On Complexity of Persian Orthography: L-Systems Approach

Nassim Taghipour
Pattern Research Center, Tehran, Iran
taghipour@iranprc.org

Hamid Haj Seyyed Javadi
Department of Mathematics and Computer Science, Shahed University
Tehran, Iran
h.s.javadi@aut.ac.ir

Mohammad Mahdi Dehshibi*
Pattern Research Center, Tehran, Iran
dehshibi@iranprc.org
**Corresponding author*

Andrew Adamatzky
Unconventional Computing Centre, University of the West of England
Bristol, UK
andrew.adamatzky@uwe.ac.uk

To understand how the Persian language developed over time, we uncover the dynamics of complexity of Persian orthography. We represent Persian words by L-systems and calculate complexity measures of these generative systems. The complexity measures include degrees of non-constructability, generative complexity, and morphological richness; the measures are augmented with time series analysis. The measures are used in a comparative analysis of four representative poets: Rudaki (858–940 AD), Rumi (1207–1273), Sohrab (1928–1980), and Yas (1982–present). We find that irregularity of the Persian language, as characterized by the complexity measures of L-systems representing the words, increases over temporal evolution of the language.

1. Introduction

Generation of culture-related motifs, for example, oriental ornaments or scripts, is among the hot topics of computer graphics and virtual reality [1–4]. The patterns can be produced in a "natural" way using simulated biological, physical, or chemical processes growing plants as L-systems [5]; precipitating reaction-diffusion media [6] are based on the actual mathematical equations designed by scientists to model the studied phenomena. Other approaches include cellular automata. Thus, Arata et al. [7] used cellular automata with a Margolus neigh-

https://doi.org/10.25088/ComplexSystems.25.2.127

borhood to model an interactive free-form scheme within a three-dimensional voxel space for designing virtual clay objects. They assumed each voxel is allocated a finite-state automaton that repeats state transitions according to the conditions of its neighbor voxels. Xu et al. [8] used a computational approach to digital Chinese calligraphy and painting. Ahuja and Loeb [9] utilized a number of geometric shapes to tessellate the “Six-Ali” pattern. In [10] a prototype for performing geometrical transformations was introduced with the aim of decorative designs using Kufic square scripts. Ma’qeli script is a sort of ancient Persian script with amazing features. Minoofam et al. [11–13] generated all forms of the Ma’qeli script with synchronous block two-dimensional asynchronous cellular automata and a Margolus neighborhood as well as three holy Islamic words using L-systems. To show the simplicity and efficiency of the proposed method, they conducted a set of experiments in which the capability of both synchronous and asynchronous cellular automata with one- and two-dimensional structures and different neighborhoods was examined. They have shown that the script, as a graphical primitive, has the capability of being utilized in creative environments such as computer games, animations, and so forth [14].

L-systems were conceived to formally describe the growth process of plants, and in this context they were extensively studied by biologists and theoretical computer scientists; see [15, 16] for classic references. Another area of applications of L-systems was discovered by Smith [17], who applied them to generate realistic images of plants and trees for computer imagery purposes. Prusinkiewicz [18] has shown that graphical applications of L-systems were not limited to plants and trees, but also included a wide range of fractal systems. Later, he presented an application of L-systems to the algorithmic generation of musical scores [19]. To this end, strings of symbols produced by an L-system are given a musical interpretation. The musical and graphical interpretations of L-systems are closely related. Because of the fractal nature of the created figures, the described approach is related to the work of Dodge and Bahn [20] and Voss and Clarke [21]. In other research, Worth and Stepney [22] showed that music can also be represented by grammars, and it is possible to interpret L-systems musically. They searched for simultaneous “pleasing” graphical and musical renderings of L-systems. This idea for forming musical scores, which are a special form of a script, also motivates researchers to work on other types of scripts.

Dehshibi et al. [23, 24] generated 180 words of a Persian poem “Neyname” with L-systems. The main reason for choosing the Persian language is its baseline, cursive structure, and agglutinative nature [25, 26]. They constructed complexity hierarchies of selected Persian words and provided insights on how the complexity of the poem changes during the poem's spatial development. In the present paper, we build on our previous work [23, 24] by producing all combinations of words in selected examples of Persian literature. The paper is structured as follows. Section 2 provides background information and an overview of algorithms proposed by Dehshibi et al. [23, 24]. The method of generating sentences is presented in Section 3. Section 4 analyzes the dynamics of complexity measures. Concluding remarks are presented in Section 5.

2. Growing Persian Words with L-Systems

An L-system is a parallel rewriting system and a type of formal grammar, which was introduced to describe rules of plant growth. The first ever representation of a language's words using L-systems was proposed by Dehshibi et al. [23, 24]. They introduced grammars and rules of a Persian poem from Rumi, the so-called "Neyname," and generated 108 words. Although the application is very interesting, this work suffers from a case-by-case graphical word formation, which is not expandable to the whole of the Persian language, much less other languages. In this paper, we aim at completing this work so as to find rules for forming all combinations of words in a language, focusing on Persian orthography. Moreover, several kinds of complexities are also defined to analyze the proposed method and find change in the constructability of the Persian language. Surprisingly enough, the results were promising and demonstrate the efficiency of the proposed analysis in showing the evolution of a language. Again, a question that may be created for researchers and readers is the bizarre and confusing connection between L-systems and orthography. To address this critical question, we can state that words when written by hand grow like L-systems. Indeed, when a person writes a symbol, he starts at one point and then "grows" the symbol on the plane, and this action is nothing more than an imitation of a regular L-system. Although this statement seems to be very intuitive, there are some references supporting our statement, and finding more appropriate references needs a deep knowledge in the history of writing systems and its bibliography; see for instance [27–29].

In order to form the poem "Neyname," first the related rules for the words should be extracted. The easiest way to achieve this aim is to draw the Persian alphabet on a sheet of grid paper; this makes the extraction of rules satisfactory. There is a critical problem: how to deal with the correct shape of Persian words that have dot(s). The fundamental turn in the rules is a 90° turn, which leads to a bad shape, as is illustrated in Figure 1.

https://doi.org/10.25088/ComplexSystems.25.2.127

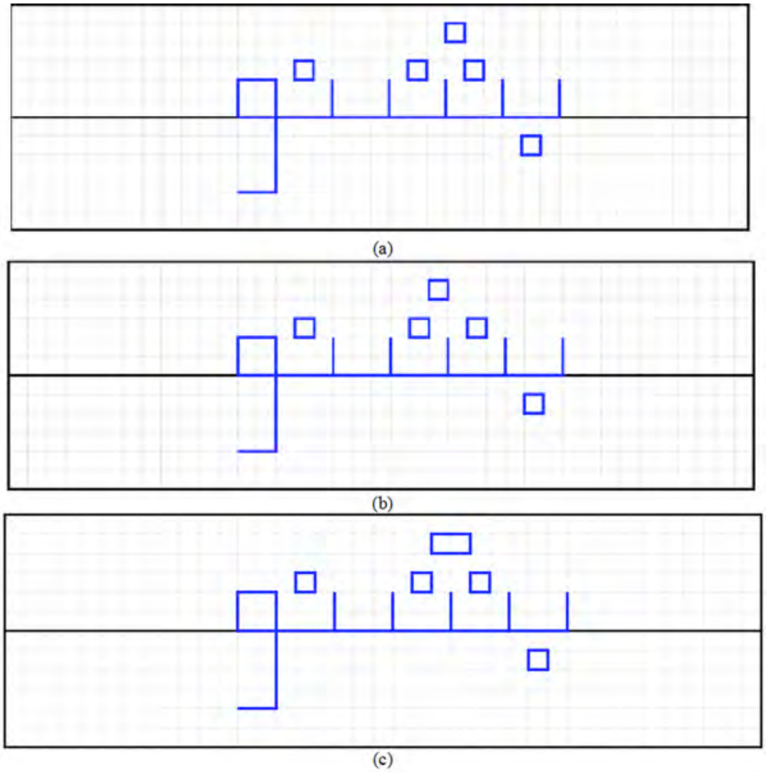


Figure 1. The problem with the dot's placement in the middle of the “ ” character. (a) Left alignment of the dot; (b) right alignment of the dot; and (c) deforming the shape of the dot.

In order to resolve this problem, two solutions are proposed. The first solution is to draw the dot in a neighbor cell and change the dot's alignment, which is the easiest solution, or to deform the shape of the dot. However, this solution leads to deformation of the Persian characters (see Figure 1(c)). The second solution is to change the angle of drawing in the string from a 90° turn to a 45° turn. This solution has two advantages, which are: (*a*) resolving the problem of drawing the diacritical mark in case of misplacement or deformation; and (*b*) better shaping of all the characters of the Persian alphabet, which makes it closer to realistic orthography. The result of using this solution is illustrated in Figure 2(a). Despite the mentioned advantages of using a 45° turn in the visual forming of Persian orthography with an L-system, a new problem emerges: one must return to the baseline in order to continue drawing the rest of the word, as many Persian words have dissociation.

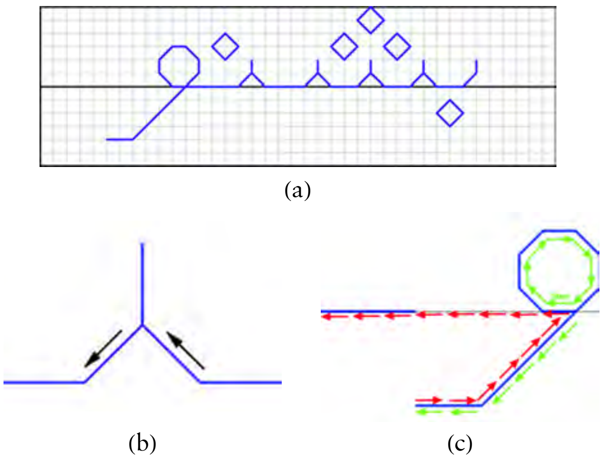


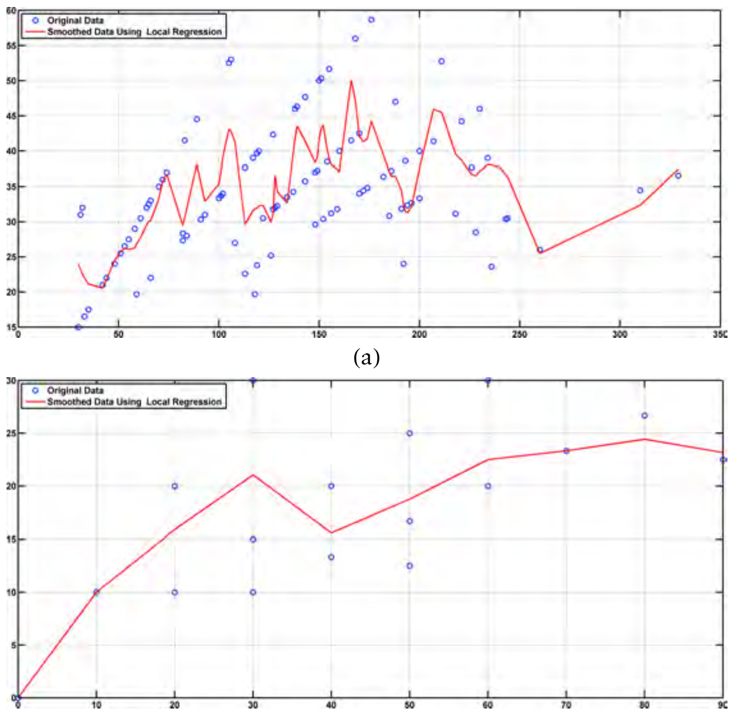
Figure 2. (a) Better orthography of the "بشنو" word using 45° turns in the rule of an L-system. (b) Procedure of synchronizing top-down and bottom-up movement for characters that have a tooth-like shape. (c) The problem of returning to the baseline using 45° turns.

This problem arises in the case of using the 45° turn, which results in the production of horizontal, vertical, and diagonal directions. The vertical and horizontal directions have length 1; however, the length of the diagonal direction is approximately 0.7. To solve this problem, we propose a synchronization between top-down and bottom-up movement; see Figure 2(b), which shows one of the tooth-like shapes of the "ـسـ" character. In some characters, when moving from the baseline to the bottom of the character orthogonally and returning to the baseline vertically, the end of the character would not be on the baseline. In this type of character, the procedure of movements is automatically synchronized; however, in a character like "و," when movement takes place to the bottom of the baseline, the only way is to move back to the baseline orthogonally on the drawn character using "*f*" (see Figure 2(c)). Afterward, other characters can be drawn based on the designed string. The differences between the methods of returning to the baseline are as follows. Although the second string is longer, it helps with drawing the Persian/Arabic characters more naturally:

Start string of the "و" character (straight movement):
F-F-F-F-F-F-FFFFF-FF--FFF++FFF
Start string of the "و" character (orthogonal method):
F-F-F-F-F-F-F-FFFFF-FF----ff+ffff+++fffffFFF.

Another novelty is in introducing a family of complexities called "absolute complexity" and "relative complexity." To calculate these complexities, first we rewrite rules in a compressed format; then counting the length of the start and rule of each grammar leads to obtaining absolute complexity, and the division of absolute complexity by the number of characters in the Persian spelling of words leads to calculating relative complexity. Figure 3 shows the absolute complexity of the words of the "Neyname" in comparison with its relative complexity.

Dehshibi et al. [23, 24] mentioned that there are several issues that cause the length of the *Start* string to increase abnormally. The characters written below the baseline have a longer *Start* string, for example, "[illegible]," "[illegible]." The main reason is that there is a return to the baseline while finishing the visual forming of the character. Persian orthography is cursive, so we were forced to use a longer *Start* string. For example, the word "[illegible]" is less chopped-off than the word "[illegible]," so it needs a shorter *Start* string.



(a) (b)

Figure 3. Absolute complexity (vertical axis) in comparison with relative complexity (horizontal axis). (a) *Start* string; (b) *Rule* string.

https://doi.org/10.25088/ComplexSystems.25.2.127

3. Persian Orthography Using L-Systems

In order to implement Persian sentences, first the rules of forming the alphabet should be extracted. The rules can be derived manually by drawing the Persian alphabet on graph paper. Thirty-two letters of the modern Persian alphabet are tabulated in Table 1. With respect to the fact that Persian script is cursive, the appearance of a letter would

Contextual Forms				IPA	Name
Isolated	Initial	Medial	Final		
ا/آ		ا		[ɒ]	alef
ب	بـ	ـبـ	ـب	[b]	be
پ	پـ	ـپـ	ـپ	[p]	pe
ت	تـ	ـتـ	ـت	[t]	te
ث	ثـ	ـثـ	ـث	[s]	se
ج	جـ	ـجـ	ـج	[d͡ʒ]	jim
چ	چـ	ـچـ	ـچ	[t͡ʃ]	che
ح	حـ	ـحـ	ـح	[h]	he-ye jimi
خ	خـ	ـخـ	ـخ	[ẍ]	khe
د		ـد		[d]	däl
ذ		ـذ		[z]	zäl
ر		ـر		[r]	re
ز		ـز		[z]	ze
ژ		ـژ		[ʒ]	jh
س	سـ	ـسـ	ـس	[s]	sin
ش	شـ	ـشـ	ـش	[ʃ]	šin
ص	صـ	ـصـ	ـص	[s]	säd
ض	ضـ	ـضـ	ـض	[z]	zäd
ط	طـ	ـطـ	ـط	[t]	tä
ظ	ظـ	ـظـ	ـظ	[z]	zä
ع	عـ	ـعـ	ـع	[ʔ]	eyn
غ	غـ	ـغـ	ـغ	[ɣ]/[G]	ġeyn
ف	فـ	ـفـ	ـف	[f]	fe
ق	قـ	ـقـ	ـق	[G]/[ɣ]/[q](in some dialects)	qäf
ک	کـ	ـکـ	ـک	[k]	käf
گ	گـ	ـگـ	ـگ	[g]	gäf
ل	لـ	ـلـ	ـل	[l]	läm
م	مـ	ـمـ	ـم	[m]	mim
ن	نـ	ـنـ	ـن	[n]	nun
و		ـو		[v]/[u:]/[o]/[ow]/[o:]	väv
ه	هـ	ـهـ	ـه	[h]	he
ی	یـ	ـیـ	ـی	[j]/[i]/[n:]/[e:]	ye

Table 1. Persian alphabet and its permutations.

https://doi.org/10.25088/ComplexSystems.25.2.127

be changed based on its position. There are four kinds of appearance (permutation), as follows: isolated, initial (joined on the left of a word), medial (joined on both sides), and final (joined on the right).

In order to form a Persian sentence with L-systems, the *Start* string related to permutations of the Persian alphabet should be extracted, which leads to the construction of 155 rules. The initial step is sketching on graph paper. However, in the case of forming words as well as sentences, several challenges are raised. We will describe these challenges as well as their solutions in Section 3.1.

3.1 Issues and Solutions in Persian Orthography

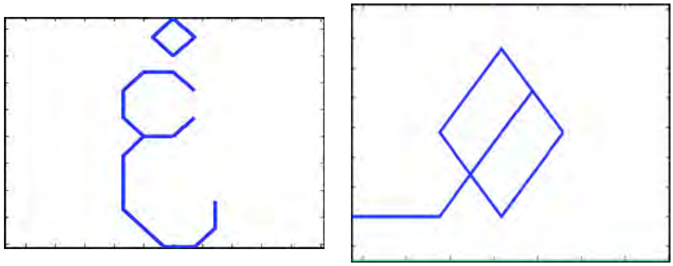
Dehshibi et al. [23, 24] proposed a 45° turn for the formation of words to reach a natural graphical orthography. The proposed method for the tooth-like letters, however, does not work for the characters that are placed in the initial position of the word, for example, “بـ.” It also does not work for sentence formation. The problem is with the diagonal lines in each character that lead to misplacement of words and sentences; that is, the word or sentence forms in such a way that the initial character is on the baseline and the rest will be placed above the baseline. Figure 4(a) illustrates the graphical form of the “بـ” character.

(a) (b)

Figure 4. (a) Jagged letters outlining initial state. (b) Tooth-like characters start with an imaginary path when they appear at the beginning of a word.

The keys to resolving these problems were considering a complete tooth-like character in which the movement from the baseline to the peak is hidden, that is, using the “*f*” character. This solution is depicted in Figure 4(b). Extending this solution to forming a sentence happens by adding a “-*f*” to the beginning of the rules. In terms of forming all Persian characters, we have characters such as “غـ” and “ﻬ,” whose return to the baseline consumes a large number of “*f*” symbols. As a consequence, the length of their *Start* string grows abnormally. Figure 5 shows these characters.

https://doi.org/10.25088/ComplexSystems.25.2.127



(a) (b)

Figure 5. (a) The letter "غ," which is produced with L-systems. *Start* string is: *F-F-F-F-F-F-f-F---f f f+F--F--F--F+f f f--f+F+F F+F F+F+F+F++++f-f-f-f f-f f-f+++f*. (b) The letter "ـط," which is produced with L-systems. *Start* string is: *-f++ F-F-F--F F++++ F F++ F+F+F++ F F++ f f++ f+++++f*.

3.2 Proposed Algorithm

Even after these problems had been overcome, there were still issues with implementation. The main problem is decoding inputs, as they are Persian characters and must be interpreted in such a way that the proposed system is a cross-platform one. Another problem is finding the permutation of the input character, that is, isolated, initial, medial, or final. Pseudocode of the proposed algorithm follows.

1. Get the input sentence and assign it to an array.
2. Define a matrix with two rows and n columns, where n is the size of the input sentence.
3. Parsing the input sentence so as to fill the first row with the ASCII of each character and the second row with a number between 1 and 4 based on the permutation of that character. 1 = initial; 2 = medial; 3 = final; 4 = isolated.
4. Extract the *Start* string of each cell.
5. Merge the *Start* strings of characters to obtain the complete rule of the sentence.
6. Graphical forming of the rule with an L-system parser.

In order to clarify the proposed algorithm, we describe it with an example. Assume the input word is "سلام," which means "Hello." Codes of the permutation of the character "ل" are tabulated in Table 2.

https://doi.org/10.25088/ComplexSystems.25.2.127

ل	ـل	ـلـ	لـ
4	3	2	1

Table 2. Codes of the character “ل” permutations.

In order to fill the second row of the defined matrix for the sentence, the following meta-rules can be used:

- If the ASCII code is 32, the character is a space; fill the entry with 0.
- If before the current character is a space and after that is a non-space character, then fill the entry with 1.
- If before the current character is a non-space and after that is a space, fill the entry with 2.
- If before and after the current character is a non-space, fill the entry with 3.
- Finally, if before and after the current character is a space or before the current character there is one of the characters “آ,” “ر,” “ز,” “ژ,” “د,” “ذ,” “و,” and after that there is a space, the entry is filled by 4. For instance the character “ف” in the “برف” word has such a situation.

According to the proposed algorithm and meta-rules, the “سلام” word will have a matrix as illustrated in Figure 6.

م	ا	ل	س
1605	1575	1604	1587
4	2	2	1

Figure 6. Code matrix of the “سلام” word.

Finally, after fetching the rule of each entry, we have a *Start* string as follows:

Start string of "سلام":
-f-F++++f-F-F-F-F++++f-F-F-F-F++++f-
F-FF--ffff++++FFFF--FFF--FFFF++++ffff--ffF-
F-F-F-F-F-F-F-fFF+F+FFFFFF----ffffff-f+++f.

As is evident from the aforementioned meta-rules and the overall procedure of the proposed algorithm, it is easy to form all Persian and also other orthography patterns. However, in the analysis of the poems, graphical forms of punctuation were not considered to be formed [30]. Adib-Soltani [31] states that punctuation entered into the Persian language from foreign languages. He explains that in ancient times, complex scientific and literary Persian books did not contain any punctuation, and reading Western books was difficult because of their dependence on punctuation. He believes that nowadays in European languages, punctuation plays such a vital role that not focusing on proper use of signs may cause misunderstanding. He also mentions that in the past there were no such problems in the Persian language, and that only in the past 100 years have these signs been adopted from foreign languages. Khayam [32] points out that for a long period of time, Persian texts were written without using any punctuation, while nowadays for writing different texts, punctuation marks are widely utilized. He emphasizes that one must pay attention to differentiating the signs from the words and also mentions that some signs may cause big problems. Yahaghi and Naseh [33] mention that the usage of punctuation in the Persian language does not have a long history, and is common in the current century based on Western texts. They emphasize that one should avoid using punctuation marks less or more than is needed. Hence, if we decide to analyze Persian literature of the current decade, it is worth considering punctuation, which has important effects on the meaning of a text. However, as our focus is to cover a long period of this language, punctuation does not have critical effects.

4. Experimental Result and Discussion

In order to reach a correct conclusion about the changes in the Persian language during different eras, a series of experiments has been conducted on four Persian poems. The main reason for selecting poems is the saying that “as a poet speaks with a language, the language will not die” and Goethe’s belief that “Persian literature is one of the four pillars of human literature” [34]. Several analyses have been done that comprise relative/absolute complexities, degree of non-constructability, generative complexity, morphological richness, and a set of time series analyses. We will describe all sorts of analyses in the forthcoming subsections. However, before starting, it is worth illustrating the results of Persian orthography based upon the defined meta-rules. Figure 7 illustrates the first verse of four poems by Rudaki, Rumi, Sohrab, and Yas [35].

Abu Abdollah Jafar ibn Mohammad Rudaki, also known as Rudaki, was a Persian poet regarded as the first great literary genius of modern Persian, who composed poems in the “New Persian” alphabet. Rumi was a thirteenth-century Persian poet, jurist, Islamic scholar, theologian, and Sufi mystic. Sohrab Sepehri was a notable Persian poet and a painter, considered to be one of the five most famous Persian poets to have practiced “New Poetry.” Yaser Bakhtiari,

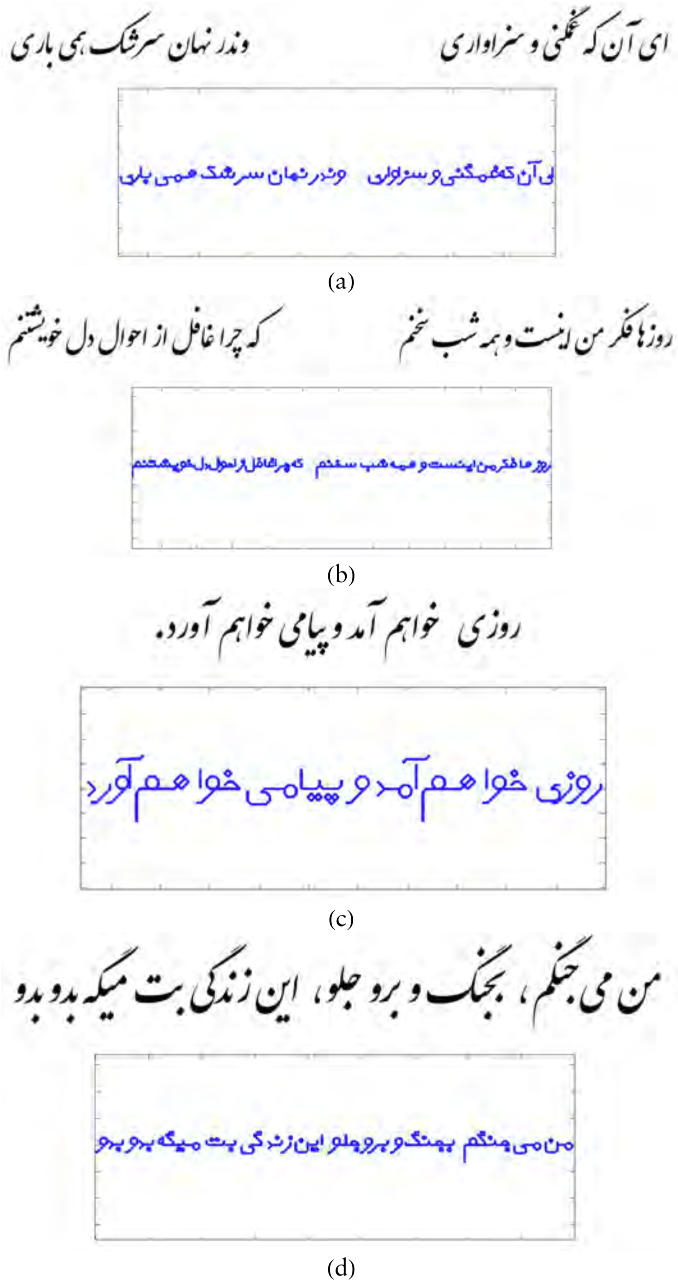


Figure 7. The first verse of the poem by: (a) Rudaki, entitled "The person who is afflicted and worthy of seldom;" (b) Rumi, entitled "Every single day I ponder over these questions;" (c) Sohrab, entitled "Message is in the way;" and (d) Yas, entitled "I am fighting."

better known by his stage name Yas, is an Iranian rapper who is the first rapper authorized to perform by the Iranian government, and who became one of the most popular rappers in Iran.

One of the main concerns of this research is to select the best samples of the Persian language, as one of the great literatures of mankind [36], to show its evolution. Persian literature has its roots in surviving works of Middle Persian and Old Persian, the latter of which date back as far as 522 BCE (the date of the earliest surviving Achaemenid inscription, the Behistun Inscription) [34]. The bulk of surviving Persian literature, however, comes from the times following the Islamic conquest of Iran, when the Iranians became the scribes and bureaucrats of the Islamic empire and, increasingly, also its writers and poets. While initially overshadowed by Arabic during the Umayyad and early Abbasid caliphates, New Persian was soon reinstated as a literary language of the Central Asian and West Asian lands. The rebirth of the language in its new form is often accredited to Rudaki and his generation, as they used pre-Islamic nationalism as a conduit to revive the language and customs of ancient Iran. So strong is the Persian aptitude for versifying everyday expressions that one can encounter poetry in almost every classical work, whether from Persian literature, science, or metaphysics. In short, the ability to write in verse form was a prerequisite for any scholar. For example, almost half of Avicenna's medical writings are in verse. Indeed, Persian poems are the most citable source of this great language, which can also help us to prove our hypothesis in this paper.

4.1 Complexity Analysis

To calculate absolute complexity, we count the length of the *Start* string as well as the *Rule* of each grammar. The relative complexity for the *Start* string is the absolute complexity divided by the number of characters in the Persian spelling of words, and the relative complexity for the *Rule* string is the absolute complexity divided by the number of characters in the Persian spelling of words that have dots. Therefore, absolute and relative complexity can be calculated by equations (1) through (3):

$$\text{AbsoluteComplexity} = \sum_{i=1}^{n} (F, f, +, -), \tag{1}$$

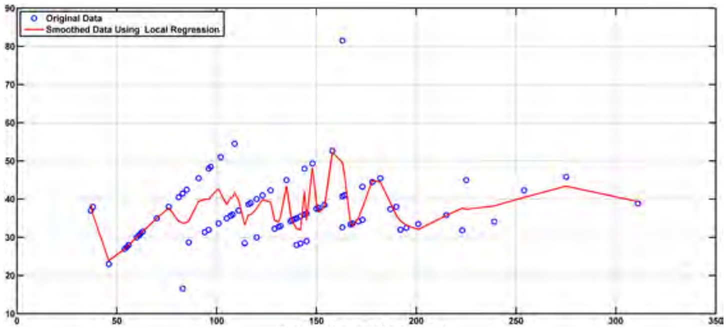
$$\text{RelativeComplexity}(Start) = \frac{\sum_{i=1}^{n} (F, f, +, -)}{\text{length}(word)}, \tag{2}$$

https://doi.org/10.25088/ComplexSystems.25.2.127

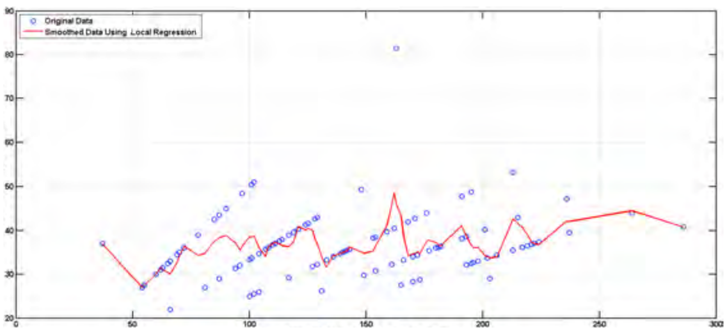
$$\text{RelativeComplexity}(Rule) = \frac{\sum_{i=1}^{n} (F, f, +, -)}{\text{count}(dots\_of\_word)}, \tag{3}$$

where n is the number of symbols in the string and $\text{length}(word)$ counts the number of characters in the Persian spelling of the word.

Figure 8 shows the absolute complexity of the words of these poems versus their relative complexity. With regard to the complexity of the whole poem, assume the poem may be expressed as $w_1, w_2, w_3, \ldots, w_n$, where w_i is the word in the i^{th} position and $1 \leq i \leq n$. We calculate the complexity of each word in the poem; for example, if the complexity of the word w_i is $C(w_i)$, we have a series of numbers $C(w_1), C(w_2), C(w_3), \ldots, C(w_n)$, which represents the "temporal" (as the poem is read) complexity dynamics of the whole poem.



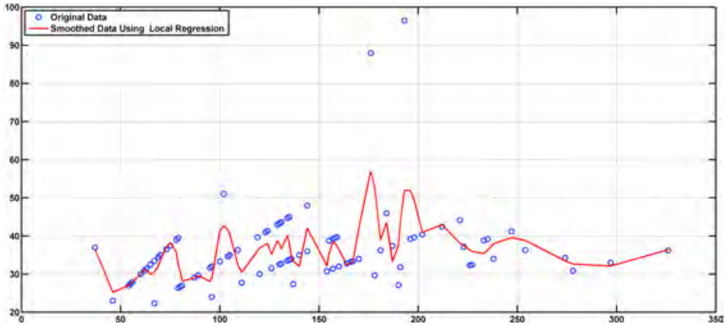
(a)



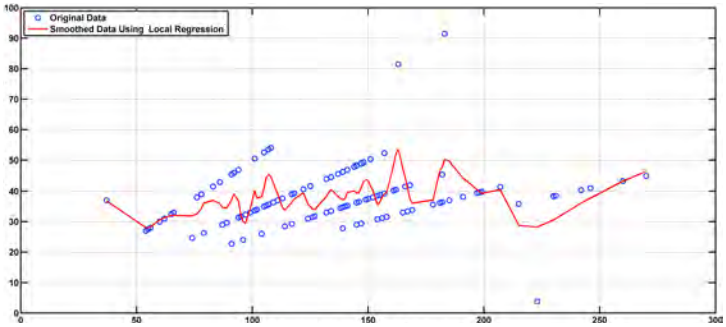
(b)

Figure 8. (*continues*).

https://doi.org/10.25088/ComplexSystems.25.2.127



(c)



(d)

Figure 8. Absolute complexity (vertical axis) versus relative complexity (horizontal axis) of whole words of the poem for the *Start* string. (a) "The person who is afflicted and worthy of seldom." (b) "Every single day I ponder over these questions." (c) "Message is in the way." (d) "I am fighting."

This kind of complexity can be approximated using two measures: "degree of non-constructability" and "generative complexity," which are calculated by equations (4) and (5), respectively:

$$D = \text{length}(\mathit{Start}), \tag{4}$$

$$G = \text{length}(\mathit{compressedRule}). \tag{5}$$

For each word w_i we calculate $D(w_i)$ and $G(w_i)$ and have two plots, which are illustrated in Figure 9 and show irregular oscillations in the course of the poem. The idea of measure D is borrowed from cellular automata theory, where a configuration is called non-constructable if it could not be reached from any other configuration by applying local rules of cell-state transitions [37–40]. In the context of our model, the degree of non-constructability shows how substantial the part of the word is that cannot be generated but must be described by a *Start* string. The generative complexity G, calculated as a length of a compressed generative rules table, shows how difficult it is to generate any particular word or pattern [41–43].

Functions approximating plots are shown by solid lines in Figures 8–11. We estimate the integral complexity of a poem by calculating the number of peaks, exceeding 10 units, on the functions. Table 3 shows that the poem "Message is in the way" has the highest overall complexity, as expressed in changes of the functions approximating plots in Figures 8, 10, and 11.

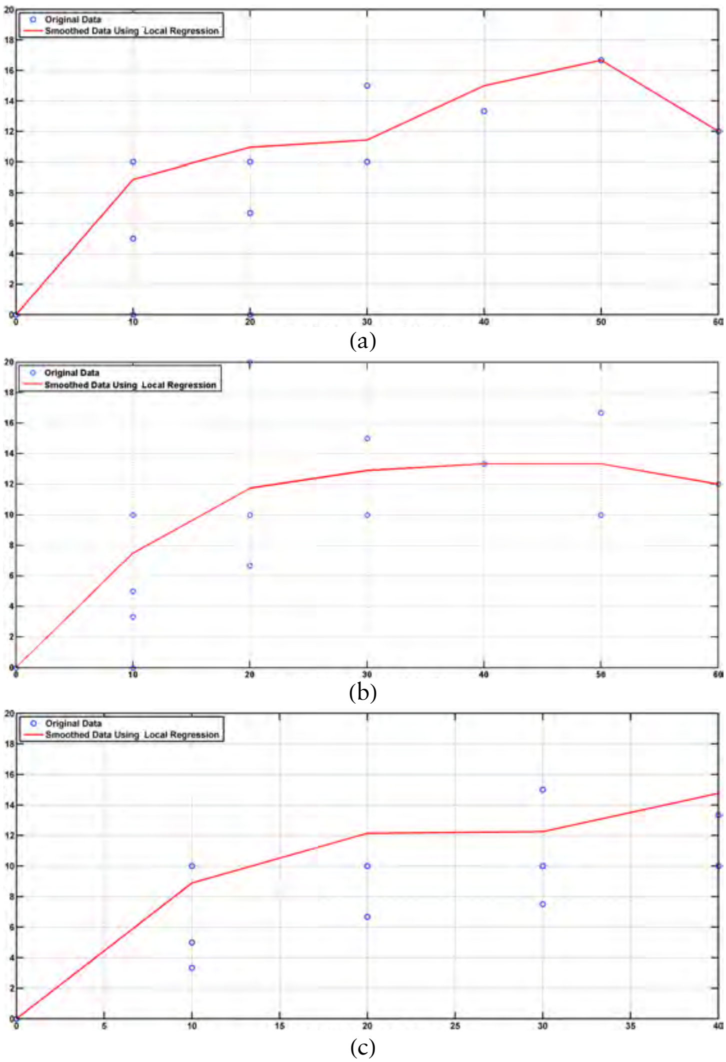
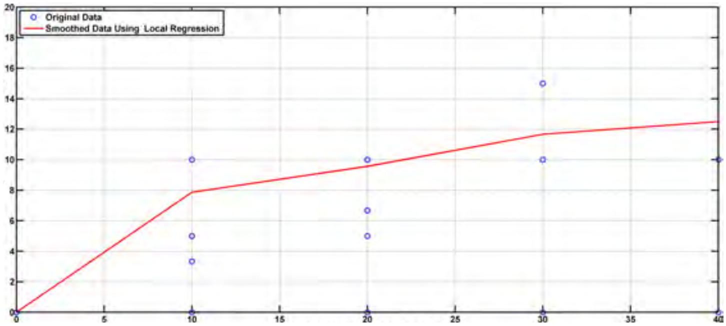
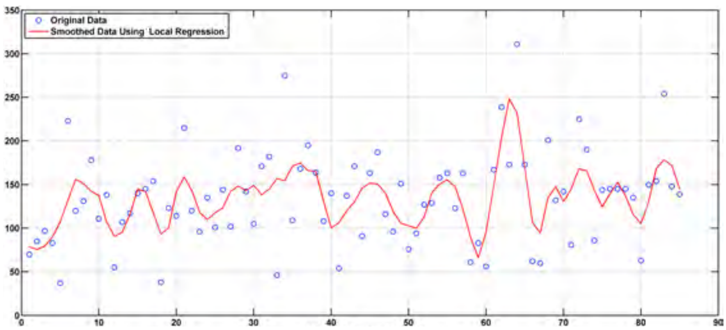


Figure 9. (*continues*).

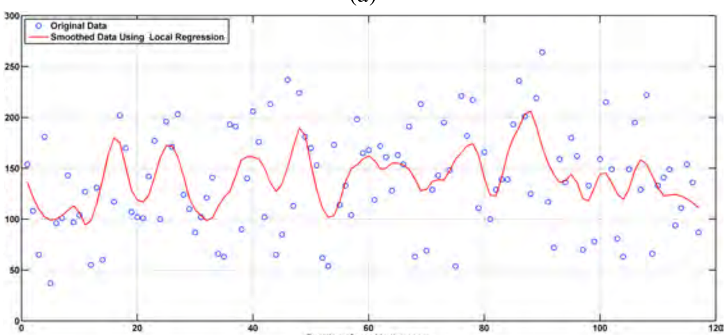


(d)

Figure 9. Absolute complexity (vertical axis) versus relative complexity (horizontal axis) of whole words of the poem for the *Rule* string. (a) "The person who is afflicted and worthy of seldom." (b) "Every single day I ponder over these questions." (c) "Message is in the way." (d) "I am fighting."

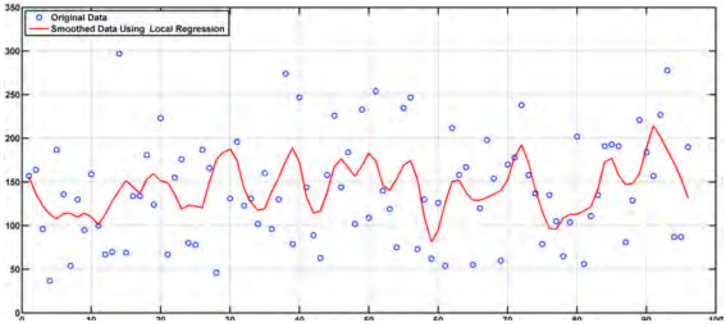


(a)

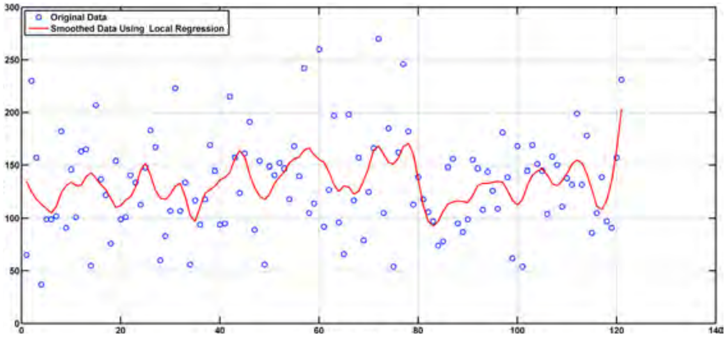


(b)

Figure 10. (*continues*).

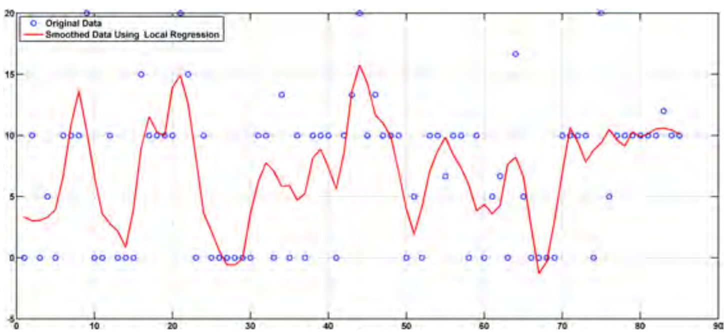


(c)



(d)

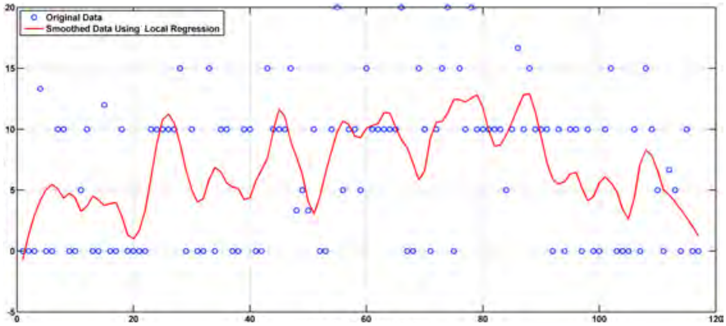
Figure 10. Degree of non-constructability of whole words of the poems. (a) "The person who is afflicted and worthy of seldom." (b) "Every single day I ponder over these questions." (c) "Message is in the way." (d) "I am fighting."



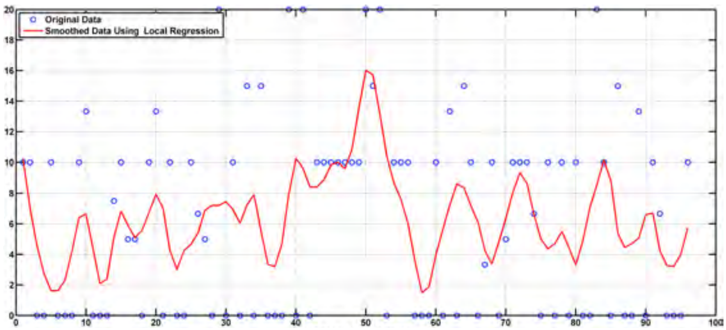
(a)

Figure 11. (*continues*).

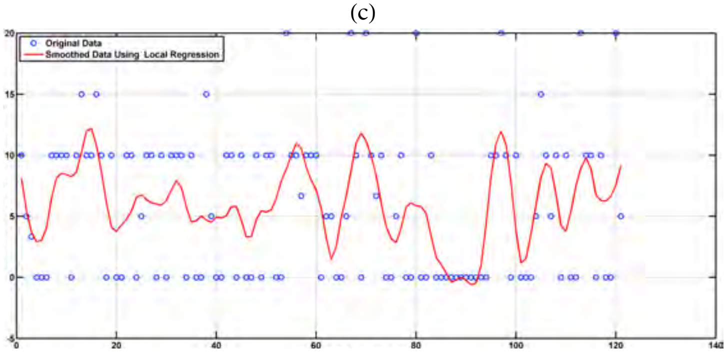
https://doi.org/10.25088/ComplexSystems.25.2.127



(b)



(c)



(d)

Figure 11. Generative complexity of whole words of the poems. (a) "The person who is afflicted and worthy of seldom." (b) "Every single day I ponder over these questions." (c) "Message is in the way." (d) "I am fighting."

https://doi.org/10.25088/ComplexSystems.25.2.127

Poem	Changes in complexity of *Start* string (Figure 8)	Changes in non-constructability (Figure 10)	Changes in generative complexity (Figure 11)
"The person who is afflicted and worthy of seldom."	9	11	8
"Every single day I ponder over these questions."	10	9	8
"Message is in the way."	12	11	10
"I am fighting."	10	10	8

Table 3. Integral complexity of poems.

If we sort values of the compressed *Start* strings in each poem in an ascending manner, then we will see three distinct classes of non-constructability (refer to Figure 10).

Based on the conducted experiments by using complexity analysis, it is worth concluding that:

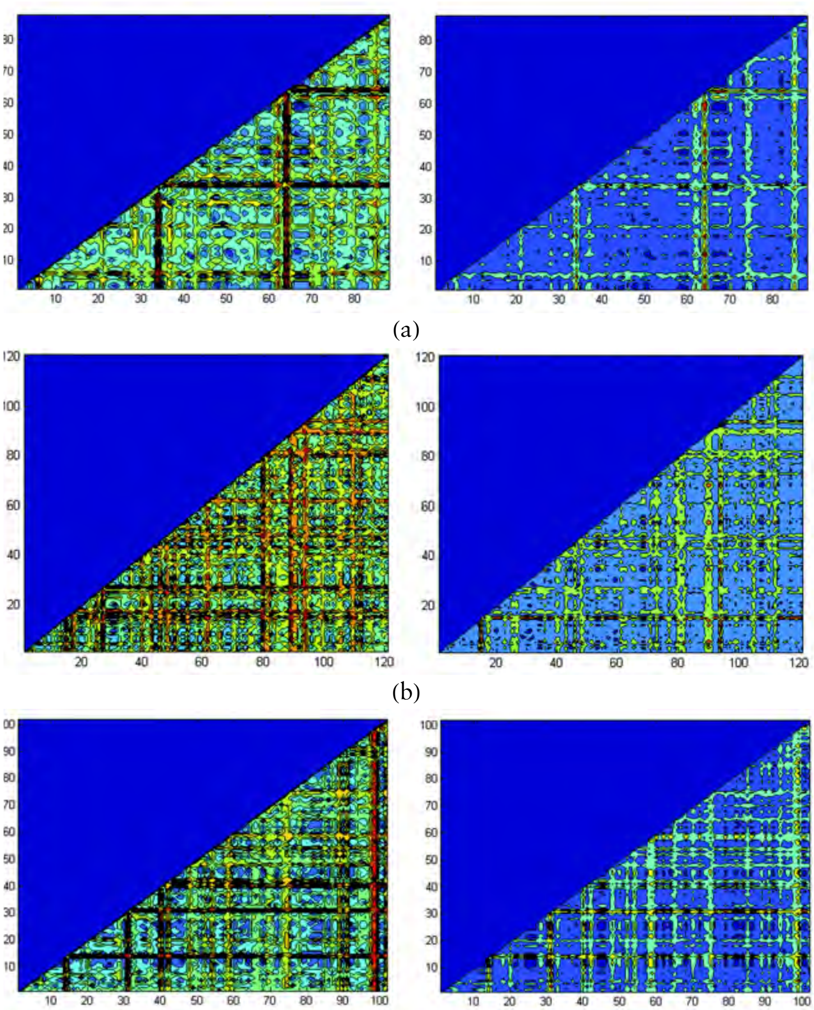
1. Surprisingly enough, about six clusters are evident in Figure 8(a)–(d), which can be the topic of further research about finding the relationship between the number of clusters in Persian viseme clustering [44–46] and Persian orthography.
2. As is evident from Figure 9, the "degree of non-constructability" graph has a chaotic visualization from which doing any inferences is hard. However, the graph of "generative complexity" is somewhat regular, and extracting four clusters is meaningful. Cluster I contains low generative complexity and non-constructability, cluster II has high generative complexity and non-constructability, cluster III contains high generative complexity and moderate non-constructability, and cluster IV has low generative complexity and high non-constructability.
3. It is reasonable to divide the plotted data in Figure 10 into three classes: (1) low degree of non-constructability; (2) moderate degree of non-constructability; and (3) high degree of non-constructability.

Although we provided a sort of complexity analysis to prove the initial hypothesis about irregularity in the temporal evolution of the Persian language, this analysis seems to be unsatisfactory. Hence, other types of analysis have been done, which will be presented in the next subsections.

4.2 Statistical Analysis

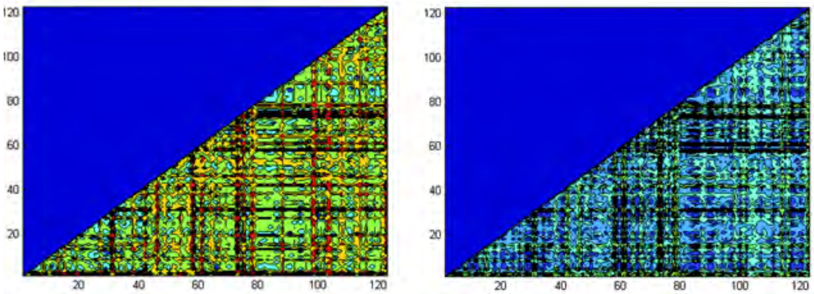
Let f and g be two rules that generate patterns (words) $W(f)$ and $W(g)$, respectively. Let $L(f, g)$ be a *Levenshtein* distance [47] between strings f and g, and $L(W(f), W(g))$ be a distance between patterns. When calculating these distances, two matrices are obtained, the contours of whose entries are depicted in Figure 11. In order to characterize the structure of the matrix $M(f, g)$ (see Figure 12) where every entry $m_{f,g} = c$ and c can be calculated by equation (6) as follows, see Figure 13:

$$L(f, g) = c \times L(W(f), W(g)). \tag{6}$$



(a) (b) (c)

Figure 12. (*continues*).



(d)

Figure 12. [left] Levenshtein distance between each two words [right]; Levenshtein distance between each two rules. (a) The person who is afflicted and worthy of seldom. (b) Every single day I ponder over these questions. (c) Message is in the way. (d) I am fighting.

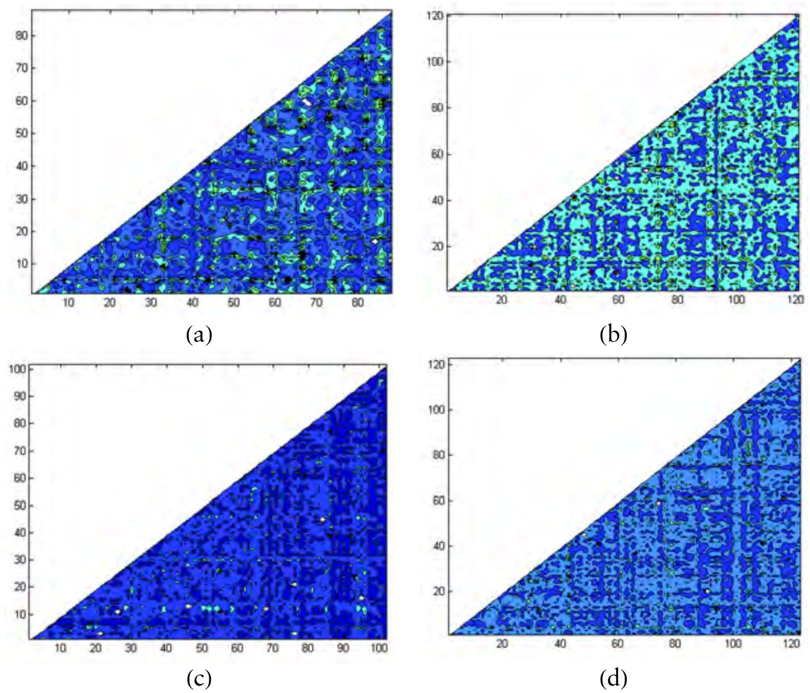


Figure 13. Structure of the coefficient matrices that are calculated from equation (6). (a) The person who is afflicted and worthy of seldom. (b) Every single day I ponder over these questions. (c) Message is in the way. (d) I am fighting.

https://doi.org/10.25088/ComplexSystems.25.2.127

Another statistical analysis that was done in our work was to calculate entropy followed by "morphological richness." In order to clarify this kind of analysis, it is worth defining morphological richness. Imagine an array of black and white pixels (in our case, each word that has been formed in a sentence). A local neighborhood of each pixel (including the pixel itself) has nine pixels (3×3). Morphological richness is calculated as the number of different configurations of 3×3 blocks divided by the number of all possible configurations (2^9). Even though the results must give us a deep sense about the restructuring of words during different eras, the chaotic nature of each plot leaves us far from the desired inference. Hence, the power spectrum [48] of the entropy of the obtained morphological richness is calculated to make the complexity analysis sensible.

One important way to look at a signal is to view its spectral density (i.e., the Fourier transform of the signal). The Fourier transform views the signal as a whole. It swaps the dimension of time with the dimension of frequency. One can think of the Fourier transform as a combination of slow and fast oscillations with different amplitudes. A very strong and slow component in the frequency domain implies that there is a high correlation between the large-scale pieces of the signal in time (macro-structures), while a very strong and fast oscillation implies correlation in the micro-structures. Therefore, if our signal $f(t)$ represents values in every single moment of time, its Fourier transform $F(\omega)$ represents the strength of every oscillation in a holistic way in that chunk of time. These two signals are related to each other by equation (7):

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \tag{7}$$

A true power spectrum of a signal has to consider the signal from $-\infty$ to ∞. However, we are not always able to observe a signal that way or derive precise functions for it. We can define $F_T(\omega)$, which is the Fourier transform of the signal in period T, and define the power spectrum as the following:

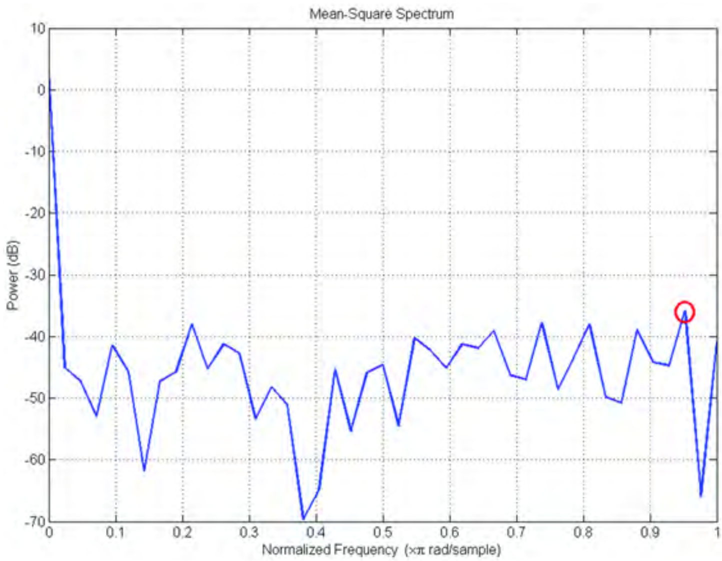
$$S_f(\omega) = \lim_{T \to \infty} \frac{1}{T} |F_T(\omega)|^2. \tag{8}$$

The power spectrum itself is the Fourier transform of the autocorrelation function. The autocorrelation function represents the relationship of long- and short-term correlation within the signal itself:

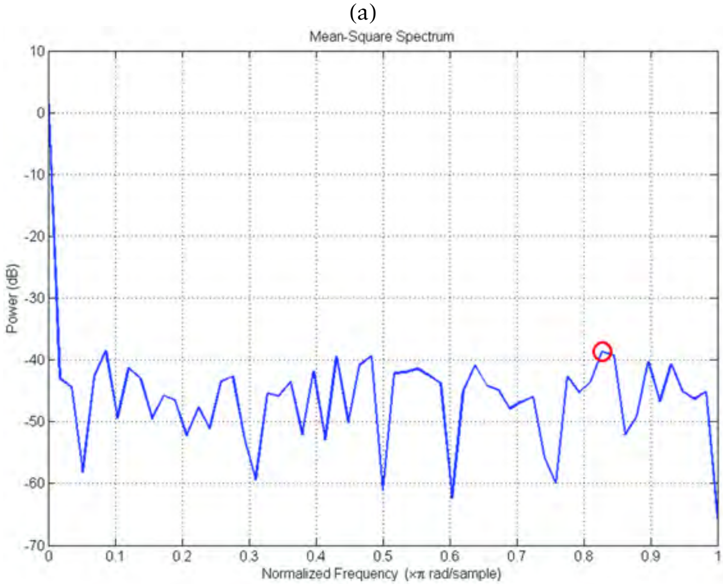
$$\langle f(t) f(t + \tau) \rangle = \frac{1}{2\pi} \int_0^{\infty} S_f(\omega) e^{j\omega t} d\omega. \tag{9}$$

https://doi.org/10.25088/ComplexSystems.25.2.127

The results of our analysis are illustrated in Figure 14. The “dominating frequencies” of the four poems, based on the time of creation, are 0.9524, 0.8276, 0.4792, and 0.4, respectively. This reduction provides a logical conclusion in which the type of words used in the verses of the poems has been changed.

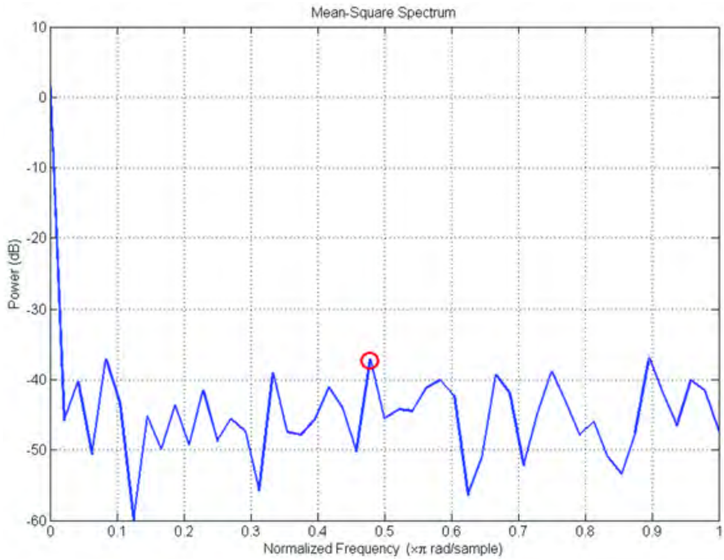


(a)

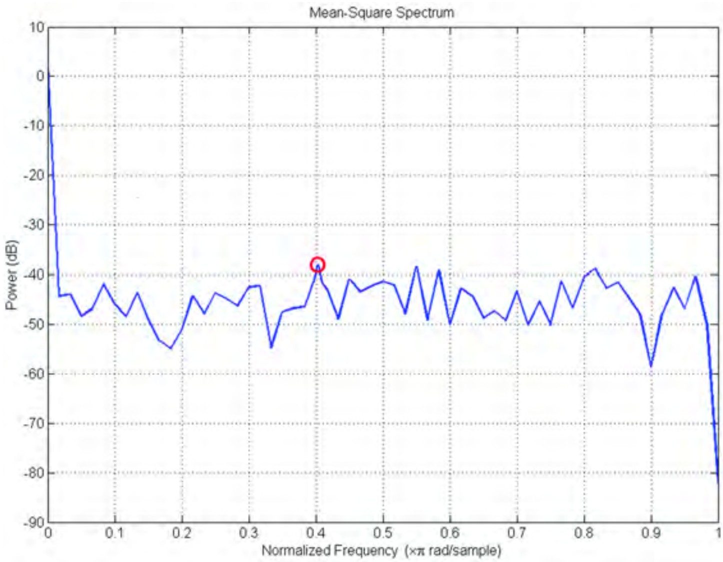


(b)

Figure 14. (*continues*).



(c)



(d)

Figure 14. Power spectrum of the entropy of the calculated morphological richness. (a) The person who is afflicted and worthy of seldom. (b) Every single day I ponder over these questions. (c) Message is in the way. (d) I am fighting.

Based on the evidence in Figures 12 (right side), 13, and 14, our findings confirm a well-known hypothesis of linguists that the Persian language is disordered over time.

5. Conclusion

We presented a method of growing sentences of the Persian language using the formalism of L-systems. We believe our method will contribute toward improved design of graphics and virtual reality components with cultural context, and design of CAD software systems used in the domain of cultural heritage, handicrafts, and the calligraphic, such as carpet weaving. Our evaluations of generative complexity might open several new venues in Persian linguistics. These are related to the complexity hierarchies of the words and the relation of the complexity hierarchies to the evolution of the Persian language. It seems that morphological richness along with entropy can derive a one-to-one mapping among the age of a word, language evolution, and the generative complexity of the word in a sentence. This analysis can then be used as a tool in a "linguistic archeology," and further, as a predictive technique for forecasting future developments in the Persian language.

We proposed L-systems as a tool for analyzing development of a written language over time. We demonstrated that L-systems can be used for graphical formation of a number of words in the Persian language and derived several measures of complexity to assist in analysis of language evolution.

Finally, we should touch on the topic of non-constructability. In cellular automaton theory [38–40] a configuration is called non-constructable or Garden-of-Eden [37, 49, 50] if it could not be reached from any other configuration by applying local rules of cell-state transitions. When adopting the concept in the generation of Persian sentences with L-systems, we can talk about a degree of non-constructability; the bigger the relative length of a *Start* string of a word, the higher the degree of non-constructability of the word. We can hypothesize that the higher the degree of non-constructibility of a word, the earlier the word emerged in the evolution of the Persian language. We plan to check this hypothesis in further research.

References

[1] M. Roussou, "The Components of Engagement in Virtual Heritage Environments," in *New Heritage: New Media and Cultural Heritage* (Y. Kalay, T. Kvan, and J. Affleck, eds.), New York: Routledge, 2008 pp. 225–241.

[2] K. B. Harandi, "Abstract Information Visualization in Interactive 3D Virtual Environments: Conceptualization and Usability Evaluation," Ph.D. thesis, University of Geneva, 2011.

[3] M. Keshavarz, A. Dehghanpour, and M. Ranjbarian, "Interactive Persepolis: A Study on Role of Interaction Design in Cultural Heritage Tourism," in *Proceedings of the Fifth Student Interaction Design Research Conference (SIDeR 'O9)*, Eindhoven University of Technology (I. Wouters, F. Kimman, R. Tieben, S. Offermans, and H. Nagtzaam, eds.), Netherlands: Eindhoven University of Technology, 2009 p. 18.

[4] D. Oreni, B. Cuca, and R. Brumana, "Three-Dimensional Virtual Models for Better Comprehension of Architectural Heritage Construction Techniques and Its Maintenance Over Time," in *Progress in Cultural Heritage Preservation*, Berlin, Heidelberg: Springer-Verlag, 2012 pp. 533–542. doi:10.1007/978-3-642-34234-9_ 55.

[5] A. Lindenmayer, "Mathematical Models for Cellular Interactions in Development I: Filaments with One-Sided Inputs," *Journal of Theoretical Biology*, **18**(3), 1968 pp. 280–299. doi:10.1016/0022-5193(68)90079-9.

[6] S. Kondo and T. Miura, "Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation," *Science*, **329**(5999), 2010 pp. 1616–1620. doi:10.1126/science.1179047.

[7] H. Arata, Y. Takai, N. K. Takai, and T. Yamamoto, "Free-Form Shape Modeling by 3D Cellular Automata," in *Shape Modeling and Applications*, Aizu-Wakamatsu, Japan, 1999, Los Alamitos, CA: IEEE, 1999 pp. 242–247. doi:10.1109/SMA.1999.749346.

[8] S. Xu, F. C. M. Lau, and Y. Pan, *A Computational Approach to Digital Chinese Painting and Calligraphy*, Berlin: Springer, 2009.

[9] M. Ahuja and A. L. Loeb, "Tessellations in Islamic Calligraphy," *Leonardo*, **28**(1), 1995 pp. 41–45. doi:10.2307/1576154.

[10] H. Moustapha and R. Krishnamurti, "Arabic Calligraphy: A Computational Exploration," in *Proceedings of Mathematics and Design 2001: The Third International Conference*, Geelong, Australia: Deakin University, 2001 pp. 294–306.

[11] S. A. H. Minoofam and A. Bastanfard, "Square Kufic Pattern Formation by Asynchronous Cellular Automata," in *Proceedings of the 9th International Conference on Cellular Automata for Research and Industry*, Ascoli Piceno, Italy (S. Bandini, S. Manzoni, H. Umeo, and G. Vizzari, eds.), Berlin, Heidelberg: Springer, 2010 pp. 79–82.

[12] S. A. H. Minoofam, M. M. Dehshibi, A. Bastanfard, and P. Eftekhari, "Ad-hoc Ma'qeli Script Generation Using Block Cellular Automata," *Journal of Cellular Automata*, 7(4), 2012 pp. 321–334.

[13] S. A. H. Minoofam, M. M. Dehshibi, A. Bastanfard, and J. Shanbehzadeh, "Pattern Formation Using Cellular Automata and L-Systems: A Case Study in Producing Islamic Patterns," in *Cellular Automata in Image Processing and Geometry* (P. Rosin, A. Adamatzky, and X. Sun, eds.), New York: Springer, 2014 pp. 233–252. doi:10.1007/978-3-319-06431-4_ 12.

[14] D. D. Hearn, M. P. Baker, and W. Carithers, *Computer Graphics with Open GL*, 4th ed., Boston: Addison Wesley, 2011.

[15] A. Salomaa, *Formal Languages*, San Diego: Academic Press, 1974.

[16] A. Lindenmayer, "Developmental Systems without Cellular Interactions, Their Languages and Grammars," *Journal of Theoretical Biology*, **30**(3), 1971 pp. 455–484. doi:10.1016/0022-5193(71)90002-6.

[17] A. R. Smith, "Plants, Fractals, and Formal Languages," *ACM SIGGRAPH Computer Graphics*, **18**(3), 1984 pp. 1–10. doi:10.1145/964965.808571.

[18] P. Prusinkiewicz, "Graphical Applications of L-Systems," in *Proceedings of Graphics Interface '86/Vision Interface '86*, Los Altos, CA: Morgan Kaufman Publishers, Inc., 1986 pp. 247–253.

[19] P. Prusinkiewicz, "Score Generation with L-Systems," in *Proceedings of the 1986 International Computer Music Conference (ICMC 1986)*, Den Haag, Netherlands, Ann Arbor, MI: Michigan Publishing, 1986.

[20] C. Dodge and C. R. Bahn, "Musical Fractals: Mathematical Formulas Can Produce Musical as Well as Graphic Fractals," *Byte*, **11**(6), 1986 pp. 185–196.

[21] R. F. Voss and J. Clarke, "1/f Noise in Music: Music from 1/f Noise," *The Journal of the Acoustical Society of America*, **63**(1), 1978 pp. 258–263. doi:10.1121/1.381721.

[22] P. Worth and S. Stepney, "Growing Music: Musical Interpretations of L-Systems," in *Applications of Evolutionary Computing*, 2005 pp. 545–550. doi:10.1007/978-3-540-32003-6_ 56.

[23] M. M. Dehshibi, A. Shirmohammadi, and A. Adamatzky, "On Growing Persian Words with L-Systems: Visual Modeling of Neyname," *International Journal of Image and Graphics*, **15**(3), 2015 1550011. doi:10.1142/S0219467815500114.

[24] M. M. Dehshibi, A. Shirmohammadi, and A. Adamatzky, "Pattern Formation Using L-Systems: A Case Study in Forming Neyname's Words," in *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2014 (ICNAAM-2014)*, Rhodes, Greece (T. E. Simos and C. Tsitouras, eds.), Melville, NY: AIP Publishing, 2015 580015. doi:10.1063/1.4912823.

[25] M. Gholamain and E. Geva, "Orthographic and Cognitive Factors in the Concurrent Development of Basic Reading Skills in English and Persian," *Language Learning*, **49**(2), 1999 pp. 183–217. doi:10.1111/0023-8333.00087.

[26] J. Cong and H. Liu, "Approaching Human Language with Complex Networks," *Physics of Life Reviews*, **11**(4), 2014 pp. 598–618. doi:10.1016/j.plrev.2014.04.004.

[27] P. T. Daniels and W. Bright (eds.), *The World's Writing Systems*, New York: Oxford University Press, 1996.

[28] G. Jean, *Writing: The Story of Alphabets and Scripts*, New York: H. N. Abrams, 1992.

[29] M. Mirdehghan, "Persian, Urdu, and Pashto: A Comparative Orthographic Analysis," *Writing Systems Research*, **2**(1), 2010 pp. 9–23. doi:10.1093/wsr/wsq005.

[30] A. Estaji and A. F. Pooresfahani, "The Investigation of Punctuation in Photographic Copies of Persian Writing," *Theory and Practice in Language Studies*, **2**(5), 2012 pp. 1090–1097. doi:10.4304/tpls.2.5.1090-1097.

[31] M.-S. Adib-Soltani, *How to Write a Book* (in Persian), Tehran: Sazman Entesharat va Amuzesh Enghelab Eslami, 1987.

[32] M. Khayam, *Future of Written Language* (in Persian), Tehran: Negah Press, 1955.

[33] M. J. Yahaghi and M. M. Naseh, *A Guide to Writing and Editing*, 3rd ed. (in Persian), Tehran: Astane Ghods Press, 2015.

[34] D. Levinson and K. Christensen (eds.), *Encyclopedia of Modern Asia*, New York: Charles Scribner's Sons, 2002.

[35] E. G. Browne, *A Literary History of Persia*, London: Routledge, 2012.

[36] A. J. Arberry (ed.), *The Legacy of Persia*, Oxford: Clarendon Press, 1953.

[37] J. Myhill, "The Converse of Moore's Garden-of-Eden Theorem," *Proceedings of the American Mathematical Society*, **14**(4), 1963 pp. 685–686. doi:10.1090/S0002-9939-1963-0155764-9.

[38] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1–2), 1984 pp. 1–35. doi:10.1016/0167-2789(84)90245-8.

[39] S. Wolfram, "Cellular Automata as Models of Complexity," *Nature*, **311**(5985), 1984 pp. 419–424. doi:10.1038/311419a0.

[40] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[41] A. Adamatzky and G. J. Martinez, "On Generative Morphological Diversity of Elementary Cellular Automata," *Kybernetes*, **39**(1), 2010 pp. 72–82. doi:10.1108/03684921011021282.

[42] A. Adamatzky, "On Diversity of Configurations Generated by Excitable Cellular Automata with Dynamical Excitation Intervals," *International Journal of Modern Physics C*, **23**(12), 2012 1250085. doi:10.1142/S0129183112500854.

[43] S. Ninagawa and A. Adamatzky, "Classifying Elementary Cellular Automata Using Compressibility, Diversity and Sensitivity Measures," *International Journal of Modern Physics C*, **25**(3), 2014 1350098. doi:10.1142/S0129183113500988.

[44] M. M. Dehshibi, M. Alavi, and J. Shanbehzadeh, "Kernel-Based Persian Viseme Clustering," in *13th International Conference on Hybrid Intelligent Systems (HIS)*, Los Alamitos, CA: IEEE, 2013 pp. 129–133. doi:10.1109/HIS.2013.6920468.

[45] M. Aghaahmadi, M. M. Dehshibi, A. Bastanfard, and M. Fazlali, "Clustering Persian Viseme Using Phoneme Subspace for Developing Visual Speech Application," *Multimedia Tools and Applications*, **65**(3), 2013 pp. 521–541. doi:10.1007/s11042-012-1128-7.

[46] M. M. Dehshibi and J. Shanbehzadeh, "Persian Viseme Classification Using Interlaced Derivative Patterns and Support Vector Machine," *Journal of Information Assurance & Security*, **9**(3), 2014 pp. 148–156.

[47] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys (CSUR)*, **33**(1), 2001 pp. 31–88. doi:10.1145/375360.375365.

[48] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, "Power Spectrum Estimation Using the FFT," *Numerical Recipes in C*, 2nd ed., v2.0, New York: Cambridge University Press, 1992.

[49] K. Sutner, "Linear Cellular Automata and the Garden-of-Eden," *The Mathematical Intelligencer*, **11**(2), 1989 pp. 49–53. doi:10.1007/BF03023823.

[50] T. Toffoli, "Computation and Construction Universality of Reversible Cellular Automata," *Journal of Computer and System Sciences*, **15**(2), 1977 pp. 213–231. doi:10.1016/S0022-0000(77)80007-X.