

On Generative Poetry: Structural, Stylistic and Lexical Features

Svetlana Anatolevna Kuchina

NOVOSIBIRSK STATE TECHNICAL UNIVERSITY, RUSSIA

ABSTRACT

This paper deals with key aspects of the Oulipo and Dada methods and their implementation in electronic generative poetry. Oulipian constraints such as acrostics, tautograms, simple numerical limitations and combinatory algorithms are easily integrated into digital environments. The analysis of structural, lexical and stylistic peculiarities of generative poetry is illustrated by permutational schemes (*Poem.exe* by Liam Cooke, *Book of all Words* by Józef Zuk Piwkowski), combinatory patterns (*Frequency* by Scott Rettberg) and syntactic templates (*Dizains* by Marcel Bénabou, *Triolets* by Paul Braffort) of electronic poems. Many combinatory and permutational electronic poems present technologically improved versions of the Oulipo constraints and Dada techniques such as open-form poetry and the use of image and graphic components in its structure. However, the electronic environment gives them an ambivalent status. Although the surface of an electronic poem looks open and random, its inward structure is preconfigured to use established parameters.

KEYWORDS

generative poetry; Oulipian constraints; Dada; aleatoric; permutational; combinatory.

RESUMO

Este artigo trata certos aspectos-chave dos métodos Oulipo e Dada e da sua implementação na poesia eletrônica gerativa. As restrições Oulipianas, como os acrósticos, tautogramas, limitações numéricas simples e algoritmos combinatórios, são facilmente integradas em ambientes digitais. A análise das peculiaridades estruturais, lexicais e estilísticas da poesia gerativa é ilustrada por esquemas permutacionais (*Poem.exe* de Liam Cooke, *Book of all Words* de Józef Zuk Piwkowski), padrões combinatórios (*Frequency* de Scott Rettberg) e modelos sintáticos (*Dizains* de Marcel Bénabou, *Triolets* de Paul Braffort) de poemas eletrônicos. Muitos dos poemas eletrônicos combinatórios e permutacionais apresentam versões tecnologicamente aprimoradas das restrições Oulipo e de técnicas Dada, tais como a poesia de forma aberta e o uso de componentes gráficos e de imagem na sua estrutura. No entanto, o ambiente eletrônico também lhes confere um estatuto ambivalente. Embora a superfície de um poema eletrônico pareça aberta e aleatória, a sua estrutura interna está pré-configurada para usar parâmetros estabelecidos.

PALAVRAS-CHAVE

poesia gerativa; restrições Oulipianas; Dada; aleatório; permutacional; combinatório.

Digital poetry was launched by poets who experimented with computers in the late 1950s. The genre of digital poetry in terms of its media, methods, and expressive intent contains diversified components (sound, graphics, animation) and techniques. All forms of digital poetry comprise a singular genre that contains multiple subcategories such as kinetic, visual, hypertext, hypermedia and generative poetry.

Generative poetry, produced by computer algorithms, is arranged as a sequence of words, or signs and symbols, according to a program. Some of the experimental features and structural principles of generative poetry were invented and practically applied long before the computing era and the World Wide Web. Some of them were originally used by Oulipo (Ouvroir de littérature potentielle), the workshop of potential literature established by French writers and mathematicians who create their works using combinatorial techniques and constrained writing (such as lipograms, alliteratives, acrostics, palindromes, anagrams, aleatory, etc.).

In this paper I will try to apply the 'genre reading' method to some generative poems and analyze them in the light of constrained writing, combinatorial and Dada traditions.

The earliest example of electronic literature in Poland is the *Book of all words* by Józef Żuk Piwowski (Piwowski, 2016). The program demonstrates the permutational possibilities of the Latin alphabet. The work represents the idea of an infinite set of words. The verbal data of the *Book of all words* is based on twenty-six letters of the Latin alphabet. The current version is an online edition in HTML.

The *Book of all words* program creates word strings that run from one letter in length to infinity where every next word is similar to the previous one apart from one letter. In this context we may say that it partly demonstrates the tautogram principle. The tautogram is a text in which all words start with the same letter. Historically, tautograms (which are a special case of alliteration) were mostly poetical forms and broadly known and used by European poets. Being one of the forms of constrained writing, tautograms can be taken to extraordinary extremes. That was demonstrated by Walter Abish in his tautogram novel *Alphabetical Africa* (Abish, 1974).

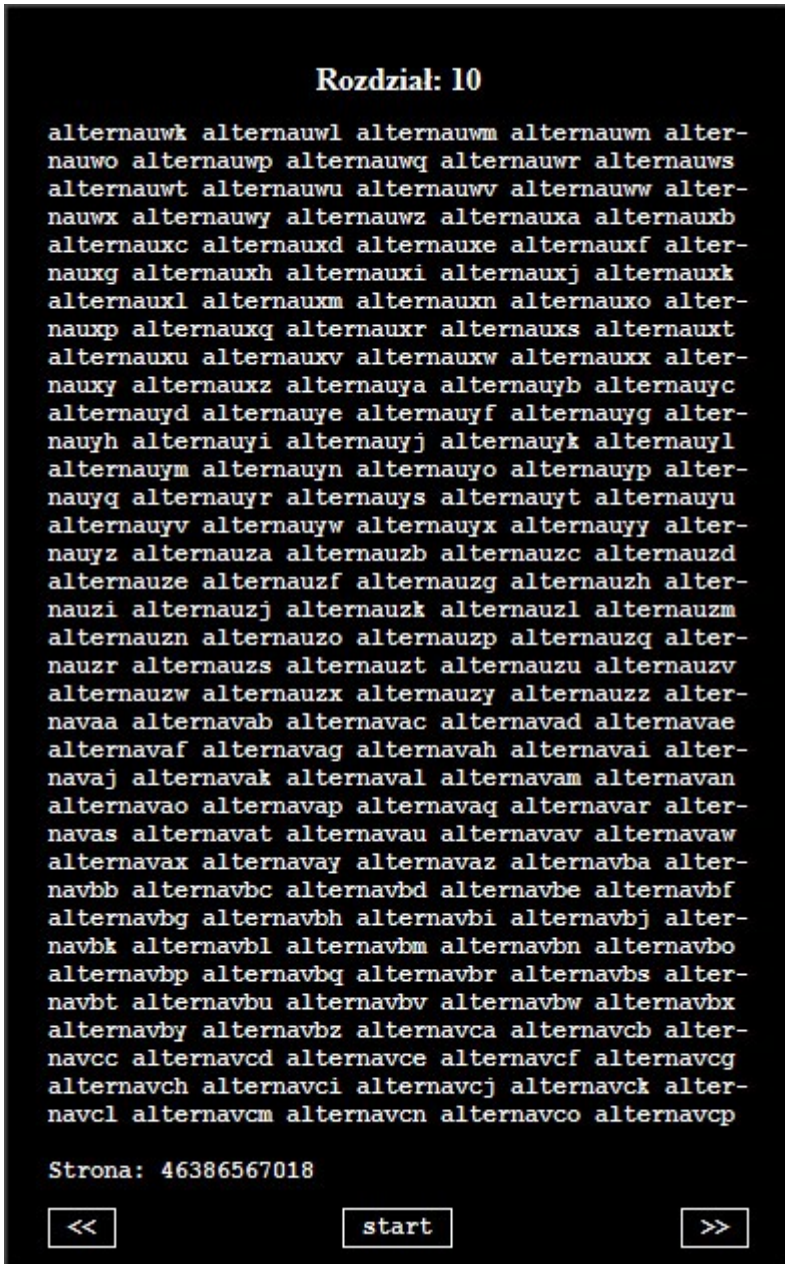


Figure 1. *Book of all words* by Józef Zuk Piwkowski.

The tautogram technique is not fully executed in the *Book of all words* program but the main principle is retained. Each page contains up to 1,800 letters, and the word sections are divided into chapters which unite the patterns starting with the same letter or even the same letter combinations. The only way to

interact with the program is to choose the word or the page number to look through. Most of the generated character combinations sound like pure language noise, thus the program generates the patterns that do not even exist in any official dictionary. It is possible to say that the *Book of all words* operates in the realm of the sublime. Its sublimity is mathematically-based because the number of generated word-like patterns grows beyond human comprehension and that is why the program cannot have a printed version.

On the one hand the tautogram principle gives the *Book of all words* the look of a system where the words are produced according to the strict alphabetic order. On the other hand, the conception of the program is very close to the Dadaist way of perceiving poetry. Dadaist artists suggested the simultaneous, noise-induced, phonetic and optophonetic poetry which was represented by random combinations of sentences, words or letters and was aimed at avoiding “any meaning in any specific language” (Wilmanski, 2009).

Generative poetry is the oldest e-poetry genre, but it remains relevant today through the new e-literary genres like bots (small autonomous programs that post to Twitter or other social media). Most Twitter bots are based on permutational schemes (recombining elements into new words or variations).

Poem.exe by Liam Cooke is a micropoetry bot that generates three or four-line haiku-like poems (Cooke, 2016). The *Poem.exe* relates to the tradition of permutational poetry. The *Poem.exe* verbal data is based on haiku poems written by Koboyashi Issa (Japanese poet, 1763-1828). The program selects random lines from Issa’s database keeping the basic syntax and lineation intact and tweets the resulting poem every two hours. The essential element of this work is its automatic interface, which needs no interaction with the reader and displays automatically generated poems.



Figure 2. *Poem.exe* by Liam Cooke.

Haiku is one of the traditional constrained writing forms. It is a very short poem, usually adopting a three-line format with seventeen syllables arranged in a five–seven–five patterns focused on some aspect of nature or the seasons. Haiku uses no punctuation or capitalization, except proper nouns that are usually capitalized. Thanks to these strict rules haiku can be easily adapted to automatic algorithm and wildly used in generative poetry.

Koboyashi Issa's poetry, which is the basis for *Poem.exe*, contains many verses on natural life, animals and insects (snails, frogs, fireflies, fleas, cicadas, etc.) The *Poem.exe* bot filters out resulting poems that do not reflect these topics and does not publish them. That is why most of the generated poems are colored by Issa's personality. Due to translation transformations not all of the generated poems correspond to the Haiku syllable scheme, and some of them are a bit longer or shorter than the original seventeen-syllable patterns.

The way the *Poem.exe* Twitter bot (and almost all bots) represents the content reflects the idea of Merz art style that traces back to the Dada movement and Kurt Schwitters. The collage principle that he suggested assumes an expanding poetry toolbox, mixing different components and producing an art object from this mixture where each element plays a creative role as a means of expression.

The most frequent type of permutation procedures of algorithmically generated poems is the combinatorial algorithm. The method was suggested by Raymond Queneau in his *A Hundred Thousand Billion Poems* where the fourteen lines from ten sonnets with the same sets of rhymes can be exchanged to produce one hundred thousand billion unique sonnets. In the poetry generator *Frequency* written by Scott Rettberg this principle is used along with the different constraints based on rhyme schemes, syllable and character count (Rettberg, 2016).

Frequency is a large constrained writing project about contemporary American life. Its verbal data is two hundred of the most common English words which are the basis for two thousand lines of all the poems. *Frequency* is written in HTML and JavaScript and runs in a web browser. The program generates poems based on different rhyme schemes (Shakespearian, Spenserian, Petrarchan and Terza rima), syllabic forms (Haiku, Tanka, Oulipian snowball) and original authors' models such as *Two towers*, *Four square* and *Doubling*.

The original models are the matter of most interest. Some of these forms have roots in the Oulipian constraints, for instance, the *Four square* and *Two towers*. The *Two Towers* poems generate two character-based stanzas next to each other. This form presents multiple ways of reading these stanzas: individually, sequentially, or line by line to form new line combinations (Flores, 2013).

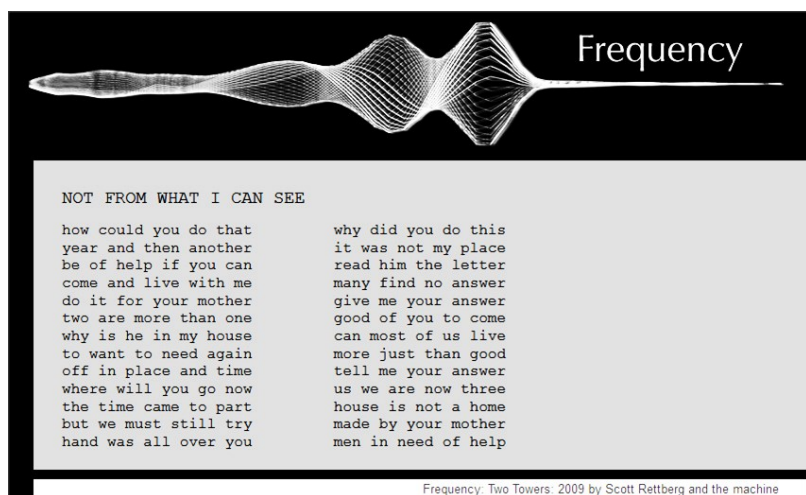


Figure 3. *Frequency* by Scott Rettberg (Two towers).

The *Four Square* poem has four four-line stanzas selected by character length. Some of the lines in the generated poems may be repeated in the first two stanzas (because they are generated from an automated array), which give the poem a refrain-like tone.

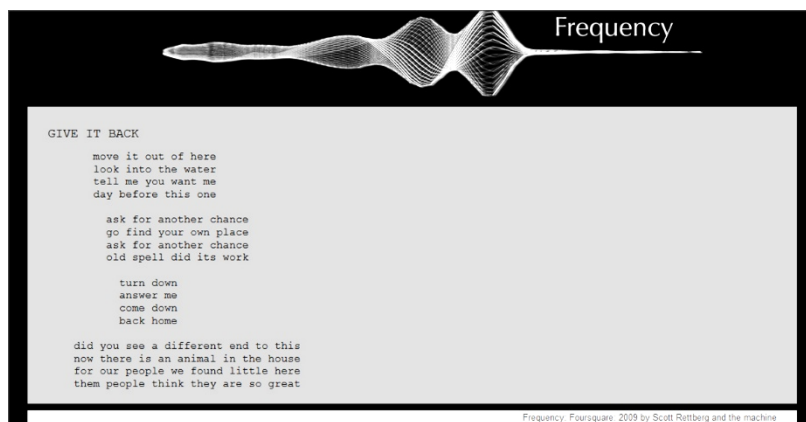


Figure 4. *Frequency* by Scott Rettberg (Four Square).

The process of *Frequency* is completely automatic and presupposes no interaction from the readers; generated poems appear at timed intervals. The number of words in the lines as well as the number of letters in the words in the *Four square* and *Two towers* poems is not fixed but the number of signs in the lines (including the spaces) is predetermined for each generated pattern. Leonardo Flores noticed that Scott Rettberg “took pictorial control over the text in these two forms which are based on the machine’s ability to count characters in a line

(including spaces), use that information to generate arrays and poems, and use a fixed-width font” (Flores, 2013).

The *Four square* model is very close to the simple numerical limitations (which are frequently used by Oulipo members): poetry texts with fixed number of verses, lines or words or even letters in the words. Oulipo members tried to complicate this task, requiring the use of equal number of letters in all words in the poem, or equal number of words in all verse lines, as it was demonstrated in the *Rail* by Georges Perec (Perec, 1976). Also the *Four square* model has its roots in Perec’s heterogrammatical poems which are based on the tradition of word square, an acrostic which consists of a set of words written out in a square grid, such that the same words can be read both horizontally and vertically. In his heterogrammatical poems Perec used eleven letters and placed them in the square table where each line was made with the help of chosen letters permutation. Then all of these permutational combinations form the line of one hundred twenty one letters. Using spaces and punctuation marks it was possible to read the result as a small poem (Bonch-Osmolovskaya, 2005).



Figure 5. “Satin, or bleu ...” by Georges Perec.

Apart from the combinatoric technique, some generative works are based on grammatical frames of fixed forms. This technique usually helps to create an “illusion of sense” in the resulting poems (Funkhouser, 2008). Fixed forms as examples of constrained writing were widely used by Oulipo members, but the origin of almost all fixed forms are peasant songs from oral traditions. Those rustic songs and dances were supposed to be repeated several times in circular motion. In the beginning of the twentieth century there was a new surge and interest in fixed forms among poets. Valery Bryusov, Konstantin Balmont, Igor Severyanin, Maximilian Voloshin put to use traditional fixed forms and invented

new ones. Some of the fixed forms are still being exploited in printed and electronic literature.

The *Dizains* by Marcel Bénabou partly explore one of the fixed form schemes: the ballade (Bénabou, 2016). As a verse form it was commonly popular in French poetry between the fourteenth and fifteenth centuries. The ballade typically consists of three eight or ten-line stanzas (*dizains*), each with a consistent meter and a particular rhyme scheme (ababbccdd). But Bénabou modified the *dizain* form into twelve syllables of French alexandrine with a medial caesura which divides the line into two half-lines of six syllables each. The *Dizains* code algorithm permutes six pairs of rhyming lines in such a way that each pair can be separated by zero, one, or two other lines. The algorithm is capable of producing 145,920 *dizain* variations.

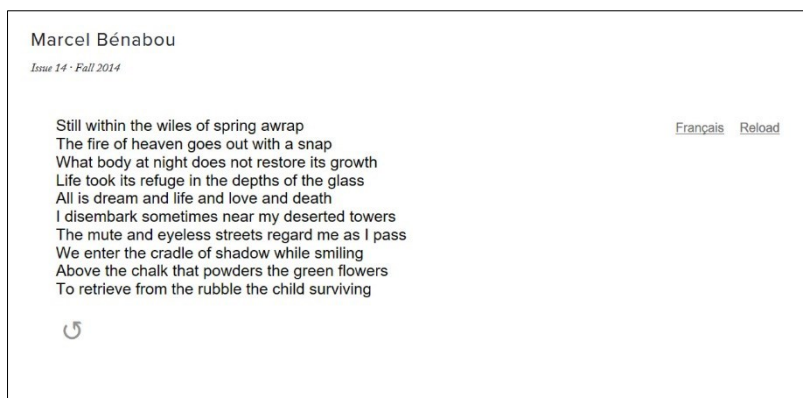


Figure 6. *Dizains* by Marcel Bénabou.

Paul Braffort's *Triolets* is based on the triolet scheme which is a French fixed form from the Middle Age (Braffort, 2016). It uses the rhyme scheme ABAA ABAB where the first, fourth and seventh lines are identical, as are the second and final lines, thereby making the initial and final couplets identical as well. Braffort's *Triolets* generator produces 7,776 unique triolets from the six compatible examples composed by the author.

Triolets by Paul Braffort, translated by Erik Stayton and Patsy Baudoin

My love is a wisp illuminate
 a beating heart of tintinnabulation
 it's a cry disproportionate
 My love is a wisp illuminate

That curtains and blinds exhibit
 one gets lost in the labyrinthian
 My love is a wisp illuminate
 a beating heart of tintinnabulation



Figure 7. *Triolets* by Paul Braffort.

Keeping the meter and lineation of the traditional fixed form (triolet and dizain) intact, Paul Braffort and Marcel Bénabou ignore the basic syntax. The lack of punctuation marks in generated poems is the essential feature of both works. The original triolet, for instance, has a fixed scheme of punctuation marks: the period is usually used at the end of the second and fourth lines and at the end of the poem. Somehow the syntactic scheme can be reconstructed with the help of capitalization used in Braffort's *Triolets*, where the first words of first, fourth, fifth and seventh lines are capitalized. In the *Dizains* by Bénabou, the first word in each line is usually capitalized.

From a lexical point of view, the generated poems, especially the triolets, are broadly in line with the style of the classical Middle Age triolets. The way the characters are presented in Braffort's *Triolets* is very close to the thematic categories of French Middle Age poetry (*la reine, le charme, le poète, d'amour*, etc.) connected with focusing on the poet's personality and its praise (*le poète en garde l'empreinte, portant rimes à mi-mollets*) and the sublimation of love affairs (*Les mots d'amour font un ballet, le charme ou l'arme ou la contrainte*). The multiple locations which are mentioned in the generated triolets also refer to France (Perros-Guirec, Virson, Zermatt) (Braffort, 2016).

One last aspect worth mentioning in describing the combinatorial algorithms is the use of Boolean algebra. The mathematical notion of "a set" was frequently reflected in theater and poetry. For example, François Le Lionnais applied this mathematical method to produce new poems on the borderland between two classical sonnets (by Georges de Brébeuf and his *Damoiselle Chercher la Vertu* and Pierre Corneille and his *Epitaphe*). The new poem by François Le Lionnais was

formed from the words that coincided, used by both authors (Bonch-Osmolovskaya, 2005). To some extent this method can be applied to generative poetry especially when we speak about combinatory algorithms based on syntactic templates where the resulting poems are generated from subsets of preselected words.

According to Christopher T. Funkhouser the permutation procedures of algorithmically generated poems can be divided into three categories (Funkhouser, 2008). The first type of these categories is a permutational scheme which is based on the algorithms recombining elements into new words (*Book of all words* by Piwowski) or new poem lines (*Poem.exe* by Cooke). The second type of poem generators is based on combinatory patterns using limited, pre-set word lists in controlled or random combinations (*Frequency generator* by Rettberg). And the third type, which is very close to the combinatory technique, is based on syntactic templates and grammatical frames to create an image of “sense” (*Dizains* by Benabou, *Triolets* by Braffort).

Dada and Oulipo postulate totally different philosophies concerning the creation of literary works. Denying any certainty, Dada tried to find freedom everywhere (language meaning, cultural independence, poetry). According to Dada such techniques as open-form poetry, the use of collage principles (mixture of sound, image, and graphic components) and irrationality free and update art. On the contrary, Oulipo uses strict constraints to investigate the potentiality of language. Both visions (Dada and Oulipo) reincarnate, in some sense, in the age of information technology and Internet. The digital environment makes it possible to implement both strategies in a form of generative poem, transforming absolute freedom into random choice of preselected elements made by the strict constraint of computer algorithm.

REFERENCES

- ABISH, Walter (1974). *Alphabetical Africa*. New York: Directions Publishing Corporation.
- BÉNABOU, Marcel (2016). “Dizains.” *Electronic Literature Collection 3*. Cambridge, MA: Electronic Literature Organization. 31 May 2017. <http://collection.eliterature.org/3/work.html?work=dizains>.
- BONCH-OSMOLOVSKAYA, Tatiana (2005). “Lectures on combinatorial literature. Lecture 12.” *Articles on combinatorial art*. 31 May 2017. http://www.ashtray.ru/main/texts/bonch_course/l12-1.htm.
- BRAFFORT, Paul (2016). “Triolets.” *Electronic Literature Collection 3*. Cambridge, MA: Electronic Literature Organization. 31 May 2017. <http://collection.eliterature.org/3/work.html?work=triolets>.
- COOKE, Liam (2016). “Poem.exe.” *Electronic Literature Collection 3*. Cambridge, MA: Electronic Literature Organization. 31 May 2017. <http://collection.eliterature.org/3/work.html?work=poem-exe>.
- FLORES, Leonardo (2013). “Frequency [Four Square, Two Towers] by Scott Rettberg and the Machine (part 5 of 5).” *I love poetry*. 31 May 2017. <http://ilovepoetry.com/?p=66>.
- FUNKHOUSER, Christopher (2008). “Digital Poetry.” *Companion to Digital Literary Studies*. Eds. Susan Schreibman and Ray Siemens. Oxford: Blackwell. 31 May 2017. http://digitalhumanities.org/3030/companion/view?docId=blackwell/9781405148641/9781405148641.xml&chunk.id=ss1-5-11&toc.depth=1&toc.id=ss1-5-11&brand=9781405148641_brand.
- PEREC, Georges (1976). *Alphabets*. Paris: Galilée.

- PIWKOWSKI, Józef Żuk (2016). "Book of all words." *Electronic Literature Collection 3*. Cambridge, MA: Electronic Literature Organization. 31 May 2017. <http://collection.eliterature.org/3/work.html?work=book-of-all-words>.
- RETTBERG, Scott (2016). "Frequency." *Electronic Literature Collection 3*. Cambridge, MA: Electronic Literature Organization. 31 May 2017. <http://collection.eliterature.org/3/work.html?work=frequency>.
- WILMANSKI, Tomasz (2009). "Visual Poetry in the context of artist's book." *Galeria At*. 31 May 2017. http://www.galeria-at.siteor.pl/text-books-ang?edit_mode=true.

© 2018 Svetlana Anatolevna Kuchina.

Licensed under the [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 International \(CC BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/).