$. Therefore, $vw^{|p|} = qp^k up'^{|w|}$ where $p' = u'u$. Thus, we have $vp'^{|w|} = vw^{|p|}$. As above, we have $w = p'$. We also have $v = qp^k u$. Thus, the last letter of v is equal to the last letter of w and since $v(|v|) \neq w(|w|)$, it is a contradiction. □

We can particularize the previous theorem in the case of CUDFAs.

Corollary 1. *Let w be a CUDFA. Therefore, w is a minimal CUDFA if and only if $w \in Q$.*

2. SOME NOTATION, DEFINITIONS AND PRELIMINARY RESULTS ON CUDFAS AND PRIMITIVE WORDS

Chapter 3

A Similarity Measure for Cyclic Unary Regular Languages

The results presented in this chapter are part of (24) in which I am first author[1].

As we said before, a measure of the similarity between CURLs is needed so as to calculate how well an individual (a language) adapts to a given environment in order to study the dynamics of the complexity of such individuals during the evolution. In this chapter, we propose a similarity measure for CURLs by modifying the Jaccard similarity coefficient and the Sørensen coefficient to measure the level of overlap between such languages. This measure computes the proportion of strings that are shared by two or more cyclic unary regular languages and is an upper bound of the Jaccard coefficient and the Sørensen coefficient. By using such similarity measure, we define a dissimilarity measure for cyclic unary regular languages that is a semimetric distance. Moreover, it can be used for the non-cyclic case.

3.1 Introduction

URLs are regular languages over a unary alphabet. Due to their relation to many number-theoretic results, as well as their difference from the general case (non-unary regular languages), they are of particular interest in the study of state complexity. Thus, some papers on state complexity of URLs have been published. For example, in

[1] authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments

(40), deterministic unary automata, nondeterministic unary automata and probabilistic unary automata accepting the same languages are compared with respect to their size. In (43), the computational complexity of the nondeterministic automaton minimization problem for finite and unary regular languages, if the input is a DFA, is investigated. In (73), the nondeterministic state complexity of URLs and of their complements are also compared. More studies related to URLs and state complexity can be seen in (95; 96; 107).

A CURL is a URL that can be represented by a cyclic automaton. In (10), the behavior of Hopcroft's algorithm for minimizing CUDFAs is analyzated. The relationships between the combinatorial properties of a circular sturmian word and the run of the Hopscroft's algorithm on its associated cyclic automaton is investigated in (21). Other properties of CURLs and unary nondeterministic finite automata have been investigated in (32; 43; 56).

However, there is a lack of results that compare two (neither cyclic nor non-cyclic) URLs based on their shared strings. As far as we know, there exist very few measures of the overlap between two URLs or between two CURLs, if we compare it with the amount of measures that there exist in the case of other types of languages. For example, in (17), an iterative procedure to compute the relative entropy between two stochastic deterministic regular grammars is proposed and in (18), approaches to compute a similarity measure between distributions generated by probabilistic tree automata is defined. In (81), mathematical distances between pairs of probabilistic context-free grammar have been investigated. A distance measure to compare distributions that are represented by stochastic DFA is presented in (79). Moreover, those few measures of the overlap between regular languages that there exist, (9; 39; 60; 63; 114), are not of interest for us, since they do not satisfy the principle: sets are more similar if they have more elements in common. This will be show this in depth in the disscusion section.

On the other hand, if one considers dynamic systems or genetic algorithms, where the populations are presented by unary regular sets (see e.g. (69; 70)), then the selection process requires a comparison of such sets in terms of portion of shared strings. Thus we are interested in a similarity measure for unary regular sets that satisfies such a principle.

In the case of finite sets A and B the Jaccard coefficient and the Sørensen (or Dice) coefficient defined by

$$JC_{A,B} = \frac{|A \cap B|}{|A \cup B|} \text{ and } SC_{A,B} = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{3.1}$$

are well-known measures of similarity (see (110), (100), (1)). Obviously, the intuitive idea behind these measures is that sets are more similar if they have more elements in common. These measures cannot directly be used for infinite sets. Since we are interested in infinite regular sets, the Jaccard and Sørensen coefficients cannot directly be used for CURLs.

In this section, we introduced modified variants of these coefficients. But their computations cannot directly be performed using a given representation of the CURLs by their minimal automata; it needs a transformation to the representation of the CURLs that has been presented in the previous section.

Thus, in this work, we propose a similarity measure for CURLs that computes the overlap between two or more CURLs directly from the given representations by minimal automata. Moreover, we prove that the similarity measure for CURLs proposed in this work is an upper bound of the Jaccard coefficient and the Sørensen coefficient for CURLs. Furthermore, if a sequence of CURLs approaches a certain CURL with respect to one of the considered similarities, then this also holds for the other ones. Thus by the relation between the measures it seems that a tendency can be seen earlier by using the newly introduced measure.

Using the similarity measure, we also define a dissimilarity measure for CURLs. That will be done in the same way as the Jaccard distance is defined by using the Jaccard coefficient (in the case of finite sets). In contrast to the Jaccard distance, such a dissimilarity measure for CURLs is not a metric distance, since the triangle inequality is not satisfied. We prove that it is a semimetric distance.

Finally, we mention that we can also use the dissimilarity measure proposed in this work in the case of non-cyclic URLs. Therefore, in general, we have a dissimilarity measure for URLs (cyclic and non-cyclic). We show that the dissimilarity measure for URLs is a symmetric distance (it does not satisfy the identity of indiscernibles) and not a semimetric distance.

3.2 A Similarity Measure for CURLs

3.2.1 Similarity between two Successions

It is natural to say that two successions have the similarity 0 if they have no numbers in common. Therefore we are interested in the cases where the intersection of the two successions of natural numbers is not empty.

Lemma 1. *For each two natural successions $A = \{a + bn\}_{n\in\mathbb{N}}$ and $B = \{c + dk\}_{k\in\mathbb{N}}$, $A \cap B \neq \emptyset$ if and only if $c - a$ is a multiple of $gcd(b, d)$ (where $gcd(b, d)$ is the greatest common divisor of b and d).*

Proof. $a + bn = c + dk$ if and only if $c - a = bn - dk$. By the Main Theorem on *gcd*, there is a solution in $\mathbb{Z}$ of this equation, if and only if $c-a$ is a multiple of $gcd(b, d)$. □

Lemma 2. *Let M be a CURL. Given $A, B \sqsubseteq M$, $A \cap B \neq \emptyset$ if and only if $A = B$.*

Proof. Let us suppose that $A \cap B \neq \emptyset$. If we assume that $A = \{a_i + bk\}_{k\in\mathbb{N}}$ and $B = \{a_j + bk\}_{k\in\mathbb{N}}$, by Theorem 1, $A \cap B \neq \emptyset$ if and only if $|a_i - a_j|$ is a multiple of $gcd(b, b) = b$. Since $a_i, a_j < b$, we have $0 \leq |a_i - a_j| < b$. Then $|a_i - a_j|$ is a multiple of $gcd(b, b) = b$ if and only if $|a_i - a_j| = 0$, that is, $A = B$. □

Let $A = \{a + bn\}_{n\in\mathbb{N}}$ and $B = \{c + dk\}_{k\in\mathbb{N}}$ be two natural successions. We use the frequency in which the overlapped elements, i.e., elements which are in A as well as in B, appear in A as the measure of the overlap (thus it reflects the portion of elements of B in A).

Let

$$T = \{k \in \mathbb{N} \mid \frac{c-a}{b} + \frac{d}{b}k \in \mathbb{N}\}$$

be the set of natural numbers such that the element $c + dk$ of B is contained in A. Furthermore, let t be the minimal number in T. We determine the amount that has to be added to t in order to obtain another element of the set T. Thus, $x \in \mathbb{N}$ with $\frac{c-a}{b} + \frac{d}{b}(t + x) \in \mathbb{N}$ is looked for. Since

$$\frac{c-a}{b} + \frac{d}{b}(t + x) \in \mathbb{N} \text{ if and only if } \frac{c-a}{b} + \frac{d}{b}t + \frac{d}{b}x \in \mathbb{N}$$

and $\frac{c-a}{b} + \frac{d}{b}t \in \mathbb{N}$, we have $\frac{c-a}{b} + \frac{d}{b}t + \frac{d}{b}x \in \mathbb{N}$ if and only if $\frac{d}{b}x \in \mathbb{N}$ if and only if $x = \frac{b}{gcd(b,d)}m$ for some $m \in \mathbb{N}$.

So, if $T = \{t + \frac{b}{gcd(b,d)}m \mid m \in \mathbb{N}\}$, then the overlapped terms belong to

$$T' = \{\frac{c-a}{b} + \frac{d}{b}(t + \frac{b}{gcd(b,d)}m) \mid m \in \mathbb{N}\}.$$

Therefore the distance of two successive elements of T' is given by

$$\frac{c-a}{b} + \frac{d}{b}(t + \frac{b}{gcd(b,d)}m) - [\frac{c-a}{b} + \frac{d}{b}(t + \frac{b}{gcd(b,d)}(m-1))] = \frac{d}{gcd(b,d)}.$$

So, starting from t can be affirmed that a natural number that belongs to T' will be found in A every $\frac{d}{gcd(b,d)}$ terms.

Therefore, $\frac{gcd(b,d)}{d}$ can be considered as the overlap of A with B. That is, we have done a partition of the succession A into d disjoint subsets and $gcd(b,d)$ words of them belong to B.

Definition 1. *The overlap of an infinite succession $A = \{a+bn\}_{n\in\mathbb{N}}$ with another one $B = \{c+dk\}_{k\in\mathbb{N}}$, that we will call $ISO_{A,B}$ (for Infinite Successions Overlap), is defined as:*

$$ISO_{A,B} = \begin{cases} \frac{gcd(b,d)}{d} & \text{if } A \cap B \neq \emptyset \\ 0 & \text{in other case} \end{cases}$$

Lemma 3. *Let $A = \{a + bn\}_{n\in\mathbb{N}}$ and $B = \{c + dk\}_{k\in\mathbb{N}}$ be two natural successions. Then $A \subseteq B$ if and only if $ISO_{A,B} = 1$.*

Proof. $ISO_{A,B} = 1$ if and only if $gcd(b,d) = d$ if and only if $b = du$ for some $u \in \mathbb{N}$. Since $ISO_{A,B} = 1$ we have $A \cap B \neq \emptyset$ and thus, by Lemma 1, $c - a = t \cdot gcd(b,d) = td$ for some $t \in \mathbb{N}$. For $n \in \mathbb{N}$, we get $a + bn = c + dt + dun = c + d(t + un)$, which proves that any element of A is contained in B or equivalently, $A \subseteq B$.

If $A \subseteq B$, then for any $n \in \mathbb{N}$, there exists $m \in \mathbb{N}$ such that $a + bn = c + dm$. In particular, it holds for $n = 1$, that is, there exists $m \in \mathbb{N}$ such that $a + b = c + dm$. Since $c - a = td$, we have $b = d(t + m)$. Thus $gcd(b,d) = d$ and $ISO_{A,B} = 1$. □

The similarity of two successions combines $ISO_{A,B}$ and $ISO_{B,A}$.

Definition 2. *The similarity measure between two infinite successions $A = \{a+bn\}_{n\in\mathbb{N}}$ and $B = \{c+dk\}_{k\in\mathbb{N}}$, that we will call $ISS_{A,B}$ (for Infinite Successions Similarity), is defined as:*

$$ISS_{A,B} = \begin{cases} \frac{ISO_{A,B} + ISO_{A,B}}{2} & \text{if } A \cap B \neq \emptyset \\ 0 & \text{in other case} \end{cases}$$

Given two infinite successions A and B, $0 \leq ISS_{A,B} \leq 1$, since $0 \leq ISO_{A,B} \leq 1$ and $0 \leq ISO_{B,A} \leq 1$ for any infinite successions A and B.

3.2.2 The Proposed Similarity Measure for CURLs

In this section, we define the similarity measure for CURLs by using the similarity measure between two successions that has been defined in the previous section.

Given two CURLs M and N, we have that $M \cap N \neq \emptyset$ if and only if there exist at least $A \sqsubseteq M$ and $B \sqsubseteq N$ such that $A \cap B \neq \emptyset$.

Definition 3. *Let M and N be two CURLs. We define the overlap of M with N, that we will call $URLO_{M,N}$ (for Unary Regular Languages Overlap), as:*

$$URLO_{M,N} = \begin{cases} \frac{1}{m} \sum\limits_{\substack{A \sqsubseteq M \\ B \sqsubseteq N}} ISO_{A,B} & \text{if } M \cap N \neq \emptyset \\ 0 & \text{in other case} \end{cases}$$

where $m \in \mathbb{N}$ is the number of successions of M.

By following the same reasoning as in the previous section, we define the similarity measure between two CURLs as follows:

Definition 4. *Let M and N be two CURLs. We define the similarity measure between M and N, that we will call $URLS_{M,N}$ (for Unary Regular Languages Similarity), as:*

$$URLS_{M,N} = \begin{cases} \frac{URLO_{M,N} + URLO_{N,M}}{2} & \text{if } M \cap N \neq \emptyset \\ 0 & \text{in other case} \end{cases}$$

Remark 1. Let $M = \{\{a_i + |v|k\}_{k\in\mathbb{N}}\}_{i=1,\dots,n}$ and $N = \{\{b_j + |w|k\}_{k\in\mathbb{N}}\}_{j=1,\dots,m}$ be the two CURLs for some $n, m \in \mathbb{N}$. In the particular case in which $gcd(|v|, |w|) = 1$, we have

$$URLS_{M,N} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1}{|v|}}{2}.$$

Moreover, in the particular case in which $gcd(|v|, |w|) = |w|$, we have

$$URLS_{M,N} = \frac{\frac{q}{|v|_1} + \frac{q}{|w|_1}}{2},$$

where q is the number of pairs $\{a_i + |v|k\}_{k\in\mathbb{N}}$ and $\{b_j + |w|k\}_{k\in\mathbb{N}}$ such that

$$\{a_i + |v|k\}_{k\in\mathbb{N}} \cap \{b_j + |w|k\}_{k\in\mathbb{N}} \neq \emptyset$$

(and that is if and only if $a_i = b_j$).

Our definitions require that the regular sets R and S are given as sets M and N of periodic sequences which are induced by the minimal automata of R and S. We now prove that any other description as sets M' and N' of periodic sequences which are induced by DFAs accepting R and S give the same similarity.

Theorem 2. *Let $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m}$ and $\overline{M} = \{\{a'_i + b'k\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m'}$ be two descriptions of the regular set R, and let $N = \{\{c_i + dk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,n}$ and $\overline{N} = \{\{c'_i + d'k\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,n'}$ be two descriptions of the regular set S. Then*

$$URLS_{M,N} = URLS_{\overline{M},\overline{N}}.$$

Proof. We first compute $URLS_{M,N}$. Let us assume that there are q pairs (i, j) such that $\{a_i + bk\} \cap \{c_j + dk\} \neq \emptyset$. Then we get

$$URLO_{M,N} = \frac{1}{m}\sum_{i,j} ISO_{\{a_i+bk\},\{c_j+dk\}} = \frac{1}{m}\cdot q \cdot \frac{gcd(b,d)}{d} \tag{3.3}$$

and an analogous result for $URLO_{N,M}$ taking n and b instead of m and d, respectively. Thus

$$URLS_{M,N} = \frac{\frac{q\cdot gcd(b,d)}{md} + \frac{q\cdot gcd(b,d)}{nb}}{2} = \frac{q\cdot gcd(b,d)(md+nb)}{2nmbd}. \tag{3.4}$$

Now we prove that other special representations of the sets R and S give the same value. Let z be the lowest common multiple of b and d. We set

$$g = \frac{z}{b} \text{ and } h = \frac{z}{d}.$$

Then we also have

$$g = \frac{d}{gcd(b,d)} \text{ and } h = \frac{b}{gcd(b,d)}. \tag{3.5}$$

We now construct the successions

$$M' = \{a_i + vb + kz \mid 1 \leq i \leq m,\ 0 \leq v \leq g-1\}_{k\in\mathbb{N}}$$

and

$$N' = \{c_j + v'd + kz \mid 1 \leq j \leq n,\ 0 \leq v' \leq h-1\}_{k\in\mathbb{N}}.$$

Thus we have ng successions in M' and mh successions in N'. Obviously, M and M', as well as N and N', describe the same regular languages. By Lemma 2, all successions of M' and N' are pairwise disjoint. As above we get

$$URLO_{M',N'} = \frac{1}{mg}\sum_{i,j,v,v'} ISO_{\{a_i+vb+kz\},\{c_j+v'd+kz\}} = \frac{q}{mg} = \frac{q\cdot gcd(b,d)}{md}$$

and an analogous result for $URLO_{N',M'}$ which leads to

$$URLS_{M',N'} = \frac{q \cdot gcd(b,d)(md+nb)}{2nmbd} = \frac{q(mg+nh)}{2nmgh} \tag{3.6}$$

Therefore we have $URLS_{M,N} = URLS_{M',N'}$.

The same argumentation can be used if we consider representations M'_u and N'_u which are based on a multiple u of z.

Let y be the lowest common multiple of b, d, b', d'. Then we get

$$URLS_{M,N} = URLS_{M'_y,N'_y} \text{ and } URLS_{\overline{M},\overline{N}} = URLS_{\overline{M}'_y,\overline{N}'_y}. \tag{3.7}$$

Since M'_y and $(M')'_y$ describe R we get that

$$U = \{a_i+sb \mid 1 \le i \le m,\ 0 \le s \le \frac{u}{b}-1\} \text{ and } U' = \{a'_i+tb' \mid 1 \le i \le m',\ 0 \le t \le \frac{u}{b'}-1\}$$

describe the set of all words in R of length at most $y-1$. Thus $U = U'$ and consequently $M'_y = (M')'_y$ (since we extend U and U' only by adding multiples of y). Analogously, $N_y = (N')_y$. Therefore, by (3.7),

$$URLS_{M,N} = URLS_{M'_y,N'_y} = URLS_{\overline{M}'_y,\overline{N}'_y} = URLS_{\overline{M},\overline{N}}.$$

□

Thus, in the sequel, we use the description which is most appropriate for our proofs.

We now present some elementary properties of the similarity measure, we particularly show that it is a value between 0 and 1 (which is a desired property).

Lemma 4. $0 \le URLS_{M,N} \le 1$ *for any two CURLs* M *and* N.

Proof. Let M and N be two CURLs. The relation $0 \le URLS_{M,N}$ is obvious.

Let $M = \{\{a_i+bk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m}$ and $N = \{\{c_j+dk\}_{k\in\mathbb{N}}\}_{j=1,2,\ldots,n}$ with $n, m \in \mathbb{N}^+$. Let z be the lowest common multiple of b and d. Moreover, let

$$g = \frac{z}{b} = \frac{d}{gcd(b,d)} \text{ and } h = \frac{z}{d} = \frac{b}{gcd(b,d)}.$$

Then we can describe M and N as

$$\begin{aligned} M &= \{A_{1,1}, A_{1,2}, \ldots, A_{1,g}, A_{2,1}, \ldots, A_{2,g}, \ldots, A_{m,1}, \ldots, A_{m,g}\} \\ &\quad \text{with } A_{i,p} = \{a_{i,p} + zk\}_{k\in\mathbb{N}}, a_{i,p} = a_i + (p-1)b \text{ for } 1 \le i \le m, 1 \le p \le g, \\ N &= \{C_{1,1}, C_{1,2}, \ldots, C_{1,h}, C_{2,1}, \ldots, C_{2,h}, \ldots, C_{n,1}, \ldots, C_{n,h}\} \\ &\quad \text{with } C_{j,l} = \{c_{j,l} + zk\}_{k\in\mathbb{N}}, c_{j,l} = c_j + (l-1)d \text{ for } 1 \le j \le n, 1 \le l \le h. \end{aligned}$$

Let $A_{i,p} \cap C_{j,t} \neq \emptyset$. Then by Lemma 1, $a_{i,p} - c_{j,t} = gcd(z,z)s = zs$ for some $s \in \mathbb{N}$. Moreover, $a_{i,p} < z$ and $c_{j,t} < z$ implies $|a_{i,p} - c_{j,t}| < z$. Thus, necessarily $a_{i,p} = c_{j,t}$. This implies immediately $A_{i,p} \subseteq C_{j,t}$. Then, by Lemma 3, $ISO_{A_{i,p},C_{j,t}} = 1$. Moreover, since all the $c_{j,t}$ are different, for any $A_{i,p} \sqsubseteq M$, there exists at most one $C_{j,t} \sqsubseteq N$ such that $A_{i,p} \cap C_{j,t} \neq \emptyset$.

If $A_{i,p} \cap C_{x,y} = \emptyset$, then $ISO_{A_{i,p},C_{x,y}} = 0$. Thus we get

$$\sum_{C_{j,t} \sqsubseteq N} ISO_{A_{i,p},C_{j,t}} = \begin{cases} 1 & \text{if } A_{i,p} \cap N \neq \emptyset \\ 0 & \text{if } A_{i,p} \cap N = \emptyset \end{cases}.$$

Now we obtain

$$\begin{aligned} URLO_{M,N} &= \frac{1}{mg} \sum_{\substack{A_{i,p} \sqsubseteq M \\ C_{j,t} \sqsubseteq N}} ISO_{A_{i,p},C_{j,t}} = \frac{1}{mg} \sum_{A_{i,p} \sqsubseteq M} \Big(\sum_{C_{j,t} \sqsubseteq N} ISO_{A_{i,p},C_{j,t}} \Big) \\ &\leq \frac{1}{mg} \sum_{A_{i,p} \sqsubseteq M} 1 = \frac{1}{mg} \cdot mg = 1. \end{aligned}$$

Analogously, we have $URLO_{N,M} \leq 1$. Thus, by the definition of $URLS_{M,N}$, we get $URLS_{M,N} \leq 1$. □

Lemma 5. *Let M and N be CURLs. $URLO_{M,N} = 1$ if and only if $M \subseteq N$.*

Proof. We consider the presentations given in the proof of Lemma 4.

Let us suppose $M \subseteq N$. Since $M \subseteq N$ if only if $A_{i,p} \sqsubseteq N$ for any $A_{i,p} \sqsubseteq M$, we get $A_{i,p} \cap N \neq \emptyset$ for any $A_{i,p} \sqsubseteq M$. Therefore, by the proof of Lemma 4, $\sum_{C_{j,t} \sqsubseteq N} ISO_{A_{i,p},C_{j,t}} = 1$ for any $A_{i,p} \sqsubseteq M$. Thus we obtain an equality in (3.8), which proves that $URLO_{M,N} = 1$.

Conversely, $URLO_{M,N} = 1$ if and only if $\sum_{C \sqsubseteq N} ISO_{A,C} = 1$ for any $A \sqsubseteq M$. Therefore, $A \cap N \neq \emptyset$ for any $A \in M$. Consequently, for any $A \in M$, there is a $C \in N$ such that $A \cap C \neq \emptyset$. As in the proof of Theorem 4, we can show that $A \cap C \neq \emptyset$ implies $A \subseteq C$. Thus, for any $A \sqsubseteq M$, there is a $C \sqsubseteq N$ with $A \subseteq C$. This implies $A \sqsubseteq N$ for any $A \sqsubseteq M$ which gives $M \subseteq N$. □

We have shown that, for any CURLs M and N, $0 \leq URLS_{M,N} \leq 1$. We will now show that also the converse holds, i.e., every number x with $0 \leq x \leq 1$ can be obtained as a similarity.

Theorem 3. *The measure URLS is dense, i.e., for any (rational) number $x \in [0,1]$ and any $\varepsilon \geq 0$, there are CURLs M and N such that*

$$|URLS_{M,N} - x| \leq \varepsilon.$$

Proof. Obviously, for the sequences $M = \{0+2k\}_{k\in\mathbb{N}}$ and $N = \{1+2k\}_{k\in\mathbb{N}}$, we get $URLS_{M,M} = 1$ and $URLS_{M,N} = 0$.

Let $0 < x < 1$. Then we choose prime numbers p and q sufficiently large such that $p < q$, $xp \leq p-1$ and $\frac{1}{2p} + \frac{1}{2q} \leq \varepsilon$. Then we also have $xq \leq q-1$. We now choose $m = \lceil xp \rceil$ and $n = \lceil xq \rceil$ and

$$M = \{\{i+pk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m} \text{ and } N = \{\{j+qk\}_{k\in\mathbb{N}}\}_{j=1,2,\ldots,n}.$$

Since the greatest common divisor of p and q is 1 and any difference $i-j$, $1 \leq i \leq m$ and $1 \leq j \leq n$, is a multiple of 1, any pair of successions $\{i+pk\}$ and $\{j+qk\}$ has an non-empty intersection. Thus we get

$$URLS_{M,N} \quad = \quad \frac{URLO_{M,N} + URLO_{N,M}}{2} = \frac{n\frac{1}{q} + m\frac{1}{p}}{2} = \frac{np+mq}{2pq}.$$

If we take into consideration that $xp \leq m \leq xp+1$ and $xq \leq n \leq xq+1$, we get

$$x = \frac{xpq+xqp}{2pq} \leq \frac{np+mq}{2pq} \leq \frac{(xp+1)q+(xq+1)p}{2pq} = x + \frac{1}{2p} + \frac{1}{2q} \leq x+\varepsilon.$$

Now the statement follows immediately. □

3.3 Jaccard Coefficient and Sørensen Coefficient for CURLs

3.3.1 Definition of a Jaccard Coefficient and Sørensen Coefficient for CURLs

The Jaccard coefficient given in the Introduction is a well-known measure for the similarity of finite sets. As we said before, we can not use this measure directly for CURLs because both sets are infinite if the intersection is non-empty. In this section, we propose an appropriate definition of the Jaccard coefficient which can be used for CURLs.

Let

$$M = \{\{a_i+bk\}_{k\in\mathbb{N}}\}_{i=1,\ldots,m} \text{ and } N = \{\{c_j+dk\}_{k\in\mathbb{N}}\}_{j=1,\ldots,n}$$

be two CURLs consisting of n and m sequences, respectively. Let

$$M_{s,t} = M \cap \{i \mid s \leq i \leq t\} \text{ and } N_{s,t} = N \cap \{i \mid s \leq i \leq t\}$$

be the subsets of M and N, consisting of all numbers greater or equal to s and smaller or equal to t. Then a natural definition of a Jaccard coefficient would be

$$\lim_{t\to\infty} \frac{|M_{0,t} \cap N_{0,t}|}{|M_{0,t} \cup N_{0,t}|}.$$

In order to use this definition we have to show that the limit exists. This will now be done. Let z be the lowest common multiple of b and d. Then it is clear that

$$M_{rz,(r+1)z-1} = M_{0,z-1} + rz = \{y + rz \mid y \in M_{0,z-1}\}$$

for all $r \geq 0$. Hence, for $t = rz + u$,

$$\begin{aligned} |M_{0,t} \cap N_{0,t}| &= r|M_{0,z-1} \cap N_{0,z-1}| + |M_{0,u} \cap N_{0,u}|, \\ |M_{0,t} \cup N_{0,t}| &= r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|. \end{aligned}$$

Therefore we get

$$\begin{aligned} &\frac{|M_{0,t} \cap N_{0,t}|}{|M_{0,t} \cup N_{0,t}|} = \frac{r|M_{0,z-1} \cap N_{0,z-1}| + |M_{0,u} \cap N_{0,u}|}{r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|} \\ &= \frac{r|M_{0,z-1} \cap N_{0,z-1}|}{r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|} + \frac{|M_{0,u} \cap N_{0,u}|}{r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|} \\ &= \frac{|M_{0,z-1} \cap N_{0,z-1}|}{|M_{0,z-1} \cup N_{0,z-1}| + \frac{|M_{0,u} \cup N_{0,u}|}{r}} + \frac{|M_{0,u} \cap N_{0,u}|}{r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|} \end{aligned}$$

which implies

$$\begin{aligned} &\lim_{t\to\infty} \frac{|M_{0,t} \cap N_{0,t}|}{|M_{0,t} \cup N_{0,t}|} \\ &= \lim_{r\to\infty} \frac{|M_{0,z-1} \cap N_{0,z-1}|}{|M_{0,z-1} \cup N_{0,z-1}| + \frac{|M_{0,u} \cup N_{0,u}|}{r}} + \frac{|M_{0,u} \cap N_{0,u}|}{r|M_{0,z-1} \cup N_{0,z-1}| + |M_{0,u} \cup N_{0,u}|} \\ &= \frac{|M_{0,z-1} \cap N_{0,z-1}|}{|M_{0,z-1} \cup N_{0,z-1}|}. \end{aligned}$$

Therefore we give the following definition.

Definition 5. *For two cyclic unary regular languages $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,\ldots,m}$ and $N = \{\{c_j + dk\}_{k\in\mathbb{N}}\}_{j=1,\ldots,n}$, we define the Jaccard coefficient $JC_{M,N}$ of M and N by*

$$JC_{M,N} = \frac{|M_{0,z-1} \cap N_{0,z-1}|}{|M_{0,z-1} \cup N_{0,z-1}|},$$

where z is the smallest common multiple of b and d.

3. A SIMILARITY MEASURE FOR CYCLIC UNARY REGULAR LANGUAGES

Let us see that the measure JC does not depend on the representation of the CURLs.

Theorem 4. *Let $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m}$ and $\overline{M} = \{\{a'_i + b'k\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,m'}$ be two descriptions of the regular set R, and let $N = \{\{c_i + dk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,n}$ and $\overline{N} = \{\{c'_i + d'k\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,n'}$ be two descriptions of the regular set S. Then*

$$JC_{M,N} = JC_{\overline{M},\overline{N}}.$$

Proof. Let z be the lowest common multiple of b and d, and u the lowest common multiple of b, b', d, d'. Then $u = tz$ for some $t \in \mathbb{N}^+$. Then

$$|M_{0,u-1} \cap N_{0,u-1}| = t|M_{0,z-1} \cap N_{0,z-1}| \text{ and } |M_{0,u-1} \cup N_{0,u-1}| = t|M_{0,z-1} \cup N_{0,z-1}|$$

and therefore

$$JC_{M,N} = \frac{|M_{0,u-1} \cap N_{0,u-1}|}{|M_{0,u-1} \cup N_{0,u-1}|}.$$

Analogously,

$$JC_{\overline{M},\overline{N}} = \frac{|\overline{M}_{0,u-1} \cap \overline{N}_{0,u-1}|}{|\overline{M}_{0,u-1} \cup \overline{N}_{0,u-1}|}.$$

Now the equality $JC_{M,N} = JC_{\overline{M},\overline{N}}$ follows because we have $M_{0,u-1} = \overline{M}_{0,u-1}$ and $N_{0,u-1} = \overline{N}_{0,u-1}$ since the same languages R and S are described. □

We now determine $JC_{M,N}$ for two CURLs $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,\ldots,m}$ and $N = \{\{c_j + dk\}_{k\in\mathbb{N}}\}_{j=1,\ldots,n}$. Let us assume that there are q pairs (i, j) such that $\{a_i + bk\} \cap \{c_j + dk\} \neq \emptyset$.

Let $g = \dfrac{d}{gcd(b,d)}$ and $h = \dfrac{b}{gcd(b,d)}$. We mention the following fact.

If the two successions M and N have a non-empty intersection, then

$$\{a_i, a_i + b, a + 2b, \ldots a_i + (g-1)b\} \cap \{c_j, c_j + d, c_j + 2d, \ldots, c_j + (h-1)d\}$$

consists only of one element.

(Assume that the intersection contains at least two elements x and y. Without loss of generality let $x < y$. Then

$$x = a + x'b = c + x''d \text{ and } y = a + x'b + y'b = c + x''d + y''d$$

which gives $y'b = y''d = p$. Since b and d are divisors of p, we have $p \geq z$. Thus $y > z$ in contrast to the choice of y.)

We construct the sets M' and N' as in the proof of Theorem 2 and show that $U = M'_{0,z-1} \cap N'_{0,z-1}$ contains exactly q elements. By the fact given above, U has at most q elements, since we have only q pairs of intersecting successions of M and N. However, if the intersection of two pairs are equal, then $a_{i_1} + v_1 b = c_{j_1} + v'_1 c = a_{i_2} + v_2 b = c_{j_2} + v'_2 c$, which gives by Lemma 2 that $a_{i_1} = a_{i_2}$ and $c_{j_1} = c_{j_2}$, i.e., the two pairs coincide.

Furthermore, $M'_{0,z-1} \cup N'_{0,z-1}$ contains $mg+nh-q$ elements because we have $mg+nh$ successions and q elements are counted twice. Hence

$$JC_{M,N} = JC_{M',N'} = \frac{q}{mg + nh - q}. \tag{3.8}$$

Obviously, $0 \leq JC_{M,N} \leq 1$ for all CURLs M and N. We now show the denseness of the measure JC.

Theorem 5. *For any rational number $r \in [0, 1]$, there are CURLs M and N such that $JC_{M,N} = r$, i.e., the measure JC is dense.*

Proof. If $r = 0$, we can choose $M = \{0 + 2k\}_{k\in\mathbb{N}}$ and $N = \{1 + 2k\}_{k\in\mathbb{N}}$ and then $JC_{M,N} = 0$.

Let $r \in (0, 1]$ be a rational number, then $r = \dfrac{x}{y}$ for any $x, y \in \mathbb{N}$ with $x \leq y$.

Let $b \in \mathbb{N}$ such that $b > y$, let us define

$$M = \{\{i + bk\}_{k\in\mathbb{N}}\}_{i=1,2,\ldots,x} \text{ and } N = \{\{j + bk\}_{k\in\mathbb{N}}\}_{j=1,2,\ldots,y}$$

Since the greatest common divisor of b and b is b, given $i \in \{1, 2, \ldots, x\}$ and $j \in \{1, 2, \ldots, y\}$, $\{i+bk\}_{k\in\mathbb{N}}$ and $\{j+bk\}_{k\in\mathbb{N}}$ have a non-empty intersection if and only if $i - j = 0$. Therefore, there are x pairs (i, j) such that $i - j = 0$.

Therefore, by the equation 3.8 taking into consideration that $g = h = 1$, we have

$$JC_{M,N} = \frac{q}{xg + yh - q} = \frac{x}{x + y - x} = \frac{x}{y}$$

Now the statement follows immediately. □

Analogously, we can consider the Sørensen coefficient as the limit (for $t \to \infty$) of the Sørensen coefficients of the initial parts $M_{0,t}$ and $N_{0,t}$. This leads to the following definition.

Definition 6. *For two cyclic unary regular languages $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,\ldots,m}$ and $N = \{\{c_j + dk\}_{k\in\mathbb{N}}\}_{j=1,\ldots,n}$, we define the Sørensen coefficient $SC_{M,N}$ of M and N by*

$$JC_{M,N} = \frac{2 \cdot |M_{0,z-1} \cap N_{0,z-1}|}{|M_{0,z-1}| + |N_{0,z-1}|},$$

where z is the smallest common multiple of b and d.

Moreover, using the same arguments as above, we show that this definition is independent of the representation and that, for two cyclic unary regular languages $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,\dots,m}$ and $N = \{\{c_j + dk\}_{k\in\mathbb{N}}\}_{j=1,\dots,n}$,

$$SC_{M,N} = SC_{M',N'} = \frac{2q}{mg + nh}, \tag{3.9}$$

where q is the number of pairs (i, j) such that $\{a_i + bk\} \cap \{c_j + dk\} \neq \emptyset$, $g = \dfrac{d}{gcd(b, d)}$ and $h = \dfrac{b}{gcd(b, d)}$.

Theorem 6. *For any rational number $r \in [0, 1]$, there are CURLs M and N such that $SC_{M,N} = r$, i.e., the measure SC is dense.*

Proof. Any rational number $r \in [0, 1]$ can be given in the form $r = \dfrac{2x}{x + y}$ with $x \leq y$ (since $r = \dfrac{x}{b} = \dfrac{2x}{2b}$ for some $x \leq b$ and then $2b = x + y$ for some $y \geq x$). Now the sets given in the proof of Theorem 5 and the considerations in that proof show the statement. □

3.3.2 Comparing the Measure URLS with the Jaccard and Sørensen Coefficients

Now, given two CURLs M and N, let us compare the similarity measure $URLS_{M,N}$ with the Jaccard coefficient $JC_{M,N}$ and the Sørensen coefficient $SC_{M,N}$ that has been defined in the previous subsection.

Theorem 7. *$URLS_{M,N} \geq SC_{M,N} \geq JC_{M,N}$ for any CURLs M and N.*

Proof. Let $M = \{\{a_i + bk\}_{k\in\mathbb{N}}\}_{i=1,\dots,m}$ and $N = \{\{c_j + dk\}_{k\in\mathbb{N}}\}_{j=1,\dots,n}$. Moreover, let us assume that there are q pairs (i, j) such that $\{a_i + bk\} \cap \{c_j + dk\} \neq \emptyset$, and let $g = \dfrac{d}{gcd(b, d)}$ and $h = \dfrac{b}{gcd(b, d)}$.

Obviously, $(mg - nh)^2 = (mg)^2 - 2mgnh + (nh)^2 \geq 0$ which implies

$$(mg + nh)^2 = (mg)^2 + 2mgnh + (nh)^2 \geq 4mgnh$$

or equivalently

$$\frac{q(mg + nh)}{2mgnh} \geq \frac{2q}{mg + nh},$$

i.e., $URLS_{M,N} \geq SC_{M,N}$ by (3.4), (3.6) and (3.9).

Furthermore, $mg \geq q$ and $nh \geq q$. Therefore $mg + nh - 2q \geq 0$. By multiplication with q and adding $mgq + nhq$, we get $2q(mg + nh) - 2q^2 \geq q(mg + nh)$ or equivalently

$$\frac{2q}{mg + nh} \geq \frac{q}{mg + nh - q},$$

i.e., $SC_{M,N} \geq JC_{M,N}$ by (3.8) and (3.8). □

Corollary 2. *Let M and N be two CURLs. Then $URLS_{M,N} = SC_{M,N} = JC_{M,N}$ if and only if $M = N$.*

Proof. Let us suppose $M = N$. By Lemma 5, we have $M = N$ if and only if $URLO_{M,N} = URLO_{N,M} = 1$. Since $URLO_{M,N} = \frac{q}{mg}$ and $URLO_{M,N} = \frac{q}{nh}$, we have $M = N$ if and only if $q = mg = nh$.

Since $q = mg = nh$ if and only if $qmg = (mg)^2$ and $qnh = (nh)^2$, we have that

$$(mg)^2 + (nh)^2 = qmg + qnh.$$

Moreover,

$$(mg)^2 + (nh)^2 = qmg + qnh$$

if and only if

$$q((mg)^2 + (nh)^2) - q^2(mg + nh) = 0$$

if and only if

$$q((mg)^2 + (nh)^2) - q^2(mg + nh) + 2qnhmg = 2qnhmg$$

if and only if

$$q(mg + nh)(mg + nh - q) = 2qnhmg$$

if and only if

$$URLS_{M,N} = \frac{q(mg + nh)}{2nhmg} = \frac{q}{mg + nh - q} = JC_{M,N}.$$

The statement follows by Theorem 7 □

Theorem 8. *Let $M_1, M_2, \ldots M_i \ldots$ be an infinite sequence of CURLS. Then the following three statements are equivalent*

i) $\lim_{i\to\infty} URLS_{M_i,N} = 1$,

ii) $\lim_{i\to\infty} SC_{M_i,N} = 1$,

iii) $\lim_{i\to\infty} JC_{M_i,N} = 1$.

Proof. iii) implies i). Assume that iii) holds. Thus, for any real number $\varepsilon \geq 0$, there is a natural number n such that $1 - JC_{M_i,N} \leq \varepsilon$ for $i \geq n$. Then by Theorem 7, $1 - URLS_{M_i,N} \leq \varepsilon$ for $i \geq n$. Therefore i) holds.

ii) implies i) and iii) implies ii) can be shown analogously.

i) implies iii). Assume that i) holds. Then for any $\varepsilon \geq 0$, there is a natural number n such that $1 - URLS_{M_i,N} < \varepsilon$ for $i \geq n$. If M_i and N contain m and n successions, respectively, we get

$$1 - \frac{q(mg+nh)}{2mgnh} = 1 - \big(\frac{qmg}{2mgnh} + \frac{qnh}{2mgnh}\big) = 1 - \frac{q}{2nh} - \frac{q}{2mg} < \varepsilon.$$

Thus

$$2 - \frac{q}{nh} - \frac{q}{mg} < 2\varepsilon \text{ and } (1 - \frac{q}{nh}) + (1 - \frac{q}{mg}) < 2\varepsilon.$$

Consequently,

$$1 - \frac{q}{nh} < 2\varepsilon \text{ and } 1 - \frac{q}{mg} < 2\varepsilon,$$

or, equivalently,

$$mg - q < 2\varepsilon mg \text{ and } nh - q < 2\varepsilon nh. \tag{3.10}$$

Now we obtain

$$\begin{aligned} 1 - JC_{M_i,N} &= 1 - \frac{q}{mg+nh-q} = \frac{mg+nh-2q}{mg+nh-q} = \frac{(mg-q)+(nh-q)}{mg+nh-q} \\ &< \frac{2\varepsilon mg + 2\varepsilon nh}{mg+nh-q} = 2\varepsilon \frac{mg+nh}{mg+nh-q} \text{ (by (3.10))} \\ &= 2\varepsilon \frac{1}{1 - \frac{q}{mg+nh}} \\ &< 4\varepsilon \qquad \text{(because } q \leq mg, q \leq nh, \text{ i. e., } \frac{q}{mg+nh} \leq \frac{1}{2}). \end{aligned}$$

Thus iii) is valid, too.

By a combination of the shown implications the assertion follows. □

3.4 A Dissimilarity Measure for CURLs

In this section, we will define a dissimilarity measure for CURLs by using the similarity measure that was defined in the previous section. That will be done in the same way as the Jaccard distance is defined by using the Jaccard coefficient.

Definition 7. *Let $n \in \mathbb{N}^+$. Let M and N be two CURLs. We define the dissimilarity measure between M and N, that we will call $URLD_{M,N}$ (for Unary Regular Languages Dissimilarity), as*

$$URLD_{M,N} = 1 - URLS_{M,N}$$

where $URLS_{M,N}$ is the similarity measure between M and N.

Then, we can say that the dissimilarity measure between CURLs is the proportion of strings that are not shared by such languages.

Given two CURLs M and N, $0 \leq URLO_{M,N} \leq 1$, we have $0 \leq URLS_{M,N} \leq 1$. Then, $0 \leq URLD_{M,N} \leq 1$ as in the Jaccard distance case.

In contrast to the Jaccard distance, the dissimilarity measure is not a metric distance since the triangle inequality is not satisfied. That can be proved by using the following counterexample: if M is the set of the odd numbers, N is the set of the even numbers and L is the set of the natural numbers, then

$$1 = URLD_{M,N} > URLD_{M,L} + URLD_{L,N} = 0.$$

However, the dissimilarity measure for CURLs is a semimetric distance, i.e a function d satisfying $d(x, y) \geq 0$, $d(x, y) = 0$ if and only if $x = y$, and $d(x, y) = d(y, x)$. Let us see that given two CURLs M and N, $URLD_{M,N}$ satisfies all the conditions to be a semimetric:

a. $URLD_{M,N} \geq 0$ has been proved in the previous section.

b. Let us see that $URLD_{M,N} = 0$ if and only if $M = N$.
   First we have to show that $URLD_{M,M} = 0$. $URLD_{M,M} = 0$ holds if and only if $URLS_{M,M} = 1$. Since $URLS_{M,M} = URLO_{M,M} = \frac{1}{m} \sum_{\substack{A \sqsubseteq M \\ B \sqsubseteq M}} ISO_{A,B}$, $A \cap B \neq \emptyset$ if and only if $A = B$ (by Lemma 2) and $ISO_{A,A} = 1$ for any $A \sqsubseteq M$, we have $URLS_{M,M} = 1$.

   Let us suppose that $URLD_{M,N} = 0$ for some $M \neq N$. Without loss of generality, let us assume that $M \not\subseteq N$, then $URLO_{M,N} < 1$ (by Lemma 5). Therefore, $URLS_{M,N} < 1$ and it implies $URLD_{M,N} \neq 0$. This is a contradiction, because we supposed $URLD_{M,N} = 0$. So, if $URLD_{M,N} = 0$, then $M = N$.

c. Let us see $URLD_{M,N} = URLD_{N,M}$. We have

$$URLS_{M,N} = \frac{URLO_{M,N} + URLO_{N,M}}{2} = \frac{URLO_{N,M} + URLO_{M,N}}{2} = URLS_{N,M}$$

3.5 Discussion

In this work, we have proposed a similarity measure for CURLs by modifying the Jaccard similarity coefficient and the Sørensen coefficient. Moreover, we have defined a dissimilarity measure for CURLs by using that similarity measure.

Moreover we can also use the similarity and dissimilarity measures defined in this work for non-cyclic URLs. In that case, we consider the infinite set of strings that is generated by the loop of its respective MUDFA, since its initial word contributes to the language with only a finite number of strings, and we follow the same strategy of the cyclic case.

For two URLs (cyclic or non-cyclic) M and N, $URLD_{M,N} = 0$ if and only if $M = N$ (the identity of indiscernibles) is not always satisfied, as can be seen from the following counterexample: If $M = \{1, 4 + 2n\}_{n\in\mathbb{N}}$ and $N = \{2n\}_{n\in\mathbb{N}}$, then $URLD_{M,N} = 0$ and $M \neq N$. Thus, the dissimilarity measure for URLs is a symmetric distance and not a semimetric distance.

As a possible application of the proposed measure we can consider grammatical inference and retrieval theory. Evolutionary computation is an example of optimization technique where the search needs to be informed by a measure that compares individuals with a target. Inferring a CURL would mean just that, and this could be done with $URLS$, JC or SC. Considering the best individual in each of the generations computed by an evolutionary algorithm we would obtained a sequence of CURLs, in the form required by Theorem 19. If the algorithm performs well, this sequence would eventually converge to N with respect to some similarity (our measure, Sørensen and Jaccard coefficient), then it tends to N with respect to the two other similarities, too. We believe that a tendency can be seen easier by the use of our measure since it is greater than the two other ones, and therefore it approaches to 1 earlier. Thus we think that the new defined measure $URLS$ is more appropriate in these circumstances, i.e., URLS could be used as an indicator of convergence, outperfoming JC and SC.

Finally we mention that there are some proposals of distances $d(R, S)$ of two (unary) regular sets R and S, however, the corresponding similarities $1 - d(R, S)$ are not of

interest for us, since the principle mentioned in the Introduction (sets are more similar if the have more elements in common) is not satisfied by them, and we wanted to have a similarity measures for CURLs which follows this intuitive idea.

As examples we mention the Bodnarchuk distance for arbitrary languages, the Baire distance for unary regular languages, the Hamming distance and the information distance for cyclic unary regular languages.

The Bodnarchuk distance $BD(R, S)$ of two sets R and S of non-empty words is defined as

$$d(R,S) = \begin{cases} 0 & \text{if } R = S \\ \dfrac{1}{\min\{|w| \mid w \in (R \setminus S) \cup (S \setminus R)\}} & \text{if } R \neq S \end{cases}$$

(see (39)). Thus the distance is the inverse of the length of the shortest word which gives a difference of the two languages. It is easy to see that $BD(A, B) = 1$ holds for

$$A = \{a^{101n+i} \mid n \geq 0, i \in \{1, 3, 4, 5, \ldots 101\}\} \text{ and } B = \{a^{101n+i} \mid n \geq 0, 2 \leq i \leq 101\},$$

i.e., their distance is maximal, and thus the similarity should be small, but these sets have 99% of their elements in common, which intuitively gives similarity.

The Baire metric of two infinite sequences $r = a_1 a_2 \ldots$ and $s = b_1 b_2 \ldots$ over some set is defined as

$$d(r,s) = \begin{cases} 0 & \text{if } r = s \\ \dfrac{1}{2^{\min\{n \mid a_n \neq b_n\}}} & \text{if } r \neq s \end{cases}$$

(see (63)). A unary regular set R of words can be represented as infinite sequences $r = a_1 a_2 \ldots$ over $\{0, 1\}$ where $a_n = 1$ if and only if $a^n \in R$. As in the the case of the Bodnarchuk metric the sets A and B given above have a relatively large distance and a large similarity, which contradicts the intuition.

In the case of CURLs, the sequences r and s can be given in the form $r = u^\omega$ and $s = v^\omega$ where u and v have the same length, i.e., they are infinite powers of some finite sequences of the same length. Then we can define the scaled Hamming distance of r and s as the number of positions where u and v differ and divided it by the length of v (by the division we ensure that the value belongs to the unit interval). However, now the sets

$$A' = \{a^{100n+1} \mid n \geq 0\} \text{ and } B' = \{a^{100n+2} \mid n \geq 0\}$$

or equivalently, $u = 10^{99}$ and $v = 010^{98}$ have a small distance $\frac{1}{50}$, but no similarity because they have no common element.

Essentially, the same holds for the information distance, which is given by the length of the minimal program (in the sense of Kolmogorov complexity) which transforms u to v (see (9))

Chapter 4

Low Disruption Transformations on Cyclic Automata

The results presented in this section are part of (25; 71) in which I am first author[1].

As we said before, the most common operations that are used in evolutionary systems where the genotypes are sequences of symbols are the edit operations of substitution, deletion, and insertion of a symbol over a word. However, there is a lack of studies concerning the disruption that such genetic operations produce in the genotype. We study such a disruption in the case of the genotypes proposed in this thesis. In this chapter, we extend the edit operations by introducing two new operations (partial copy and partial elimination) inspired by biological gene duplication. We define a disruption measure for an operation over a word by using the similarity measure defined in the previous chapter and prove that whereas the traditional edit operations are disruptive, partial copy and partial elimination are non-disruptive. Moreover, we show that the application of only edit operations does not generate (with low disruption) all the words over a binary alphabet, but this can indeed be done by combining partial copy and partial elimination with the substitution operation.

4.1 Introduction

Edit operations of substitution, deletion, and insertion of a symbol over a word have been extensively studied in literature and have been applied to many different kinds

[1] in (25), authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments, and in (71), the rest of coauthors gave hints, suggestions, and comments

of problems. These are biologically inspired operations that are also known as point mutation operations (20; 23).

They have been applied to the problem of transforming a word of finite length into another word. Moreover, this case has been studied expanding the set of edit operations. For example, in (86), the set of edit operations is extended to include the squashing and expansion operations. Whereas in the squashing operation two (or more) contiguous symbols of the first word can be transformed into a single symbol of the second word, in the expansion operation a single symbol in the first word may be expanded into two or more contiguous symbols of the second word. In (87; 88), the edit operations together with the straightforward transposition of adjacent symbols are used in pattern recognition. The theory of error-correcting codes of variable lengths treats errors that can be modeled as substitutions, insertions or deletions of symbols ((28; 57)). In (6), two novel operations, called node fusion and edge fusion, are introduced and are used join with the tree edit operations to compare two RNA secondary structures coded in the form of trees. Papers in which edit operations are used to compare sequences of symbols can be seen in (80; 106).

Furthermore, there are many studies that endeavor to explain a number of bioinspired evolutionary processes using edit operations. In (23), the concept of an evolutionary system is introduced. This is a language generating device inspired by the evolution of cell populations, and it is based on edit operations and string divisions. The purpose of this system is to model some properties of evolving cell communities at the syntactical level. In (20), a computational device called network of evolutionary processors is proposed. It is based on evolutionary rules and communication within a network. Such evolutionary rules are substitution, deletion, and insertion rules. The generative power of evolutionary networks and many other properties have been widely analyzed, (4; 5; 19; 27; 66; 67; 78). There have been several studies of molecular evolution models that incorporate base substitutions, insertions, and deletions ((74; 103)). In (113), transposition and gene duplication are used join with the edit operations in gene regulation studies.

However, to our knowledge, there are not many studies that analyze the disruptive effects of the edit operations. Since non-random search methods benefits from a low disruption in the application of operations to refine solutions, an analysis of how

disruptive these operations are, and the proposal of new low disruptive operations is necessary.

In this section, such a study of disruption of the edit operations when they are applied over the genotypes proposed in this thesis is done. As we said before, we use CUDFAs as genotypes and the corresponding accepted sets of words as phenotypes. Moreover, we showed that the automata of this type can be described by words over $\{0,1\}$, where the zeros represent the non-accepting states of the automaton, and the ones represent the accepting states of the automaton. Thus we can use the edit operations to obtain changes of the genotypes which model the evolution. Obviously, by biological reasons, the changes cannot be too drastic. Since any change of the genotype results in a change of the phenotype, i. e., in a change of the corresponding regular sets, we can measure the size of the change on the phenotype side. We use the similarity measure between regular sets over unary alphabets, which was introduced in (24) and has been presented in the previous chapter, to define the disruption given by operations applied to genotypes. Intuitively, the disruption of an operation O with respect to a word w is a pair (a,b) with $a,b \in \mathbb{R}$, where a is the portion of words that are accepted by w and are not accepted by $O(w)$ and b is the portion of words that are accepted by $O(w)$ and are not accepted by w.

Now we are interested in the genotypes which can be obtained by iterated applications of edit operations where each application is accompanied by low disruption. We determine the set of all such words which can be generated from a given word. The result shows that not all words can be obtained.

However, if we use in addition two new bioinspired operations which have no disruption, more precisely disruption $(0,0)$, with respect to all words, then starting from any $w \in \{0,1\}^+$, we can obtain all the words $v \in \{0,1\}^+$ that accept a non-empty language where each step has low disruption. The proposed non-disruptive operations have been inspired by gene duplication, an important genetic mechanism that plays an important role in evolution (85; 116). Considering the binary word as a genome, duplication simply adds redundant information (in our case, to $w \in \{0,1\}^+$), keeping the associated phenotype (the language accepted by w) unchanged. The genomic portion gained after gene duplication provides a substrate for coding new functions (proteins, in biology) by future alterations: substitutions, additions, deletions, or even being totally or par-

tially copied/eliminated again. In particular, partial copy/elimination may introduce significant differences in the genome, but keeping the fitting level of the phenotype.

4.2 Definitions

We first define some operations over CUDFAs which are inspired by substitutions, insertions, deletions and copying of molecules which occur in the evolution of biological systems.

Throughout this section, $V = \{0, 1\}$ and h is the mapping $V \to V$ with $h(1) = 0$ and $h(0) = 1$.

For any natural numbers $m, p > 0$, we set

$$T(m, p) = \{w \mid w = (x_1 x_2 \dots x_m)^p,\ x_i \in V \text{ for } 1 \leq i \leq m\}.$$

Definition 8. *For any natural numbers $n, m, p > 0$, j with $0 \leq j \leq n$, i with $1 \leq i \leq n$, $q > 1$, and $y \in V$, we define*

- *the addition operation $A_{j,y} : V^n \longrightarrow V^{n+1}$ as*

$$A_{j,y}(x_1 x_2 \dots x_n) = x_1 x_2 \dots x_j y x_{j+1} \dots x_n,$$

- *the partial copy operation $PC_p : T(m, p) \longrightarrow T(m, p + 1)$ as*

$$PC_p((x_1 x_2 \dots x_m)^p) = (x_1 x_2 \dots x_m)^{p+1},$$

- *the elimination operation $E_i : V^n \longrightarrow V^{n-1}$ as*

$$E_i(x_1 x_2 \dots x_n) = x_1 x_2 \dots x_{i-1} x_{i+1} \dots x_n,$$

- *the partial elimination operation $PE_q : T(m, q) \longrightarrow T(m, q - 1)$ as*

$$PE_q((x_1 x_2 \dots x_m)^q) = (x_1 x_2 \dots x_m)^{q-1},$$

- *the substitution operation $S_i : V^n \longrightarrow V^n$ as*

$$S_i(x_1 x_2 \dots x_n) = x_1 x_2 \dots h(x_i) \dots x_n.$$

Let $\mathcal{A}$, $\mathcal{E}$, $\mathcal{S}$, $\mathcal{PC}$, and $\mathcal{PE}$, be the sets of all addition, elimination, substitution, partial copy, and partial elimination operations, respectively. The operations in $\mathcal{A}$, $\mathcal{E}$ and $\mathcal{S}$ are called the edit operations.

In the sequel we assume $|w| \geq 2$ if we apply the elimination operation since otherwise we yield the empty word which does not correspond to a CUDFA.

In order to define the disruption of an operation transforming a word w into w', which are the genotypes, we need a measure which compare the corresponding phenotypes $L(w)$ and $L(w')$. We use the measure of similarity for CURLs defined in (24) that has been presented in the previous chapter, which is intuitively the portion of the words of $L(w)$ but not in $L(w')$.

Taking into account Definition 3, we can define a notion which measures the change of the phenotype $L(w)$ to the phenotype $L(w')$ if w' is obtained from w by the application of an operation. We define it by two rational numbers where the first one gives the difference from w to w' and the second that from w' to w. This is analogous to the concepts of Recall and Precision in Information Retrieval. The precision is the fraction of the documents retrieved that are relevant to the user's information needs, while the recall is the fraction of the documents that are relevant to the query and are successfully retrieved.

Definition 9. *Let $w \in V^+$ be a CUDFA and $O \in \mathcal{S} \cup \mathcal{A} \cup \mathcal{E} \cup \mathcal{PC} \cup \mathcal{PE}$ be an operation such that $O(w)$ is defined. We define the disruption $D(O, w)$ of the operation O over w as*

$$D(O, w) = (1 - URLO_{L(w),L(O(w))}, 1 - URLO_{L(O(w)),L(w)}).$$

That is, the disruption of an operation O over w is a pair (a, b) with $a, b \in \mathbb{R}$, where a is the portion of words that are accepted by w and are not accepted by $O(w)$ and b is the portion of words that are accepted by $O(w)$ and are not accepted by w.

If $D(O, w) = (0, 0)$ for a given operation O and for all w, we will say that the operation O is not disruptive.

4.3 Determination of the Disruption of the Operations

In this section, we study the disruption of the operations that have been defined in the previous section.

4. LOW DISRUPTION TRANSFORMATIONS ON CYCLIC AUTOMATA

We start with a lemma which enables us to show the non-disruptiveness of partial copy and partial elimination.

Lemma 6. *For a CUDFA $w \in V^+$ and $n \in \mathbb{N}^+$, $L(w) = L(w^n)$.*

Proof. Let $B(w) = \{a_1, a_2, \ldots, a_m\}$. Then $L(w) = \{a_i + |w|k\}_{k\in\mathbb{N}, 1\leq i\leq m}$. Therefore, we get

$$B(w^n) = \{a_1, a_2, \ldots, a_m, a_1 + |w|, a_2 + |w|, \ldots, a_m + |w|, \ldots$$

$$\ldots, a_1 + (n-1)|w|, a_2 + (n-1)|w|, \ldots, a_m + (n-1)|w|\}.$$

Thus

$$L(w^n) = \bigcup_{j=0}^{n-1} \{(a_i + j|w|) + n|w|k\}_{k\in\mathbb{N}, 1\leq i\leq m}.$$

Obviously, $L(w^n) = L(w)$. □

The next corollaries follow immediately.

Corollary 3. *For any CUDFA $w \in V^+$ and $n, m \in \mathbb{N}^+$, $L(w^n) = L(w^m)$.*

Corollary 4. *For any $p \geq 1$ and $q > 1$, the operations PC_p and PE_q are not disruptive.*

Let us study the disruption of the remaining operations.

Lemma 7. *Let $w \in V^+$ be a CUDFA and i a natural number with $1 \leq i \leq |w|$. If $|w|_1 = m$, then*

i) $D(S_i, w) = (0, \frac{1}{m+1})$ if we mutate a zero into a one,

ii) $D(S_i, w) = (\frac{1}{m}, 0)$ if we mutate a one into a zero.

Proof. Let $B(w) = \{a_1, a_2, \ldots, a_m\}$. Then $L(w) = \{a_j + |w|k\}_{k\in\mathbb{N}, 1\leq j\leq m}$.

i) If we mutate a zero at the position i, then $B(S_i(w)) = B(w) \cup \{i-1\}$ and the CURL represented by $S_i(w)$ is

$$L(S_i(w)) = \{a_j + |w|k\}_{k\in\mathbb{N}, 1\leq j\leq m} \cup \{(i-1) + |w|k\}_{k\in\mathbb{N}}.$$

In this case, since a non-accepting state has been changed into an accepting state in w, a portion of new words has been added to $L(w)$. Since $gcd(|w|, |w|) = |w|$, for $A \sqsubseteq L(w)$ and $B \sqsubseteq L(S_i(w))$, we have

$$ISO_{A,B} = \begin{cases} 1 & \text{if } A \cap B \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

by Definition 1. Since the intersection of $\{a_s + |w|k\}_{k\in\mathbb{N}}$ and $\{a_t + |w|k\}_{k\in\mathbb{N}}$ is non-empty if and only if $t = s$ and $\{a_j + |w|k\}_{k\in\mathbb{N}} \cap \{(i-1) + |w|k\}_{k\in\mathbb{N}} = \emptyset$ for $1 \leq j \leq m$, we get

$$URLO_{L(S_i(w)),L(w)} = \frac{m}{m+1} \text{ and } URLO_{L(w),L(S_i(w))} = 1.$$

and then

$$D(S_i, w) = (0, \frac{1}{m+1}).$$

ii) If we mutate a one at the position i, then $i-1 = a_t$ for some t, $1 \leq t \leq m$. Thus, we obtain $B(S_i(w)) = B(w) \setminus \{a_t\}$ and

$$L(S_i(w)) = \{a_1+|w|k\}_{k\in\mathbb{N}} \cup \cdots \cup \{a_{t-1}+|w|k\}_{k\in\mathbb{N}} \cup \{a_{t+1}+|w|k\}_{k\in\mathbb{N}} \cup \cdots \cup \{a_m+|w|k\}_{k\in\mathbb{N}}.$$

By an analogous argumentation as in i), we obtain

$$URLO_{L(w),L(S_i(w))} = \frac{m-1}{m} \text{ and } URLO_{L(S_i(w)),L(w)} = 1$$

and thus

$$D(S_i, w)) = (\frac{1}{m}, 0).$$

□

Lemma 8. *For any CUDFA $w \in V^+$ with $|w|_1 = m$, any natural number i with $0 \leq i \leq |w|$, and any $y \in V$, the disruption of the operation $A_{i,y}$ is*

$$D(A_{i,y}, w) = (1 - \frac{m+y}{|w|+1}, 1 - \frac{m}{|w|}).$$

Proof. Let $B(w) = \{a_1, a_2, \ldots, a_m\}$, again. Then $L(w) = \{a_j + |w|k\}_{k\in\mathbb{N}, 1\leq j\leq m}$. We first discuss the case that $a_t < i \leq a_{t+1}$ for some t, $1 \leq t \leq m-1$.

Let us assume that $y = 1$. Then

$$B(A_{i,1}(w)) = \{a_1, a_2, \ldots, a_t, i, a_{t+1}+1, a_{t+2}+1, \ldots, a_m+1\}$$

and the CURL represented by $A_{i,1}(w)$ is

$$\begin{aligned} L(A_{i,1}(w)) &= \{a_j + (|w|+1)k\}_{k\in\mathbb{N}, 1\leq j\leq t} \cup \{i + (|w|+1)k\}_{k\in\mathbb{N}} \\ &\quad \cup \{(a_j+1) + (|w|+1)k\}_{k\in\mathbb{N}, t+1\leq j\leq m}. \end{aligned}$$

Due to $gcd(|w|, |w|+1) = 1$, for any $A \sqsubseteq L(w)$ and any $B \sqsubseteq L(A_{i,1}(w))$, we have $A \cap B \neq \emptyset$ by Lemma 1, as well as $ISO_{A,B} = \frac{1}{|w|+1}$ and $ISO_{B,A} = \frac{1}{|w|}$ by Definition 1.

Since we have m successions in $L(w)$ and $m+1$ successions in $L(A_{i,1}(w))$ and therefore $m(m+1)$ different pairs (A, B) with $A \sqsubseteq L(w)$ and $B \sqsubseteq L(A_{i,1}(w))$, we get

$$URLO_{L(w),L(A_{i,y}(w))} = \frac{m+1}{|w|+1} \text{ and } URLO_{L(A_{i,y}(w)),L(w)} = \frac{m}{|w|}$$

by Definition 3. Thus, the disruption of $A_{i,1}$ on w is

$$D(A_{i,1}, w) = (1 - \frac{m+1}{|w|+1}, 1 - \frac{m}{|w|}) = (1 - \frac{m+y}{|w|+1}, 1 - \frac{m}{|w|}).$$

Let us now assume that $y = 0$. Then

$$B(A_{i,0}(w)) = \{a_1, a_2, \ldots, a_t, a_{t+1}+1, a_{t+2}+1, \ldots, a_m+1\}$$

and

$$L(A_{i,0}(w)) = \{a_j + (|w|+1)k\}_{k\in\mathbb{N}, 1\leq j\leq t} \cup \{(a_j+1) + (|w|+1)k\}_{k\in\mathbb{N}, t+1\leq j\leq m}.$$

As above, we get $A \cap B \neq \emptyset$ by Lemma 1, $ISO_{A,B} = \frac{1}{|w|+1}$ and $ISO_{B,A} = \frac{1}{|w|}$ for any $A \sqsubseteq L(w)$ and any $B \sqsubseteq L(A_{i,0}(w))$. This implies $URLO_{L(w),L(A_{i,0}(w))} = \frac{m}{|w|+1}$ and $URLO_{L(A_{i,0}(w)),L(w)} = \frac{m}{|w|}$. Therefore,

$$D(A_{i,0}, w) = (1 - \frac{m}{|w|+1}, 1 - \frac{m}{|w|}) = (1 - \frac{m+y}{|w|+1}, 1 - \frac{m}{|w|}).$$

If $i \leq a_1$ or $a_m < i$, we can prove analogously the statement. □

Lemma 9. *Let $w \in V^+$ be a CUDFA, $|w| \geq 2$, $|w|_1 \geq 1$, i a natural number with $1 \leq i \leq |w|$, and y the i-th letter of w. Then $D(E_i, w) = (1 - \frac{|w|_1 - y}{|w|-1}, 1 - \frac{|w|_1}{|w|})$.*

Proof. Again, let $B(w) = \{a_1, a_2, \ldots, a_m\}$ and $L(w) = \{a_j + |w|k\}_{k\in\mathbb{N}, 1\leq j\leq m}$.

Let us assume that $y = 1$. Then $i - 1 = a_t$ for some t, with $1 \leq t \leq m$, $B(E_i(w)) = B(w) \setminus \{a_t\}$, and the CURL represented by $E_i(w)$ is

$$L(E_i(w)) = \{a_j + (|w|-1)k\}_{k\in\mathbb{N}, 1\leq j\leq t-1} \cup \{(a_j-1) + (|w|-1)k\}_{k\in\mathbb{N}, t+1\leq j\leq m}.$$

Since $gcd(|w|, |w|-1) = 1$, we again have $A \cap B \neq \emptyset$, $ISO_{A,B} = \frac{1}{|w|-1}$, and $ISO_{B,A} = \frac{1}{|w|}$ for any $A \sqsubseteq L(w)$ and any $B \sqsubseteq L(E_i(w))$. Analogously to the preceding proof ($L(w)$ has m successions and $L(E_i(w))$ has $m-1$ succcessions) we get

$$URLO_{L(w),L(E_i(w))} = \frac{m-1}{|w|-1} \text{ and } URLO_{L(E_i(w)),L(w)} = \frac{m}{|w|}$$

which implies

$$D(E_i, w) = (1 - \frac{m-1}{|w|-1}, 1 - \frac{m}{|w|}) = (1 - \frac{m-y}{|w|-1}, 1 - \frac{m}{|w|}).$$

Let us now assume that $y = 0$. First we consider the case that $a_t < i - 1 < a_{t+1}$ for some t, $1 \leq t \leq m-1$. Then we get $B(E_i(w)) = B(w)$ and

$$L(E_i(w)) = \{a_j + (|w|-1)k\}_{k\in\mathbb{N}, 1\leq j\leq t} \cup \{(a_j - 1) + (|w|-1)k\}_{k\in\mathbb{N}, t+1\leq j\leq m}.$$

Again, due to $gcd(|w|, |w|-1) = 1$ we obtain as above

$$URLO_{L(w),L(E_i(w))} = \frac{m}{|w|-1} \text{ and } URLO_{L(E_i(w)),L(w)} = \frac{m}{|w|}$$

and

$$D(E_i, w) = (1 - \frac{m}{|w|-1}, 1 - \frac{m}{|w|}) = (1 - \frac{m-y}{|w|-1}, 1 - \frac{m}{|w|}).$$

The cases $y = 0$ and $i - 1 < a_1$ or $a_m < i - 1$ can be handled analogously. □

Therefore, the edit operations are disruptive operations. Moreover, for an edit operation, the disruption is decreasing as the number of ones in the word is increasing.

4.4 Low Disruptions and Iterated Application of Operations

We now define the central notion of the chapter.

Definition 10. *Let a CUDFA $w \in V^+$, $\mathcal{O} \subseteq \mathcal{S} \cup \mathcal{A} \cup \mathcal{E} \cup \mathcal{PC} \cup \mathcal{PE}$, and a real number λ, $0 < \lambda < 1$, be given.*

i) We say that a word v can be obtained with a disruption less than λ from w using $\mathcal{O}$ if there exist operations $O_1, O_2, \ldots, O_p \in \mathcal{O}$, $p \geq 0$, such that

- $v = O_p(O_{p-1} \ldots (O_2(O_1(w))) \ldots)$ *and*
- $D(O_i, O_{i-1}(\ldots(O_2(O_1(w))\ldots)) < (\lambda, \lambda)$ *for any* $1 \leq i \leq p$.

ii) By $LD(w, \mathcal{O}, \lambda)$ we denote the set of all words v which can be obtained with a disruption strictly less than λ from w using operations from $\mathcal{O}$.

An important branch of the biological community supports the idea that during evolution gradual accumulations of small genetic changes occur resulting in producing small alterations in the phenotype; this permits the individual to stay adapted to the

environment. From this point of view, those words which can be obtained in such a way that in each step a low disruption occurs are the most interesting of the set of all words which can be obtained from w by iterated applications of operations from $\mathcal{O}$ (e.g. (23), (4) and other papers).

In Definition 10, we have made the natural supposition $0 < \lambda < 1$. If $\lambda = 1$, then any sequence of operations is an evolution with disruption at most 1, i.e., we allow all sequences which coincides with the situation studied in previous papers. If $\lambda = 0$, no change of the phenotype is possible, which is not of interest from the biological point of view. By the biological motivation, we are only interested in the case of small λ, for instance $\lambda = \frac{1}{100}$. In the sequel, we require $0 < \lambda \leq \frac{1}{2}$, which is sufficient from the mathematical point of view to guarantee a low disruption.

The aim of this section is the determination of $LD(w, \mathcal{O}, \lambda)$ for some choices of $\mathcal{O}$. We start with two easy examples.

Let $w = 10^n$ for some $n \geq 2$, $0 < \lambda < \frac{1}{2}$ and $\mathcal{O} = \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$. Then by Lemmas 7, 8, and 9, for any operation O from $\mathcal{O}$, we have $D(O, w) = (a, b)$ with $a \geq \frac{1}{2}$ or $b \geq \frac{1}{2}$. Thus, from w no word can be obtained with a disruption at most λ using $\mathcal{O}$. Since we allow that no operation has to be used, we obtain $LD(w, \mathcal{O}, \lambda) = \{w\}$.

Let $w = 0^n$ for some $n \geq 1$, $0 < \lambda < \frac{1}{2}$ and $\mathcal{O} = \mathcal{S} \cup \mathcal{A} \cup \mathcal{E} \cup \mathcal{PC} \cup \mathcal{PE}$. It is easy to see that operations from $\mathcal{S}$ and of the form $A_{i,1}$ applied to w have a disruption at least $\frac{1}{2}$. Moreover, by operations from $\mathcal{PC}$ and $\mathcal{PE}$ we can get all words only consisting of zeros with no disruption (see Corollary 4). Hence $LD(w, \mathcal{O}, \lambda) = \{0^m \mid m \geq 1\}$.

Obviously, the reason that in the first example no operation has low disruption comes from the very small number of ones. If we change this situation, $LD(w, \mathcal{O}, \lambda)$ can be different from $\{w\}$ and can contain infinitely many words, as can be seen from the following theorem.

Theorem 9. *Let $w \in V^+$ be a CUDFA and $0 < \lambda \leq \frac{1}{2}$ such that $\frac{1}{|w|_1 + 1} < \lambda$. Further, let $\mathcal{O} = \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$. Then*

$$LD(w, \mathcal{O}, \lambda) = \{v \mid |v|_0 > 0,\ \frac{1}{|v|_1 + 1} < \lambda\} \cup \{1^m \mid m \geq 1\} \cup \{w\}.$$

Proof. Since $\frac{1}{|w|_1 + 1} < \lambda \leq \frac{1}{2}$, w contains at least one letter 1. Let $|w|_0 = t$.

Let $Z = \{v \mid |v|_0 > 0,\ \frac{1}{|v|_1 + 1} < \lambda\} \cup \{1^m \mid m \geq 1\} \cup \{w\}$. We first show that any word of Z belongs to $LD(w, \mathcal{O}, \lambda)$.

This is obvious for w since no operation has to be applied ($p = 0$ in Definition 10).

Let us now suppose that $v \in Z$ and $|v|_0 = q$ for some $q \geq 1$, i.e., $v \neq 1^m$ for all $m \geq 0$. Moreover, by the definition of Z, we have $\frac{1}{|v|_1 + 1} < \lambda$. Then we consider the following finite sequence of operations:

- By appropriate substitution operations $O_1, O_2, \ldots, O_t \in \mathcal{S}$ we mutate all zeros of w. Therefore, $O_t(O_{t-1} \ldots (O_1(w)) \ldots) = 1^{|w|}$.
- Let $b = ||v| - |w||$.
  - If $|w| \leq |v|$, we choose $O_{t+1}, O_{t+2}, \ldots, O_{t+b} \in \mathcal{A}$ to get
  $$O_{t+b}(\ldots (O_{t+1}(1^{|w|})) \ldots) = 1^{|v|}.$$
  - If $|w| > |v|$, we choose $O_{t+1}, O_{t+2}, \ldots, O_{t+b} \in \mathcal{E}$ to get
  $$O_{t+b}(\ldots (O_{t+1}(1^{|w|})) \ldots) = 1^{|v|}.$$
- By $O_{t+b+1}, O_{t+b+2}, \ldots, O_{t+b+q} \in \mathcal{S}$ we mutate all the positions in which $1^{|v|}$ has a one and v has a zero and get $O_{t+b+q}(\ldots (O_{t+b+1}(1^{|v|})) \ldots) = v$.

Therefore, we have $O_{t+b+q}(O_{t+b+q-1}(\ldots (O_2(O_1(w))) \ldots)) = v$.

It remains to show that the disruption of each operation O_i, $1 \leq i \leq t + b + q$, over the word obtained immediately before is smaller than (λ, λ).

If $1 \leq i \leq t$, then O_i increases the numbers of ones by 1. Thus, for $1 \leq i \leq t$, we have $|O_{i-1}(\ldots (O_2(O_1(w))) \ldots)|_1 > |w|_1$ and hence, by Lemma 7

$$\begin{aligned} D(O_i, O_{i-1}(\ldots (O_2(O_1(w))) \ldots)) &= (0, \frac{1}{|O_{i-1}(\ldots (O_2(O_1(w))) \ldots)|_1 + 1}) \\ &\leq (0, \frac{1}{|w|_1 + 1}) < (\lambda, \lambda). \end{aligned}$$

Let $t + 1 \leq i \leq t + b$. If $|w| \leq |v|$, $O_i \in \mathcal{A}$ adds a one to a word 1^k for some k. Thus O_i can be interpreted as a partial copy. By Corollary 4,

$$D(O_i, O_{i-1}(\ldots (O_2(O_1(w))) \ldots)) = (0, 0) < (\lambda, \lambda).$$

If $|w| > |v|$, $O_i \in \mathcal{E}$ can be interpreted as a partial elimination. By Corollary 4,

$$D(O_i, O_{i-1}(\ldots (O_2(O_1(w))) \ldots)) = (0, 0) < (\lambda, \lambda).$$

For $t+b+1 \leq i \leq t+b+q$, the operation O_i does not change the $|v|_1$ ones of v. Thus $|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1 \geq |v|_1+1$ and hence, by Lemma 7

$$\begin{aligned} D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) &= (\frac{1}{|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1}, 0) \\ &\leq (\frac{1}{|v|_1+1}, 0) < (\lambda, \lambda), \end{aligned}$$

where the last inequality holds by $v \in Z$.

Therefore, for $1 \leq i \leq t+b+q$, we have $D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) < (\lambda, \lambda)$.

If $v = 1^m$ for some $m \geq 1$, then we only consider the operations $O_1, O_2, \dots, O_{t+b}$ from above, which produce 1^m from w.

Thus it is shown that all words $v \in Z$ belong to $LD(w, \mathcal{O}, \lambda)$, i.e., we have $Z \subseteq LD(w, \mathcal{O}, \lambda)$.

In order to prove $Z = LD(w, \mathcal{O}, \lambda)$ it remains to show that $Y = LD(w, \mathcal{O}, \lambda) \setminus Z$ is empty.

We introduce a partial order on V^* by $v_1 \prec v_2$ if and only if $|v_1| < |v_2|$ or the conditions $|v_1| = |v_2|$ and $|v_1|_1 < |v_2|_1$ hold.

If Y is not empty, then there is a minimal word y with respect to $\prec$ in Y. Since $y \notin Z$, $|y|_0 > 0$ and $\frac{1}{|y|_1+1} \geq \lambda$ are valid.

Let $O_1, O_2, \dots, O_p$ be operations from $\mathcal{O}$ such that
– $O_p(\dots O_2(O_1(w))\dots) = y$ and
– $D(O_j, O_{j-1}(\dots O_2(O_1(w))\dots)) < (\lambda, \lambda)$ for $1 \leq j \leq p$.
We consider the step $y = O_p(x)$ where $x = O_{p-1}(\dots O_2(O_1(w))\dots)$. Let $m = |x|_1$.

We discuss the possible cases for O_p.

Case 1. $O_p = A_{i,0}$ for some i. If $x \neq 1^g$ for all $g \geq 1$, then $|x| < |y|$, $|x|_0 > 0$, and $\frac{1}{|x|_1+1} = \frac{1}{|y|_1+1} \geq \lambda$. Therefore $x \in Y$ and $x \prec y$ in contrast to our choice of a minimal y. Therefore $x = 1^g$ for some $g \geq 1$. Then $|x| = |x|_1 = g = m$ and by Lemma 8

$$D(O_p, x) = (1 - \frac{m}{m+1}, 1 - \frac{m}{m}) = (\frac{1}{m+1}, 0) \geq (\lambda, 0)$$

(because $\frac{1}{m+1} = \frac{1}{|y|_1+1} \geq \lambda$ by $y \in Y$), i.e., the last step does not satisfy the requirement for a disruption less than λ.

Case 2. $O_p = A_{i,1}$. Then x satisfies $|x| < |y|$ and $|x|_0 = |y|_0 > 0$ and $\frac{1}{|x|_1+1} = \frac{1}{|y|_1} > \frac{1}{|y|_1+1} \geq \lambda$ which contradicts the minimality of y.

Case 3. $O_p = E_i$ for some i.

If we cancel a letter 1, then $m = |y|_1+1$ and $|x|_0 = |y|_0 \geq 1$ and $|x| \geq |x|_1+1 = m+1$. Since $(m-1)(|x|-m-1) \geq 0$ or equivalently $1 - \frac{m-1}{|x|-1} \geq \frac{1}{m}$, the first component of $D(O_p, x)$ satisfies

$$1 - \frac{m-1}{|x|-1} \geq \frac{1}{m} = \frac{1}{|y|_1+1} \geq \lambda$$

in contrast to the choice of the operations.

If we cancel a zero, then $|x|_0 \geq 2$ and hence $|x| \geq m+2$. Furthermore, $|y|_1 = m$. Since $m(|x|-m-1) \geq 0$ or equivalently $1 - \frac{m}{|x|} \geq \frac{1}{m+1}$, the second component of $D(O_p, x)$ satisfies

$$1 - \frac{m}{|x|} \geq \frac{1}{m+1} = \frac{1}{|y|_1+1} \geq \lambda,$$

which contradicts our assumption again.

Case 4. $O_p = S_i$. By the choice of y, we have to change a one into a zero. Hence $m = |y|_1 + 1$. Moreover, the first component of $D(O_p, x)$ satisfies $\frac{1}{m} = \frac{1}{|y|_1+1} \geq \lambda$. We have a contradiction, again.

Further operations have not to be discussed by the choice of $\mathcal{O}$. Since we got a contradiction in each case, our assumption that Y is non-empty is false. □

From a biological point of view, the tendency of the complexity through the evolution has been a increasing tendency. For that reason, we could think that in order to find a parallelism with biology, it is logical that we have to increase the length of the words. Therefore we give the following corollary.

Corollary 5. *i) Let $w \in V^+$ be a CUDFA and $0 < \lambda \leq \frac{1}{2}$ such that $\frac{1}{|w|_1+1} < \lambda$, and let $\mathcal{O} = \mathcal{S} \cup \mathcal{A}$. Then*

$$LD(w, \mathcal{O}, \lambda) = \{v \mid |w| \leq |v|, |v|_0 > 0, \ \frac{1}{|v|_1+1} < \lambda\} \cup \{1^m \mid m \geq |w|\} \cup \{w\}.$$

ii) Let $w \in V^+$ be a CUDFA and $0 < \lambda \leq \frac{1}{2}$ such that $\frac{1}{|w|_1+1} < \lambda$, and let $\mathcal{O} = \mathcal{S} \cup \mathcal{PC}$. Then

$$LD(w, \mathcal{O}, \lambda) = \{v \mid |w| \leq |v|, |v|_0 > 0, \ \frac{1}{|v|_1+1} < \lambda\} \cup \{1^m \mid m \geq |w|\} \cup \{w\}.$$

Proof. i) For$|w| \leq |v|$, we have used only operations from $\mathcal{S} \cup \mathcal{A}$ in the proof of Theorem 9.

ii) The addition operations used add a 1 to a word only consisting of ones. Hence, there is an operation from $\mathcal{PC}$ which has the same effect.

Let

$$Z' = \{v \mid |w| \leq |v|, |v|_0 > 0, \ \frac{1}{|v|_1 + 1} < \lambda\} \cup \{1^m \mid m \geq |w|\} \cup \{w\}.$$

If $\mathcal{O}_1 \subseteq \mathcal{O}_2$, then $LD(w, \mathcal{O}_1, \lambda) \subseteq LD(w, \mathcal{O}_2, \lambda)$ for all $w \in V^*$ and $0 < \lambda < 1$. All words that belong to $LD(w, \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}, \lambda)$ but not to Z' are shorter than w. Since the operations from $\mathcal{O}$ do not decrease the length, those words cannot be obtained. Hence, $LD(w, \mathcal{O}, \lambda) = Z'$. □

If we allow operations of $\mathcal{PE}$ in addition to those from $\mathcal{S} \cup \mathcal{PC}$, we get a case where all words of interest (i.e., all words describing a CUDFA which accepts a non-empty language) can be obtained with low disruptions from a given word w.

Theorem 10. *Let $w \in V^+$ be a CUDFA with $|w|_1 \geq 1$ and $0 < \lambda \leq \frac{1}{2}$, and let $\mathcal{O} = \mathcal{S} \cup \mathcal{PC} \cup \mathcal{PE}$. Then*

$$LD(w, \mathcal{O}, \lambda) = V^+ \setminus \{0^m \mid m \geq 1\}.$$

Proof. Let $w \in V^+$ be a word with $|w| = m$ and $|w|_1 = r > 0$, and let $v \in V^+$ be a word with $|v| = n$ and $|v|_1 = s > 0$.

For a multiple $y = lcm(m, n)z$, $z \in \mathbb{N}^+$, of the lowest common multiple of m and n, we set $z' = \frac{y}{m}$ and $z'' = \frac{y}{n}$. We choose y sufficiently large, i.e., z sufficient large, such that $\frac{1}{rz'} < \lambda$ and $\frac{1}{sz''} < \lambda$. Then we construct the following finite sequence of operations.

- We choose $O_1, O_2, \ldots, O_{\frac{y}{m}-1} \in \mathcal{PC}$ such that any O_i adds a copy of w. Therefore we obtain $O_{\frac{y}{m}-1}(O_{\frac{y}{m}-2} \ldots (O_1(w)) \ldots) = w^{z'}$.
- Let t be the number of positions in which $w^{z'}$ has a zero and $v^{z''}$ has a one. We choose the operations $O_{\frac{y}{m}}, O_{\frac{y}{m}+1}, \ldots, O_{\frac{y}{m}+t-1} \in \mathcal{S}$, such that those zeros are changed into ones. Let us obtain $\overline{w} = O_{\frac{y}{m}+t-1}(\ldots (O_{\frac{y}{m}}(w^{z'})) \ldots)$.
- Let q be the number of positions in which $\overline{w}$ has a one and $v^{z''}$ has a zero. We choose the operations $O_{\frac{y}{m}+t}, O_{\frac{y}{m}+t+1}, \ldots, O_{\frac{y}{m}+t+q-1} \in \mathcal{S}$ such that those ones are mutated into zeros and obtain

$$v^{z''} = O_{\frac{y}{m}+t+q-1}(\ldots (O_{\frac{y}{m}+t}(\overline{w})) \ldots).$$

- We choose $O_{\frac{y}{m}+t+q}, O_{\frac{y}{m}+t+q+1}, \ldots, O_{\frac{y}{m}+t+q+\frac{y}{n}-2} \in \mathcal{PE}$ such that any operation $O_{\frac{y}{m}+t+q+j}$ cancels one copy of v. Obviously, then

$$v = O_{\frac{y}{m}+t+q+\frac{y}{n}-2}(\ldots (O_{\frac{y}{m}+t+q}(v^{z''})) \ldots).$$

Therefore, $O_{\frac{y}{m}+t+q+\frac{y}{n}-2}(O_{\frac{y}{m}+t+q+\frac{y}{n}-3}(\dots(O_2(O_1(w)))\dots)) = v$.

We now calculate the disruption for any step where we apply one of the previous operations.

Let $1 \leq i \leq \frac{y}{m} - 1$. Then $D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) = (0,0) < (\lambda,\lambda)$ by Corollary 4.

Let $\frac{y}{m} \leq i \leq \frac{y}{m} + t - 1$. Since any operation O_i changes a zero into a one, i.e., we increase the number of ones, and $|w^{z'}|_1 = rz'$, we get $|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1 \geq rz'$ and

$$\begin{aligned} D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) &= (0, \frac{1}{|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1 + 1}) \\ &< (0, \frac{1}{rz'}) < (\lambda,\lambda). \end{aligned}$$

Let $\frac{y}{m} + t \leq i \leq \frac{y}{m} + t + q - 1$. Since O_i changes a one into a zero, but sz'' ones of $\overline{w}$ are not changed, we have $|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1 \geq sz''$ and

$$\begin{aligned} D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) &= (\frac{1}{|O_{i-1}(\dots(O_2(O_1(w)))\dots)|_1}, 0) \\ &\leq (\frac{1}{sz''}, 0) < (\lambda,\lambda). \end{aligned}$$

Let $\frac{y}{m} + t + q \leq i \leq \frac{y}{m} + t + q + \frac{y}{n} - 2$. Then, by Corollary 4

$$D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) = (0,0) < (\lambda,\lambda).$$

Therefore, we have

$$D(O_i, O_{i-1}(\dots(O_2(O_1(w)))\dots)) < (\lambda,\lambda)$$

for $1 \leq i \leq \frac{y}{m} + t + q + \frac{y}{n} - 2$.

It remains to show that we cannot obtain words 0^m with $m \geq 1$. If we assume the contrary, then there is a number k such that there are some operations $O_1, O_2, \dots, O_p \in \mathcal{PC} \cup \mathcal{PE} \cup \mathcal{S}$ with

$$0^k = O_p(O_{p-1}\dots(O_2(O_1(w)))\dots)$$

and

$$D(O_i, O_{i-1}(\dots(O_2(O_1(w))\dots)) < (\lambda,\lambda) \text{ for } 1 \leq i \leq p. \tag{4.1}$$

Without loss of generality we can assume that

$$O_j(O_{j-1}\dots(O_2(O_1(w)))\dots) \notin \{0^m \mid m \geq 1\}$$

holds for $1 \leq j < p$ (otherwise the word $O_j(O_{j-1}\dots(O_2(O_1(w)))\dots) \in \{0^m \mid m \geq 1\}$ is considered instead of 0^k). Thus O_p is a substitution operation which replaces a one by a zero and $O_{p-1}(O_{p-2}\dots(O_2(O_1(w)))\dots)$ contains exactly once the letter 1. Hence, we have $D(O_p, O_{p-1}(O_{p-2}\dots(O_2(O_1(w)))\dots)) = (1,0)$ by Lemma 7 which is a contradiction to (4.1). □

We note that the operations of $\mathcal{PE}$ are not so common in biology as the edit operations and those from $\mathcal{PC}$. Thus we now look for a result where we only use the edit operations together with that of $\mathcal{PC}$.

Theorem 11. *For any word w with $|w|_1 > 0$ and any λ with $0 < \lambda \leq \frac{1}{2}$,*

$$LD(w, \mathcal{PC} \cup \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}, \lambda) = \{v \mid |v|_0 > 0,\ \frac{1}{|v|_1 + 1} < \lambda\} \cup \{1^m \mid m \geq 1\} \cup \{w\}.$$

Proof. Let $|w|_1 = m \geq 1$. Then there is a number $r \in \mathbb{N}^+$ such that $\frac{1}{mr} < \lambda$. Using $r-1$ times operations from $\mathcal{PC}$ which copy w, we get w^r. Obviously, $|w^r|_1 = mr$, and thus $\frac{1}{|w^r|_1} < \lambda$. All the disruptions of these operations are $(0,0)$ by Corollary 4.

Again, let $Z = \{v \mid |v|_0 > 0,\ \frac{1}{|v|_1 + 1} < \lambda\} \cup \{1^m \mid m \geq 1\} \cup \{w\}$. Starting from the word w^r, by Theorem 9, for any word $v \in Z$, we can construct a sequence of operations $O_1, O_2, \dots, O_p$ from $\mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$ such that

- $v = O_p(O_{p-1}\dots(O_2(O_1(w^r)))\dots)$ and
- $D(O_i, O_{i-1}(\dots(O_2(O_1(w))\dots)) < (\lambda, \lambda)$ for any $1 \leq i \leq p$.

This proves $v \in LD(w, \mathcal{PC} \cup \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}, \lambda)$. Therefore $Z \subseteq LD(w, \mathcal{PC} \cup \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}, \lambda)$. As in the part of the proof of Theorem 9, we can show that $LD(w, \mathcal{PC} \cup \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}, \lambda) \setminus Z = \emptyset$ (in the notation of that proof, the operations from $\mathcal{PC}$ cannot be used for O_p by the choice of y). □

In this section, we have proved that the expressive capability of the set of operations $\{\mathcal{S}, \mathcal{PC}, \mathcal{PE}\}$ is higher than the expressive capability of $\{\mathcal{S}, \mathcal{A}, \mathcal{E}\}$, $\{\mathcal{S}, \mathcal{A}\}$, $\{\mathcal{S}, \mathcal{PC}\}$, and $\{\mathcal{S}, \mathcal{A}, \mathcal{E}, \mathcal{PC}\}$ if low disruption is kept. This is because with the set of operations $\{\mathcal{PC}, \mathcal{PE}\}$ any length can be obtained without disruption, and then with the operations $\mathcal{S}$, that have a very low disruption in most cases, we get the correct symbol at each position.

4.5 Comparing the Operations with respect to Disruption

In the preceding section, for two given words w and v and a given set of operations, we have constructed a sequence of operations $O_1, O_2, \ldots, O_t$ such that

$$O_t(O_{t-1}(\ldots(O_2(O_1(w)))\ldots)) = v$$

and, for each operation O_i, $1 \leq i \leq t$, the disruption of O_i over

$$O_{i-1}(O_{i-1}(\ldots(O_1(w))\ldots))$$

is limited by (λ, λ) for some small λ. From an algorithmic point of view, one can be interested in short such sequences. Ignoring the lowness of the disruption, there are some nice algorithms which determine a shortest sequence $O_1, O_2, \ldots, O_t$ of operations from $\mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$ such that $O_t(O_{t-1}(\ldots(O_2(O_1(w)))\ldots)) = v$ (see, e.g., (44)). Obviously, a greedy algorithm, which also considers the disruption, chooses an operation with a disruption as large as possible in each step. In this section, we show that our choices of operations do the converse, i.e., we mostly choose operations with almost minimal disruption on the current intermediate word. Hence, it seems that our choices are not optimal with respect to the length of sequence of operations which transform a word w into a word v with limited disruption in each step.

In order to show the above mentioned aspect of our choices, we compare the operations with respect to disruption.

Definition 11. *Let O and P be two operations over a CUDFA $w \in V^+$ with disruption $D(O, w) = (a, b)$ and $D(P, w) = (c, d)$ with $a, b, c, d \in \mathbb{R}$. We say that O is at most as disruptive as P over w if $a \leq c$ and $b \leq d$, and we write $D(O, w) \leq D(P, w)$. If $a < c$ and $b \leq d$ or $a \leq c$ and $b < d$, then we say O is less disruptive than P over w and write $D(O, w) < D(P, w)$.*

In the sequel, let S, A, and E be arbitrary operations in $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{E}$, respectively.

Lemma 10. *For any CUDFA $w \in V^+$ with $|w| \neq |w|_1 > 1$, we have*

$$D(S, w) \leq D(E, w).$$

Proof. Let us assume that $|w|_1 = m$. Then $|w| \geq m + 1$ by assumption.

By Lemma 9, we have $D(E, w) = (1 - \frac{m - y}{|w| - 1}, 1 - \frac{m}{|w|})$ where $y \in V$ is the eliminated symbol.

Let S be an operation which mutates a zero into a one. According to Lemma 7, we know that $D(S,w) = (0, \frac{1}{m+1})$. It is trivial that $0 \leq 1 - \frac{m-y}{|w|-1}$. Moreover, $\frac{1}{m+1} \leq 1 - \frac{m}{|w|}$ if and only if $|w| \geq m+1$.

Let S be an operation which mutates a one into a zero. Then $D(S,w) = (\frac{1}{m}, 0)$. It is trivial that $0 \leq 1 - \frac{m}{|w|}$. Moreover,

$\frac{1}{m} \leq 1 - \frac{m-y}{|w|+1}$ if and only if $(m-1)|w| \geq m(m-y-1)+1$ if and only if $|w| > m$

(note that $m > 1$).

Therefore the inequality $D(S,w) \leq D(E,w)$ holds for any CUDFA $w \in V^+$ with $|w| \neq m$. □

If, in addition to the suppositions of Lemma 10, we assume that $|w| \geq |w|_1 + 2$, then we get the stronger relation $D(S,w) < D(E,w)$. This follows from the fact that, in this case, $0 < 1 - \frac{m-y}{|w|-1}$ and $0 < 1 - \frac{m}{|w|}$.

Lemma 11. *For any CUDFA $w \in V^+$ with $|w| \geq |w|_1 + 3$ and $|w|_1 > 1$ or with $|w| \geq |w|_1 + 2$ and $|w|_1 > 2$, $D(S,w) < D(A,w)$.*

Proof. Let us assume that $|w|_1 = m$.

By Lemma 8, $D(A,w) = (1 - \frac{m+y}{|w|+1}, 1 - \frac{m}{|w|})$ where $y \in V$ is the added symbol.

Let S be an operation which mutates a zero into a one. According to Lemma 7, we know that $D(S,w) = (0, \frac{1}{m+1})$. It is trivial that $0 < 1 - \frac{m+y}{|w|+1}$. Moreover, $\frac{1}{m+1} \leq 1 - \frac{m}{|w|}$ if and only if $|w| \geq m+1$ which is satisfied by our assumptions concerning $|w|$ and $m = |w|_1$.

Let S be an operation which mutates a one into a zero. Then we have

$$D(S,w) = (\frac{1}{m}, 0).$$

It is trivial that $0 < 1 - \frac{m}{|w|}$. Moreover,

$$\frac{1}{m} \leq 1 - \frac{m+y}{|w|+1} \text{ if and only if } |w| \geq m + \frac{my+1}{m-1},$$

and the latter condition is satisfied by our assumptions concerning $|w|$ and $m = |w|_1$.

□

By Lemmas 10 and 11, to ensure low disruption, substitution operations have to be preferred to addition and elimination operations. This justifies the choices in the proof of Theorem 9.

We now compare elimination and addition operations.

Lemma 12. *For any CUDFA $w \in V^+$ with $|w|_0 \geq 1$ and $|w|_1 \geq 1$, we have*
i) $D(E,w) < D(A,w)$ if the eliminated and the added symbol is a zero, and
ii) $D(A,w) < D(E,w)$ if the eliminated and the added symbol is a one.

Proof. Let $m = |w|_1$. By our assumption, $|w| \geq |w|_1 + 1$ and hence $|w| \geq 2$.

i) Since 0 is the added and eliminated symbol, by Lemmas 8 and 9, we have the disruptions $D(A,w) = (1 - \frac{m}{|w|+1}, 1 - \frac{m}{|w|})$ and $D(E,w) = (1 - \frac{m}{|w|-1}, 1 - \frac{m}{|w|})$. Since $1 - \frac{m}{|w|-1} < 1 - \frac{m}{|w|+1}$ if and only if $1 \leq m$ and $2 \leq |w|$, the inequality of the statement holds.

ii) Since 1 is the added and eliminated symbol, by Lemmas 8 and 9, we have the relations $D(A,w) = (1 - \frac{m+1}{|w|+1}, 1 - \frac{m}{|w|})$ and $D(E,w) = (1 - \frac{m-1}{|w|-1}, 1 - \frac{m}{|w|})$. Since $1 - \frac{m+1}{|w|+1} < 1 - \frac{m-1}{|w|-1}$ if and only if $m < |w|$, the inequality holds. □

The following statements show that it is natural to use partial copies before substitutions and partial eliminations after substitutions in order to ensure small disruptions (see the proof of Theorem 10).

Theorem 12. *For any CUDFA $w \in T(n,p)$ with $|w|_0 \geq 1$, $|w|_1 \geq 1$, $n \geq 1$, and $p > 0$, we have*

- $D(S, PC_p(w)) < D(S,w)$,
- $D(E, PC_p(w)) < D(E,w)$ *if the eliminated symbol is a one and* $D(E, PC_p(w)) > D(E,w)$ *if the eliminated symbol is a zero,*
- $D(A, PC_p(w)) < D(A,w)$ *if the added symbol is a zero and* $D(A, PC_p(w)) > D(A,w)$ *if the added symbol is a one.*

Proof. We know $D(PC_p, w) = (0,0)$. Let us suppose $w = v^p$ for some $v \in V^+$ and $p > 0$, and let us assume $|v| = n$ and $|v|_1 = m$. Then $|w| = np$, $|w|_1 = mp$, $|PC_p(w)| = np + n$ and $|PC_p(w)|_1 = mp + m$. Since $|w|_1 < |PC_p(w)|_1$, it is trivial that $D(S, PC_p(w)) < D(S,w)$ by Lemma 7.

Let the eliminated symbol be one. Since $1 - \frac{mp+m-1}{np+n-1} < 1 - \frac{mp-1}{np-1}$ if and only if $m < n$, the inequality $D(E, PC_p(w)) < D(E, w)$ holds by Lemma 9.

Let the eliminated symbol be zero. Since $1 - \frac{mp+m}{np+n-1} > 1 - \frac{mp}{np-1}$ if and only if $m > 0$, the inequality $D(E, PC_p(w)) > D(E, w)$ holds by Lemma 9.

Let the added symbol be zero. Since $1 - \frac{mp+m}{np+n+1} < 1 - \frac{mp}{np+1}$ if and only if $m > 0$, the inequality $D(A, PC_p(w)) < D(A, w)$ holds by Lemma 8.

Let the added symbol be one. Since $1 - \frac{mp+m+1}{np+n+1} > 1 - \frac{mp+1}{np+1}$ if and only if $m < n$, the inequality $D(A, PC_p(w)) > D(A, w)$ holds by Lemma 8. □

Theorem 13. *For any CUDFA $w \in T(n, p)$ with $|w|_0 \geq 1$, $|w|_1 \geq 1$, $n \geq 1$, and $p > 1$, we have*

- $D(S, PE_p(w)) > D(S, w)$,
- $D(E, PE_p(w)) < D(E, w)$ *if the eliminated symbol is a zero and* $D(E, PE_p(w)) > D(E, w)$ *if the eliminated symbol is a one,*
- $D(A, PE_p(w)) < D(A, w)$ *if the added symbol is a one and* $D(A, PE_p(w)) > D(A, w)$ *if the added symbol is a zero.*

Since the proof of this theorem is similar to the proof of the previous one, it is left to the reader.

4.6 Discussion

In this paper, we started the investigation of iterated applications of some bioinspired operations with the additional requirement that the disruption is (very) small in each step. In one case (Theorem 10) we were able to generate all words which correspond to non-empty regular languages. However, from a biological point of view, the other results are also satisfactory because the genotypes have to contain a lot of information, i.e., the words under consideration have to be long and to contain a sufficiently large number of ones. This means that the assumptions of Theorem 9 are satisfied and all words of biological interest can be obtained by Theorems 9 and 11.

In the literature, one can find nice algorithms to determine the minimal number of edit operations which transform a given word w into another given word v. It remains to search for good algorithms where the additional requirement of low disruption in any

step is satisfied. Note that the sequences proving the existence of such transformations with low disruptions (constructed in the proofs of Theorems 9, 10, and 11) seem to be not optimal by Section 4.5.

Finally, a future research line will be to study whether the results presented in this paper are also satisfied for more complex devices than CUDFAs.

Chapter 5

Generating Primitive Words

In the previous chapter, we have shown that in spite of the edit operations are the most common genetic operations that are used in evolutionary systems, they produce a high disruption in the genotypes defined until now in this thesis (binary words). For that reason, we extended them by introducing two operations inspired by biological gene duplication. Despite reducing such a disruption, these operations do not preserve the minimality of automata, thus individuals with the same complexity can be represented by automata with very different number of states and this seems not to be very logical from a biological point of view.

For that reason, a representation of the genotypes over which the genetic operations do not cause such problems was one of the most important aims of the thesis. In this chapter, two different ways of generating primitive words are presented. The first approach consists of the definition of a set of operations inspired by biological gene duplication that preserve primitivity of words. A large subset of binary primitive words can be obtained by using sequences of these operations as genotypes. Since genetic operations can be applied over these sequences and the minimality of the automata is preserved, these will be the genotypes used to study the complexity during the evolution. The second approach consists in a non-grammatical method that is based on a characterization of the non-primitive words. By using this approach, the application of the genetic operations is not as natural as in the previous approach. In spite of this approach is not used to define the genotypes, it is still interesting as it provides a relation between number theory and primitive words.

5.1 Introduction

Primitive words have been widely studied in the literature. There are a lot of papers on relations of Q_V to other language families such as the families of the Chomsky hierarchy (e.g. in (33) and (93), it has been shown that Q_V is not a deterministic as well as not an unambiguous context-free language, in (32; 94), it has been shown that Q_V is not a regular language, in (54) some connections to regular languages are given), polyslender languages (see (34)) and to some languages and language families related to codes (see e.g. (108)). Also, context-free languages have been proposed consisting of non-primitive words (53; 62). Moreover, there are papers on combinatorial properties of Q_V and subsets of primitive words, (7; 16; 45). Some grammars that generate Q_V have also been proposed (e.g. Kunimochi proposed a monotone grammar for Q_V in (61), and in (35) a Marcus contextual grammar for Q_V is defined).

There are some papers where it was investigated whether the application of homomorphisms to primitive words leads to primitive words in all cases or leads to primitive words with a finite number of exceptions or to non-primitive words in all cases; we refer to (50; 76; 77; 90). In (99), homomorphisms are studied which preserve the property to be a Lyndon word or to be border-free (a word w is a Lyndon word if and only if any non-empty proper suffix of w is greater than w with respect to the lexicographic order; it is border-free if there is no non-empty word which is a proper prefix as well as a proper suffix of w); it is shown that such homomorphisms preserve primitivity, too. Substitutions form another operation which was investigated with respect to preservation of primitivity. There were substitutions of very short subwords in the focus, especially point mutations (deletions, insertions and substitutions of one letter) were studied. We refer to (89) for details. A further study in this direction concerns insertions (see (49; 58)).

As we said before, in this chapter, two different ways of generating primitive words are proposed. The first one is based on applying a sequence of operations preserving primitivity of words and the second one is based on a relation between Q_V and number theory.

5.2 Some Operations Preserving Primitivity of Words

The results presented in this section are part of (26) in which I am first author[1].

As it is shown before, there is only a small number of results concerning the closure of Q_V under operations. Obviously, there is a large variety of operations from which one can expect that Q_V is closed under them (since the portion of primitive words is very high). In this section we consider some operations where essentially, from a given word w, the word ww' is constructed where w' is a modified copy of w or a modified mirror image of w. The modifications are of such a form that the edit distance of w and w' is very small or very large (i.e., it is very near to the length of w).

We have two reasons for this investigation. The first one is of combinatorial nature. Obviously, ww is not primitive for all w. We are interested in conditions for changes of the second copy w to w' such that ww' is primitive for all w. Especially, how many changes or deletions or insertions of letters are necessary and how many such operations are possible. For example, we shall determine all possible transformation where the edit distance of w and w' is at most two and primitivity is preserved.
The second reason comes from the theory of dynamic systems (as we have explained before). In the papers (69; 70) dynamical systems based on regular languages has been proposed. The regular languages are essentially described by primitive words. Since in dynamical systems one needs mutations in order to develop the system, one is interested in devices which describe primitive words and allow mutations. Here the use of operations which preserve primitivity is of interest. Then a primitive word can be given as a sequence of operations; and a mutation is the replacement of one operation by another one or a deletion or insertion of an operation in the sequence. This ensures primitivity of the word obtained from the mutated sequence of operations. Obviously, it is not necessary to generate all primitive words, however, the set of generated primitive words should contain a good approximation of any primitive word where the quality of approximations is determined by the dynamic system (especially its fitness function). We have chosen the operations under which Q_V is closed in such a way that, if the underlying alphabet V consists of two letters, then by the operations we can generate

[1] authors by alphabetical order, the rest of coauthors are my supervisors and gave hints, suggestions, and comments

all primitive words of length ≤ 11 (as can be shown by computer calculations) and a sufficient large amount of primitive words of the length up to twenty.

Thus this section can also be considered as a continuation of the investigations of devices generating only primitive words (see e.g. (31)).

5.2.1 Some Notation and Facts

For a word $w = x_1x_2 \dots x_n \in V^+$ with $x_i \in V$ for $1 \leq i \leq n$, we define the mirror image w^R by $w^R = x_nx_{n-1} \dots x_1$. Given two words $w = x_1x_2 \dots x_n \in V^+$ and $w' = y_1y_2 \dots y_n \in V^+$ with $x_i, y_i \in V$ for $1 \leq i \leq n$, the Hamming distance $d(w, w')$ is defined by $d(w, w') = |\{i \mid x_i \neq y_i\}|$ and the edit distance $ed(w, w')$ of w and w' is the minimal number of changes, deletions and insertions of letters in order to transform w into w'.

Throughout this section we assume that V has at least two elements. If V is understood from the context we omit the index V and write simply Q.

We recall three facts (see (64), (108), (7)) which will be used in the sequel.

Lemma 13. *For any words $v, v' \in V^*$, $vv' \in Q$ if and only if $v'v \in Q$.*

Lemma 14. *For two non-empty words u and v, $uv = vu$ if and only if there is a word z such that $u = z^n$ and $v = z^m$ for some natural numbers n and m.*

Lemma 15. *In a free monoid V^*, the equation $a^mb^n = c^p$, where $a, b, c \in V^*$ and $m, n, p \geq 2$, has only trivial solutions, where a, b and c are powers of some word in V^*.*

Lemma 16. (Fine-Wilf Theorem) *Let $u, v \in V^+$ and $n, m \geq 2$. If u^n and v^m have a common prefix of length at least $|u| + |v| - gcd(|u|, |v|)$, then u and v are powers of the same primitive word.*

The following statement holds trivially.

Lemma 17. *If $w \in Q$, then also $w^R \in Q$.*

Lemmas 13 and 17 can be interpreted as follows: If we apply a cyclic shift or the mirror image to a primitive word, then we obtain a primitive word, again. Thus cyclic shifts and reversal are operations which preserve primitivity.

Lemma 18. *For any $x \in V$, $y \in V$ and $z \in V^*$, if $xz = zy$, then $x = y$.*

Proof. If $z = \lambda$, then $x = y$ immediately. If $z = a_1 a_2 \dots a_n$ with $a_i \in V$ for $1 \leq i \leq n$, then $x = a_1, a_1 = a_2, a_2 = a_3, \dots a_{n-1} = a_n, a_n = y$ and consequently $x = y$. □

In the sequel we shall use the following notation. If $w = w_1 w_2 \dots w_r = z_1 z_2 \dots z_s$ for some words $w_1, \dots w_r, z_1, \dots, z_s \in V^*$ such that $|w_1 w_2 \dots w_i| = |z_1 z_2 \dots z_j|$ for some i and j, we write

$$w_1 w_2 \dots w_i | w_{i+1} w_{i+2} \dots w_r = z_1 z_2 \dots z_j | z_{j+1} z_{j+2} \dots z_s,$$

i.e., by the symbol | we mark a certain position in the word. (Some authors write $(w, w') = (z, z')$ instead of $w|w' = z|z'$.) Mostly, | will mark the middle of a word of even length, or it will be put after the m-th letter if the word has odd length $2m - 1$.

5.2.2 Operations with an Almost Duplication

Obviously, the word ww obtained from w by a duplication leads from any word w to a non-primitive word. In order to obtain primitive words from a primitive word w one has to perform some changes in the second occurrence of w, i.e., one has to consider words of the form ww' where w' differs only slightly from w. In most cases the edit distance of w and w' will be at most 2, and thus ww' can be considered as an almost duplication of w.

We start with the case where we only change some letters to obtain w' from w.

Theorem 14. *i) Let w be a primitive word of some length n and w' an arbitrary word of length n such that the Hamming distance $d(w, w')$ is a power of 2, then ww' is primitive, too.*
ii) If d is not a power of 2, then there are a primitive word w and a word w' with $d(w, w') = d$ such that ww' is not a primitive word.

Proof. i) Obviously, $|ww'|$ is even. Let us suppose $ww' \notin Q$, that is, there exists $p \in \mathbb{N}$, $p > 1$, and $v \in V^+$ of length at least 2 such that $ww' = v^p$.

If p is even, then $w = w' = v^{\frac{p}{2}}$ since $|w| = |w'|$. Thus $d(w, w') = 0$ which contradicts the assumption on the Hamming distance of w and w'.

If p is odd, i.e., $p = 2m + 1$ for some $m \geq 1$, then $|v|$ is even (since otherwise $|v|p = |ww'|$ would be odd). Thus there are words v' and v'' of length $\frac{|v|}{2}$ such that $v = v'v''$. Then we get $w = v^m v' = (v'v'')^m v'$ and $w' = v''v^m = v''(v'v'')^m$.

Then $d(w, w') = (2m + 1)d(v', v'')$. Since $2m + 1$ is an odd number, $d(w, w')$ is not a power of 2 in contrast to our supposition.

ii) Let d be not a power of 2. Then there is an odd number $q > 1$ and a number p such that $d = qp$. Let $q = 2m + 1$ for some $m \geq 1$. We now set $v' = 10^p$, $v'' = 11^p$, $w = (v'v'')^m v'$, and $w' = (v''v')^m v''$. Obviously, w is primitive,

$$d(w, w') = (2m + 1)d(v', v'') = (2m + 1)p = qp = d$$

and $ww' = (v'v'')^{2m+1} \notin Q$. □

By part ii) of the preceding theorem, if w is a primitive word and $d(w, w')$ is not a power of 2, in general, ww' is not a primitive word. However, if we require that the changes occur in special positions it is possible to obtain preservation of primitivity. As an example we give the following operation.

Definition 12. *For any odd natural numbers $n \geq 3$, any alphabet V, and any mapping $h : V \to V$ with $h(a) \neq a$ for all $a \in V$, we define the operation $O_{n,h} : V^n \to V^{2n}$ by*

$$O_{n,h}(x_1x_2 \dots x_n) = x_1x_2 \dots x_nh(x_1)x_2 \dots x_{i-1}h(x_i)x_{i+1} \dots x_{n-1}h(x_n)$$

where $i = \dfrac{n+1}{2}$.

Theorem 15. *For any odd natural number $n \neq 5$, any primitive word q of length n, and any mapping $h : V \to V$ with $h(a) \neq a$ for all $a \in V$, $O_{n,h}(q)$ is a primitive word.*

Proof. Let $w = x_1x_2 \dots x_n$ with $x_j \in V$ for $1 \leq j \leq n$ and $i = \dfrac{n+1}{2}$. Then $O_{n,h}(x_1x_2 \dots x_n)$ has an even length.

Let us suppose that $O_{n,h}(w) \notin Q$, that is, there exist a $p \geq 2$ and $v \in Q$ such that $O_{n,h} = v^p$.

If p is even then

$$v^{\frac{p}{2}} = x_1x_2 \dots x_{n-1}x_n = h(x_1)x_2x_3 \dots x_{i-1}h(x_i)x_{i+1}x_{i+2} \dots x_{n-1}h(x_n).$$

Thus $x_i = h(x_i)$, which is a contradiction.

Thus p is odd, say $p = 2m + 1$ for some $m \geq 1$. As above there are words v, v_1 and v_2 such that $v = v_1v_2$ and $|v_1| = |v_2|$ and

$$x_1 \dots x_{n-1}x_n|h(x_1)x_2 \dots x_{i-1}h(x_i)x_{i+1} \dots x_{n-1}h(x_n) = (v_1v_2)^m v_1|v_2(v_1v_2)^m.$$

Since v_1 starts with x_1 (first occurrence) and ends with x_n (last occurrence in the first part), $v_1 = x_1v_1'x_n$ and analogously, $v_2 = h(x_1)v_2'h(x_n)$. Therefore we have that $O_{n,h}(w)$ has the form

$$(x_1v_1'x_nh(x_1)v_2'h(x_n))^m x_1v_1'x_n|h(x_1)v_2'h(x_n)(x_1v_1'x_nh(x_1)v_2'h(x_n))^m.$$

Since the letters x_i and x_n do not occur in the first occurrence of v, by the definition of $O_{n,h}$, the last letter of the first occurrence of v_1 (in the first part of the word) and last letter of the first occurrence of v_2 in the second part coincide, i.e., $x_n = h(x_n)$ which is a contradiction. □

The supposition $n \geq 5$ in Theorem 15 is necessary since the statement does not hold for $n = 3$ as can be seen from the following example. Let $q = aba \in Q$. Then $O_{3,h}(q) = ababab = (ab)^3 \notin Q$.

We now discuss some operations where the edit distance of w to w' is at most 2 and at least one deletion or one insertion is performed to obtain w'; more precisely, we consider

(a) the deletion of an arbitrary letter,

(b) the deletion of an arbitrary letter and the change of an arbitrary remaining letter,

(c) the insertion of an arbitrary letter,

(d) the insertion of an arbitrary letter and the change of an arbitrary letter of w.

We now give the formal definition of these operations.

Definition 13. *For any natural numbers n, i, j, i' with $1 \leq i \leq n$, $0 \leq i' \leq n$, $1 \leq j \leq n$ and $i \neq j$, letters $x, y, z \in V$ with $x \neq y$, and a word $w = x_1x_2 \dots x_n$, $x_i \in V$, of length n, we define the following operations*

$$D_{n,i},\ D_{n,i,j,x,y} : V^n \to V^{2n-1} \text{ and } I_{n,i',z},\ I_{n,i',z,j,x,y} : V^n \to V^{2n+1}$$

by

$$\begin{aligned}
D_{n,i}(x_1x_2 \dots x_n) &= x_1x_2 \dots x_nx_1x_2 \dots x_{i-1}x_{i+1}x_{i+2} \dots x_n, \\
D_{n,i,j,x,y}(x_1 \dots x_n) &= \begin{cases} x_1 \dots x_nx_1 \dots x_{i-1}x_{i+1} \dots x_{j-1}yx_{j+1} \dots x_n & x_j = x, i < j \\ x_1 \dots x_nx_1 \dots x_{j-1}yx_{j+1} \dots x_{i-1}x_{i+1} \dots x_n & x_j = x, i > j \\ \textit{undefined} & \textit{otherwise} \end{cases}, \\
I_{n,i',z}(x_1x_2 \dots x_n) &= x_1x_2 \dots x_nx_1x_2 \dots x_{i'}zx_{i'+1}x_{i'+2} \dots x_n, \\
I_{n,i',z,j,x,y}(x_1 \dots x_n) &= \begin{cases} x_1 \dots x_nx_1 \dots x_{i'}zx_{i'+1} \dots x_{j-1}yx_{j+1} \dots x_n & x_j = x, i' < j \\ x_1 \dots x_nx_1 \dots x_{j-1}yx_{j+1} \dots x_{i'}zx_{i'+1} \dots x_n & x_j = x, i' > j \\ \textit{undefined} & \textit{otherwise} \end{cases}.
\end{aligned}$$

Theorem 16. *If $n \geq 2$, $1 \leq i \leq n$, and q is a primitive word of length n, then $D_{n,i}(q) \in Q$ also holds.*

Proof. Let $q = uav$ for some $u, v \in V^*$ and $a \in V$. If $|q| = 1$, i.e., $q = a$, then $D_{n,i}(q) = a \in Q$.

If $|q| \geq 2$, then $D_{|q|,|u|+1}(q) = uavuv$. Let us suppose that $uavuv \notin Q$. Then $(vu)^2a \notin Q$ by Lemma 13. Let $(vu)^2a = z^m$ for some $z \in V^+$ and some $m \geq 2$. Thus $(vu)^2$ is a common prefix of $(vu)^2$ and z^m. Since

$$|uv| + |z| = |uv| + \frac{2|uv|+1}{m} \leq |vu| + \frac{2|uv|+1}{2} < 2|vu| + 1,$$

we have

$$|vu| + |z| - gcd(|vu|, |z|) < (2|vu| + 1) - 1 = 2|vu| = |(vu)^2|.$$

By Lemma 16, we obtain $(vu)^2 = w^k$ and $(vu)^2a = z^m = w^l$ for some $w \in V^+$ and some numbers k and l. Obviously, $w = a$. Hence u and v are powers of a and thus q is a power of a. This contradicts the primitivity of q. □

Theorem 17. *If $w \in V^+$ and $D_{n,i,j,x,y}(w)$ is defined, then $D_{n,i,j,x,y}(w) \in Q$ holds.*

Proof. We first discuss $D_{n,n,j,x,y}$. Let $w = x_1x_2 \dots x_n$. Then

$$D_{n,n,j,x,y}(w) = x_1x_2 \dots x_{j-1}xx_{j+1}x_{j+2} \dots x_nx_1x_2 \dots x_{j-1}yx_{j+1}x_{j+2} \dots x_{n-1}.$$

Let us assume that $D_{n,n,j,x,y}(w) \notin Q$. Then there is a word $v \in V^+$ such that $D_{n,n,j,x,y}(w) = v^p$ for some $p \geq 2$. Since $D_{n,n,j,x,y}(w)$ has odd length, p and the length of v are odd numbers. Let $p = 2m + 1$ for some $m \geq 1$. Thus there are words $v_1 \in V^+$ and $v_2 \in V^+$ such that $v = x_1v_1v_2$, $k - 1 = |v_1| = |v_2|$ and

$$x_1x_2 \dots x_{j-1}xx_{j+1}x_{j+2} \dots x_n|x_1x_2 \dots x_{j-1}yx_{j+1}x_{j+2} \dots x_{n-1} = v^mx_1v_1|v_2v^m.$$

Then $|v| = 2k - 1$. We set $s = 2k - 1$. We distinguish some cases.

Case 1. Let $1 \leq j \leq k - 1$. Then by definition of $D_{n,n,j,x,y}$,

$$x_1v_1 = x_1x_2 \dots x_{j-1}xx_{j+1} \dots x_{k-1}x_k = z_1xz_2x_k$$

and

$$v_2 = x_1x_2 \dots x_{j-1}yx_{j+1} \dots x_{k-1} = z_1yz_2.$$

Thus, we get,

$$v = z_1xz_2x_kz_1yz_2.$$

If $m \geq 2$, the first part of the word is

$$z_1xz_2x_kz_1yz_2z_1xz_2x_kz_1yz_2v^{m-2}z_1xz_2x_k \tag{5.1}$$

and that of the second part is

$$z_1yz_2z_1xz_2x_kz_1yz_2z_1xz_2x_kz_1yz_2v^{m-2} \tag{5.2}$$

and these two words differ in the $(|z_1xz_2x_kz_1yz_2z_1|+1)$-st letter, which contradicts the definition of $D_{n,n,j,x,y}$. If $m=1$, the first and second part are

$$z_1xz_2x_kz_1yz_2z_1xz_2x_k \text{ and } z_1yz_2z_1xz_2x_kz_1yz_2,$$

respectively, and we get a contradiction as above.

Case 2. Let $j=k$. Then the k-th letter in the second part is y. On the other hand, it is x_1 since there starts the word v. Thus $x_1=y$. This gives

$$x_1v_1=x_1x_2\dots x_{k-1}x_k=yzx,\ v_2=x_1x_2\dots x_{k-1}=yz \text{ and } v=yzxyz$$

with $z=x_2x_3\dots x_{k-1}$. Then the first and second part are

$$yzxyzyzxyzv^{m-2}yzx \text{ and } yzyzxyzyzxyzv^{m-2},$$

respectively. We obtain $zx=yz$ by looking on the words starting in the position $|z|+3$. Thus by Lemma 18, $x=y$ in contrast to the definition of $D_{n,n,j,x,y}$.

Case 3. Let $k+1\le j\le 2k-1$. Then $v=x_1v_1v_2'xv_2''$. Moreover, $|v_2'|=j-k-1$. Furthermore, y stands in the j-th position of $v_2'xv_2''x_1v_1$, i.e., $x_1v_1=x_1v_1'yv_1''$ with $|v_1'|=j-k-1$. Therefore $v=x_1v_1'yv_1''v_2'xv_1''$ and $|v_1'|=|v_2'|$ and $|v_1''|=|v_2''|$. Then we get for the second part

$$x_1v_1'yv_1''v_2'yv_2''x_1v_1'yv_1''v_2'xv_2''x_{2s-1}x_{2s}\dots x_n$$

by the definition of $D_{n,n,j,x,y}$ and from the form

$$v_2'xv_2''x_1v_1'yv_1''v_2'xv_2''v^{m-1}$$

given by our assumption.

Considering the words which start in the position $(|x_1v_1'yv_1''|+1)$ and in the position $(|x_1v_1'yv_1''v_2'y|+1)$, respectively, we see that $v_1'=v_2'=z$ and $v_1''=v_2''=z'$. Looking on the subwords starting in the first position and in the position $|v_1'|+2$, we get $x_1z=zx$ and $yz'=xx_1$. By Lemma 18, $x_1=x$ and $y=x_1$, which contradicts $x\neq y$.

Case 4. Let $j=hs+q$ for some $h\ge 1$ and $1\le q\le k-1$. Then $x_j=x$ is the q-th letter of v. Thus $v=v_1'xv_1''v_2$ with $|v_1'|=q-1$.

We now compute the position of y in v. Since the second part starts with v_2 of length $k-1$ and $hs+q=k-1+(h-1)s+s+q-(k-1)=k_1+(h-1)s+k+q$,

y is the $(k+q)$-th letter of v. Therefore $v = v_1' x v_1'' v_2' y v_2''$ with $|v_1'| = |v_2'|$. Moreover, $|v_1''| = |v_2''| + 1$. Now we get easily the same situation as in Case 1; thus we get (5.1) and (5.2) and a difference in the $(|z_1|+1)$-st position.

Case 5. Let $j = hs + k$ for some $h \geq 1$. Then x is the k-th letter of v. We compute the position of y in v. Since the second part starts with v_2 of length $k-1$ and we have $hs + k = k - 1 + hs + k - (k-1)$, y is the first letter of v. Therefore we get $v = yzxyz$ as in Case 2, which leads to a contradiction.

Case 6. Let $j = hs + q$ for some $h \geq 1$ and $k+1 \leq q \leq 2k-1$. Then $x_j = x$ is the q-th letter of v. Thus $v = x_1 v_1 v_2' x v_2''$ with $|x_1 v_1 v_2'| = q - 1 \geq k$. Furthermore, $|v_2''| = 2k - 1 - q$.

We now compute the position of y in v. Since the second part starts with v_2 of length $k-1$ and $hs + q = k - 1 + hs + q - (k-1)$, y is the $(q-k+1)$-st letter of v. Therefore

$$v = x_1 v_1' y v_1'' v_2' x v_2'' \text{ with } |x_1 v_1'| = q - k.$$

Therefore $|v_1''| = k - (q - k + 1) = 2k - 1 - q$. Hence $|v_1''| = |v_2''|$ and consequently we have $|v_1'| = |v_2'|$. Therefore we have exactly the situation of Case 3, which leads to contradiction.

Let us now consider $i = 1$, i.e., the operation $D_{n,1,j,x,y}$. By the first part of this proof

$$D_{n,n,n-j+1,x,y}(w^R) = x_n x_{n-1} \dots x_1 x_n x_{n-1} \dots x_{j+1} y x_{j-1} x_{j-2} \dots x_2 \in Q,$$

by Lemma 17,

$$x_2 x_3 \dots x_{j-1} y x_{j+1} x_{j+2} \dots x_n x_1 x_2 \dots x_n \in Q,$$

and by Lemma 13

$$x_1 x_2 \dots x_n x_2 x_3 \dots x_{j-1} y x_{j+1} x_{j+2} \dots x_n = D_{n,1,j,x,y}(w) \in Q.$$

We now consider the case $j < i$. We set

$$\overline{w} = x_{i+1} x_{i+2} \dots x_n x_1 x_2 \dots x_i.$$

Moreover, let $x_j = x$. By the first part of this proof we get

$$D_{n,n,n-i+j,x,y}(\overline{w}) = x_{i+1} \dots x_n x_1 \dots x_i x_{i+1} \dots x_n x_1 \dots x_{j-1} y x_{j+1} \dots x_{i-1} \in Q.$$

Hence, by Lemma 13

$$x_1 \dots x_i x_{i+1} \dots x_n x_1 \dots x_{j-1} y x_{j+1} \dots x_{i-1} x_{i+1} \dots x_n = D_{n,i,j,x,y}(w) \in Q.$$

If $i < j$ we can prove that $D_{n,i,j,x,y}(w) \in Q$ analogously to the case $j < i$ using $D_{n,1,j,x,y}$ instead of $D_{n,n,j,x,y}$. □

Theorem 18. *If q is a primitive word of length n, $0 \leq i \leq n$ and $z \in V$, then $I_{n,i,z}(q) \in Q$.*

Proof. Let q be a primitive word of length n and $a \in V$. Let u be the prefix of q of length i and $q = uv$. Then $I_{n,i,a}(w) = uvuav$. If $uvuav \notin Q$, we can derive a contradiction as in the proof of Theorem 16. □

Theorem 19. *If $q \in Q$ and $I_{n,i,z,j,x,y}(q)$ is defined, then $I_{n,i,z,j,x,y}(q) \in Q$.*

Proof. Let $w = x_1x_2 \dots x_{j-1}xx_{j+1}x_{j+2} \dots x_n$. Then

$$I_{n,n,a,j,x,y} = x_1x_2 \dots x_nx_1x_2 \dots x_{j-1}yx_{j+1}x_{j+2} \dots x_na.$$

If we assume that $I_{n,n,a,j,x,y}$ is not in Q, then

$$x_1 \dots x_{j-1}yx_{j+1} \dots x_nax_1 \dots x_n = D_{n+1,n+1,j,y,x}(x_1 \dots x_{j-1}yx_{j+1} \dots x_na) \notin Q,$$

which is a contradiction to Theorem 17. The general case can be obtained using Lemmas 13 and 17. □

Let ww' be given with $ed(w, w') = 1$. Then w' is obtained by a change (i.e., $d(w, w') = 1 = 2^0$), either by a deletion or by an insertion. By the Theorems 14, 16 and 18, ww' is in Q provided that $w \in Q$. If $ed(w, w') = 2$ we have again $ww' \in Q$ if two changes, or a deletion and a change, or a change and an insertion are performed (by Theorems 14, 17 and 19). In the remaining cases, in general, primitivity is not preserved. Performing two deletions we can get a non-primitive word, as can be seen from $w = 110^p1$ which results in 110^p1110^p1 and gives $110^p110^p = (110^p)^2 \notin Q$ if we delete the first and last letters of the second copy (note that the statement holds for any length $n \geq 4$ since it holds for any $p \geq 1$). The same holds for two insertions; e.g. the duplication 10^p10^p of $w = 10^p \in Q$ yields $10^p110^p1 = (10^p1)^2$ by inserting a 1 before and after the second copy of 10^p. Furthermore, if we cancel the first letter and insert a 1 before the last 0 in the duplication 110110 of $110 \in Q$, we get $110110 = (110)^2 \notin Q$, again.

Therefore we have a complete picture for the case that the edit distance is at most 2.

5.2.3 Concatenation of an Almost Mirror Image

In this section, again, we consider words of the form ww'. However, instead of an almost copy w' of w we choose w' in such a way that the Hamming/edit distance of w' and the mirror image w^R is small.

We start with the remark that, in general, for a primitive word w, ww^R is not a primitive word. For example, if we concatenate 100110 and its mirror image, we obtain $100110011001 = (1001)^3 \notin Q$. Moreover, if we delete one letter in w^R, the obtained operation is not primitivity preserving as can be seen from the following counterexample. Let $w = 01001$. Since $w^R = 10010$, $ww^R = 0100110010$. If we delete the first letter of w^R, then we obtain $010010010 = (010)^3 \notin Q$.

We define formally three operations which are analogous to some with a small Hamming distance $d(w, w')$ considered in the preceding section.

Definition 14. *For any natural numbers n, i, j with $1 \leq i \leq n$ and $2 \leq j \leq n$, all letters $x, y \in V$ with $x \neq y$, and a word $w = x_1x_2 \dots x_n$, $x_i \in V$, of length n, we define the following operations*

$$M_{n,i,x,y} : V^n \to V^{2n}, \text{ and } M'_{n,j,x,y} : V^n \to V^{2n-1}$$

by

$$M_{n,i,x,y}(x_1x_2 \dots x_n) = \begin{cases} x_1x_2 \dots x_nx_nx_{n-1} \dots x_{i+1}yx_{i-1}x_{i-2} \dots x_1 & x_i = x \\ \textit{undefined} & \textit{otherwise} \end{cases},$$

$$M'_{n,j,x,y}(x_1x_2 \dots x_n) = \begin{cases} x_1x_2 \dots x_nx_nx_{n-1} \dots x_{j+1}yx_{j-1}x_{j-2} \dots x_2 & x_j = x \\ \textit{undefined} & \textit{otherwise} \end{cases}.$$

For all odd natural numbers n, all mappings $h : V \to V$ with $h(a) \neq a$ for all $a \in V$, and all words $w = x_1x_2 \dots x_n$, $x_i \in V$, of length n, we define the operation $O'_{n,h} : V^n \to V^{2n}$ by

$$O'_{n,h}(x_1x_2 \dots x_n) = x_1x_2 \dots x_nh(x_n)x_{n-1} \dots x_{i+1}h(x_i)x_{i-1}x_{i-2} \dots x_2h(x_1)$$

where $i = \dfrac{n+1}{2}$.

Theorem 20. *If $w \in Q$ such that $M_{n,i,x,y}(w)$ is defined, then $M_{n,i,x,y}(w) \in Q$ also holds.*

Proof. Let $w = x_1x_2 \dots x_n$. Then

$$w' = M_{n,i,x,y}(w) = x_1x_2 \dots x_{i-1}xx_{i+1}x_{i+2}x_nx_nx_{n-1} \dots x_{i+1}yx_{i-1}x_{i-2} \dots x_1.$$

Let $u_1 = x_1 \dots x_{i-1}$ and $u_2 = x_{i+1} \dots x_n$. Then

$$w = u_1xu_2 \text{ and } w' = u_1xu_2u_2^Ryu_1^R.$$

Let us assume that $w' \notin Q$. Then $w' = v^p$ for some $p \geq 2$ and some word $v \in V^+$.

If p is even, then

$$v^{\frac{p}{2}} = u_1xu_2 = u_2^Ryu_1^R. \tag{5.3}$$

We now count the number of occurrences of x and get

$$|u_1xu_2|_x = |u_1|_x + 1 + |u_2|_x$$

and

$$|u_2^Ryu_1^R|_x = |u_2^R|_x + |u_1^R|_x = |u_2|_x + |u_1|_x.$$

Thus

$$|u_1xu_2|_x \neq |u_2^Ryu_1^R|_x$$

which contradicts (5.3).

If p is odd, say $p = 2m + 1$ for some $m \geq 1$, then $w' = v^mv_1v_2v^m$ where $v = v_1v_2$ and $|v_1| = |v_2|$. If $i > |v|$, then by the construction of w' we get $w' = vzv^R$ with $z = v^{m-1}v_1v_2v^{m-1}$ and by our assumption ($w' = v^{2m+1}$) we have $w' = vzv$. Therefore $v = v^R$. Now let $i \leq |v|$. Then v_1 and v_2 and v satisfy the following conditions:

- $v_2 = v_1^R$ (by construction),
- $v_2^R = ((v_1)^R)^R = v_1$,
- $v^R = (v_1v_2)^R = v_2^Rv_1^R = v_1v_2 = v$.

Hence in both cases we have $v = v^R$. This implies

$$(w')^R = (v^p)^R = (v^R)^p = v^p = w'.$$

Thus $x = y$ in contrast to our supposition. □

Theorem 21. *If $w \in Q$ such that $M'_{n,j,x,y}(w)$ is defined, then $M'_{n,i,x,y}(w) \in Q$ also holds.*

Proof. Let $w = x_1x_2 \dots x_n$. Then

$$M'_{n,j,x,y}(w) = x_1x_2 \dots x_nx_nx_{n-1} \dots x_{j+1}yx_{j-1}x_{j-2} \dots x_2.$$

Obviously, $|M'_{n,j,x,y}(w)| = 2n - 1$, i.e., the length of $M'_{n,j,x,y}(w)$ is odd.

If $M'_{n,j,x,y}(w)$ is not a primitive word, then $M'_{n,j,x,y}(w) = v^p$ for some primitive word v of odd length and some odd number p with $p \geq 3$, say $p = 2m + 1$ with $m \geq 1$. As in the preceding proofs we get $v = v_1x_nv_2$ with

$$M'_{n,j,x,y}(w) = v^mv_1x_n|v_2v^m = (v_1x_nv_2)^mv_1x_n|v_2(v_1x_nv_2)^m$$

and $|v_1| = |v_2|$. Let $|v_1| = q$, i.e., $|v| = 2q + 1$.

Let $2 \leq j \leq 2q+1$. Then considering the $(m+1)$-st factor v of $M'_{n,j,x,y}(w)$, we obtain $v = v_1x_n|v_2 = x_1x_2 \dots x_qx_n|x_nx_q \dots x_2$. Let $z = x_2x_3 \dots x_qx_n$. Then $v = x_1zz^R$. On the other hand, for $2 \leq j \leq 2q + 1$, by definition of $M'_{n,j,x,y}(w)$, $M'_{n,j,x,y}(w)$ does not end with $(zz^R)^R = zz^R$. Thus we have a contradiction to the fact that $M'_{n,j,x,y}(w)$ ends with v and therefore with zz^R.

Let $j = 2q + 2$. Then the $(2q + 2)$-nd letter of w is x. Moreover, the $(2q + 2)$-nd letter of w is the first letter of the second factor v of $M'_{n,j,x,y}(w)$ which is x_1. Hence $x = x_1$. On the other hand, by the definition of $M'_{n,j,x,y}(w)$, counting from the end, y is the $(2q + 1)$-st letter of $M'_{n,j,x,y}(w)$, which means that y is the first letter of the last factor v of $M_{n,j,x,y}(w)$. Thus $y = x_1$. Hence we get $x = y$ in contradiction to the definition of $M'_{n,j,x,y}$.

Let $2q + 3 \leq j \leq n$. Then we can derive a contradiction by analogous argument (if $m(2q + 1) < j \leq n$, then we get $v = v_1x_nv_2 = x_1zz^R$ by considering the first factor v_1 and the last factor v_2 in $M'_{n,j,x,y}(w)$). □

Finally in this section, we give a result which is the counterpart of Theorem 15. We omit the proof which can be given in analogy to the proof of Theorem 15.

Theorem 22. *For any odd natural number $n \geq 5$, any primitive word q of length n, and any mapping $h : V \to V$ with $h(a) \neq a$ for all $a \in V$, $O'_{n,h}(q)$ is a primitive word.* □

5.2.4 Further Operations with an Almost Duplication of Length

First in this section, we discuss the situation where w' in ww' is obtained from w or w^R by large changes.

If we change all letters in the second part, primitivity is not preserved in general. For instance, if we take the primitive word $w = 100110$, then by changing all letters of w we obtain $100110011001 = (1001)^3 \notin Q$; and starting with the primitive word $w = 10010110$ and changing all letters of w^R we get $1001011010010110 = w^2 \notin Q$.

Theorem 23. *Let w and w' be two words of length n such that $n - d(w, w')$ is a power of 2, then ww' is a primitive word.*

Proof. The proof can be given in a way analogous to the proof of Theorem 14. □

The following definition and result are analogies to $D_{n,n}$ and Theorem 16.

Definition 15. *For any natural numbers n, any natural number i with $1 \leq i \leq n$, and any homomorphism $h : V^* \to V^*$ with $h(a) \neq a$ and $h(h(a)) = a$ for all $a \in V$, we define the operation $D_{n,h} : V^n \to V^{2n-1}$ by*

$$D_{n,h}(x_1x_2 \dots x_n) = x_1x_2 \dots x_nh(x_1x_2 \dots \dots x_{n-1}).$$

Theorem 24. *For any natural numbers n, any natural number i with $1 \leq i \leq n$, any homomorphism $h : V^* \to V^*$ with $h(a) \neq a$ and $h(h(a)) = a$ for all $a \in V$, and any $w \in Q$, $D_{n,h}(w) \in Q$ also holds.*

Proof. Let $w = x_1x_2 \dots x_n$ with $x_j \in V$ for $1 \leq j \leq n$. Then

$$D_{n,h}(x_1x_2 \dots x_n) = x_1x_2 \dots x_nh(x_1 \dots x_{n-1})$$

has an odd length.
Let us suppose that $D_{n,h}(w) \notin Q$, that is, there exist a $p \geq 2$ and $v \in Q$ such that $D_{n,h}(w) = v^p$.

Thus p is odd, say $p = 2m + 1$ for some $m \geq 1$. As above there are words v, v_1 and v_2 such that $v = v_1x_nv_2$ and

$$x_1x_2 \dots x_n|h(x_1 \dots x_{n-1}) = (v_1x_nv_2)^mv_1x_n|v_2(v_1x_nv_2)^m.$$

Since $|(v_1x_nv_2)^mv_1| = |v_2(v_1x_nv_2)^m|$, $|v_1| = |v_2|$.

Furthermore $v_2 = h(v_1)$ by definition of $D_{n,h}$. Therefore we get

$$x_1x_2 \dots x_n|h(x_1 \dots x_{n-1}) = (v_1x_nh(v_1))^mv_1x_n|h(v_1)(v_1x_nh(v_1))^m.$$

Thus $(h(v_1)h(x_n)v_1)^mh(v_1) = h(v_1)(v_1x_nh(v_1))^m$, that is,

$$(h(v_1)h(x_n)v_1)^mh(v_1) = (h(v_1)v_1x_n)^mh(v_1).$$

Hence $h(x_n)v_1 = v_1x_n$. Therefore, by Lemma 18, $h(x_n) = x_n$ in contrast to the supposition concerning h. □

By Theorem 23, from a word $w \in Q$ we obtain a primitive word ww' where w' is constructed from w by changing all letters except one letter. This result does not hold for the mirror image, i.e., if one concatenates w with its mirror image and changes all letters of the mirror image besides one letter, in general, one does not obtain a primitive word. For example, if $w = 11100 \in Q$ and $i = 3$, then we obtain

$$1110011100 = (11100)^2 \notin Q.$$

However, if we restrict to special positions, then the corresponding statement is true, as shown by the following two theorems.

Definition 16. *For any natural numbers n and i with $1 \leq i \leq n$ and any homomorphism $h : V^* \to V^*$ with $h(a) \neq a$ for all $a \in V$, we define the operations*

$$M_{n,1,h},\ M_{n,n,h} : V^n \to V^{2n}$$

by

$$\begin{aligned} M_{n,1,h}(x_1x_2\dots x_n) &= x_1x_2\dots x_nx_nh(x_{n-1}x_{n-2}\dots x_1), \\ M_{n,n,h}(x_1x_2\dots x_n) &= x_1x_2\dots x_nh(x_nx_{n-1}\dots x_2)x_1. \end{aligned}$$

Theorem 25. *For any $n \geq 2$, any homomorphism $h : V^* \to V^*$ with $h(a) \neq a$ for all $a \in V$ and any $w \in Q$, $M_{n,1,h}(w) \in Q$ also holds.*

Proof. Let $w = x_1x_2\dots x_n$, where $x_i \in V$. Then

$$M_{n,1,h}(w) = x_1x_2\dots x_{n-1}x_nx_nh(x_{n-1}x_{n-2}\dots x_1)$$

has an even length.

Let us suppose that $M_{n,1,h}(w) \notin Q$, that is, there exists a $p \in \mathbb{N}$, $p > 1$, and $v \in Q$ such that $x_1x_2\dots x_{n-1}x_nx_nh(x_{n-1}x_{n-2}\dots x_1) = v^p$.

If p is even and $p > 2$, then $v^{\frac{p}{2}} = w$ and $\dfrac{p}{2} \geq 2$, which contradicts $w \in Q$. If $p = 2$, then $x_1x_2\dots x_{n-1}x_nx_nh(x_{n-1}x_{n-2}\dots x_1) = v^2$, that is,

$$v = x_1x_2\dots x_{n-1}x_n = x_nh(x_{n-1}x_{n-2}\dots x_1).$$

Then $x_n = x_1$ and $x_n = h(x_1)$, which is a contradiction.

If p is odd, then $p = 2m + 1$ for some $m \geq 1$ and $v = x_1v'x_nv''$ with $v', v'' \in V^*$, which can be shown as in the proof of Theorem 17. Since

$$x_1\dots x_{n-1}x_n|x_nh(x_{n-1}x_{n-2}\dots x_1) = v^mx_1v'|x_nv''v^m, |v'| = |v''|.$$

We distinguish the cases $v' \neq \lambda \neq v''$ and $v' = \lambda = v''$.

Supposing $v' \neq \lambda \neq v''$ and $v' = y_1\dots y_r$ and $v'' = z_1\dots z_r$. Then

$$x_1 \dots x_{n-1}x_n | x_n h(x_{n-1}x_{n-2} \dots x_1)$$
$$= (x_1y_1 \dots y_r x_n z_1 \dots z_r)^m x_1 y_1 \dots y_r | x_n z_1 \dots z_r (x_1 y_1 \dots y_r x_n z_1 \dots z_r)^m$$

and $y_r = x_n$. Since $h(x_1y_1y_2 \dots y_r) = z_r z_{r-1} \dots z_1 x_n$ by construction, $h(y_r) = x_n$, which contradicts $y_r = x_n$

Supposing $v' = \lambda = v''$, we get

$$x_1 \dots x_{n-1}x_n | x_n h(x_{n-1}x_{n-2} \dots x_1) = (x_1x_n)^m x_1 | x_n (x_1x_n)^m,$$

which implies $x_n = x_1$ and $x_n = h(x_1)$, so it is a contradiction.

Therefore $Q_{n,1,h}(w) \in Q$. □

Theorem 26. *For any $n \geq 2$, any homomorphism $h : V^* \to V^*$ with $h(a) \neq a$ for all $a \in V$ and any $w \in Q$, $M_{n,n,h}(w) \in Q$ also holds.*

Proof. Let $w = x_1x_2 \dots x_n$. Let us assume that $M_{n,n,h}(w) \notin Q$. Then there is a word $v \in V^+$and a natural number $p \geq 2$ such that $M_{n,n,h}(w) = v^p$.

If $p = 2$, then $v = x_1x_2 \dots x_n = h(x_nx_{n-1} \dots x_2)x_1$. Hence $x_1 = h(x_n)$ and $x_n = x_1$, which is a contradiction. If $p > 2$ and even, then $w = v^{\frac{p}{2}} \in Q$ in contrast to our supposition.

If p is odd, i.e., $p = 2m + 1$ for some $m \geq 1$, then there are words v_1 and v_2 with $v = v_1v_2$, $|v_1| = |v_2|$ and

$$x_1x_2 \dots x_n | h(x_nx_{n-1} \dots x_2)x_1 = v^m v_1 | v_2 v^m.$$

Let $k = |v_1|$. Then

$$v_1 = x_1x_2 \dots x_k \quad \text{and} \quad v_2 = h(x_kx_{k-1} \dots x_2)x_1$$

by definition of $M_{n,n,h}$. Thus $x_{2k+1} = x_1$ and $h(x_{2k+1}) = x_1$ in contrast to the required property of h that $h(a) \neq a$ for all $a \in V$. □

We now define an operation where we duplicate the word, but the copy is shifted some positions to the left. Hence, on one hand, no change is done in the copy, but on the other hand, the position of the letters are changed essentially. An analogous operation is performed where we shift an almost completely changed version of the word.

Definition 17. *For any natural numbers n and i with $1 \leq i \leq n - 1$, we define the operation $S_{n,i} : V^n \to V^{2n}$ by*

$$S_{n,i}(x_1x_2 \dots x_n) = x_1x_2 \dots x_i x_1 x_2 \dots x_n x_{i+1} x_{i+2} \dots x_n.$$

Theorem 27. *For any natural numbers $n \geq 2$ and i with $1 \leq i \leq n-1$ and any word $q \in Q$ of length n, $S_{n,i}(q) \in Q$ also holds.*

Proof. Let $q = ww' \in Q$ with $w = x_1x_2 \dots x_{i-1}$ and $w' = x_ix_{i+1} \dots x_n$, where $x_j \in V$ for $1 \leq j \leq n$. Then $S_{n,i}(q) = www'w'$.

Assume $www'w' \notin Q$, that is, there exist a $p \in \mathbb{N}$, $p > 2$ and $v \in Q$ such as $www'w' = v^p$, that is, $w^2(w')^2 = v^p$. It is known, by Lemma 15, $w = u^k$, $w' = u^l$, and $v = u^m$. Since $ww' \in Q$ and $ww' = u^{k+l}$, we have a contradiction.

Therefore $www'w' \in Q$. □

We mention that an analogous statement does not hold, if one uses the mirror image instead of a copy. The following example shows that then primitivity is not preserved. Let $w = 01$ and $i = 1$; using the mirror image and shifting it by one position to the left we get $0101 \notin Q$.

Finally in the following theorem we present some operations which, together with the above operations, allow the generation of all primitive words of length ≤ 11 (as can be shown by computer calculations) and of a considerable amount of primitive words of length up to twenty.

Theorem 28. *Let $w \in Q$ be a primitive word of length $n \geq 2$ and $x \in V$ and $y \in V$ two different letters of V.*
i) Then wx^n and wx^{n-1} and wxy^{n-2} are in Q, too.
ii) If n is even, then $w(xy)^{(n-2)/2}x$ and $w(xy)^{(n-2)/2}y$ are primitive words, too.

Proof. We omit the easy proofs for i).
ii) We only prove the statement for $w(xy)^{(n-2)/2}x$; the other proof can be given analogously.

Let us assume that $w(xy)^{(n-2)/2}x \notin Q$. Then there is a word $v \in V^+$ such that $w(xy)^{(n-2)/2}x = v^p$ for some $p \geq 2$. Since $w(xy)^{(n-2)/2}x$ has odd length, p and the length of v are odd numbers. Let $p = 2m+1$ for some $m \geq 1$. Thus there are $v_1, v_2 \in V^+$ such that

$$v = v_1v_2,\ |v_1| = |v_2| + 1 \text{ and } w|(xy)^{(n-2)/2}x = v^mv_1|v_2v^m.$$

By $w(xy)^{(n-2)/2}x = v^{2m+1}$, $v = (xy)^kx$ for some $k \geq 1$, and then $v_1 = (xy)^r$, $v_2 = (xy)^{r-1}x$ and

$$w|(xy)^{(n-2)/2}x = ((xy)^kx)^m(xy)^r|(xy)^{r-1}x((xy)^kx)^m.$$

Since the $(n + 2(r-1) + 2)$-nd letters in both representations differ, we have a contradiction. □

5.2.5 Conclusions

In this section, some operations that preserve primitivity of words have been presented. They are summed up in Table I.

Thus, we have the next theorem.

Theorem 29. *For any operation O given in Table I, if $w \in Q$, then $O(w) \in Q$.*

5.3 A Non-Grammatical Method to Generate the Set of Primitive Words

The results presented in this section are part of (68) in which I am first author[1].

Since the main interest in generating Q_V comes from the possibility of figuring out the classification of Q_V, most efforts to generate Q_V materialized as grammatical methods. In order to open a new line of research, we propose a non-grammatical generative method for Q_V that is based on a characterization of the non-primitive numbers. For such a purpose, we will define a non-primitive number in a given base and of a given length as a number whose representation in that base is a non-primitive word with that length.

In the characterization of the non-primitive numbers we will show that a number is primitive if and only if it and zero are not congruent modulo certain numbers. Moreover, we will identify the generalized Fermat-Mersenne numbers with such moduli (that is, moduli under which a primitive number and zero are not congruent).

5.3.1 Some Notation

Let V be an alphabet such that $|V| = q$, with $q \in \mathbb{N}^+$. Then, the largest possible number of words of length n over V is q^n.

Let $q \in \mathbb{N}^+$. We notate the set of the q distinct symbols of the base-q numeral system as $B(q)$, that is to say, $B(q) = \{0, 1, \ldots, q-1\}$. Since, for any non-trivial alphabet V, with $|V| = q$, we can define a bijection $V \to B(q)$, we can transform any word $w \in V^+$ into the base-q representation of an integer. Moreover, since, for any $q \in \mathbb{N}^+$, we can transform the base-q representation of a given integer into that integer, we can transform any word $w \in V^+$ into an integer.

[1] the other author is my supervisor and gave hints, suggestions, and comments

Without loss of generality, we can assume that $V = B(q)$ for any $q \in \mathbb{N}^+$. Therefore, the words over the alphabet V, with $|V| = q$, will be base-q representation of integers.

Definition 18. *Let $m, q \in \mathbb{N}^+$ and $n \in \mathbb{N}$. We say that the integer n is a primitive number in base q and length m if its base-q representation is a primitive word of length m over the alphabet V with $|V| = q$. We will say that $n \in PN_{q,m}$.*

We say that n is a non-primitive number in base q and length m if n is not a primitive number in base q and length m. We will say that $n \in nPN_{q,m}$.

In the sequel we shall use the following notation. We denote the base-q representation of an integer by $(n)_q$ and we will say that $(n)_q \in B(q)^+$. Let $(n)_q, (m)_q \in B(q)^+$, then $(n)_q(m)_q = (nm)_q$ is the multiplication in base-q.

If $(n)_q \in B(q)^+$ and $(n')_{q'} \in B(q')^+$ are the base-q and the base-q' representations of an integer, respectively, then we will say that $(n)_q = (n')_{q'}$. Moreover, in the sequel, we assume $n = (n)_{10}$ for any integer n.

Remark 2. *Let $q, n \in \mathbb{N}^+$. Let V be the alphabet with $|V| = q$. Let V_n^* the set of all the possible words of length n over V. The words of V_n^* are the base-q representations of the set of integers $\{0, \ldots, q^n - 1\}$.*

From now on, we will consider that adding zeros on the left side of a number in any base does not modify its value.

Finally, in (30) the generalized Fermat-Mersenne numbers have been defined as

$$G_{q,p,n} = q^{(p-1)n} + q^{(p-2)n} + \cdots + q^n + 1 = \frac{q^{pn} - 1}{q^n - 1}. \tag{5.4}$$

where $q, p, n \in \mathbb{N}^+$.

5.3.2 Some Remarks and First Results

In this section, we will present some results which are used in the sequel.

Theorem 30. *Given $q, p, n \in \mathbb{N}^+$, $G_{q,p,n} \in nPN_{q,pn}$, that is to say, the base-q representation of $G_{q,p,n}$ is a word $w \in V$ with $|V| = q$, $w = v^p$ with $v \in V^+$ and p a natural number, $p > 1$, and $|v| = n$.*

Proof. If we generalize Equation 13 in (30), for any base q, then we get

$$(G_{q,p,n})_{10} = ((10^{n-1})^{p-1}1)_q.$$

Moreover, $((10^{n-1})^{p-1}1)_q = ((0^{n-1}1)^p)_q$, then

$$(G_{q,p,n})_{10} = ((0^{n-1}1)^p)_q.$$

Therefore, it is clear that the base-q representation of $G_{q,p,n}$ is a word $w \in V$ with $|V| = q$, $w = v^p$ ($v = 0^{n-1}1$) and $|v| = n$. □

Theorem 31. *Given $q, p, n \in \mathbb{N}^+$ and $k \in \mathbb{N}^+$ with $k < q^n$, then $kG_{q,p,n} \in nPN_{q,pn}$.*

Proof. First of all, let us see that if $k < q^n$, then $kG_{q,p,n}$ is a number in base q and length pn. By Remark 2, that is to say, $kG_{q,p,n} \leq q^{pn} - 1$.

By Equation 5.4, $G_{q,p,n} = \dfrac{q^{pn}-1}{q^n-1}$, then we get

$$kG_{q,p,n} \leq (q^n - 1)\frac{q^{pn}-1}{q^n-1} = q^{pn} - 1.$$

Now, we will see that $kG_{q,p,n}$ is a non-primitive number, that is to say, its base-q representation is a non-primitive word.

Let us suppose that $(k)_{10} = (x_1 \cdots x_n)_q$ with $x_i \in B(q)$ for any $i = 1, \ldots, n$ (that is, $k < q^n$). Since $(G_{q,p,n})_{10} = ((10^{n-1})^{p-1}1)_q$, we have

$$(x_1 \cdots x_n)_q((10^{n-1})^{p-1}1)_q = ((x_1 \cdots x_n)^p)_q.$$

Therefore, if $k < q^n$, then $kG_{q,p,n} \in nPN_{q,pn}$. □

Corollary 6. *Let $q, p, n, k \in \mathbb{N}^+$. If $k \geq q^n$, $kG_{q,p,n}$ is not a number of length pn.*

Proof. Since $kG_{q,p,n} \geq q^n \dfrac{q^{pn}-1}{q^n-1} > q^{pn} - 1$, by Remark 2, we have $kG_{q,p,n}$ is not a number of length pn. □

Theorem 32. *Let $q, p, n, p', n' \in \mathbb{N}^+$. If $n = kn'$ for some $k \in \mathbb{N}^+$ and $pn = p'n'$, then*

$$G_{q,p',n'} = G_{q,k,n'}G_{q,p,n}$$

that is, $G_{q,p',n'}$ is a multiple of $G_{q,p,n}$.

Proof. Let $k \in \mathbb{N}^+$ be such that $n = kn'$. Since $pn = p'n'$, we get

$$G_{q,k,n'}G_{q,p,n} = \frac{q^{kn'}-1}{q^{n'}-1} \cdot \frac{q^{pkn'}-1}{q^{kn'}-1} = G_{q,p',n'}.$$

□

5.3.3 The Proposed Generative Method

In this section, we show a characterization of the non-primitive numbers that will be used as a method to generate the set of primitive words.

5.3.3.1 Characterization of the Non-Primitive Numbers

Given $n \in \mathbb{N}^+$, we will classify the non-primitive numbers of length n. We define the set of all the divisors of n that are different from n as $div(n) = \{m < n \mid m \in \mathbb{N}^+ \text{ and } m|n\}$.

Theorem 33. *Let $m, q \in \mathbb{N}^+$ and $n \in \mathbb{N}$. If $n \in nPN_{q,m}$, then there exist $k, p, d \in \mathbb{N}^+$, with $k < q^d$ and $m = pd$, such that $n = kG_{q,p,d}$.*

Proof. Since $n \in nPN_{q,m}$, we have $(n)_{10} = (v^p)_q$ for some $v \in B(q)^+$ with $|v| = d$ and some $p \in \mathbb{N}^+$ such that $m = pd$. Let us suppose $v = x_1 \dots x_d$ with $x_i \in B(q)$ for any $i \in \{1, \dots, d\}$.

Then, we have

$$(n)_{10} = ((x_1 \dots x_d)^p)_q = (x_1 \dots x_d)_q((10^{d-1})^p 1)_q = (k)_{10}(G_{q,p,d})_{10}$$

with $(k)_{10} = (x_1 \dots x_d)_q$. □

Now, we enunciate a corollary that is more precise than the previous theorem.

Corollary 7. *Let $m, q, s \in \mathbb{N}^+$, $n \in \mathbb{N}$ and $div(m) = \{d_1, d_2, \dots, d_s\}$. If $n \in nPN_{q,m}$, then there exist $k, p_i \in \mathbb{N}^+$ and $d_i \in div(m)$, with $k < q^{d_i}$, $m = p_i d_i$ and $d_i \nmid d_j$ for any $d_j \in div(m)$ with $i \neq j$, such that $n = kG_{q,p_i,d_i}$.*

Proof. By using Theorem 33, we know that if $n \in nPN_{q,m}$, then there exist $k, p, d \in \mathbb{N}^+$, with $k < q^d$ and $m = pd$, such that $n = kG_{q,p,d}$. Since $m = pd$, we trivially have $d \in div(m)$.

On the other hand, by using Theorem 32, we know that given $q, p, n, p', n' \in \mathbb{N}^+$. If $n = k'n'$ for some $k' \in \mathbb{N}^+$ and $pn = p'n'$, then $G_{q,p',n'} = G_{q,k',n'}G_{q,p,n}$.

Therefore, if $n = kG_{q,p_i,d_i}$ for any $d_i \in div(m)$ such that $d_i \mid d_j$ for some $d_j \in div(m)$ with $i \neq j$, then

$$n = kG_{q,p_i,d_i} = kG_{q,k',d_i}G_{q,p_j,d_j} = k''G_{q,p_j,d_j}$$

where $k'' = kG_{q,k',d_i}$. Let us see that $k'' < q^{d_j}$. Since $k < q^{d_i}$, we have

$$k'' = kG_{q,k',d_i} \leq (q^{d_i} - 1)\frac{q^{k'd_i} - 1}{q^{d_i} - 1} = q^{k'd_i} - 1 = q^{d_j} - 1.$$

If $d_j \nmid d_k$ for any $d_k \in div(m)$ with $d_j \neq d_k$, then we have finished. If $d_j \mid d_k$ for some $d_k \in div(m)$ with $j \neq k$, then we repeat the same process as before. □

Now, we can enunciate a characterization of the non-primitive numbers by using the previous results and the following definition.

Definition 19. *Let $m, q, s \in \mathbb{N}^+$, $n \in \mathbb{N}$ and $div(m) = \{d_1, d_2, \ldots, d_s\}$. We define the representative numbers of base q and length m as*

$$R_{q,m} = \{G_{q,p_i,d_i} \mid m = p_i d_i \text{ and } d_i \nmid d_j \text{ for any } d_j \in div(m) \text{ with } i \neq j\}.$$

Theorem 34 (Characterization Theorem)**.** *Let $m, q \in \mathbb{N}^+$ and $n \in \mathbb{N}$. Then the natural number $n \in nPN_{q,m}$ if and only if $n = kG_{q,p,d}$ for some natural number k with $k < q^d$ and some $G_{q,p,d} \in R_{q,m}$.*

5.3.3.2 Selection of the Moduli

In this section, we will define the moduli under which a primitive number and zero are not congruent. This will provide a generative method for the set of primitive numbers. It will be used to define a non-grammatical method to generate the language of the primitive words.

By using the characterization theorem of the non-primitive numbers, we can enunciate the next theorem, that is a characterization of the primitive numbers.

Theorem 35. *Let $m, q \in \mathbb{N}^+$ and $n \in \mathbb{N}$. Then $n \in PN_{q,m}$ if and only if n is not a multiple of any $G_{q,p,d} \in R_{q,m}$.*

Let $m, n \in \mathbb{N}^+$. Since m being a multiple of n means that $m \equiv 0 \pmod{n}$, we can enunciate the previous theorem using moduli.

Theorem 36. *Let $m, q \in \mathbb{N}^+$, $n \in \mathbb{N}$ and $R_{q,m} = \{r_1, \ldots, r_t\}$. Then $n \in PN_{q,m}$ if and only if $n \not\equiv 0 \pmod{r_i}$ for any $i \in \{1, \ldots, t\}$.*

Given $m, q \in \mathbb{N}^+$ and using the previous theorem, we can generate $PN_{q,m}$. Since, for any $n \in PN_{q,m}$, we have $(n)_{10} = (w)_q$ for some $w \in V^+$ with $|V| = q$, we can generate all the primitive words of length m over such an alphabet V.

5.3.4 A Property of Symmetry for the Non-Primitive Numbers

We have defined a non-grammatical method to generate the language of the primitive words over any non-trivial alphabet, that is the main aim of this work. In this Section, we will describe a property of the non-primitive numbers.

Let us suppose $m, q \in \mathbb{N}^+$. We will see that the distribution of the distances between two consecutive non-primitive numbers in base q and length m is symmetric.

Let $R_{q,m} = \{r_1, \ldots, r_t\}$. By Theorem 34, we know that $n \in \mathbb{N}$ belongs to $nPN_{q,m}$ if and only if $n = kr_i$ for some natural number k with $k < q^{d_i}$, with $d_i \in div(m)$, and some $r_i \in R_{q,m}$. Moreover, by Remark 2, the words in V_m^*, with $|V| = q$, are the base-q representations of the set of integers $\{0, \ldots, q^m - 1\}$.

Therefore, since we want to see that the distribution of the distances between two consecutive numbers in $nPN_{q,m}$ is symmetric, let us see that $q^m - (x+1)$ belongs to $nPN_{q,m}$ for any number x in the first half of $\{0, \ldots, q^m - 1\}$ that belongs to $nPN_{q,m}$.

Let x be a number in the first half of $\{0, \ldots, q^m - 1\}$ that belongs to $nPN_{q,m}$. Since $x \in nPN_{q,m}$, there exists a natural number k_i, with $k_i < q^{d_i}$ and $m = p_i d_i$, and $r_i \in R_{q,m}$ such that $x = k_i r_i$. Let us see that $q^m - (x+1) \in nPN_{q,m}$, that is, there exist a $r_j \in R_{q,m}$ and $k_j < q^{d_j}$ such that $q^m - (x+1) = k_j r_j$.

Let us suppose that $r_i = G_{q,p_i,d_i}$ for some $p_i, d_i \in \mathbb{N}^+$.

Let $k_j = q^{d_i} - (k_i + 1)$ and $r_j = r_i$. Then

$$k_j r_j = k_j r_i = (q^{d_i} - (k_i+1))r_i = (q^{d_i} - 1)r_i - k_i r_i = (q^m - 1) - k_i r_i = q^m - (x+1).$$

Therefore, we can say that the distribution of the distances between two consecutive numbers in $nPN_{q,m}$ is symmetric. In Figure 5.1, we show two graphic representations of this symmetry. The distances between any two consecutive numbers in $nPN_{q,m}$ can be observed in the y-axis of the figures.

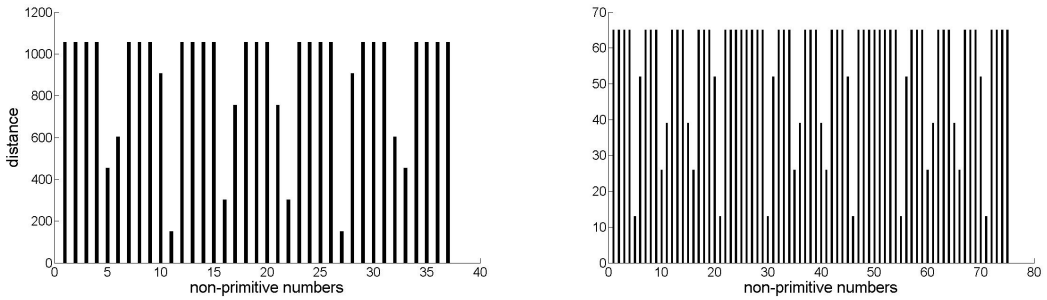


Figure 5.1: Distances between pairs of consecutive non-primitive numbers in base 2 and lengths 15 and 12, respectively.

As a complementary result, it can be demonstrated that the distribution of the distances between two consecutive numbers in $PN_{q,m}$ is symmetric.

5.3.5 Conclusions

In this section, we have defined a non-grammatical method to generate the language of the primitive words over any non-trivial alphabet. Such a method is different from the other well-known methods to generate Q based on grammatical methods. Moreover, since our method is based on the generation of primitive numbers by using moduli, it provides a relation between Q and number theory.

As we pointed out, the main interest in generating Q comes from the interest in establishing the class of the language Q, and for that reason, all the generative methods of Q that have been defined until now, are grammatical methods. What it is proposed here goes beyond a new way to generate the language Q. Since basic arithmetic operations can be formulated in a grammatical way (so are moduli), it is possible to convert this method to a grammar that generates Q. This grammar can shed light about the class of the language Q in Chomsky's hierarchy. In general, different numerical methods (i.e., other connections between number theory and Q) could be sketched, and converted to grammatical formalisms, yielding new insights on this intricate problem of formal languages.

Table I: These operations are defined for any natural numbers n, i, j, k, i' and i'' with $1 \leq i \leq n$, $1 \leq j \leq n$, $2 \leq k \leq n$, $0 \leq i' \leq n$, $1 \leq i'' \leq n-1$ and $i \neq j$, letters $x, y, z \in V$ with $x \neq y$, a word $w = x_1 x_2 \dots x_n \in V^+$ of length n, and the mapping $h : V \to V$ with $h(1) = 0$ and $h(0) = 1$.

$I_{n,i',z}(w) = x_1 x_2 \dots x_n x_1 x_2 \dots x_{i'} z x_{i'+1} x_{i'+2} \dots x_n$ for $n \geq 2$
$I_{n,i',z,j,x,y}(w) = \begin{cases} x_1 \dots x_n x_1 \dots x_{i'} z x_{i'+1} \dots x_{j-1} y x_{j+1} \dots x_n & x_j = x, i' < j \\ x_1 \dots x_n x_1 \dots x_{j-1} y x_{j+1} \dots x_{i'} z x_{i'+1} \dots x_n & x_j = x, i' > j \\ \text{undefined} & \text{otherwise} \end{cases}$
$D_n(w) = x_1 x_2 \dots x_n h(x_1 x_2 \dots \dots x_{n-1})$ for $n \geq 2$
$M_{n,i,x,y}(w) = \begin{cases} x_1 x_2 \dots x_n x_n x_{n-1} \dots x_{i+1} y x_{i-1} x_{i-2} \dots x_1 & x_i = x \\ \text{undefined} & \text{otherwise} \end{cases}$
$M'_{n,k,x,y}(w) = \begin{cases} x_1 x_2 \dots x_n x_n x_{n-1} \dots x_{k+1} y x_{k-1} x_{k-2} \dots x_2 & x_k = x \\ \text{undefined} & \text{otherwise} \end{cases}$ for $n \geq 2$
$S_{n,i''}(w) = x_1 x_2 \dots x_{i''} x_1 x_2 \dots x_n x_{i''+1} x_{i''+2} \dots x_n$, for $n \geq 2$
$D_{n,i}(w) = x_1 x_2 \dots x_n x_1 x_2 \dots x_{i-1} x_{i+1} x_{i+2} \dots x_n$, for $n \geq 2$
$D_{n,i,j,x,y}(w) = \begin{cases} x_1 \dots x_n x_1 \dots x_{i-1} x_{i+1} \dots x_{j-1} y x_{j+1} \dots x_n & x_j = x, i < j \\ x_1 \dots x_n x_1 \dots x_{j-1} y x_{j+1} \dots x_{i-1} x_{i+1} \dots x_n & x_j = x, i > j \\ \text{undefined} & \text{otherwise} \end{cases}$ for $n \geq 2$
$M_{n,1}(w) = x_1 x_2 \dots x_n x_n h(x_{n-1} x_{n-2} \dots x_1)$ for $n \geq 2$
$M_{n,n}(w) = x_1 x_2 \dots x_n h(x_n x_{n-1} \dots x_2) x_1$ for $n \geq 2$
$z_1(w) = w x^n$ for $n \geq 2$
$z_2(w) = w x^{n-1}$ for $n \geq 2$
$z_3(w) = w x y^{n-2}$ for $n \geq 2$
$O_n(w) = x_1 x_2 \dots x_n h(x_1) x_2 \dots x_{i-1} h(x_i) x_{i+1} \dots x_{n-1} h(x_n)$, where $i = \dfrac{n+1}{2}$, for odd $n \geq 5$
$O'_n(w) = x_1 x_2 \dots x_n h(x_n) x_{n-1} \dots x_{i+1} h(x_i) x_{i-1} x_{i-2} \dots x_2 h(x_1)$, where $i = \dfrac{n+1}{2}$, for odd $n \geq 5$
$z_4(w) = w(xy)^{(n-2)/2} x$ for even n
$z_5(w) = w(xy)^{(n-2)/2} y$ for even n

Chapter 6

Dynamics of the Complexity

Now, we have enough elements to study the dynamics of the complexity during the evolution of CURLs that try to adapt to a given environment.

In Section 6.1, we show some preliminary studies on the behavior of the complexity by using a simple framework and two different approaches. The first one uses a greedy algorithm and the edit operations over binary words. The second one uses a genetic algorithm and the edit operations over sequences of the operations that preserve the minimality that were defined in the previous chapter. We will see that although the behavior of the complexity is increasing in both of them, in the second approach, the correlation between the similarity to the environment and the complexity of the individual is stronger. This will be the definitive push to choose the sequences of operations preserving primitivity as the representation of the genotypes.

In Section 6.2, an artificial ecosystem of CURLs is defined by using the elements presented in the previous chapters, where genotypes are sequences of the operations that preserve primitivity. By using this complex ecosystem, we can study, as well as other features, the different components that affect such an increasing behavior of the complexity during the evolution. For example, we have found: (1) a strong correlation between the evolving complexity of the population and the complexity of the environment, and (2) that predatory behavior promotes a higher complexity of the individuals.

6.1 Dynamics of the Complexity in the Evolution of Finite Automata

The results presented in this section are part of (70) in which I am first author[1].

6.1.1 Introduction

The evolution of formal systems and, in particular, of finite automata, has been widely studied. This study can be done from either an analytical or empirical point of view. In (20), a computational device called network of evolutionary processors is proposed. It is based on evolutionary rules and communication within a network. The generative power of evolutionary networks is discussed in (4; 5; 19; 27; 66; 67; 78). Grammatical inference is the problem of inferring a grammar from a set of positive samples which the grammar should generate, and a set of negative samples which the grammar should not generate (see (29)). Evolutionary computation (see (8)), mainly genetic algorithms, are widely used with this purpose. For example, in (65; 92), genetic algorithms infer regular languages, and in (55), context-free languages are inferred.

The study of the evolution of a formal system can also be biologically motivated. In (23), the concept of an evolutionary system is introduced. This is a language generating device inspired by the evolution of cell populations, and it is based on edit operations and string divisions. The purpose of this system is to model some properties of evolving cell communities at the syntactical level. On the other hand, finite state machines (FSMs, for short) have been applied to model organisms. In (97), a new approach to evolve such artificial organisms is presented. FSMs learn a navigation and searching task in heterogeneous environments. The authors report on the formation of different species. Moreover, grammatical inference methods are expected to find some grammatical structures hidden in biological sequences, (104).

Although (as the previous papers show) many properties in the evolution of formal systems have been analyzed, as far as we know, the behavior of the complexity during such an evolution has not been studied. The main reason for that lack of results is that the complexity is a complex concept in itself. The term "complexity" presents so many variations that it is only valid in specific situations. This means that measuring the complexity is an abstract estimation that depends on the context in which it is

[1] the rest of coauthors are my supervisors and gave hints, suggestions, and comments

used. For that reason, in the course of history, different complexity measures have been considered. Some well-known approaches are Kolmogorov-Chaitin-Solomonoff's measure (59) and Shannon's entropy (105). However, none of them provide a good method to address the problem of measuring the complexity. Shannon's entropy is defined as the amount of information of the genome of an organism, but that definition has a problem: a text with the same length as the human genome that has been obtained by combining the letters A, T, C and G might represent more information than the genome. Kolmogorov-Chaitin-Solomonoff's measure is defined as the length of the shortest program that is necessary to generate a text, but such a measure is not computable.

Since we need an objective complexity measure to study the behavior of the complexity in the evolution of a formal system, as we explained before, we will use the set of CURLs. In this way, state complexity can be used.

Thus, we will start with a population of CURLs (individuals) that will try to adapt to a given environment (represented by another CURL). In order to calculate how well an individual adapts to the environment, we use the measure of the similarity for CURLs defined in (24) and has been introduced in Section 3.

In this section, we propose two different approaches to study the behaviour of the complexity during the evolutionary process in which a set of CURLs adapts to a given environment (another CURL). The first one will use a greedy algorithm and the edit operations (substitution, elimination and addition of a symbol over a word) over binary words. The second one will use a genetic algorithm and the edit operations over a sequence of operations that preserve the minimality of the CUDFAs (genotypes) of the population during the evolution (consequently, in this case, the CUDFAs of the initial population and of the environment will be also minimal) that was presented in the previous chapter.

6.1.2 Edit Operations

The usual operations over words are the edit operations, i.e., addition, elimination and substitution of a symbol. They were introduced in Definition 4.2 in Chapter 4.

As we mentioned there, we represent the sets of all possible additions, eliminations and substitutions as $\mathcal{A}$, $\mathcal{E}$ and $\mathcal{S}$, respectively. Thus, the operations in $\mathcal{A}$, $\mathcal{E}$ and $\mathcal{S}$ are called the edit operations.

6.1.3 Evolution of Similarity and Complexity under Edit Operations over Binary Words

Since we have a formal framework, in which an objective complexity measure (the state complexity) is defined, if a dynamics over this framework is introduced, then a study about the dynamics of the complexity can be done.

The main objective in this section is the study of the behaviour of the complexity during the process in which CURLs (the individuals) are approaching to a much more complex CURL (the environment) by applying some operations over their genotypes. We will say that a given CUDFA is close to another CUDFA if the language represented by both of them is similar.

As a first attempt, we propose the next method that uses a greedy algorithm and the edit operations over the binary words that represents the CUDFAs, that is to say, the genotypes (in this section, $V = \{0, 1\}$). As we said before, the CURL represented by a given CUDFA will be considered as its corresponding phenotype.

6.1.3.1 Greedy Algorithm

The similarity measure for CURLs introduced in (24) is used. Given a CUDFA $w \in V^+$, we notate the CURL represented by w as $L(w)$. We start with the following definition.

Definition 20. *Let $v, w \in V^+$ be two CUDFA and $\mathcal{O} \subseteq \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$. We define the highest similarity set of operations from $\mathcal{O}$ for v and w, $HS(v, w, \mathcal{O})$, as*

$$\begin{aligned} HS(v, w, \mathcal{O}) &= \{O \in \mathcal{O} \mid URLS_{L(O(v)),L(w)} \\ &\quad = \max_{O' \in \mathcal{O}}(URLS_{L(O'(v)),L(w)})\}. \end{aligned}$$

Let us see that by using substitution and addition operations, we can obtain an individual that is completely adapted to the environment by using a greedy algorithm.

Theorem 37. *Let p a prime number and $\mathcal{O} = \mathcal{S} \cup \mathcal{A}$. For any $v, w \in V^+$ with $|v| + 1 < |w| = p$ and $|w|_1 < |w|$, there exists a sequence of operations*

$$O_1, O_2, \ldots, O_x \in \mathcal{O} \text{ with } x = |v|_0 + |w|_0 + |w| - |v| - 1$$

such that

$$w = O_x(O_{x-1} \ldots (O_2(O_1(v))) \ldots)$$

and

$$O_i \in HS(O_{i-1}(\dots(O_2(O_1(v))\dots), w, \mathcal{O})$$

for $1 \leq i \leq x$.

Proof. Let $L(v) = \{\{a_i + |v|k\}_{k\in\mathbb{N}}\}_{i=1,\dots,n}$ and $L(w) = \{\{b_j + |w|k\}_{k\in\mathbb{N}}\}_{j=1,\dots,m}$ for some $n, m \in \mathbb{N}$.

Since $|w| = p$, $|v| < |w|$ and p is a prime number, we have $gcd(|v|, |w|) = 1$. Thus, $URLS_{L(v),L(w)} = \dfrac{\frac{|w|_1}{|w|} + \frac{|v|_1}{|v|}}{2}$ (see Remark 1).

Given an operation $O \in \mathcal{O}$, we notate $O(v) = v'$.

Let us suppose $O \in \mathcal{S}$.

- If we change a zero into a one, then $|v'|_1 = |v|_1 + 1$ and $|v'| = |v|$. Therefore,

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1+1}{|v|}}{2} > URLS_{L(v),L(w)}.$$

- If we change a one into a zero, then $|v'|_1 = |v|_1 - 1$ and $|v'| = |v|$. Therefore,

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1-1}{|v|}}{2} < URLS_{L(v),L(w)}.$$

Let us suppose $O \in \mathcal{A}$.

- If we add a one, then $|v'|_1 = |v|_1 + 1$. Therefore,

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1+1}{|v|+1}}{2} \geq URLS_{L(v),L(w)}.$$

The equality is satisfied when $|v|_1 = |v|$, that is, when $v = 1^k$ for some $k \in \mathbb{N}$.

- If we add a zero, then $|v'|_1 = |v|_1$. Therefore,

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1}{|v|+1}}{2} < URLS_{L(v),L(w)}.$$

We now compare the similarities obtained after applying any of the operations over v. Let O and O' be a substitution of a zero into a one and an addition of a one, respectively. Then, by the above calculations,

$$URLS_{L(O(v)),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|v|_1+1}{|v|}}{2} > \frac{\frac{|w|_1}{|w|} + \frac{|v|_1+1}{|v|+1}}{2} = URLS_{L(O'(v)),L(w)}.$$

Taking into consideration that substitutions of a one into a zero and additions of a zero, lead to smaller similarities, we get

$$HS(v, w, \mathcal{O}) = \{O \in \mathcal{S} \mid \text{ O changes a zero into a one}\},$$

Following the same reasoning and using the sequence of operations

$$O_1, O_2, \ldots, O_{|v|_0} \in \mathcal{S}$$

that changes a zero into a one in each step, we get

$$O_{|v|_0}(\ldots(O_2(O_1(v)))\ldots) = 1^{|v|}$$

and

$$O_i \in HS(O_{i-1}(\ldots(O_2(O_1(w))\ldots), w, \mathcal{O})$$

for $1 \leq i \leq |v|_0$.

If we have a word of the form 1^s for some s, then we cannot change a zero into a one. Thus, by the above calculations,

$$HS(1^s, w, \mathcal{O}) = \{O \in \mathcal{A} \mid O \text{ adds a one}\}$$

for any $|v| \leq s \leq |w| - 1$. Therefore, using the sequence of operations

$$O_{|v|_0+1}, \ldots, O_{|v|_0+|w|-|v|-1} \in \mathcal{A}$$

that adds a one in each step, we get

$$O_{|v|_0+|w|-|v|-1}(\ldots(O_{|v|_0+1}(1^{|v|})\ldots) = 1^{|w|-1}.$$

Now, we have $r = 1^{|w|-1}$.

- If we add a one, then $v' = 1^{|w|}$ and $q = |w|_1$. Therefore, by Remark 1

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|w|_1}{|w|_1}}{2} = \frac{\frac{|w|_1}{|w|} + \frac{|w|-1}{|w|-1}}{2} = URLS_{L(r),L(w)}.$$

- If we add a zero in a position in which w has a zero (we know that there exists at least a zero in w because $|w|_1 < |w|$), then $q = |w|_1$. Therefore, by Remark 1

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|-1} + \frac{|w|_1}{|w|_1}}{2} > \frac{\frac{|w|_1}{|w|} + \frac{|w|-1}{|w|-1}}{2} = URLS_{L(r),L(w)}.$$

- If we change a one into a zero, we have $URLS_{L(v'),L(w)} < URLS_{L(v),L(w)}$ as above.

Therefore,

$$HS(1^{|w|-1}, w, \emptyset) = \{O \in \mathcal{A} \mid O \text{ adds a zero in a position in which } w \text{ has a zero}\}.$$

Thus, $O_{|v|_0+|w|-|v|} \in \mathcal{A}$.

Now, we have $r = 1^t01^u$ for some $t, u \in \mathbb{N} \cup \{0\}$ such that $t+u+1 = |w|$. Therefore, $|r| = |w|$, $|r|_1 = |w| - 1$ and $q = |w|_1$. Thus,

$$URLS_{L(r),L(w)} = \frac{v\frac{|w|_1}{|w|-1} + \frac{|w|_1}{|w|_1}}{2}.$$

- If we change a one into a zero, then $|v'|_1 = |r|_1 - 1 = |w| - 2$.
  - If w has a zero in that position, then $q' = |w|_1$. Therefore,
$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|-2} + \frac{|w|_1}{|w|_1}}{2} > URLS_{L(r),L(w)}.$$
  - If w has not a zero in that position, then $q' = |w|_1 - 1$. Therefore,
$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1-1}{|w|-2} + \frac{|w|_1-1}{|w|_1}}{2} < URLS_{L(r),L(w)}.$$
- If we change a zero into a one, then $|v'|_1 = |r|_1 + 1 = |w|$.
  - If w has a one in that position, then $q' = |w|_1 + 1$. Therefore,
$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1+1}{|w|} + \frac{|w|_1+1}{|w|_1}}{2} > URLS_{L(r),L(w)}.$$
    This is not our case of study, since r has an only one and it is in a position in which w also has a zero.
  - If w has not a one in that position, then $q' = |w|_1$. Therefore,
$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|w|_1}{|w|_1}}{2} < URLS_{L(r),L(w)}.$$
- If we add a one, then $|v'|_1 = |r|_1 + 1 = |w|$ and $|v'| = |w| + 1$. Therefore,
$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|w|}{|w|+1}}{2} > URLS_{L(r),L(w)}.$$

- If we add a zero, then $|v'|_1 = |r|_1 = |w| - 1$ and $|v'| = |w| + 1$. Therefore,

$$URLS_{L(v'),L(w)} = \frac{\frac{|w|_1}{|w|} + \frac{|w|-1}{|w|+1}}{2} > URLS_{L(r),L(w)}.$$

Let us compare the similarities obtained after applying any of the operations over r. Let $O \in \mathcal{S}$ change a one into a zero in a position in which w has a zero.

- Let $O' \in \mathcal{A}$ be such that a one is added. Then

$$URLS_{L(O(r)),L(w)} = \frac{v\frac{|w|_1}{|w|-2} + \frac{|w|_1}{|w|_1}}{2} > \frac{\frac{|w|_1}{|w|} + v\frac{|w|}{|w|+1}}{2} = URLS_{L(O'(r)),L(w)}.$$

- Let $O' \in \mathcal{A}$ be such that a zero is added. Then

$$URLS_{L(O(r)),L(w)} = \frac{\frac{|w|_1}{|w|-2} + \frac{|w|_1}{|w|_1}}{2} > \frac{\frac{|w|_1}{|w|} + \frac{|w|-1}{|w|+1}}{2} = URLS_{L(O'(r)),L(w)}.$$

Therefore,

$$HS(r, w, \mathcal{O}) = \{O \in \mathcal{S} \mid O \text{ changes a one in a position in which } w \text{ has a zero}\}.$$

Thus, using the sequence of operations

$$O_{|v|_0+|w|-|v|+1}, \ldots, O_{|v|_0+|w|-|v|+|w|_0-1} \in \mathcal{S}$$

that change a one into a zero in a position in which w has a zero, we get

$$O_{|v|_0+|w|-|v|+|w|_0-1}(\ldots(O_{|v|_0+|w|-|v|+1}(r))\ldots) = w$$

and, for $1 \leq i \leq |w|$, $O_{|v|_0+|w|-|v|+i}$ is in the highest similarity set of

$$O_{|v|_0+|w|-|v|+i-1}(\ldots(O_{|v|_0+|w|-|v|+1}(r))\ldots).$$

□

We have observed computationally the same behavior when w has not a prime length. For the theoretical proof of the general case, we would have to consider a lot of different cases in dependence of the *gcd* between the length of the environment and the length of the currently evolved individual. Thus, we take into account the computational results to assert that the previous theorem seems to be satisfied in the general case.

Let us study the case in which $\mathcal{O} \subseteq \mathcal{S} \cup \mathcal{A} \cup \mathcal{E}$, that is, the elimination operation is also included. Let us see that in this case, we can not ensure that we will be able to be adapted completely to the environment, at least in a reasonable time. This is because the algorithm can get stuck in the set of words consisting of only ones whose lengths are between one and $|w| - 1$. Given $O \in \mathcal{S}$ and $O' \in \mathcal{E}$,

$$URLS_{L(O(v)),L(w)} > URLS_{L(O'(v)),L(w)}$$

can be proved as in the previous theorem. Therefore, we obtain the word $1^{|v|}$ by using the previous greedy algorithm. Since given $O' \in \mathcal{E}$ and $O'' \in \mathcal{A}$, we have $URLS_{L(O'(v)),L(w)} = URLS_{L(O''(v)),L(w)}$, thus the greedy algorithm chooses words consisting of only ones whose lengths can variate without limits between one and $|w| - 1$ and in some cases, it does not reach the length of w.

Therefore, using $\mathcal{O} = \mathcal{S} \cup \mathcal{A}$ any individual can be completely adapted to any environment. The behaviour of the complexity by using this approach is always as figure 6.1 shows. We start with a individual of a certain complexity (usually very low) and after applying the substitutions of all the zeros into ones, a word with only ones that have state complexity one is obtained. This complexity remains during a lot of steps until the length of the environment is reached. In this point, we obtain a word $r = 1^t 0 1^u$ for some $t, u \in \mathbb{N} \cup \{0\}$ and the complexity experiments a huge jump. Thus, very similar individuals show very different states complexities (individuals in the generation 29 and 30 in figure 6.1). The main reason for this is that the edit operations do not preserve the primitivity of words. Thus, we propose a second method that preserves primitivity.

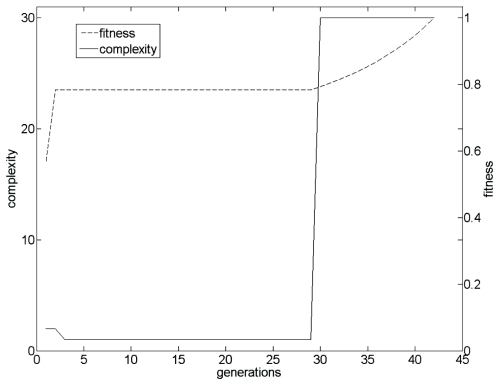


Figure 6.1: Complexity and fitness shown by the greedy algorithm which transforms the CUDFA 01 into the CUDFA 00111011101010110111010101010100. Each time an operation is applied the graphic on the right axis shows the similarity (fitness) between the corresponding word and the environment and the graphic on the left axis shows the complexity of the corresponding word.

6.1.4 Evolution of Similarity and Complexity under Edit Operations over Sequences of Operations Preserving Primitivity

We have shown that by using the previous method, the behaviour of the complexity is quite limited. Moreover, in some cases, very similar individuals show very different states complexities. The main reason for this is that the edit operations over binary words do not preserve the minimality of the genotypes (the primitivity of the words) during the process. For that reason, we propose another method in which operations that preserve the primitivity are used.

We will use a genetic algorithm and the edit operations over sequences of operations that preserve the primitivity defined in Section 5.2 (and summed up in Table I) that represent the minimal CUDFA.

6.1.4.1 Genetic Algorithm

Genetic algorithms are a particular branch of evolutionary computation that were originally introduced by Holland in (48). They are implemented in a computer simulation in which a population of abstract representations (called genotypes) of candidate solutions (called individuals or phenotypes) to an optimization problem evolves toward better solutions.

Let us see the representation of a minimal CUDFA that we will use in this section, i.e, in this method, the genotypes will not be binary words as in the previous one.

Thus, the genotype of an individual is defined by a vector $v \in C \times O^k$ where

- $C = \{0, 1\}$ is the zygote of the individual (i.e., its initial cell),
- O is the set of operations listed in Table I plus the operation $N(w) = w$ for any word $w \in V^+$. These operations preserve primitivity of words and had been defined in (26). O^k defines the k operations, $k \geq 1$, which apply to the zygote of the individual to develop into its phenotype.

Thus, the genotype of an individual is a vector $(x, O_1, \ldots, O_p)$ of size at least two (i.e., at least one operation has to be applied) where $w = O_p(\ldots(O_2(O_1(x)))\ldots)$. Each component of such a vector is a gene.

For example, the vector $(0, M_{1,1,0,1}, I_{2,1,1}, D_{5,3})$ is the genotype of the individual $L(011010101)$ since

$$D_{5,3}(I_{2,1,1}(M_{1,1,0,1}(0))) = 010110111.$$

Since the zygotes are primitive words, and the operations in O preserve primitivity, the genotypes are primitive words, and consequently, by Corollary 3, all the genotypes are minimal CUDFAs.

Thus, in this method, we restrict the framework to the subset of all minimal CUDFAs which can be generated by the operations. First of all, a set of 500 CURLs (the initial population) with a small complexity (2 or 3) is randomly generated, i.e, the sequence of operations that represents each minimal CUDFA (genotype) is generated with a uniform distribution such that each operation is equally likely to be chosen. By using a genetic algorithm (GA, for sort), the population (where each individual is a minimal CUDFAs) gets adapted to a given environment (a CURL with a much higher complexity, around 20 states, which is also randomly generated by using the operations). The adaptation of an individual to the environment (i.e. its fitness) is measured as the similarity between the languages that they represent. Thus, the behaviour of the complexity during this process will be studied.

Mutation is the only genetic operation which is used in the GA. Crossover has not been included, because we have empirically seen that it introduces a high disruption in the descendants, which differ 75% in average from their parents.

Three types of mutation have been implemented: changing, deleting and adding a gene (the edit operations). The position of the gene that will be mutated is selected with uniform probability. In the case in which a gene is added, or changed, the new gene will be chosen randomly with uniform probability.

In each generation, the population is ranked by fitness. Thus, those individuals in the upper half of the rank will be selected to be mutated. At most one mutation is applied to each selected individual in each generation. The mutation that will be applied is chosen in the following way: whereas in every 5 generations one of the three mutations is randomly chosen with uniform distribution, in the intervening generations, changing a gene is the only mutation that can be chosen. A mutated individual is considered as an offspring. If the offspring obtained after mutating a selected individual is not a valid individual (that is to say, some operation of its sequence of genes is not defined for its corresponding word), then the selected individual will be considered as the offspring.

The individuals in the lower half of the rank are replaced by the mutated individuals (the offspring). Selection will leave 500 individuals for the next generation. The GA stops when the best individual of the population is, at least, 90% similar to the environment.

Simulation results (see figure 6.2) show how the adaptation of the individuals to the environment increases continuously. Frequently, a run ends up after some few generations with a solution that approaches 95% of the environment. Furthermore, the behaviour of the average complexity of the population is increasing. In particular, in the first half of the generations, an increment of the complexity of the fittest individual of the population can be observed and it correlates with the increment of the maximum fitness (similarity). That is to say, almost every time the fitness increases, the complexity increases too. Thus, until this point, there exists a strong correlation between the fitness and the complexity of the best individual. Finally, when an individual is found which equals the complexity of the environment, the complexity remains unchanged while the maximum fitness continues increasing.

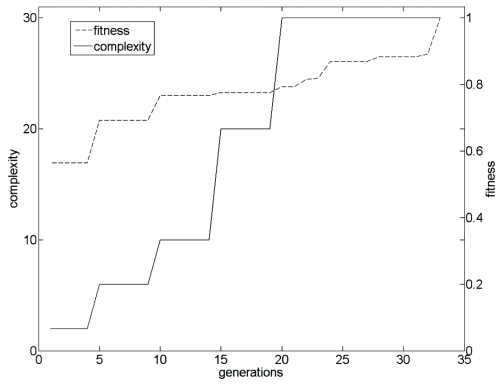


Figure 6.2: Complexity and fitness shown by the genetic algorithm using 00111011101010110111010101010100 as the environment. The graphics on the right and the left axis respectively show the fitness and the complexity of the best individual in each generation.

6.2 Complexity Dynamics in Evolving Populations of Cyclic Automata

The results presented in this section are part of (69) in which I am first author[1].

6.2.1 Introduction

Darwin described the process of natural selection more than 150 years ago. Although natural selection does not guarantee that organisms will increase in complexity as they evolve, it is apparent that the complexity of certain lineages has increased during evolution (12; 47; 75). For that reason, measuring the tendency of the complexity during the evolution is a problem that has been tried to be solved for many years by artificial life researchers, (11). To study such a tendency, complexity needs to be both rigorously defined and measurable (2).

Many scientists have modeled evolving artificial ecosystems that exhibit dynamics comparable in some way to the dynamics of biological evolution to make a case for or against a trend in the evolution of complexity. The main reason for this is that some of the multiple features of biological systems, included complexity, can be explained by

[1] the other author is my supervisor and gave hints, suggestions, and comments

using quantifiable measures over such artificial ecosystems. Partial differential equations systems have been the most common approach to ecological modeling despite presenting a big number of limitations (mainly related to the difficulty of adding new features and the requirement of deep mathematical knowledge to analyze them). As alternative to these limitations, individual-based models (IbMs, for short) has been extensively used in ecological modeling (see (13; 14; 15; 22; 37; 41; 42; 46; 91; 98; 101)), mainly, since the review of Huston et al. appeared two decades ago, (51). IbMs can simulate very complex populations due to the huge storage ability and processing speeds available in nowadays computer simulations.

Different complexity measures have been considered in IbMs when the tendency of the complexity has been studied. This is due to complexity is a complex concept in itself. The term "complexity" presents so many variations that it is only valid in specific situations. This means that measuring the complexity is an abstract estimation that depends on the context in which it is used. Many papers that refer to complexity in IbMs use the number of genes as measure of complexity, (52; 111). Although, they usually obtain that under certain conditions the tendency of the complexity is increasing, whether or not the number of genes in an organism's genome is an appropriate measure of biological complexity has been questioned. In (109), it is mentioned that the recent flurry of completed genome sequences, including our own, suggests that this is not necessarily the case. Rather surprisingly, it turns out that the Caenorhabditis elegans worm has $18,424$ genes in its genome, the Drosophila melanogaster fruit fly $13,601$, the Arabidopsis plant about $25,498$, and humans about $35,000$.

The amount of information of the genome of an organism has been quantified using the Shannon's entropy, but this approach has a problem: an accurate calculation would require unavailable information on fitness weight of each nucleotide in the genomic DNA. A modified Shannon's entropy has been used as complexity measure in (3; 84), but still it is mentioned that it can be measured only approximately. More statistical measures are collected in (38).

McShea presented complexity in (72) as a broad term covering four independent types. For each type, some measures are described. This division makes more difficult to study the tendency of the complexity. In (112), the complexity of individual tasks is measured by using the shortest possible coding for an algorithm capable of resolving

the tasks, that is to say, the algorithmic complexity. Since such a measure is not computable, a rough measure of the algorithmic complexity is used.

A definition of complexity based on algebraic automata theory and a mathematical axiomatization of complexity developed is proposed by Nehaniv in (82). Thus, the complexity measure is defined as the number of levels in a hierarchical decomposition for a given automaton. Since decompositions are not unique, the shortest possible decomposition is chosen. The proposed definition has the benefit that it is mathematically rigorous, however, determining the size of the shortest decomposition turns out to be a very difficult problem. To avoid this problem, the holonomy decomposition method is used, (36). It is a particular decomposition method that is computationally accessible although in not the shortest decomposition.

Thus, it is seems that none of the proposed complexity measures provides a rigorous method to address the problem of measuring the complexity. However, state complexity over regular languages is a well-known and an objective complexity measure. Since, independently of the chosen representation of a given regular language, the minimal deterministic finite automaton that represents it can be always deterministically calculated, state complexity constitutes a rigorous complexity measure.

For that reason, as we explained before, we propose a formal framework in which state complexity can be used. In this way, by using this rigorous complexity measure, objective studies of the complexity dynamics will be done. Thus, in the proposed artificial ecosystem, a population of CURLs with low complexity is placed in an artificial ecosystem that is compounded of a set of CURLs (usually with high complexity) randomly placed in a torus, we have called them subenvironments. The individuals compete by inert (obtained from the subenvironments) and non-inert resources (depredation between individuals).

Thus, in this section, we propose a IbM consisting of CURLs in which objective studies of complexity and population dynamics can be done.

6.2.2 Description of the Model

In general lines, in the proposed model, individuals attempt to process resources from the environment to get enough energy to leave offspring.

6.2.2.1 Individuals

As we said before, an individual is a CURL $L(w)$ where $w \in Q$. The genotype of an individual $L(w)$ is defined as in Section 6.1.4.1 by a vector $v \in C \times O^k$.

As we said before, all the genotypes are minimal CUDFAs.

6.2.2.2 Environment

In the proposed model, space is represented as a discrete two-dimensional torus where its side length is a parameter in the model, D. Each point of the space can contain at most one individual. According to a parameter N_e of the model, N_e CURLs are randomly located in the space. They will provide the resources of the environment that will be called inert resources. Each of them sets a local environment (subenvironment) and has associated a natural number n. This natural number will be used to locate in the environment the inert resources that will be provided by the subenvironment. We will say that it is the derivation of the subenvironment.

By equation (2.3), we know that the set of strings that belong to a given CURL is a subset of the natural numbers. Thus, N_c random natural numbers of each subenvironment are randomly located in the environment according to a two-dimensional normal distribution with mean and standard derivation the position and the derivation of the corresponding subenvironment, respectively. We will say that each subenvironment has generated N_c inert resources.

We will say that an individual $L(w)$ processes a given inert resource, when the inert resource is a natural number that belongs to the CURL $L(w)$ represented by the minimal CUDFA w. If an individual processes an inert resource, then the inert resource is removed and the corresponding subenvironment generates another inert resource that is randomly located in the same way as before.

6.2.2.3 Energy

Individuals gain and lose energy during evolution by processing resources and leaving offspring.

Processing resources Let us see some definitions.

Definition 21. *Let $w \in Q$ be a CUDFA and let $L(w)$ be the CURL (individual) represented by w. Given an inert resource $n \in \mathbb{N}$, we say that $L(w)$ processes n if and only if $n \in L(w)$.*

An example can be seen in Figure 6.3, where black circles are the accepting states of the automaton. The automaton transits in the counterclockwise direction.

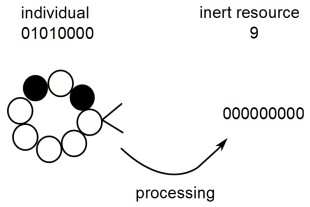


Figure 6.3: Since the CURL represented by 01010000 is $L(01010000) = \{1+8k, 3+8k\}_{k\in\mathbb{N}}$ and $9 \in L(01010000)$, the individual $L(01010000)$ processes 9.

Definition 22. *Given an individual $L(w)$ with $w \in Q$ and an inert resource $n \in \mathbb{N}$, we define the energy consumed by $L(w)$ when it attempts to process n as*

$$E_c(L(w), n) = \alpha\frac{\beta}{n}$$

where α is the energy that the individual consumes when transits to a non-accepting state (it is a parameter of the model) and β is the number of non-accepting states to which the individual transits before accepting or rejecting the string of length n. We define the energy acquired by $L(w)$ from n as

$$E_a(L(w), n) = 1 - E_c(L(w), n).$$

After attempting to process an inert resource, the energy of the individual will be updated in the following way.

Definition 23. *Given an individual $L(w)$ with $w \in Q$ and an inert resource $n \in \mathbb{N}$, the energy of $L(w)$ after attempting to process n in the evaluation t is defined as*

$$E_t(L(w)) = E_{t-1}(L(w))p(L(w), n)^{1/\gamma} + R$$

where $p(L(w), n) = 1 - d(L(w), n)$ being $d(L(w), n)$ the Euclidean distance between the positions of $L(w)$ and n, $\gamma \in \mathbb{N}$ is a parameter of the model and

$$R = \begin{cases} E_a(L(w), n) & \text{if } n \in L(w) \\ -E_c(L(w), n) & \text{in other case} \end{cases}$$

Note that the nearer the individual is to the resource, the smaller the amount of energy that the individual loses in the journey to approach the inert resource is.

Having offspring After an individual has attempted to process a resource, it will try to leave offspring. This will be possible if it has enough energy to do it.

Definition 24. *Given an individual $L(w)$ with $w \in Q$, its number of descendants in the evaluation t is defined as*

$$ND_t(L(w)) = [E_t(L(w))]$$

where $[x]$ is the nearest integer function.

If $ND_t(L(w)) > 0$ for some individual $L(w)$ in the evaluation t, then $L(w)$ is replaced by its offspring in the population. Descendants are mutated copies of their predecessors (we will see this in the next section).

Let us suppose that an individual $L(w)$ has p descendants $\{d_1, \ldots, d_p\}$ in the evaluation t, then

$$E_t(d_i) = \frac{E_t(L(w))}{ND_t(L(w))}$$

for any $i = 1, \ldots, p$.

Offspring are randomly located in the environment according to a two-dimensional normal distribution with mean the predecessor position and standard derivation $\frac{D}{100}$ (that is to say an 1% of D). If the position that has been randomly assigned to a certain descendant is already occupied by another individual, then the descendant dies.

6.2.2.4 Mutation Operations

Since descendants are mutated copies of the genotypes of their predecessors, let us see the types of mutations that can be applied and their rates. As in the previous section, since we have empirical evidences of crossover introduces a high disruption in the descendants, mutation is the only genetic operation which is used in the GA.

Three types of mutation have been implemented: changing a gene, deleting a gene and adding a gene. The position of the gene that will be mutated is selected with uniform probability between the second and the last component of the vector that represents the genotype of the individual (that is, the zygote can not be mutated). In the case in which a gene is added or changed, the new gene will be chosen from O randomly with uniform probability.

In order to obtain an descendant, at most one mutation is applied to the corresponding predecessor. The mutation that will be applied is randomly chosen with uniform

distribution. Once the operation has been chosen, it will be applied depending on its application rate.

If the descendant obtained after mutating a selected individual is not a valid individual (that is to say, some operation of its sequence of genes is not defined for its corresponding word), then the selected individual will be considered as the descendant.

6.2.3 Experiments

Matlab programming language has been used to implement the algorithm and simulations have been performed in a computer cluster of 32 CPUs (2 GHz).

N_c has been settled to 100 (variations of this parameter have not significant effect in the simulation results as processed inert resources are replaced by new ones). For each subenvironment, its derivation is randomly generated in the range

$$\{[D/50], [D/50]+1, \ldots, 2[D/50]\}.$$

α has been settled to 0.5 (since smaller and higher values of α cause massive extinction of the population with a higher probability). The application rates of the operations are 1 in the case of the changing operation and 0.01 in the case of both the deleting and the adding operations (since these are more disruptive than the changing operation). γ has been settled to 2 (if $\gamma = 1$, then individuals do not need to be near to the resource to get enough energy to leave offspring and if $\gamma > 2$, then the energy that individuals lose in the journey to approach a resource is too much and hardly they get enough energy to leave offspring).

Once N_e CURLs have been randomly located in the environment and $N_e \cdot N_c$ natural numbers have been generated by them, the initial population is generated. This consists of 500 individuals such that their genotypes are vectors $v \in C \times O$ (that is to say, only one operation is applied over the zygote). Moreover, $E_0(L(w)) = 0$ for any individual $L(w)$ in the initial population.

Each time, an individual is randomly chosen from the population. The selected individual attempts to process a resource to get enough energy to leave offspring. After processing a resource the individual gains energy, the inert resource is removed from the environment and a new inert resource is generated by the same subenvironment that generated the removed inert resource. Moreover, if the individual has enough energy

to leave offspring, then its descendants will be included in the population and the predecessor will be removed. In the case in which the individual has not enough energy to leave offspring, the energy of the individual will be updated (according to definition 23) and the set of inert resources in the environment does not change. Individuals with negative energy are removed from the population. All this process, from the individual is selected to its energy is updated and descendants (if they exist) are located in the environment, will be called evaluation.

The algorithm will stop either when the size of the population does not increase in 20.000 evaluations or when the size of the population has increased, but the individuals distribution remains almost the same.

In order to use the usual terminology of evolutionary algorithm, we will consider that one generation has gone by when 1000 evaluations have been performed.

6.2.3.1 Simulation Results

Since individuals can only gain energy from inert resources, the individuals of the population tend to assemble around the subenvironments. Thus, when there is a unique subenvironment, then finally the whole population is concentrated in the same area of the environment (around the unique subenvironment). This can be observed in Figure 6.4, where the circle represents two standard deviations from the position of the subenvironment (it accounts for about 95% of the inert resources that can be generated by the subenvironment) and the black points are the individuals of the population.

Moreover, population size undergoes a big decreasing in the first generations and after that, its behavior is increasing until it remains almost unchanged at the end (it can be seen in Figure 6.5). This is due to both the spatial limitation (each point of the space can contain at most one individual) and individuals that are far from subenvironments need a lot of energy to cross the path to reach an inert resource.

The state complexity of a CURL is defined as the number of states of the minimal CUDFA that represents it. Thus, we can study the behavior of the complexity of the population during the evolution. We have observed that the behavior of the average complexity is increasing (it can be seen in Figure 6.5).

Similar results have been obtained when $N_e > 1$. Figure 6.6 shows how the individuals are concentrated around the different subenvironments. Moreover, some subenvironments are empty due to none individual has been able to process the inert resources

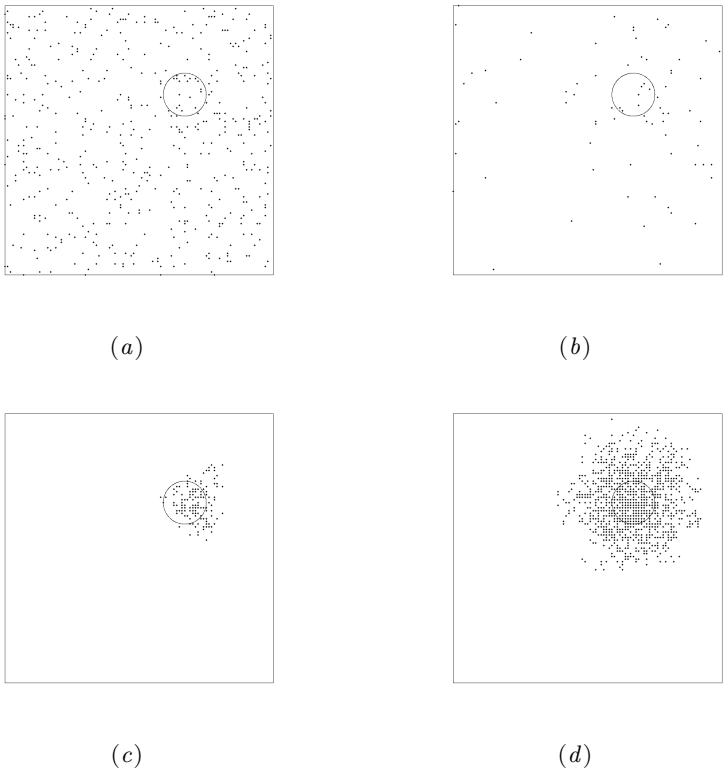


Figure 6.4: Figures a, b, c and d show generation 0, 2, 5 and 69, respectively, in an experiment where $D = 100$ and $N_e = 1$.

generated by it in the first evaluations and the individuals which are fitted to other subenvironments hardly can move far from them because they would lose energy in the journey. In this case, although the average complexity and the population size have the same behavior as when N_e is smaller, an increment in the population size and a decrement in the average complexity have been observed. That is to say, the bigger the number of subenvironments is, the slower the average complexity increases. We will study this later.

Moreover, similar results can be observed in experiments in which $D > 100$.

In order to study the diversity dynamics, some preliminary studies of the dynamics of the species during the evolution have been done. We say that two individuals belong to the same species if and only if they are at least 90% similar. For this purpose, the similarity measure for CURLs defined in (24) has been used. In general, it has been obtained that the number of species increases during the evolution until it remains

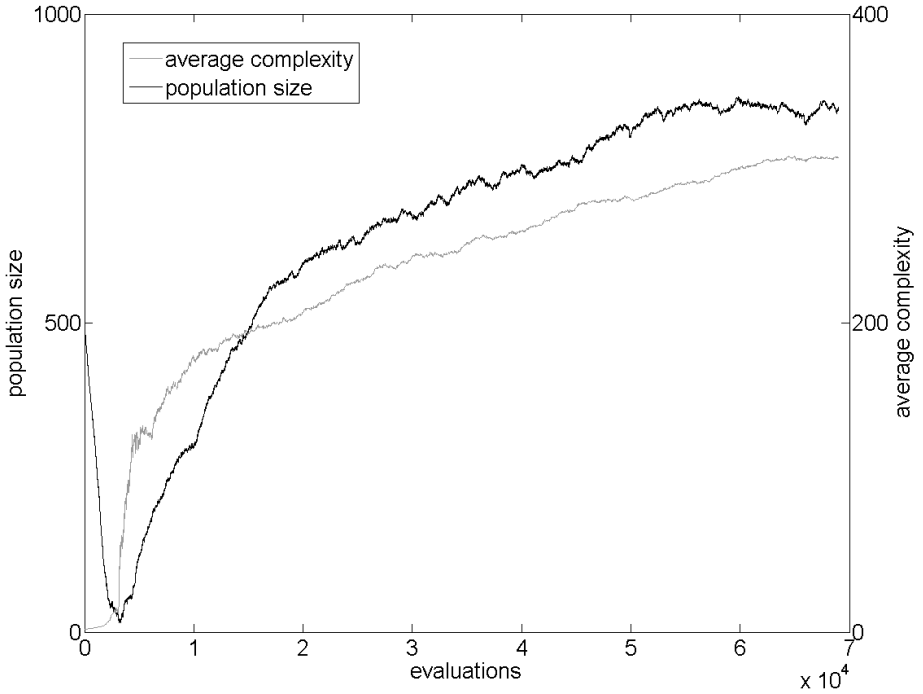


Figure 6.5: The graphics on the right and the left axis respectively show average complexity of the population and population size during the evolution of the experiment shows in Figure 6.4.

almost constant. In general, this behaviour is directly proportional to the behaviour of the population size. This can be seen in Figure 6.7.

6.2.3.2 Inserting Depredation in the Model

In this section, we will introduce depredation in the model as a way of studying the effects of having interactions between individuals. That is to say, individuals not only are able to gain energy from inert resources, but also from processing other individuals. Therefore, we will say that individuals are non-inert resources. Thus, in this case, there are two types of resources available in the environment: non-inert resources (individuals) and inert resources (generated by the subenvironments).

Therefore, in this case, along with the processing of inert resources, an individual $L(w)$ can:

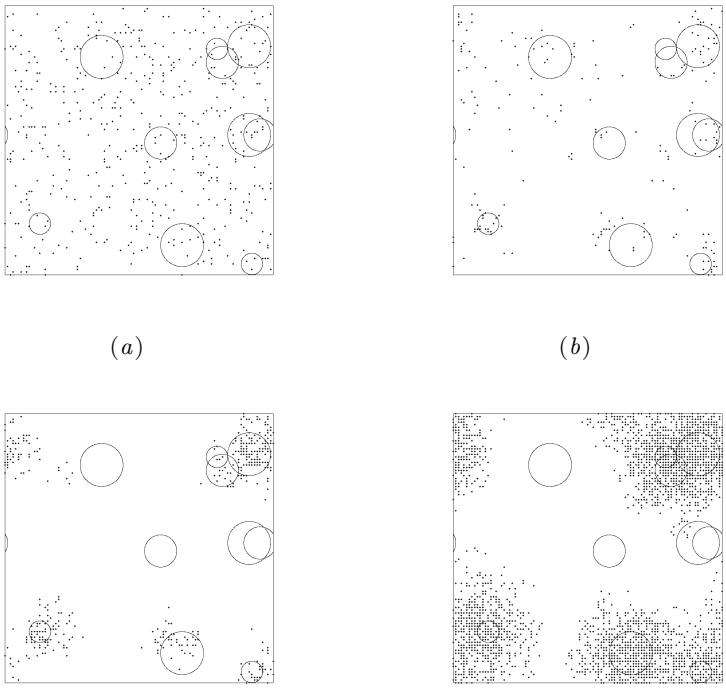


Figure 6.6: Figures a, b, c and d show generation 0, 3, 10 and 136, respectively, in an experiment where $D = 100$ and $N_e = 10$.

- attack another individual: $L(w)$ attempts to process another individual (non-inert resource) of the population.

- react before the attack of another individual: $L(w)$ also attacks the individual that is attacking it.

The previous definition of genotype is extended by adding two new genes related to the predatory behavior of individuals. Now, the genotype of an individual $L(w)$ is defined by a vector $v \in C \times O^k \times T^2$ where $T = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. T^2 defines the predatory behavior of $L(w)$. The first component indicates the depredation rate, that is to say, the probability of $L(w)$ attempts to process either an inert or a non-inert resource and the second component indicates the reaction rate, that is to say, the probability of $L(w)$ reacts before the attack of another individual.

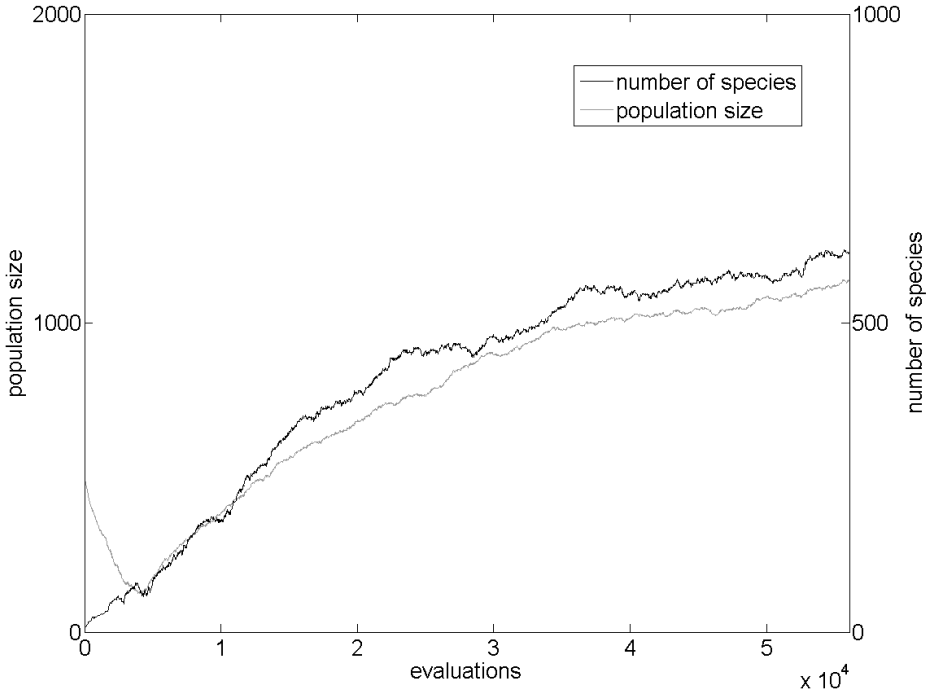


Figure 6.7: Population size and number of species in an experiment with ten subenvironments.

Thus, the genotype of an individual $L(w)$ is a vector $(x, O_1, \ldots, O_p, t_1, t_2)$ of size at least four where $w = O_p(\ldots(O_2(O_1(x)))\ldots)$ with depredation rate t_1 and reaction rate t_2.

For example, the individual $L(011010101)$ with depredation rate 0.9 (it tries to process mostly non-inert resources) and reaction rate 0.5 (it reacts only half the time of being attacked) is represented by the vector $(0, M_{1,1,0,1}, I_{2,1,1}, D_{5,3}, 0.9, 0.5)$.

An individual $L(w)$ attempts to process either an inert or a non-inert resource depending on its depredation rate. Since, in this case, individuals can also gain energy from non-inert resources, let us see some definitions of how to calculate it.

Definition 25. *Let $L(w)$ be an individual, with $w \in Q$, and let $L(w')$ be a non-inert resource, with $w' \in Q$. Let $S = \{n_i \in \mathbb{N} \mid n_i \in L(w') \text{ for any } i = 1, \ldots, \delta\}$, i.e., S is a sample set of $L(w') = \{b_i + |w'|k \mid b_i \in B(w'),\ k \in \mathbb{N}\}$, the numbers are randomly chosen from $\{b_i + |w'|k \mid b_i \in B(w'),\ 1 \leq k \leq m\}$ with uniform probability, where δ and m are a parameters in the model. We define the energy consumed by $L(w)$ when*

attempt to process $L(w')$ *as*

$$E'_c(L(w), L(w')) = \frac{\sum_{n \in N} E_c(L(w), n)}{||N||}$$

where $N = \{n_i \in M \mid n_i \notin L(w)\}$. *We define the energy acquired by* $L(w)$ *from* $L(w')$ *as*

$$E'_a(L(w), L(w')) = \frac{\sum_{n \in A} E_a(L(w), n)}{||A||}$$

where $A = M - N$.

In the experiments, δ and m have been settled to 10 and 100, respectively. Higher values of δ and m have not significant effects in the simulation results but the computation time increases noticeably. However, smaller values of δ and m promote that the complexity of the individuals remains almost unchanged during the whole evolution, that is to say, complex individuals hardly can emerge.

An individual $L(w')$ does react or not before the attack of another individual $L(w)$ depending on its reaction rate. If $w' = w$, then nothing happens.

Definition 26. *Given an individual* $L(w)$ *and a non-inert resource* $L(w')$, *with* $w \neq w'$,

a. if $L(w')$ *does not react, then the energy of* $L(w)$ *after attempting to process* $L(w')$ *in the evaluation* t *is defined as*

$$E_t(L(w)) = E_{t-1}(L(w))p(L(w), L(w'))^{1/\gamma} + R$$

where

$$R = \begin{cases} E'_a(L(w), L(w')) & \text{if } E'_a(L(w), L(w')) \geq 0 \\ -E'_c(L(w), L(w')) & \text{in other case} \end{cases}$$

Moreover, if $E'_a(L(w), L(w')) \geq 0$, *then the individual* $L(w')$ *is removed from the population.*

b. if $L(w')$ *reacts, then the energy of* $L(w)$ *after attempting to process* $L(w')$ *in the evaluation* t *and the energy of* $L(w')$ *after being attacked by* $L(w)$ *in the evaluation* t *are defined as*

- *if* $E'_a(L(w), L(w')) \geq E'_a(L(w), L(w'))$, *then*

$$E_t(L(w)) = E_{t-1}(L(w))p(L(w), L(w'))^{1/\gamma} + E'_a(L(w), L(w'))$$

$$E_t(L(w')) = E_{t-1}(L(w')) - E'_c(L(w), L(w'))$$

- *if* $E'_a(L(w), L(w')) \leq E'_a(L(w), L(w'))$, *then*

$$E_t(L(w')) = E_{t-1}(L(w')) + E'_a(L(w), L(w'))$$

$$E_t(L(w)) = E_{t-1}(L(w))p(L(w), L(w'))^{1/\gamma} - E'_c(L(w), L(w'))$$

- *if* $E'_a(L(w), L(w')) = E'_a(L(w), L(w'))$, *then*

$$E_t(L(w)) = E_{t-1}(L(w))p(L(w), L(w'))^{1/\gamma} - E'_c(L(w), L(w'))$$

$$E_t(L(w')) = E_{t-1}(L(w')) - E'_c(L(w), L(w'))$$

where $d(L(w), L(w'))$ *is the Euclidean distance between the positions of* $L(w)$ *and* $L(w')$.

We take note that the attacked individual, $L(w')$, is always stationary (the depredator approaches it), therefore, it does not lose energy by displacement. In Figure 6.8, an example in which an individual $L(w)$ acquires energy by processing a non-inert resource that reacts can be seen.

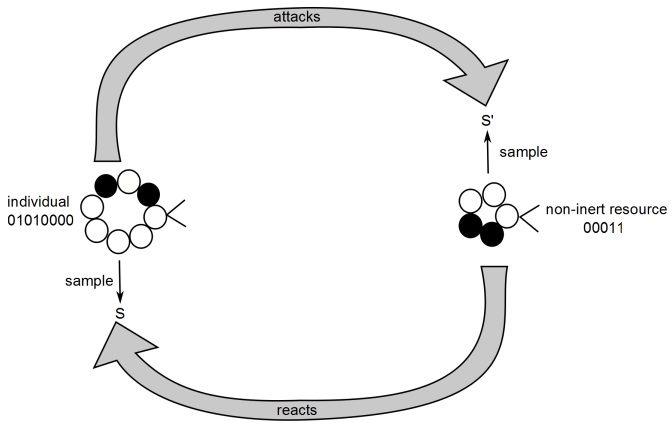


Figure 6.8: The individual $L(w)$ with $w = 01010000$ attacks the non-inert resource $L(w')$ with $w' = 000111$ and it reacts. S and S' are the sample sets of $L(w)$ and $L(w')$, respectively. Thus, the energy of both individuals will be updated by using definition 26*b*.

Essentially, we divide the experiments with depredation into two groups:

- non-fixed rates: both the depredation and the reaction rate of the individuals in the initial population are greater or equal to zero and can be mutated during the evolution by the mutation operations.

- fixed rates: both the depredation and the reaction rate of the individuals in the initial population are greater than zero and fixed during the whole evolution.

Now, mutations will be applied in the following way. Let v be the genotype of an individual $L(w)$. As before, the position of the gene that will be mutated is selected with uniform probability between the second and the last component of v. In the case in which the selected position is between the second and the $(\mid v \mid -2)$-nd component of v, the mutation is applied as showing before. However, in the case in which the selected position is between the last two component of v, only the changing operation can be applied and the new gene will be chosen from T randomly with uniform probability.

Thus, in this section, we will study some properties of the model when depredation happens (either starting with individuals with rates greater than zero or with rates zero but mutating them during the evolution).

In this case, individuals can gain energy from both inert and non-inert resources (it is shown in Figure 6.9). For that reason, although the individuals continue placed in the neighborhood of some subenvironments, they are not so condensed in the center of them than before. The reason for this is that in this case, with depredation, some of them do not need to go near to a subenvironment to find a resource as they can attempt to process other individuals that are in their neighborhoods. However, there are not isolated communities of only depredation, that means, it is necessary that in the community both depredators and non-depredators coexist. In order to differentiate the depredation rate of each individual of the population, we will use points of different colors depending on the depredation rate of the individual. The used palette of colors goes from blue to red passing for green that correspond to depredation rate values from 0 to 1, respectively.

Figure 6.10 shows the last generation (the behavior during the evolution is similar to the case without depredation and for that reason is not shown) of an experiment in which the same environment as in Figure 6.4 has been used, but in this case, both the depredation and the reaction rate of the individual in the initial population are zero and they are mutated during the evolution. It can be observed that the individuals accumulation around the subenvironment is smaller and moreover, there are individuals that are placed further from the subenvironment than before. Similar results has been obtained in experiments where both the depredation and the reaction rate are greater than zero in the initial population and they are mutated during the evolution.

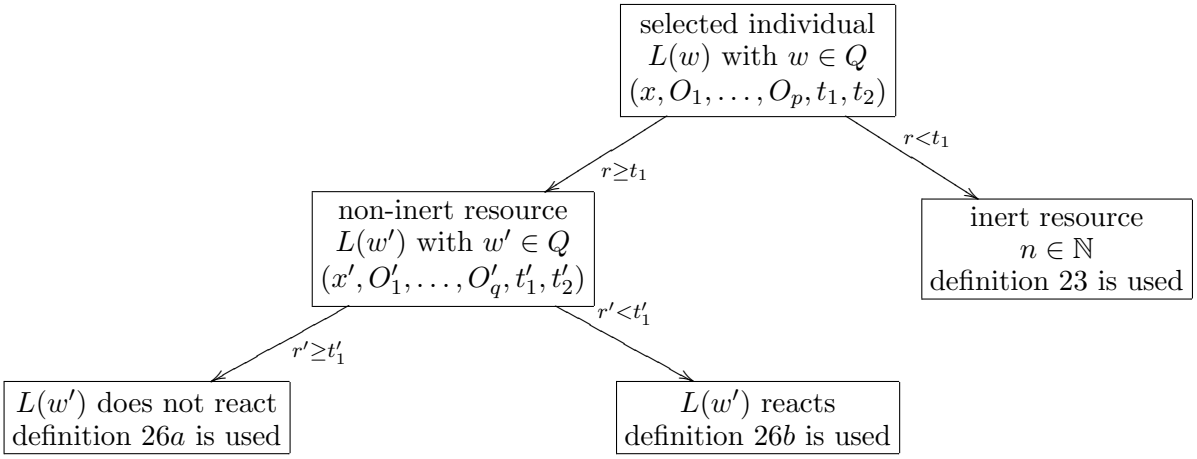


Figure 6.9: The diagram shows how the energy of a selected individual $L(w)$ is updated where $r, r' \in [0, 1]$ are randomly generated in each evaluation.

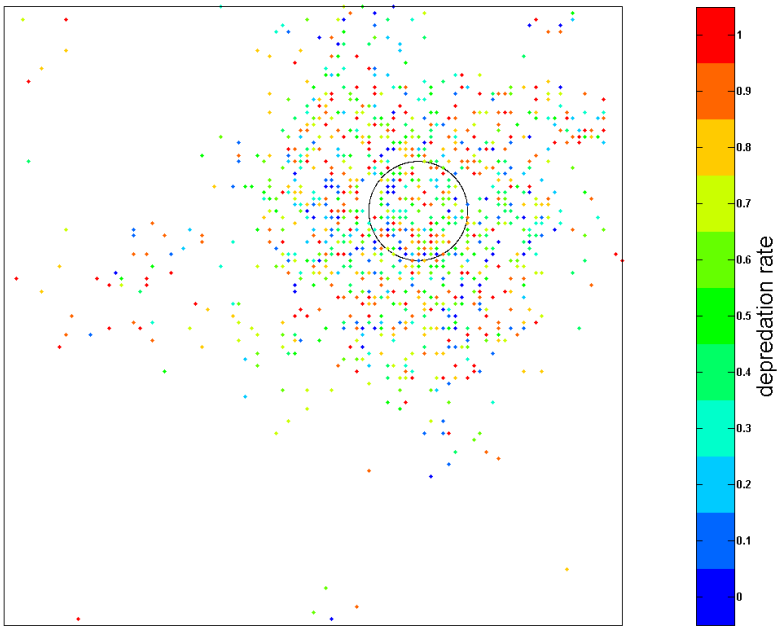


Figure 6.10: Generation 76 of an experiment in which the same environment as in Figure 6.4 has been used, but depredation has been introduced.

The behavior of population size is the same as in experiments without depredation (see Figure 6.11). Moreover, the behavior of the average complexity of the individuals

in the population is also increasing, but in this case, more complex individuals than before are obtained (see Figure 6.11). That is to say, the non-fixed predatory component propitiates individuals more complex. This will be studied later.

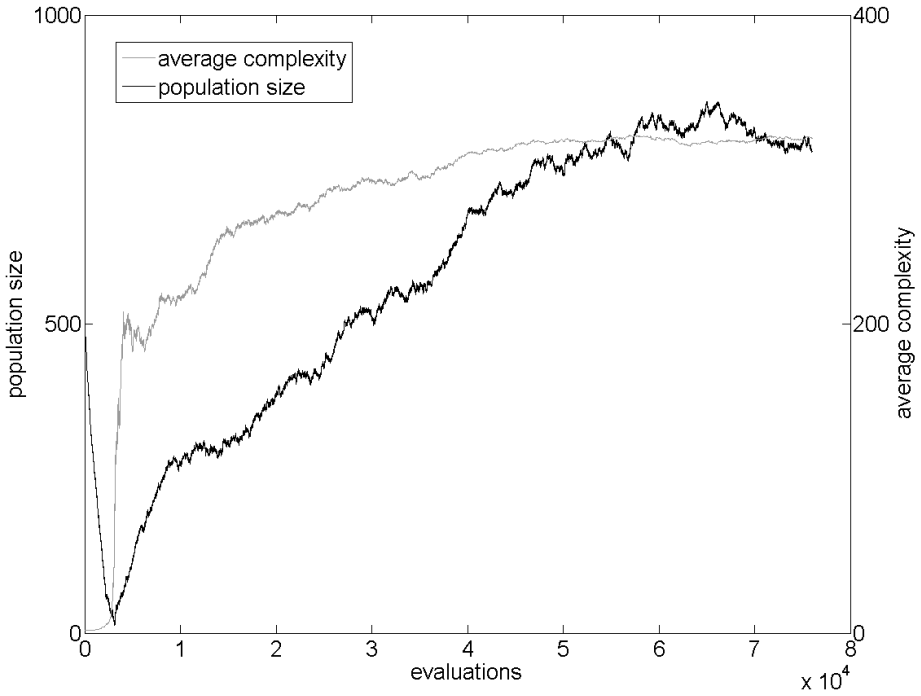


Figure 6.11: The graphics on the right and the left axis respectively show average complexity of the population and population size during the evolution of the experiment shows in Figure 6.10.

Similar results have been obtained when $N_e > 1$. Figure 6.12 shows the last generation of an experiment in which the same environment as in Figure 6.6 has been used, but in this case, both the depredation and the reaction rate of the individual in the initial population are zero and they are mutated during the evolution. It can be observed how, in this case, the individuals are located not only around the different subenvironments. Moreover, although in the first evaluations individuals are coexisting in a certain subenvironment, they have the possibility of moving into other subenvironments during the evolution. This is due to depredators do not need to be near to a subenvironment to gain energy. In this case, more generations are necessary to the population size stops growing. Moreover, as in the case without depredation, although

the average complexity and the population size have the same behavior as when N_e is smaller, an increment in the population size and a decrement in the average complexity have been observed.

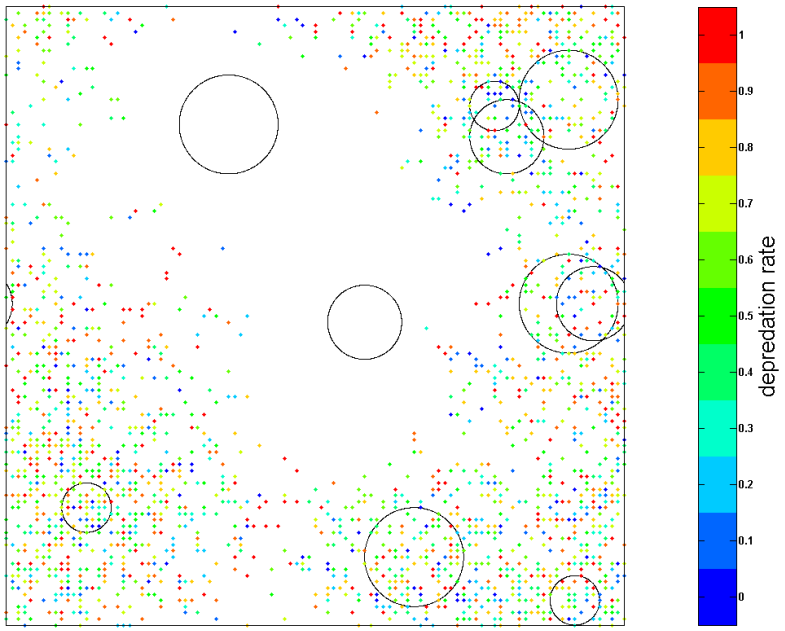


Figure 6.12: Generation 140 of an experiment in which the same environment as in Figure 6.6 has been used.

In experiments with non-fixed depredation and reactive rate, we have observed that the population evolves to an attractor. This attractor moves to the left in the phase space (where the degrees of freedom are the average depredation rate and the average reaction rate) when the number of subenvironments increases. Figure 6.13 shows how in an experiment with $N_e = 1$ the attractor is the set in which the average depredation rate is between 0.4 and 0.6 and the average reaction rate is between 0.5 and 0.7.

Finally, we have observed that when both the depredation and the reaction rate are greater than zero in the initial population and fixed during the evolution, the nearer the rates are to one, the smaller the average complexity of the population is. Moreover, the higher the rates are, the more groups of individuals coexisting far away from the subenvironments exist. Thus, the increasing of population size does not stop until the

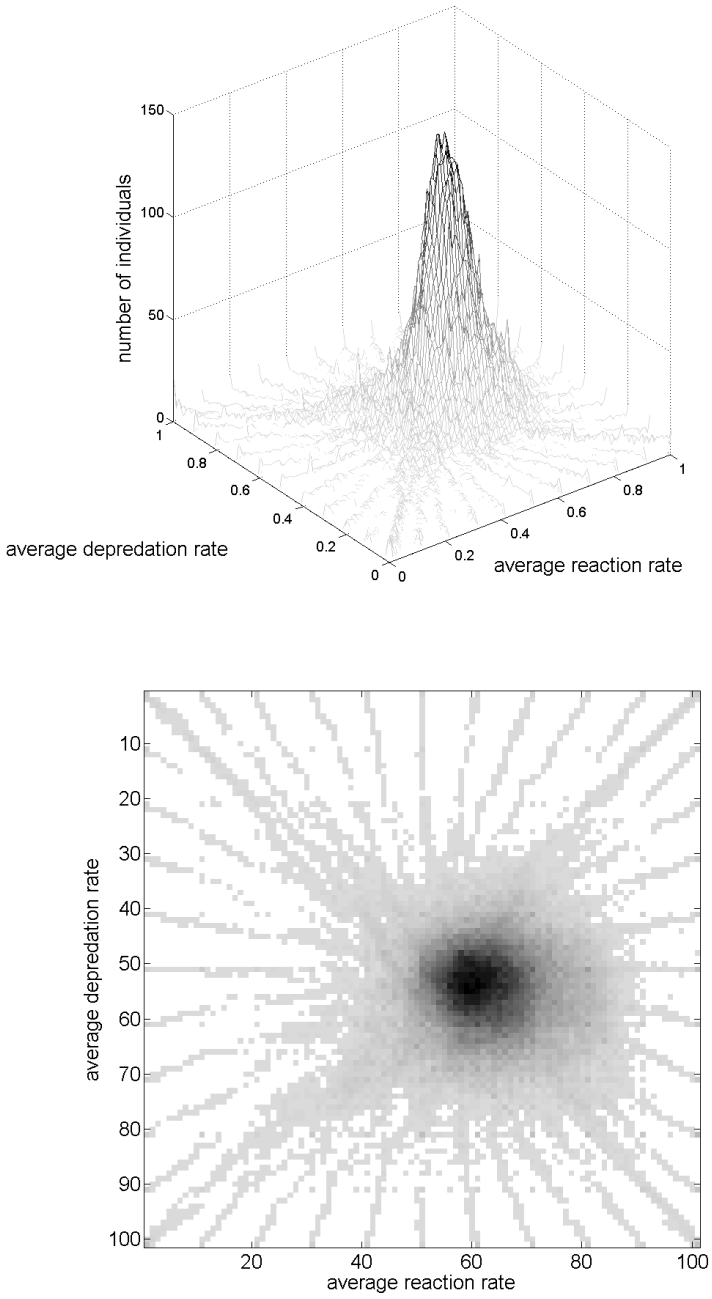


Figure 6.13: 3D and 2D histograms: average depredation rate vs average reaction rate. Experiments with $N_e = 1$.

whole space is completed and is much faster than before. This can be observed in Figure 6.14 and Figure 6.15, where the same environment as in Figure 6.4 has been used and both the depredation and reaction rate are one and fixed during the whole evolution.

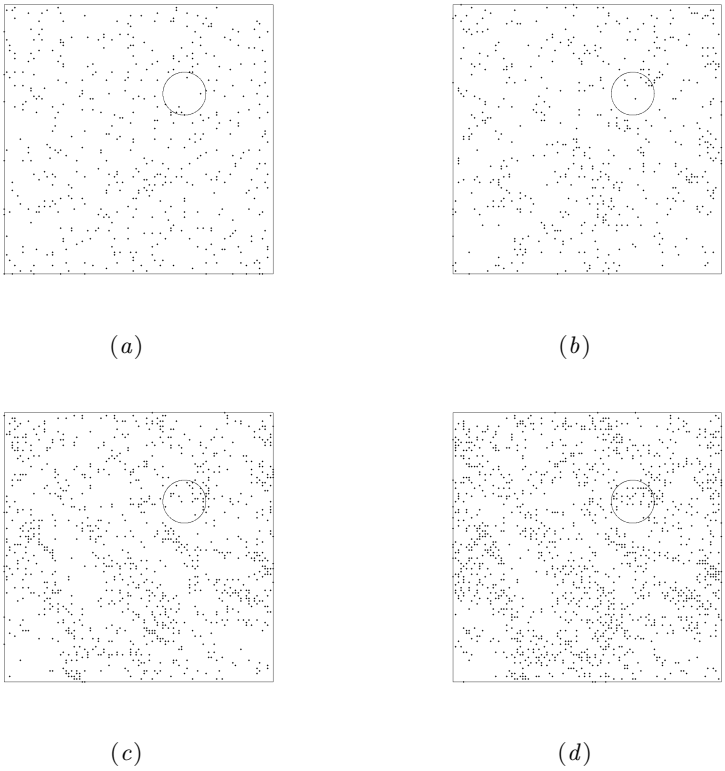


Figure 6.14: Figures *a*, *b*, *c* and *d* show generation 0, 2, 4 and 6, respectively, in an experiment in which the same environment as in Figure 6.4 has been used, but a fixed depredation rate has been introduced.

Moreover, we have observed a high percentage of massive extinctions when both the depredation and reaction rate are greater than 0 and mainly, when they can be mutated during the evolution.

6.2.3.3 Dynamics of the Complexity of the Population

In the previous sections, we have shown that the behavior of the average complexity of the population is increasing. Let us see to what extent the complexity

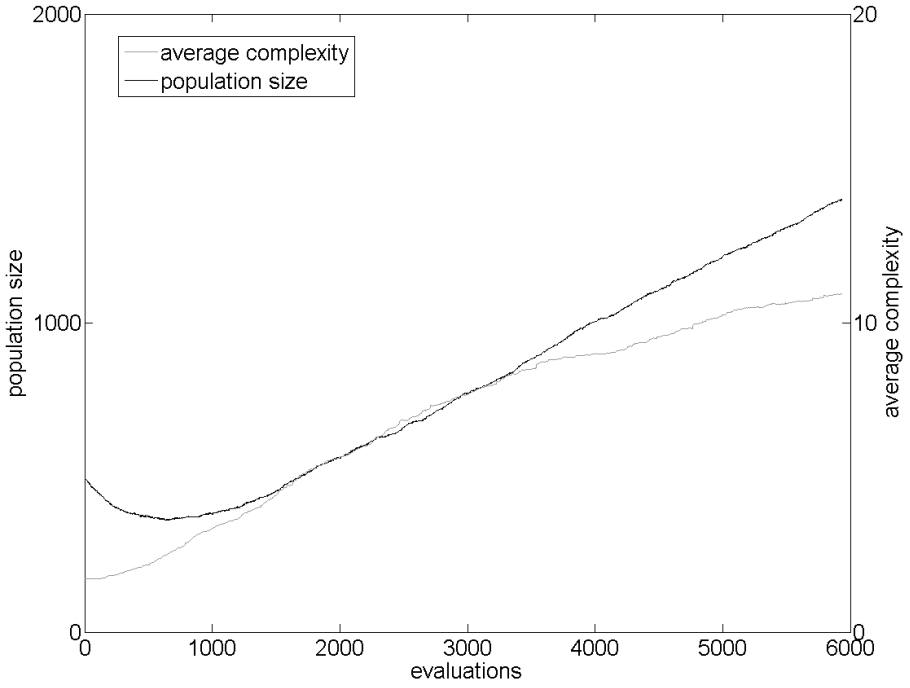


Figure 6.15: The graphics on the right and the left axis respectively show average complexity of the population and population size during the evolution of the experiment shows in Figure 6.14.

of the environment and the predatory behavior of individuals affect the complexity of the population.

We can understand by complexity of the environment either the average complexity of its subenvironments or the number of subenvironments that compose it. Figure 6.16 shows how the complexity of the subenvironments affects the complexity of the population. Three different cases have been studied: complexity of the subenvironment smaller than 300, between 300 and 700 and higher than 700. For each of them, the mean of the average complexity in a set of 10 experiments with $N_e = 1$ is showed. The results show that the more complex the environment is, the more complex the individuals of the population are (although the difference is not very strong).

In the case in which we consider the number of subenvironments as measure of the complexity of the environment, we have obtained that, in general, the bigger the number of subenvironments is, the slower the average complexity increases (although,

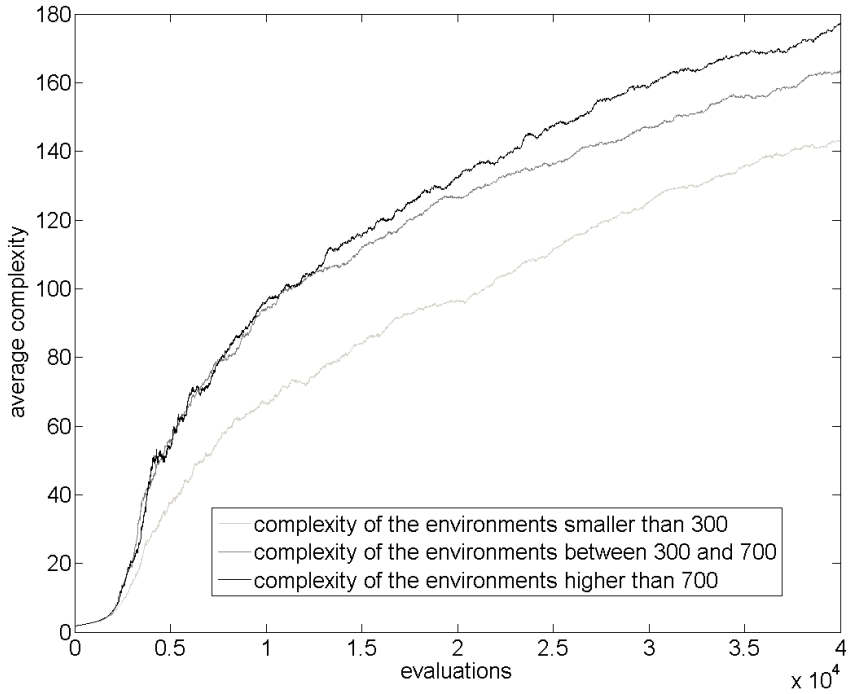


Figure 6.16: Average complexity of the population depending on the complexity of the environment in experiments with $N_e = 1$.

as in the previous case, the difference is not very strong). It can be seen in Figure 6.17, where ten different cases have been studied (from one to ten subenvironments). For each of them, the mean of the average complexity in a set of 10 experiments is showed.

Finally, let us study to what extent the predatory behavior of individuals affects the complexity of the population. Figure 6.18 shows that individuals are much more complex when there exists depredation in the model (for a set of 10 different environments, the mean of their average complexities in simulations with and without depredation is showed).

6.2.4 Conclusions

In this section, an IbM consisting of CURLs has been presented. Two different strategies have been proposed to study how the predatory behavior of individuals affects both the complexity and population dynamics. That is to say, the effects of the interactions between individuals in the model has been analyzed.

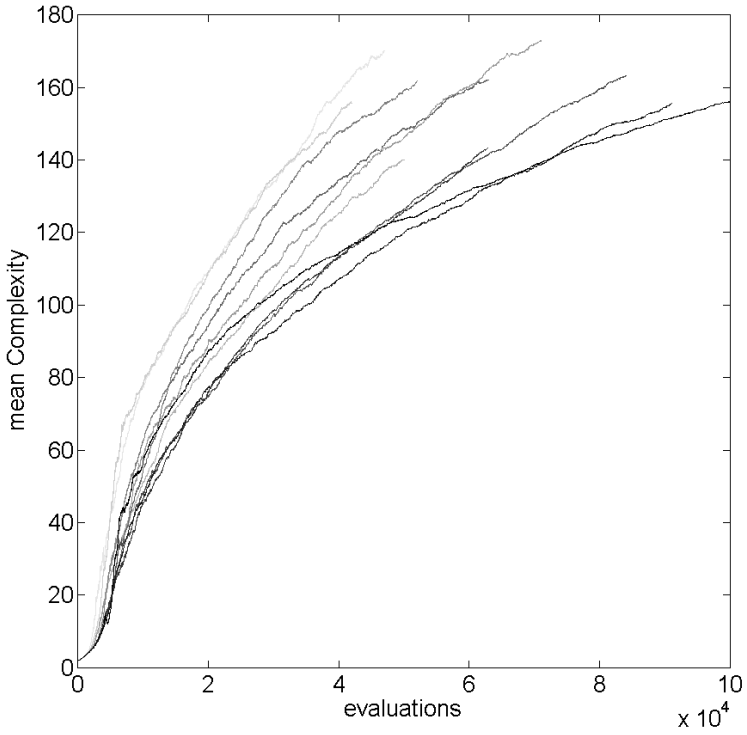


Figure 6.17: Average complexity of the population depending on the number of subenvironments. A gray scale is used: the more subenvironments exist, the darker the line is. Experiments in which there are from one to ten subenvironments have been used.

Although natural selection does not guarantee that organisms will increase in complexity as they evolve, it is widely accepted that complexity of individual systems has increased during evolution. In the presented model, an increasing behavior of the average complexity of the population has been observed in experiments with and without depredation. Moreover, it has been shown that individuals are more complex when there exists depredation in the model. This could mean that coevolution directly affects the emergence of complex individuals, although it is not the only reason for that (since the complexity is also increasing in experiments without depredation).

To what extent the complexity of the environment affects the complexity of the individuals has been also studied. As a result of the experiments, we have obtained that there exists a strong correlation between the complexity of the population and the complexity of the environment. To study this, we took into account two different

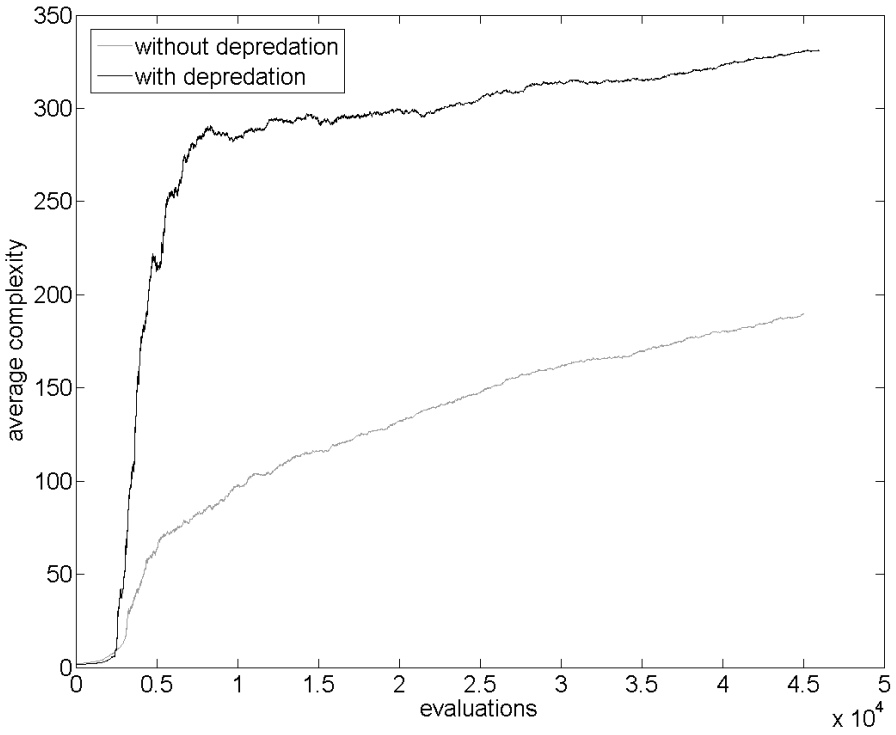


Figure 6.18: Average complexity of the population depending on the predatory behavior of individuals in experiments with $N_e = 1$.

measures of complexity of the environment. In the first one, we considered the average complexity of its subenvironments and the results showed that the more complex the environment is, the more complex the individual of the population are. In the second one, we considered the number of subenvironments and the results showed that, in general, the bigger the number of subenvironments, the slower the average complexity of the population increases. This points out that the complexity of an individual is locally affected by the surrounding subenvironment more that by the global environment. Moreover, it has been proved that the predatory component also affects the complexity of the individuals promoting a higher complexity of the individuals.

Concerning the population dynamics, the following results have been obtained in experiments with depredation. The population evolves towards an attractor when both depredation and reaction rate can be mutated during the evolution. This attractor moves to the left in the phase space (where the degrees of freedom are the average

depredation rate and the average reaction rate) when the number of subenvironments increases (when there is only one subenvironment, then the attractor is the set in which the average depredation rate is between 0.4 and 0.6 and the average reaction rate is between 0.5 and 0.7). Studying whether the introduction of some changes in the environment (that is to say, a dynamical environment) could change the position of such an attractor in the phase space would be interesting. That is to say, studying the stability of the population in a dynamical environment.

In section 6.2.3.1, some preliminary studies of the dynamics of the species is commented. By using the framework that has been proposed in this section, studies on how the number of species are affected by diverse factors could be done. This factors could be either the predatory behavior of individuals or the complexity of the environment or the existence of a dynamical environment, among others. Finally, studies of the cumulative number of persisting species and the relative species abundance could be also done.

Finally, we have pointed out that, by using the proposed framework, a variety of experiments in Theoretical Ecology could be done in a rigorous way.

Chapter 7

Conclusions

7.1 English Version

As it was said at the beginning, the main aim of this thesis was concerned with defining a formal framework where the evolution of biological complexity can be studied in a rigorous way. Throughout this thesis, it has been shown that such an aim has been executed during these almost four years and moreover, in the process of defining that framework, contributions to fields like formal language theory and evolutionary computation have been made. As it has been detailed previously, many of the contributions of this thesis are supported by publications in international journals listed in the JCR index and those non-published are in preparation to be submitted.

Before focusing in the particular contributions of each chapter, the two more relevant aspects of the thesis are highlighted. On the one hand, ecological features have been studied by introducing evolution through evolutionary computation in a field as traditional as the formal language theory is. That is to say, proper elements from evolutionary computation have been translated into formal language terms: CURLs as phenotypes, CUDFAs as genotypes, sequences of operations preserving primitivity as representation of such genotypes over which traditional genetic operations can be applied, similarity measure as fitness function and for speciation, and so on. On the other hand, the other even more important contribution comes from the fact that rigorous studies of the evolution of the complexity can be done in the proposed framework. This rigorous aspect is due to it counts on state complexity that is a well-known and objective measure for CURLs. By using state complexity, we run away from the most

common problem that papers dealing with complexity studies have: rigorous methods to address the problem of measuring the complexity are not provided.

By using the description of CUDFAs by words over $\{0,1\}$, where the zeros represent the non-accepting states of the automaton, and the ones represent the accepting states of the automaton, a characterization of minimal CUDFAs has been presented in Section 2.2: a CUDFA $w \in \{0,1\}^+$ is minimal if and only if w is a primitive word. This introduces a novel relation between CUDFAs and primitive words. For this reason, many properties of primitive words have been studied during this thesis. Since the set of primitive words has been widely studied in the literature, these results (even isolated from the evolutionary side of this thesis) are of interest for a big part of the formal language community.

In spite of the high interest that primitive words cause, in the literature there are only a small number of results concerning the closure of Q_V under operations. In Section 5.2, some operations inspired by biological gene duplication that preserve primitiviy of words have been defined. Essentially, from a given primitive word w, the word ww' is constructed where w' is a modified copy of w or a modified mirror image of w. The operations are chosen in such a way that in the case of a two-letter alphabet, all primitive words of length ≤ 11 can be obtained from single letters. That is to say, a large subset of binary primitive words can be obtained by using them. Obviously, for the ultimate aim of this thesis (studying the evolution of the complexity), generating the whole set of primitive words is not necessary.

Up to now, and despite the attention that the set of primitive words has attracted, whether the language Q_V is context-free remains an open question. Thus, one of the most interesting problems over primitive words is figuring out the classification of Q_V in Chomsky's hierarchy. For this purpose, different generative methods of Q_V have been proposed in the literature, mainly grammatical methods. In Section 5.3, a non-grammatical generative method of Q_V based on basic arithmetic operations has been presented. This method not only provides a way of generating primitive words, but also a relation between Q_V and number theory. Although the aim of this thesis is not the classification of Q_V in Chomsky's hierarchy, since basic arithmetic operations can be formulated in a grammatical way, it has been mentioned that it is possible to convert this method to a grammar that generates Q_V. Thus, this grammar could shed light about the classification of the language Q_V.

In order to study the evolution of the complexity, evolutionary computation has been used to introduce dynamics in a population of CURLs. That is to say, genetic operations have been applied over the genotypes to obtain changes which model the evolution. The most common operations that are used in evolutionary systems are the edit operations of substitution, deletion and insertion of a symbol. However, to our knowledge, there are not many studies that analyze the disruptive effects of such operations when they are applied over the genotypes. That is to say, representations of the genotypes over which the operations cause low disruption are not proposed in most of the cases. From our point of view, this is not logical for two reasons, one biological and other computational: (1) in nature, the rate of fixation of those low-disruption mutations is higher that the rate of fixation of those mutations that change the original phenotype too much, (2) non-random search methods benefit from a low-disruption in the application of operations to refine solutions. One of the main reasons why these studies on disruption are not usually done is the lack of appropriate disruption measures. In Chapter 4, a disruption measure for an operation over a word has been defined by using the similarity measure for CURLs defined in Chapter 3. Thus, studies on the disruption that edit operations produce when they are applied over the genotypes as binary words can be done. Intuitively, the disruption of an operation O with respect to a word w is a pair (a, b) with $a, b \in \mathbb{R}$, where a is the portion of words that are accepted by w and are not accepted by $O(w)$, and b is the portion of words that are accepted by $O(w)$ and are not accepted by w. The results show that edit operations cause a high disruption and consequently, not all words over $\{0, 1\}$ can be obtained by iterated applications of edit operations where each application is accompanied by low disruption. Therefore, an extension of the edit operations has been proposed in order to reduce the disruption. Two new operations (partial copy and partial elimination) inspired by biological gene duplication with no disruption have been introduced. By combining them with edit operations, all words over $\{0, 1\}$ can be obtained by low disruption. Thus, this seems to show to what extent biological gene duplication contributes to reduce the disruption caused by mutations during evolution.

As we said in the previous paragraph, the disruption measure has been defined by using the similarity measure for CURLs defined in Chapter 3. This measure has been defined because the similarity measures that have been proposed in the literature between regular set (such that Bodnarchuk, Baire, Hamming or information distance)

cannot be applied here, since the following principle is not satisfied: sets are more similar if they have more elements in common. Thus, the measure proposed in this thesis computes the portion of words that are shared by two CURLs. It has also been proved that it is an upper bound of the well-known Jaccard coefficient and the Sørensen coefficient. The proposed measure was mainly defined because a measure that calculates the disruption of genetic operations over the genotypes was needed. Moreover, how well an individual (CURL) adapts to a given environment was also needed to study the dynamics of the complexity during the evolution of such individuals and this can be calculated by using this measure. In this thesis, such a similarity measure has also been used in Chapter 6 to study the dynamics of the species. However, other applications, such as grammatical inference and retrieval theory, can be found for this measure. Finally, it has been proved that the dissimilarity measure for CURLs that has been defined by using the proposed similarity measure is a semimetric distance. Moreover, we have shown that both of them can be used also for URLs, although in this case the dissimilarity measure is not a semimetric distance, but a symmetric distance.

Thus, we have identified all the necessary components for evolutionary computation in formal language terms, in particular, we have a formal framework based on CURLs in which objective studies on the evolution of the complexity can be done. The main interest in such studies comes from the accepted idea that the biological complexity of certain lineages has increased during evolution (although natural selection described by Charles Darwin does not guarantee such an increasing). Although many scientists have modeled evolving artificial ecosystems in order to make a case for, or against, a trend in the evolution of complexity, most of these results have been questioned because the way in which the complexity is calculated is not rigorous enough or even measurable (this is the case of the number of genes, Shannon's entropy, Kolmogorov complexity, and others). In the framework proposed in this thesis, we have not this problem, since state complexity can be used and it is a well-known and an objective complexity measure. In order to study the tendency in the evolution of complexity, two preliminary studies on the behavior of the complexity by using a simple framework have been shown in Section 6.1. The first one uses a greedy algorithm and the edit operations over binary words. The second one uses a genetic algorithm and the edit operations over sequences of the operations that preserve the minimality. It has been observed that the behavior of the complexity increases in both of them, but in the second approach, the correlation

between the adaptation level (similarity) to the environment and the complexity of the individual is stronger. Taking into account this result, and the fact that the minimality of automata is preserved when the genetic operations are applied over these sequences of operations (individuals with the same complexity have the same number of states), the determination of representing the genotypes as sequences of operations preserving primitivity was unavoidable.

Finally, since knowing the factors that propitiate such an increasing behavior of the complexity is important, in Section 6.2, an IbM consisting of CURLs has been presented. Essentially, a population of CURLs with low complexity is placed in an artificial ecosystem that is compounded of a set of CURLs (subenvironments) randomly placed in a torus. The individuals compete by inert (obtained from the subenvironments) and non-inert resources (depredation between individuals). By using this model, the effects of the interactions between individuals and between the individuals and the environment have been analyzed. While the individuals tend to assemble around the subenvironments when there is not depredation in the model, they are placed further from the subenvironments when depredation is introduced. In general, we have concluded that the more predatory the model is, the smaller the concentration around the subenvironments. In both cases (with and without depredation), population size undergoes a big decreasing in the first generations and after that its behavior is increasing until it remains almost unchanged at the end. When both depredation and reaction rate can be mutated during the evolution, we have observed that the population evolves towards an attractor that moves to the left in the phase space when the number of subenvironments increases. Concerning complexity, we have shown that there exists a dependency between the complexity of the environment and the complexity of the individuals. Thus, when we consider the average complexity of the subenvironments as the complexity measure for the environment, we have obtained that the more complex the environment is, the more complex the individuals are. On the other hand, when we consider the number of subenvironments as the complexity measure for the environment, we have obtained that, in general, the bigger the number of subenviornments, the slower the average complexity of the population increases. It can be understood as the complexity of an individual is locally affected by the surrounding subenvironment more than by the global environment. Moreover, it has been proved that the predatory

component also affects the complexity of the individuals, obtaining more complex individuals when there exists depredation in the model. By using the similarity measure defined in Chapter 3, some preliminary studies on the dynamics of the species have been done (two individuals belong to the same species if and only if they are at least 90% similar). It has been shown that, in general, there exists a strong correlation between the population size and the number of species.

By using the framework proposed in this thesis, a variety of experiments in Theoretical Ecology can now be proposed and performed rigorously.

7.2 Spanish Version

Como se dijo al principio, el objetivo principal de esta tesis era la definición de un marco de trabajo formal donde la evolución de la complejidad biológica pudiera estudiarse de una forma rigurosa. A lo largo de esta tesis, se ha podido comprobar que este objetivo ha sido alcanzado en el transcurso de estos casi cuatro años y además, en el proceso de definición de tal marco de trabajo, se han realizado diversas contribuciones en el ámbito de la teoría de lenguajes formales y de la computación evolutiva. Como ha sido detallado anteriormente, muchas de las contribuciones presentadas en esta tesis están avaladas por publicaciones en revistas internacionales indexadas en el JCR y aquellas contribuciones aún no publicadas, están en preparación para ser enviadas.

Antes de centrarnos en las contribuciones particulares de cada capítulo, señalaremos los dos aspectos de mayor relevancia de esta tesis. En primer lugar, se han hecho experimentos ecológicos mediante la introducción de evolución usando computación evolutiva en un campo tan tradicional como es el campo de la teoría de lenguajes formales. Es decir, elementos propios de la computación evolutiva han sido definidos en términos de lenguajes formales: CURLs como fenotipos, CUDFAs como genotipos, secuencias de operadores que preservan primitividad como representación de tales genotipos sobre las cuales se pueden aplicar los operadores genéticos tradicionales, medida de similitud para los CURLs como función de bondad y para hacer estudios de especiación, etc. En segundo lugar, la otra contribución, de mayor importancia si cabe, viene dada por el hecho de que en el marco de trabajo propuesto pueden hacerse estudios rigurosos de la evolución de la complejidad. Esta rigurosidad se debe a que la conocida y objetiva medida de complejidad de estados definida para lenguajes regulares puede usarse como

medida de complejidad en nuestros estudios. De esta manera, se evita el problema más común que tienen la mayoría de los artículos que presentan estudios de complejidad: no proporcionan métodos rigurosos para abordar el problema de medir la complejidad.

Gracias a la descripción propuesta de los CUDFAs como palabras binarias sobre el alfabeto $\{0, 1\}$, donde los ceros representan los estados de no aceptación del autómata, y los unos representan los estados de aceptación del autómata, en la Secion 2.2 se presenta una caracterización de los CUDFAs mínimos: un CUDFA $w \in \{0, 1\}^+$ es mínimo si y sólo si w es una palabra primitiva. Este resultado introduce una relación novedosa entre los CUDFAs y las palabras primitivas que ha provocado que en esta tesis se estudien diversas propiedades de las palabras primitivas. Estos resultados (incluso aislados de la parte evolutiva de esta tesis) son de interés para una gran parte de la comunidad que estudia los lenguajes formales debido a que el lenguaje de las palabras primitivas ha sido ampliamente estudiado en la literatura.

A pesar del gran interés que suscitan las palabras primitivas, el número de resultados que pueden encontrarse en la literatura relativos al cierre de Q bajo ciertas operaciones es pequeño. En la Secion 5.2, se han definido algunas operaciones inspiradas por la duplicación biológica de genes que preservan la primitividad de las palabras. En esencia, para una palabra primitiva w dada, se construye la palabra ww' donde w' es una copia de w modificada o una copia espejo de w modificada. Las operaciones han sido elegidas de tal manera que en el caso de alfabetos de dos letras, todas las palabras primitivas de longitud ≤ 11 pueden ser generadas a partir de una sola letra. Es decir, usando tales operaciones puede obtenerse un subconjunto grande de palabras primitivas. Claramente, no es necesario generar el conjunto total de palabras primitivas para alcanzar el principal objetivo de esta tesis (estudiar la evolución de la complejidad).

Hasta ahora, y a pesar de la gran atención que ha generado el conjunto de las palabras primitivas, se desconoce si el lenguaje Q es o no independiente del contexto. Asimismo, encontrar la clasificación de Q dentro de la jerarquía de Chomsky es de gran interés. Con este propósito, una gran cantidad de diferentes métodos generativos de Q han sido propuestos en la literatura, principalmente métodos gramaticales. En la Secion 5.3, se ha presentado un método generativo de Q que no es gramatical y está basado en operaciones aritméticas básicas. Este método no sólo proporciona una manera de generar palabras primitivas, sino también proporciona una relación entre Q y la teoría de números. Aunque el objetivo de esta tesis no es la clasificación de Q

dentro de la jerarquía de Chomsky, resaltamos que sería posible convertir este método a una gramática, ya que las operaciones aritméticas básicas pueden formularse de manera gramatical. Esta gramática podría contribuir a esclarecer la clasificación del lenguaje Q.

Para el estudio de la evolución de la complejidad, es necesario introducir una dinámica en una población de CURLs. La computación evolutiva es usada con este propósito, es decir, se aplican operadores genéticos sobre los genotipos para obtener los cambios que modelarán la evolución. Los operadores más comunes que se usan en sistemas evolutivos son los operadores *edit* de substitución, eliminación e inserción de un símbolo. Sin embargo, hasta donde sabemos, no hay muchos estudios que analicen el efecto disruptivo de estos operadores al ser aplicados sobre los genotipos. Es decir, en la mayoría de los casos, no se proponen representaciones de los genotipos sobre las cuales los operadores causen una disrupción baja. Desde nuestro punto de vista, esto no tiene sentido debido a dos razones, una biológica y otra computacional: (1) en la naturaleza, la tasa de fijación de aquellas mutaciones que causan una disrupción baja es más alta que la tasa de fijación de aquellas mutaciones que cambian mucho el fenotipo original, (2) los métodos de búsqueda no aleatoria se benefician de una baja disrupción en la aplicación de los operadores para refinar las soluciones. Una de las principales razones por las que normalmente no se llevan a cabo estos estudios de disrupción es la falta de medidas de disrupción apropiadas. En el Capitulo 4, se ha definido una medida de disrupción para un operador sobre una palabra, para ello se ha usado la medida de similitud para CURLs definida en el Capitulo 3. De esta manera, pueden realizarse estudios sobre la disrupción que los operadores *edit* producen en los genotipos (palabras binarias) que son objeto de estudio en esta tesis. De manera intuitiva, se define la disrupción de un operador O con respecto a una palabra w como un par (a, b) con $a, b \in \mathbb{R}$, donde a es la porción de palabras que son aceptadas por w y no por $O(w)$ y b es la porción de palabras que son aceptadas por $O(w)$ y no por w. Los resultados muestran que los operadores *edit* causan una disrupción alta al ser aplicados sobre los genotipos y por lo tanto, aplicándolos iterativamente no se pueden obtener todas las palabras sobre el alfabeto $\{0, 1\}$ con disrupción baja en cada aplicación de un operador. Como estrategia para reducir la disrupción, se ha propuesto una extensión de los operadores *edit*. Asimismo, se han propuesto dos nuevos operadores (copia parcial y eliminación parcial) que están inspirados por la duplicación biológica de genes y no son

disruptivos. Además, se ha demostrado que usándolos junto con los operadores *edit*, se pueden obtener todas las palabras sobre $\{0, 1\}$ con disrupción baja en la aplicación de cada operador. Este resultado parece mostrar que la duplicación biológica de genes reduce la disrupción causada por las mutaciones durante la evolución.

Como se ha mencionado en el párrafo anterior, la medida de disrupción propuesta está definida usando la medida de similitud para CURLs definida en el Capitulo 3. Esta medida de similitud ha sido definida ya que las medidas de similitud existentes para conjuntos regulares (tales como la de Bodnarchuk, la de Baire, la de Hamming o la relativa a la teoría de información) no nos sirven. Esto es debido a que no satisfacen el siguiente principio: dos conjuntos son más similares, si tienen más elementos en común. Asimismo, la medida de similitud propuesta en esta tesis calcula la porción de palabras que comparten dos CURLs dados. También se ha demostrado que esta medida de similitud es una cota superior de los conocidos coeficientes de Jaccard y de Sørensen. Como se ha explicado antes, esta medida fue definida porque era necesaria una medida que calculara la disrupción de los operadores genéticos. Pero además, dado que para estudiar la dinámica de la complejidad durante la evolución de una población de individuos se necesita calcular cómo de bien adaptados están estos individuos a un entorno dado, esta medida de similitud es también necesaria y puede ser usada en este caso. Esta medida también es usada en la tesis para definir el concepto de especies y estudiar su dinámica. Además de las aplicaciones que esta medida de similitud ha tenido en esta tesis, existen otras muchas aplicaciones como son la inferencia gramatical y la teoría de la recuperación de información. Por último, se ha demostrado que la distancia que puede definirse a partir de la medida de similitud propuesta es una distancia semimétrica. Además, estas medidas también pueden usarse para los URLs, aunque en este caso no es una distancia semimétrica si no una distancia simétrica.

En este punto, tenemos identificados todos los componentes necesarios para la computación evolutiva en términos de lenguajes formales. En particular, tenemos un marco de trabajo formal basado en CURLs en el cual pueden hacerse estudios objetivos sobre la evolución de la complejidad. El principal interés en tales estudios viene de la idea ampliamente aceptada de que, durante la evolución, la complejidad biológica de ciertos linajes ha sido creciente (aunque la selección natural descrita por Charles Darwin no garantiza tal crecimiento). Aunque muchos científicos han modelado ecosistemas artificiales evolutivos con el objetivo de posicionarse a favor o en contra de una tendencia

en la evolución de la complejidad, la mayoría de esos resultados han sido cuestionados porque la forma en la que se calculaba la complejidad no era lo suficientemente rigurosa o incluso medible (este es el caso del número de genes, la entropía de Shannon, la complejidad de Kolmogorov, y otras). En el marco de trabajo propuesto en esta tesis no tenemos este problema, ya que la complejidad de estados puede ser usada y es una medida de complejidad conocida y objetiva. Para estudiar la tendencia en la evolución de la complejidad, en la Sección 6.1 se muestran dos estudios preliminares sobre el comportamiento de la complejidad mediante el uso de un marco de trabajo muy simple. El primer estudio usa un algoritmo voraz y los operadores *edit* sobre palabras binarias. El segundo estudio usa un algoritmo genético y los operadores *edit* sobre secuencias de operadores que preservan la minimalidad. Se ha observado que el comportamiento de la complejidad es creciente en ambos estudios, pero en el segundo la correlación entre la similitud de los individuos con el entorno y la complejidad de estos es mayor. En este punto, la determinación de representar los genotipos como secuencias de operadores que preservan la primitividad se hizo aún más fuerte. Finalmente, y debido a que es importante conocer los factores que propician tal comportamiento hacia una complejidad cada vez mayor, en la Sección 6.2, se ha propuesto un modelo basado en individuos formado por CURLs. En términos generales, una población de CURLs con complejidad baja es colocada en un ecosistema artificial que está compuesto de un conjunto de CURLs (subentornos) aleatoriamente posicionados en un toro. Los individuos compiten por recursos inertes (obtenidos de los subentornos) y por recursos no inertes (depredación entre individuos). Usando este modelo, se analizan los efectos de las interacciones entre individuos y de las interacciones entre los individuos y el entorno. Mientras los individuos tienden a agruparse alrededor de los subentornos cuando no hay depredación en el modelo, cuando esta es introducida, los individuos se posicionan más lejos de los subentornos. En general, mientras mayor es el nivel depredativo del modelo, menor es la acumulación de los individuos alrededor de los subentornos. En ambos casos (con o sin depredación), el tamaño de la población experimenta un gran decrecimiento en las primeras generaciones y después su comportamiento es creciente hasta que se estabiliza al final. Cuando tanto la tasa de depredación como la de reacción pueden ser mutadas durante la evolución, se ha observado que la población evoluciona hacia un atractor que se mueve a la izquierda en el espacio de fases cuando el número de subentornos crece. Con respecto a la complejidad, se ha mostrado que existe una

relación entre la complejidad del entorno y la complejidad de los individuos. De esta manera, cuando se considera como medida de complejidad del entorno la media de la complejidad de los subentornos, se obtiene que mientras más complejo sea el entorno, más complejos son los individuos. Por otro lado, cuando se considera como medida de complejidad del entorno el número de subentornos, se obtiene que mientras mayor sea el número de subentornos, menor es la velocidad de crecimiento de la complejidad de los individuos. Esto puede entenderse como que la complejidad de un individuo está localmente afectada por el subentorno que lo rodea más que goblamente por el entorno. Además, se ha probado que la componente de depredación también afecta a la complejidad de los individuos, obteniendo individuos más complejos cuando existe depredación en el modelo. Usando la medida de similitud definida en el Capitulo 3, se han realizado estudios preliminares sobre la dinámica de las especies (dos individuos son de la misma especie si y sólo si son similares, al menos, un 90%). Se ha mostrado que en general existe una fuerte relación entre el tamaño de la población y el número de especies, creciendo y estabilizandose practicamente en el mismo intervalo.

Además de todos los estudios presentados en la tesis, mediante el uso del marco de trabajo propuesto, pueden realizarse una gran cantidad de experimentos ecológicos de forma rigurosa.

List of Figures

List of Tables

LIST OF TABLES

References

[1] Wikipedia articles on Jaccard coefficient and Sørensen coefficient. 29

[2] C. Adami. What is complexity? *BioEssays*, 24(12):1085–1094, 2002. 107

[3] C. Adami, D. Ofria, and T. Collier. Evolution of Biological Complexity. *In Proc. Nat. Acad. Sci.*, 97(9):4463–4468, 2000. 108

[4] A. Alhazov, J. Dassow, C. Martín-Vide, Yu. Rogozhin, and B. Truthe. On Networks of Evolutionary Processors with Nodes of Two Types. *Fundamenta Informaticae*, 91:1–15, 2009. 48, 56, 96

[5] A. Alhazow, C. Martín-Vide, and Y. Rogozhin. On the Number of Nodes in Universal Networks of Evolutionary Processors. *Acta Informatica*, 43:331–339, 2006. 48, 96

[6] J. Allali and M. F. Sagot. Novel Tree Edit Operations for RNA Secondary Structure Comparison, 2004. 48

[7] J.P. Allouche and J. Shallit. *Automatic Sequences. Theory, Applications, Generalizations.* Cambridge University Press, 2003. 13, 70, 72

[8] T. Bäck, D.B. Fogel, and Z. Michalewics. *Handbook of Evolutionary Computation.* Published in cooperation with the Institute of Physics, 1997. 96

[9] C.M. Bennett, P. Gács, M. Li, P.M.B Vitanyi, and W.H Zurek. Information distance. *IEEE Transactions on Information Theory*, pages 1407–1423, 1998. 28, 46

[10] J. Berstel, L. Boasson, and O. Carton. Hopcroft's automaton minimization algorithm and Sturnian words. In *DMTCS*, pages 351–362, 2008. 28

[11] M. Bleda, H. Brandt, I. Gebeshuber, M. MacPherson, T. Matsuguchi, and S. Számadó. Does complexity always increase during major evolutionary transitions? In *CSSS*, 2003. 107

[12] J. Bonner. *The evolution of complexity by means of natural selection.* Princeton, NJ: Princeton University Press, 1988. 7, 107

[13] S. Bornhofen and C. Lattaud. Outlines of Artificial Life: A Brief History of Evolutionary Individual Based Models. *Lecture notes in computer science*, 3871:226–237, 2006. 108

[14] B. Breckling, F. Muller, H. Reuter, F. Holker, and O. Franzle. Emergent Properties in Individual-Based Ecological Models: Introducing Case Studies in an Ecosystem Research Context. *Ecological modeling*, 186(4):376–388, 2005. 108

[15] M.S. Burtsev. Measuring the Dynamics of Artificial Evolution. *Lecture notes in computer science*, 2801(580-587), 2003. 108

[16] D. Callan. The value of a primitive word. *Math. Monthly*, 107:88–89, 2000. 13, 70

[17] R. Carrasco. Accurate Computation of the Relative Entropy between Stochastic Regular Grammars, RAIRO. *Theoretical Informatics and Applications*, 31(5):437–444, 1997. 10, 28

[18] R.C. Carrasco and J.R. Rico. A Similarity between Probabilistic Tree Languages: Application to XML Document Families. *Pattern Recognition*, 36(9):2197–2199, 2003. 10, 28

[19] J. Castellanos, C. Marín-Vide, V. Mitrana, and J. Sempere. Networks of Evolutionary Processors. *Acta Informatica*, 38:517–529, 2003. 48, 96

[20] J. Castellanos, C. Martín-Vide, V. Mitrana, and J. Sempere. Solving NP-complete Problems with Networks of Evolutionary Processors. *Lecture Notes in Computer Science*, 2084:621–628, 2001. 9, 48, 96

[21] G. Castiglionea, A. Restivo, and M. Sciortinoa. Circular sturmian words and Hopcroft's algorithm. *Theoretical Computer Science*, 410(43):4372–4381, 2009. 28

[22] K. Christensen, S.A. di Collobiano, M. Hall, and H.J. Jensen. Tangled Nature: A Model of Evolutionary Ecology. *Journal of theoretical biology*, 216(1):73–84, 2002. 108

[23] E. Csuhaj-Varjú and V. Mitrana. Evolutionary Systems: A Language Generating Device Inspired by Evolving Communities of Cells. *Acta Informatica*, 36:913–926, 2000. 9, 48, 56, 96

[24] J. Dassow, G.M. Martín, and F.J. Vico. A Similarity Measure for Cyclic Unary Regular Languages. *Fundamenta Informaticae*, 96(1):71â88, 2009. 3, 4, 27, 49, 51, 97, 98, 115

[25] J. Dassow, G.M. Martín, and F.J. Vico. Evolving under Small Disruption. *In Proceedings of the Workshop on Non-Classical Models of Automata and Applications (NCMA09). 17th International Symposium on Fundamentals of Computation Theory, 31/08-1/09, Wroclaw, (Poland)*, pages 91–106, 2009. 2, 5, 47

[26] J. Dassow, G.M. Martín, and F.J. Vico. Some operations preserving primitivity of words. *Theoretical Computer Science*, 410(30-32):2910–2919, 2009. 2, 3, 5, 71, 105

[27] J. Dassow and B. Truthe. On the Power of Networks of Evolutionary Processors. *Lecture notes in computer science*, 4664:158–169, 2007. 48, 96

[28] M. Davey and D. Mackay. Reliable Communication over Channels with Insertions, Deletions, and Substitutions. *IEEE Transactions on Information Theory*, 47:687–698, 2001. 9, 48

[29] Colin de la Higuera. *Grammatical Inference*. Cambridge University Press, 2010. 96

[30] V.S. Dimitrov, T.V. Cooklev, and B.D. Donevsky. Generalized Fermat-Mersenne number theoretic transform. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 41:133–139, 1994. 88

[31] P. Dömösi, D. Hauschidt, G. Horváth, and M. Kudlek. Some results on small context-free grammars generating primitive words. *Publ. Math. Debrecen*, 54:667–686, 1999. 72

[32] P. Dömösi and G. Horváth. The language of primitive words is not regular: two simple proofs. *European Association for Theoretical Computer Science*, 87:191–194, 2005. 13, 28, 70

[33] P. Dömösi, S. Horváth, and M. Ito. Formal languages and primitive words. *Publ. Math. Debrecen*, 42:315–321, 1999. 70

[34] P. Dömösi, S. Horváth, M. Ito, and M. Katsura. Some results on primitive words, palindroms and polyslender languages. *Publ. Math. Debrecen*, 65:13–28, 2004. 70

[35] P. Dömösi, M. Ito, and S. Marcus. Marcus contextual languages consisting of primitive words. *Discrete Mathematics*, 308:4877–4881, 2008. 13, 70

[36] P. Dömösi and C.L. Nehaniv. *Algebraic Theory of Finite Automata Networks: An Introduction*, chapter 3. SIAM Series on Discrete Mathematics and Applications, 2005. 109

[37] L. Eamounts and H.S. Permeabilities. Towards a General Theory of Biodiversity. *Nature*, 410:923–926, 2001. 108

[38] D.P. Feldman and J.P. Crutchfield. Measures of Statistical Complexity: Why? *Physics Letters A*, 283(4-5):244–252, 1998. 7, 108

[39] F. Gecseg and I. Peak. *Algebraic Theory of Automata.* Academiai Kiado, Budapest, 1972. 28, 45

[40] G. Gregor. Probabilistic and Nondeterministic Unary Automata. *In: Proceedings of Mathematical Foundations of Computer Science. Lecture Notes in Computer Science*, 2747:460–469, 2003. 28

[41] V. Grimm. Ten Years of Individual-Based modeling in Ecology: What have we learned and what could we learn in the future. *Ecological modeling*, 115:129–148, 1999. 108

[42] V. Grimm and S.F. Railsback. *Individual-Based Modeling and Ecology.* Princeton, NJ: Princeton University Press., 2005. 108

[43] H. Gruber and M. Holzer. Computational complexity of NFA minimization for finite and unary languages. In *LATA*, pages 261–272, 2007. 28

[44] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, New York, 1997. 63

[45] T. Harju and D. Nowitzki. Counting bordered and primitive words of a fixed weight. *Theor. Comp. Sci.*, 340:273–279, 2005. 13, 70

[46] C. Herraiz, J. Merelo, S. Olmeda, and Et Al. An Individual-Based Model that Reproduces Natural Distribution of Species Abundance and Diversity. In *in Proceedings of Artificial Life V*, 1997. 108

[47] F. Heylighen. The growth of structural and functional complexity during evolution. In F. Heylighen D. Aerts (Eds.), editor, *The evolution of complexity*, pages 17–44, Dordrecht, The Netherlands: Kluwer, 1999. 7, 107

[48] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* The MIT Press, 1992. 104

[49] H. K. Hsiao, Y. T. Yeh, and S. S. Yu. A note of ins-primitive words. *International Journal of Computer Mathematics*, 81(8):917–929, agosto 2004. 70

[50] H.K. Hsiao, Y.T. Yeh, and S.S. Yu. Square-free-preserving and primitive-preserving homomorphisms. *Acta Math. Hungar.*, 37:45–52, 1988. 12, 70

[51] M. Huston, D. DeAngelis, and W. Post. New Computer Models Unify Ecological Theory. *BioScience*, 38:682–691, 1988. 108

[52] T.J. Hutton. Evolvable Self-Reproducing Cells in a Two-Dimensional Artificial Chemistry. *Artificial life*, 13(1):11–30, 2007. 7, 108

[53] M. Ito and M. Katsura. Context-free languages consisting of non-primitive words. *International Journal of Computer Mathematics*, 40:157–167, 1991. 13, 70

[54] M. Ito, M. Katsura, H.J. Shyr, and S.S. Yu. Automata accepting primitive words. *Semigroup Forum*, 37:45–52, 1988. 70

[55] F. Javed, B. R. Bryant, M. Črepinšek, M. Mernik, and A. Sprague. Context-free grammar induction using genetic programming. *ACM Southeast Regional Conference*, 2004. 96

[56] T. Jiang, E. McDowell, and B. Ravikumar. The Structure and Complexity of Minimal NFA's over a Unary Alphabet. *Int. J. Found. Comp. Sci.*, 2(2):163–182, 1991. 28

[57] H. Jürgensen and S. Konstantinidis. Error Correction for Channels with Substitutions, Insertions, and Deletions. *Lecture notes in computer science*, 1133:149–163, 1996. 9, 48

[58] L. Kari and G. Thierrin. Word insertions and primitivity. *Utilitas Mathematica*, 53:49–61, 1998. 12, 70

[59] A. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965. 97

[60] S. Konstantinidis. computing the edit distance of a regular language. *Information and Computation*, 205(9):1307–1316, 2007. 28

[61] Y. Kunimochi. A context sensitive grammar generating the set of all primitive words. *RIMS Kokyuroku. Algebras, Languages, Computations and their Applications*, 1562:143–145, 2007. 13, 70

[62] P. Leupold. Some Properties of Context-Free Languages Related to Primitive Words. In *WORDS'03*, 2003. 70

[63] A. Levi. *Basic Set Theory*. Dover, 1979. 28, 45

[64] M. Lothaire. *Combinatorics on words*. Adison-Wesley, 1983. 25, 72

[65] Simon M. Lucas and T. Jeff Reynolds. Learning Deterministic Finite Automata with a Smart State Labeling Evolutionary Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7), 2005. 96

[66] F. Manea and V. Mitrana. All NP-problems can be solved in polynomial time by accepting hybrid networks of evolutionary processors of constant size. *Information Processing Letters*, 103(3), 2007. 48, 96

[67] M. Margenstern, V. Mitrana, and M.J. Pérez-Jiménez. Accepting Hybrid Networks of Evolutionary Processors. *DNA Computing*, 3384:235–246, 2005. 48, 96

[68] G.M. Martín and F.J. Vico. A Non-Grammatical Method to Generate the Set of Primitive Words. *(in preparation).* 3, 5, 87

[69] G.M. Martín and F.J. Vico. Complexity Dynamics in Evolving Populations of Cyclic Automata. *(in preparation).* 3, 6, 28, 71, 107

[70] G.M. Martín, F.J. Vico, and J. Dassow. Dynamics of the Complexity in the Evolution of Finite Automata. *(in preparation).* 2, 3, 4, 6, 23, 28, 71, 96

[71] G.M. Martín, F.J. Vico, J. Dassow, and B. Truthe. Low Disruption Transformations on Cyclic Automata. *Fundamenta Informaticae (to appear).* 2, 5, 47

[72] D.W. McShea. Metazoan Complexity and Evolution: Is There a Trend? *Evolution*, 50(2):477–492, 1996. 7, 108

[73] F. Mera and G. Pighizzini. Complementing Unary Nondeterministic Automata. *Theor. Comput. Sci.*, 330(2):349–360, 2005. 28

[74] P. Messer, P. Arndt, and M. Lässig. Solvable Sequence Evolution Models and Genomic Correlations. *Phys. Rev. Lett.*, 94(13):art. no. 138103, 2005. 48

[75] T. Miconi. Evolution and Complexity: The Double-Edged Sword. *Artificial Life*, 14(3):325–344, 2008. 7, 107

[76] V. Mitrana. Primitive morphisms. *Inform. Proc. Lett.*, 64:277–281, 1997. 12, 70

[77] V. Mitrana. Some remarks on morphisms and primitivity. *Bull. EATCS*, 62:213–216, 1997. 12, 70

[78] V. Mitrana and B. Truthe. On Accepting Networks of Evolutionary Processors with at Most Two Types of Nodes. In Adrian Horia Dediu, Armand Mihai Ionescu, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications*, volume 5457 of *Lecture Notes in Computer Science*, pages 588–600, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 48, 96

[79] T. Murgue and C. Higuera. Distances between Distributions: Comparing Language Models. *Lecture notes in computer science*, 3138:269–277, 2004. 10, 28

[80] S. Muthukrishnan and S.C. Sahinalp. Simple and practical sequence nearest neighbors with block operations. *Lecture notes in computer science*, pages 262–278, 2002. 48

[81] M.J. Nederhof and G. Satta. Computation of distances for regular and context-free probabilistic languages. *Theoretical Computer Science*, 395(2-3):235–254, 2008. 10, 28

[82] C.L. Nehaniv and J.L. Rhodes. The Evolution and Understanding of Hierarchical Complexity in Biology from an Algebraic Perspective. *Artificial life*, 6:45–67, 2000. 7, 109

[83] A. Nerode. Linear Automaton Transformations. *Proceedings of the American Mathematical Society*, 9(4):541 – 544, 1958. 8, 20

[84] C. Ofria, W. Huang, and E. Torng. On the Gradual Evolution of Complexity and the Sudden Emergence of Complex Features. *Artificial Life*, 263:255–263, 2008. 108

[85] S. Ohno. *Evolution by Gene Duplication*. Springer-Verlag, Berlin, 1970. 49

[86] B. Oommen. String Alignment with Substitution, Insertion, Deletion, Squashing, and Expansion Operations. *Information Sciences*, 83:89–107, 1995. 9, 48

[87] B. Oommen and R. Loke. Pattern Recognition of Strings with Substitutions, Insertions, Deletions and Generalized Transpositions. *Pattern Recognition*, 30:789–800, 1997. 9, 48

[88] B. J. Oommen and R. K. S. Loke1. Noisy subsequence recognition using constrained string editing involving substitutions, insertions, deletions and generalized transpositions. *Lecture notes in computer science*, 1024:116–123, 1995. 9, 48

[89] G. Pãun, N. Santean, G. Thierrin, and S.S. Yu. On the robustness of primitive words. *Discrete Appl. Math.*, 117:239–252, 2002. 12, 70

[90] G. Pãun and G. Thierrin. Morphisms and primitivity. *Bull. EATCS*, 61:85–88, 1997. 12, 70

[91] E. Pachepsky, T. Taylor, and S. Jones. Mutualism Promotes Diversity and Stability in a Simple Artificial Ecosystem. *Artificial Life*, 8(1):5–24, 2002. 108

[92] P. Pawarand and G. Nagaraja. Regular Grammatical Inference: A Genetic Algorithm Approach. *Lecture Notes in Computer Science*, 2275:295–339, 2002. 96

[93] H. Petersen. On ambiguity of primitive words. *Lecture Notes in Computer Science*, 117:679–690, 1994. 70

[94] H Petersen. On the language of primitive words. *Theoretical Computer Science*, 161(1-2):141–156, julio 1996. 70

[95] G. Pighizzini. Unary Language Concatenation and Its State Complexity. *5th International Conference on Implementation and Application of Automata*, 2088:252–262, 2001. 28

[96] G. Pighizzini and J. Shallit. Unary Language Operations, State Complexity and Jacobsthal's Function. *Int. J. Found. Comp. Sci.*, 13:145–159, 2002. 28

[97] A. Rasek, W. Dörwald, M. Hauhs, and A. Kastner-Maresch. Species Formation in Evolving Finite State Machines. *Lecture Notes in Computer Science*, 1674:139–148, 1999. 96

[98] H. Reutera, F. Hölkerb, U. Middelhoffc, F. Jopp, C. Eschenbache, and B. Breckling. The Concepts of Emergent and Collective Properties in Individual-Based Models. *Ecological modeling*, 186:489–501, 2009. 108

[99] G. Richomme. On morphisms preserving infinite Lyndon words. *Discrete Math. Theor. Compt. Sci.*, 9:89–107, 2007. 12, 70

[100] C.J.Van Rijsbergen. *Information Retrieval*. London, 1979. 29

[101] M. Rönkkö. An Artificial Ecosystem: Eergent Dynamics and Lifelike Properties. *Artificial Life*, 13(2):159–187, 2007. 108

[102] G. Rozenberg and A. Salomaa. *Handbook of Formal Languages*. Springer-Verlag, Berlin, 1997. 19

[103] D. Saakian. Evolution Models with Base Substitutions, Insertions, Deletions, and Selection. *Phys. Rev.*, 78(6):art. no. 061920, 2008. 48

[104] Y. Sakakibara. Grammatical Inference in Bioinformatics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7), 2005. 96

[105] C. Shannon and N. Weaver. *A mathematical theory of communication*. 1949. 97

[106] D. Shapira and J.A. Storer. Edit distance with move operations. *Journal of Discrete Algorithms*, 5(2):380–392, junio 2007. 48

[107] D. Shen and T. Liu. The Fooling Set Problem for Unary Regular Language Is in NP. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, pages 23–27, 2008. 28

[108] H.J. Shyr. *Free monoids and languages*. Hon Min Book Co., Taichung, 1991. 70, 72

[109] E. Szathmary. Molecular Biology and Evolution: Can Genes Explain Biological Complexity? *Science*, 292(5520):1315–1316, 2001. 108

[110] P.N Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006. 29

[111] T. Taylor. Niche Construction and the Evolution of Complexity. In *In Proceedings of Artificial Life IX*, pages 375–380, 2004. 7, 108

[112] R. Walker. Niche Selection and the Evolution of a Complex Behavior in a Changing Environment: a Simulation. *Artificial life*, 5(3):271–289, 1999. 7, 108

[113] J. Watson, N. L. Geard, and J. Wiles. Towards more biological mutation operators in gene regulation studies. *BioSystems*, 76(1-3):239–248, 2004. 9, 48

[114] S. Wu, U. Manbe, and G Myers. A subquadratic algorithm for approximate limited expression matching. *Algorithmica*, 15(1):50–67, 1996. 28

[115] S. Yu. State complexity of regular languages. *Journal of Automata, Languages and Combinatorics*, 6(2):221–234, 2001. 8, 20

[116] J. Zhang. Evolution by Gene Duplication: An Update. *Trends in Ecology and Evolution*, 18:292–298, 2003. 49