MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **82** (2019) 549-560

ISSN 0340 - 6253

Grammars of Organic Chemistry

Grzegorz Matyszczak

Faculty of Chemistry, Warsaw University of Technology

Noakowskiego street 3, 00-664 Warsaw, Poland.

gmatyszczak@ch.pw.edu.pl

ORCID: 0000-0001-7521-2284

(Received June 22, 2019)

Abstract

Automata theory has been used in the subject of organic chemistry. The alphabet of the language of all organic compounds has been proposed and discussed, furthermore, universal system of notation has been presented that allows writing certain organic compound as a string. Deterministic finite-state automaton have been constructed to decide whether the organic compound can be synthesized in the esterification reaction. Then deterministic finite-state automaton for more synthetically useful organic reaction, which is the Radziszewski reaction, has been presented as well as a discussion of problems that arise while constructing finite-state automata for more complex organic reactions. A proof that the grammar of organic compounds language is not regular has been made based on the contrapositive of the Pumping Lemma. The algebraic structure of organic chemistry has been briefly discussed. Directions of further investigations that would allow more applications of automata theory in chemistry are outlined, especially in chemical informatics and organic synthesis planning.

1 Introduction

History of automata theory starts in the 1930s when A. Turing studied an abstract machine with computing capability the same as modern computers have, then in the 1940s and 1950s concept of finite automata has been introduced [1]. Automata theory, which is a discipline on the border of mathematics and informatics and underlies modern computer science, is useful in many other sciences. In biology among applications modeling of biological development can be mentioned [2]. In biochemistry automata theory has been used to analyze the algebraic structure of biochemical reactions on the example of Krebs citric acid cycle [3]. It was the first time when such analysis was proposed. Wang and others have built finite automata with 16 states based on 3 types of logic gates, that showed memory [4]. Oishi and Klavins showed a method to build any finite-state automaton with repressing transcription factors and also equivalence of finite-state machine and Boolean model of gene regulatory networks [5]. Finite-state automaton have been also used (together with cell signaling) in the design of new types of multicellular behavior in synthetic biology [5]. Finite state machines have been as well applicated in modeling and recognition of gestures utilizing fact, that certain gesture composes of ordered number of states [6]. Thanks to that it was possible to create an application, which in real time could recognize human gestures [6]. In informatics finite automata are a good model for many important kinds of software and hardware, also they are base of many algorithms, among others to text mining and Web page scanning [1]. Furthermore, it has been proved, that finite state machines can be utilized in the same way as recurrent neural networks [7 – 10]. From a properly trained neural network, it is possible to extract deterministic finite-state automaton which performance is even better according to the same task in comparison to neural network [7]. Also, it has been proved, that in the case of an adequately long training period of neural network it can become true finite state automata [8]. An algorithm for constructing recurrent neural networks that implements a finite state machine has been proposed and proved [10]. It should be emphasized that finite-state automata, unlike artificial neural networks, do not require any training procedure. From a physical point of view, finite-state automata have been investigated in terms of irreversibility and dissipation [11].

In chemical informatics, many methods of artificial intelligence have found to be useful, among other support-vector machines, genetic algorithms and neural networks [12 - 14]. Particularly important are applications of these algorithms in new drugs and materials design

[15 - 19]. Finite state machines in chemical informatics for the first time have been used to implement fast text-based molecular similarity searching (LINGO) [20]. Sayle and others have utilized finite state machines to the recognition of chemical compounds names in patents of pharmaceutical interest [21]. Furthermore, the concept of Chemical Reaction Automata, which in a maximally parallel manner are computationally equivalent to Turing machines, has been proposed [22]. Also, chemical kinetic implementations of neural networks and finite-state machines have been shown [23, 24]. As can be seen, there is a lot of interesting connections between chemistry and informatics.

In this article automata theory has been used in problems of organic chemistry. Organic compounds are threatened as words in the language of organic chemistry, generated by respective grammars. Finite-state automata that decide if an organic compound can be synthesized by a certain chemical reaction are constructed. Some aspects of grammars of organic chemical compounds are discussed based on the contrapositive of the Pumping Lemma. The system of linear notation of organic compounds, allowing to process them through finite-state machines, is presented and discussed. This work makes foundations for further applications of mathematical linguistics and automata theory in chemistry.

2 Theoretical basis

2.1 A brief introduction to automata theory

Basic concepts of the theory of automata are: alphabet, strings, and language. An alphabet is a finite, nonempty set of symbols, which here is denoted by $\sum$. Examples of the alphabet are the binary alphabet ($\sum = \{0, 1\}$) and the Latin alphabet. A string is a finite sequence of symbols that belongs to some alphabet. For example, the word "alphabet" is a string from the Latin alphabet. A language is a set of strings composed of symbols belonging to that same alphabet.

A grammar G is a tuple G = (V, T, P, S), where V – a finite set of nonterminal symbols, T – a finite set of terminal symbols, S – the starting symbol of grammar ($S \in V$), P – a finite set of productions. Grammars generate strings that belong to a certain language. There are four classes of grammars: regular (RgL), context-free (CFL), context-sensitive (CSL) and unrestricted (REL). They are ordered in the following hierarchy:

$$\mathrm{RgL} \subset \mathrm{CFL} \subset \mathrm{CSL} \subset \mathrm{REL}$$

Regular grammars generate regular languages and so on. Strings that belong to regular language are accepted (recognized) by finite-state automata (FSA). FSA is a system composed of 5 components, $A = (Q, \sum, \delta, q_0, F)$ where: Q – set of states, $\sum$ - alphabet, q_0 – starting state ($q_0 \in Q$), F – set of accepting states, $\delta: Q \times \sum \rightarrow Q$ – transition function. Finite-state automata recognize strings only from regular languages. For more information about automata theory and mathematical linguistics see the handbook of Hopcroft and others [1].

2.2 System of notation and alphabet

System of notation that allows writing organic compounds as strings is needed to make them suitable for proceeding by finite-state automata. In system used in this article each symbol in string has a general form $X^{(a)}_{b,\ c}$ where X is a chemical element symbol, (a) means maximal order of each bond in which certain atom is involved (for example (a) = (2) for sp^2 carbon atom or (1) for any hydrogen atom), b is stoichiometric index and c points out if certain atom begins (c = s) or ends (c = e) ring. Including typical chemical occurrence of elements in organic compounds, the following alphabet has been proposed:

carbon: $C^{(1)}$, $C^{(2)}$, $C^{(3)}$, $C^{(1)}_s$, $C^{(2)}_s$, $C^{(3)}_s$, $C^{(1)}_e$, $C^{(2)}_e$, $C^{(3)}_e$

oxygen: $O^{(1)}$, $O^{(2)}$, $O^{(1)}_s$, $O^{(1)}_e$

hydrogen: $H^{(1)}$, $H^{(1)}_2$, $H^{(1)}_3$

nitrogen: $N^{(1)}$, $N^{(2)}$, $N^{(3)}$, $N^{(1)}_s$, $N^{(2)}_s$, $N^{(1)}_e$, $N^{(2)}_e$

sulphur: $S^{(1)}$, $S^{(2)}$, $S^{(1)}_s$, $S^{(1)}_e$

halogens: $F^{(1)}$, $F^{(1)}_2$, $F^{(1)}_3$, $Cl^{(1)}$, $Cl^{(1)}_2$, $Cl^{(1)}_3$, $Br^{(1)}$, $Br^{(1)}_2$, $Br^{(1)}_3$, $I^{(1)}$, $I^{(1)}_2$, $I^{(1)}_3$

Arabic numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

a symbol denoting bridges: b

symbols denoting respectively begin and end of branch: (,)

empty symbol: ε

Expansions of this basic alphabet can be considered, for example, to include metal organics, radicals, a charge of atoms in the molecule and even stereochemistry. The presented alphabet makes a foundation for modifications leading to describe more complex organic compounds. It must be mentioned that the nature of chemical bonding is so far not well understood, so it is better to denote bonding by maximal order, not by hybridization.

The string is started by any C atom, then it's ligands are denoted in the order of decreasing atomic number, then goes another C atom and so on. Oxygen, sulfur, and nitrogen atoms are threatened in the same way as carbon atoms. Branches in organic compounds are denoted by (i…….)i where i is an integer describing the number of the branch in the molecule. If there is only one branch in compound i is omitted for simplicity. Notation of rings is following: begin of the ring must be set (f. e. $C^{(2)}{}_{s}$), then goes every atom in the ring, and then end of the ring must be denoted (f. e. $C^{(2)}{}_{e}$). If the molecule has multiple rings, the described procedure is applied for the most outer ring and remaining chains are threatened as branches and bridges (see example f below). Strings of some organic compounds presented in Figure 1 are as follows:

a) $C^{(1)}\ H^{(1)}{}_{3}\ C^{(2)}\ O^{(2)}\ O^{(1)}\ H^{(1)}$

b) $C^{(1)}\ H^{(1)}{}_{3}\ C^{(2)}\ O^{(2)}\ O^{(1)}\ C^{(1)}\ H^{(1)}{}_{3}$

c) $C^{(2)}{}_{s}\ H^{(1)}\ C^{(2)}\ H^{(1)}\ C^{(2)}\ H^{(1)}\ C^{(2)}\ H^{(1)}\ C^{(2)}\ H^{(1)}\ C^{(2)}{}_{e}\ H^{(1)}$

d) $C^{(1)}\ H^{(1)}{}_{3}\ C^{(2)}\ O^{(2)}\ O^{(1)}\ C^{(1)}{}_{s}\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}{}_{e}\ H^{(1)}{}_{2}$

e) $C^{(2)}\ O^{(2)}\ O^{(1)}\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(2)}{}_{s}\ C^{(2)}\ (1\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(2)}\ O^{(2)}\ O^{(1)}\ H^{(1)})1\ C^{(2)}\ H^{(1)}\ N^{(1)}\ H^{(1)}\ C^{(2)}{}_{e}\ (2\ C^{(1)}\ H^{(1)}{}_{2}\ N^{(1)}\ H^{(1)}{}_{2})2$

f) $C^{(1)}{}_{s}\ H^{(1)}{}_{2}\ C^{(1)}b1\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}b2\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}b2\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}b3\ H^{(1)}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}\ H^{(1)}\ (C^{(1)}b1\ H^{(1)}\ C^{(1)}b3\ H^{(1)}\ C^{(1)}b2\ H^{(1)})\ C^{(1)}\ H^{(1)}{}_{2}\ C^{(1)}{}_{e}\ H^{(1)}{}_{2}$

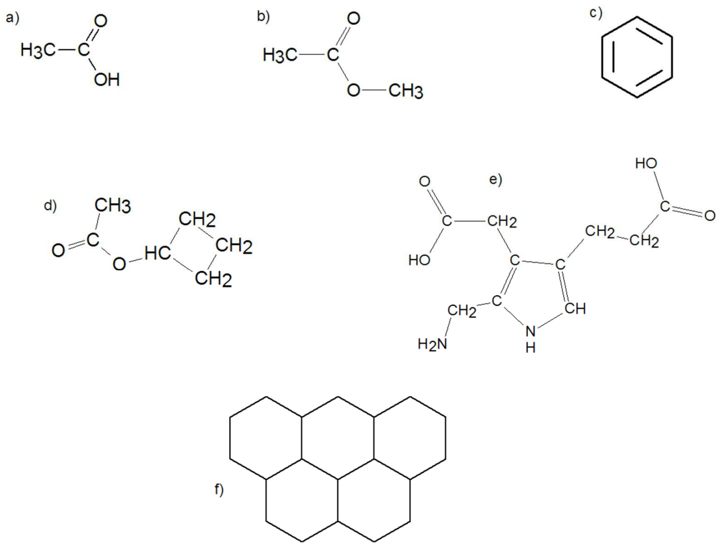

Figure 1. Examples of organic compounds for which strings are given.

The presented notation allows generating strings for a given structure of the molecule. It is obvious that one molecule can be written as many strings, depending on which carbon atom will be selected for start. However, this system of notation is unambiguous and two different molecules must have different strings. It is because two compounds would have same strings only if they would have the same summary formula, but presented system of notation deals with structural isomers by denoting branches – two compounds with different structure must have different strings. Also, stereochemistry can be easily included by adding two more symbols to the alphabet, S and R, to denote the absolute configuration of an atom. Note that strings generated by presented in this paper notation system can be utilized in the same way as SMILES strings, but are as well more chemically intuitive than SMILES strings.

3 Results

3.1 Finite-state automata for example reactions

System of notation allows to write certain organic molecule as a string, so it can be proceeded by finite-state automata. FSA can be utilized in the planning of organic synthesis. Let's consider the following problem: one has to plan synthesis of certain organic compound characterized by promising biological activity. However, this compound has a very complicated structure. It would be very useful to have computer algorithms which would say "Use certain reaction". Thanks to the presented system of notation it is possible to construct finite-state automaton that decides whether the organic compound can be synthesized with certain organic reaction. Let's consider simply esterification reaction. Figure 2 presents finite-state automaton that accepts only compounds that can be obtained in that reaction. FSA reads the input string (organic compound) symbol by symbol and moves from state to state in a way described by transition function δ. If FSA ends up on accepting state (q_1 in Figure 2) it means that this organic compound can be obtained in the esterification reaction.

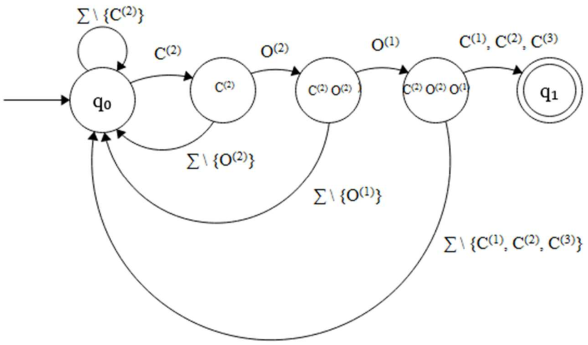

Figure 2. Finite-state automaton accepting esters.

The esterification reaction is a somewhat trivial example. The more complex and synthetically useful reaction is now considered: the Radziszewski reaction. The Radziszewski reaction is a multicomponent universal reaction of any dicarbonyl, any aldehyde, and ammonia leading to the creation of imidazoles, which are compounds with varying biological properties (Figure 3) [25]. Furthermore, it is a very efficient method that is investigated in continuous microreactors and is still used recently [26 – 28]. Also, many expansions and modifications of this reaction have been discovered [26, 29]. General scheme of the classical Radziszewski reaction and corresponding finite-state automaton are presented in Figure 3 and 4 respectively.

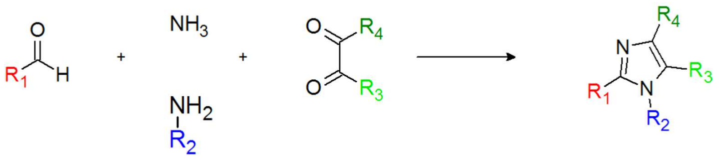

Figure 3. General scheme of the classical Radziszewski reaction.

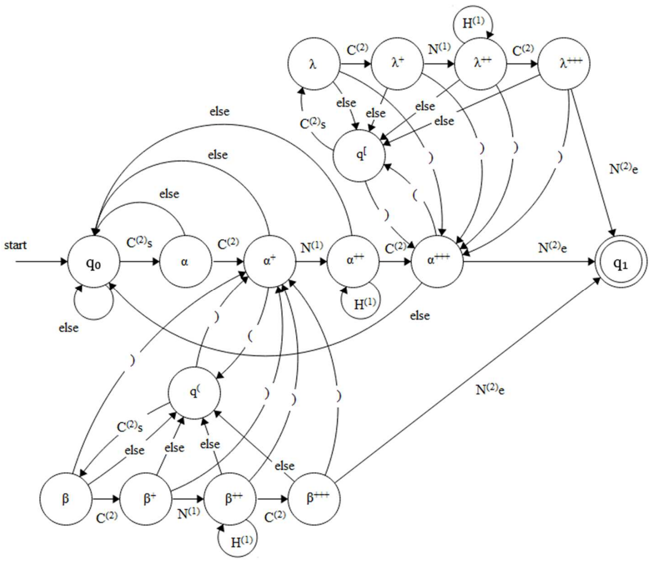

Figure 4. Finite-state automaton that accepts compounds which can be obtained with the Radziszewski reaction.

Finite-state automaton for the Radziszewski reaction is much more complicated in comparison with the case of the esterification reaction. While constructing this automaton it was assumed that R^2 in Figure 3 is hydrogen atom for simplicity and that the ring is entered through R^4. Furthermore, it was assumed that there is only one branch in the whole molecule, so ‘i' after ‘(‘ and ‘)' is omitted. The assumption that there are 2 branches and that they can occur as well as R^1, R^2 and R^3 (in other words that sp^3 nitrogen atom does not have to bind with hydrogen) leads to much more complicated automaton. For exercise, one can try to construct that automaton and also universal finite-state automaton for the Radziszewski reaction. It can be easily proved that universal FSA for this reaction does not exist.

4 Discussion

A fact from automata theory is that every finite-state automaton accept words from regular languages. Finite-state automaton that would accept words from at least context-free language (in other words: that string would be produced by at least context-free grammar) does not exist.

Lemma 4.1. *(the Pumping Lemma)*:

If language L is regular then exists constant n_L such that for every string $z \in L$ following statement is true:

$$|z| \geq n_L \Rightarrow (\exists\ u, v, w \in \sum^*, z = uvw, |uv| \leq n_L, |v| \geq 1)\ (\forall\ i = 0, 1, 2, 3, \ldots)\ z_i = uv^iw \in L$$

where: $\sum^*$ - set of all words over alphabet $\sum$, v^i – v repeated i times, $|z|$ - length of word z.

Theorem 4.2. *The contrapositive of the Pumping Lemma*:

If for any constant n exists word $z \in L$ such that:

$$|z| \geq n \text{ and } (\forall\ u, v, w \in \sum^*, uvw = z, |uv| \leq n, |v| \geq 1)\ (\exists\ i \in \{0, 1, 2, 3, \ldots\})\ z_i = uv^iw \neg\in L$$

then language L is not regular.

With the help of contrapositive of the Pumping Lemma, it is easy to prove that the language of all organic compounds over the alphabet proposed in this article, in general, is not regular. Let's take a word (string) from organic compounds language that has the symbol ‘(‘ or ‘)' inside. Now let's take v = (. Then $|v| = 1$, which meets the requirements of Theorem 4.2. It's also obvious that $v \in \sum^*$ and that constant n can be chosen so $|uv| \leq n$. Iterating of v will lead to a string that does not describe any real organic compound, in other words there exists $i \in \{0, 1, 2, 3 \ldots\}$ such that $z_i = uv^iw$ does not belong to the language of organic compounds. Thus this language is not regular. This is the reason why universal finite-state automaton for the Radziszewski reaction can’t be constructed. Parentheses ‘(‘ or ‘)’ are necessary to denote branches (R^1, R^2, R^3 and R^4 in Figure 3) which number generally is not known – in branches can exist other branches and so on.

However finite-state automaton has been constructed for simply esterification reaction. It means that the language of esters is regular. In fact, if the presence of a linear group (such as ester group –COO-) in an organic molecule is crucial for recognition by automaton then finite-state automaton can be constructed. Less complicated reactions are equivalent to regular languages and more complicated reactions are equivalent to non-regular languages. This statement shows some structure of organic chemistry, as regular languages are a subset of context-free languages. Nevertheless, it is still possible to construct finite-state automaton for the Radziszewski reaction, but some assumptions are necessary (Figure 4). In particular, knowledge about numbers of branches is needed.

Finite-state automata can be utilized in computational organic synthesis. FSA can be constructed for every reaction, at last with the help of some assumptions. Knowing that FSA outperforms recurrent neural networks it is possible to create a very powerful and useful computational tool not only for organic synthesis planning but also even for drug design. Applications of automata in this subject should be explored.

5 Conclusions

Automata theory is widely used not only in computer science but also in other sciences such as biology or chemistry. This article shows a new approach in applications of automata theory in organic chemistry. Linguistics basics for organic compounds language are proposed that allows the construction of finite-state automata for either simple and complex organic reactions. Furthermore, the algebraic structure of organic chemistry has been shown, as more complex organic reactions are equivalent to more complicated languages. This conclusion points out the potential application of automata theory in complex systems physics, as many of the problems in chemistry belongs to that discipline. In this term connection between artificial neural networks, which are equivalent to finite-state automata, and nonlinear dynamics should be mentioned. Further exploration of grammars of organic compounds would lead to some constructive and interesting conclusions.

Chemical informatics requires efficient computational algorithms. Thus finite-state automata utilized in problems of organic chemistry offers a promise of fast and efficient computational tools for drug design and synthesis planning. It would be interesting to compare the performance of an artificial neural network and finite-state automaton in terms of recognizing compounds that could be obtained in a certain synthetically important organic reaction.

References

[1] J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Pearson, Boston, 2007.

[2] I. C. Baianu, Computer models and automata theory in biology and medicine, *Math. Model.* **7** (1986) 1513–1577.

[3] A. Egri-Nagy, C. L. Nehaniv, J. L. Rhodes, M. J. Schilstra, Automatic analysis of computation in biochemical reactions, *Biosystems* **94** (2008) 126–134.

[4] Z. G. Wang, J. Elbaz, F. Remade, R. D. Levine, I. Willner, All-DNA finite-state automata with finite memory, *Proc. Nat. Acad. Sci. USA* **107** (2010) 21996–22001.

[5] P. Hong, M. Turk, T. S. Huang, Gesture modeling and recognition using finite state machines, in: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Computer Society, Los Alamitos, 2000, pp. 410–415.

[6] K. Oishi, E. Klavins, Framework for engineering finite state machines in gene regulatory networks, *ACS Synth. Biol.* **3** (2014) 652–665.

[7] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, Y. C. Lee, Learning and extracting finite state automata with second-order recurrent neural networks, *Neural Comput.* **4** (1992) 393–405.

[8] P. Manolios, R. Fanelli, First-order recurrent neural networks and deterministic finite state automata, *Neural Comput.* **6** (1994) 1155–1173.

[9] M. Casey, The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction, *Neural Comput.* **8** (1996) 1135–1178.

[10] C. W. Omlin, C. L. Giles, Constructing deterministic finite-state automata in recurrent neural networks, *J. ACM* **43** (1996) 937–972.

[11] N. Ganesh, N. G. Anderson, Irreversibility and dissipation in finite-state automata, *Phys. Lett. A* **377** (2013) 3266–3271.

[12] M. Kadukova, S. Grudinin, Knodle, a support vector machines-based automatic perce-ption of organic molecules from 3D coordinates, *J. Chem. Inf. Model.* **56** (2016) 1410–1419.

[13] M. Tsuchiya, J. Ross, Application of genetic algorithm to chemical kinetics: systematic determination of reaction mechanism and rate coefficients for a complex reaction network, *J. Phys. Chem. A* **105** (2001) 4052–4058.

[14] D. Fooshee, A. Mood, E. Gutman, A. Tavakoli, G. Urbam, F. Liu, N. Huynh, D. Van Vranken, P. Baldi, Deep learning for chemical reaction prediction, *Mol. Syst. Des. Eng.* **3** (2018) 442–452.

[15] W. Duch, K. Swaminathan, J. Meller, Artificial intelligence approaches for rational drug design and discovery, *Curr. Pharm. Des.* **13** (2007) 1497–1508.

[16] O. Engkvist, P. O. Norrby, N. Selmi, Y. H. Lam, Z. Peng, E. C. Sherel, W. Amberg, T. Erhard, L. A. Smyth, Computational prediction of chemical reactions: current status and outlook, *Drug Discov. Today* **23** (2018) 1203–1218.

[17] C. J. Bartel, S. L. Millican, A. M. Deml, J. R. Rumptz, W. Tumas, A. W. Weimer, S. Lany, V. Stevanović, C. B. Musgrave, A. M. Holder, Physical descriptor for the Gibbs energy of inorganic crystalline solids and temperature-dependent materials chemistry, *Nat. Commun.* **9** (2018) 4168.

[18] D. Jha, L. Ward, A. Paul, W. Liao, A. Choudhary, C. Wolverton, A. Agrawal, ElemNet: deep learning the chemistry of materials from only elemental composition, *Sci. Rep.* **8** (2018) #17593.

[19] L. Cao, C. Li, T. Mueller, The use of cluster expansions to predict the structures and properties of surfaces and nanostructured materials, *J. Chem. Inf. Model.* **58** (2018) 2401–2413.

[20] J. A. Grant, J. A. Haigh, B. T. Pickup, A. Nicholls, R. A Sayle., Lingos, finite state machines, and fast similarity searching, *J. Chem. Inf. Model.* **46** (2006) 1912–1918.

[21] R. Sayle, P. H. Xie, S. Muresan, Improved chemical text mining of patents with infinite dictionaries and automatic spelling correction, *J. Chem. Inf. Model.* **52** (2012) 51–62.

[22] F. Okubo, T. Yokomori, The computational capability of chemical reaction automata, *Nat. Comput.* **15** (2016) 215–224.

[23] A. Hjelmfelt, E. D. Weinberger, J. Ross, Chemical implementation of neural networks and Turing machines, *Proc. Nat. Acad. Sci. USA* **88** (1991) 10983–10987.

[24] A. Hjelmfelt, E. D. Weinberger, J. Ross, Chemical implementation of finite-state machines, *Proc. Nat. Acad. Sci. USA* **89** (1992) 383–387.

[25] J. A. Hernandez Munoz, J. J. Junior, F. Martins da Silva, Radziszewski reaction: an elegant, easy, simple and efficient method to synthesise imidazoles, *Curr. Org. Synth.* **11** (2014) 824–834.

[26] D. R. J. Acke, R. V. A. Orru, C. V. Stevens, Continuous synthesis of tri- and tetrasubstituted imidazoles via a multicomponent reaction under microreactor conditions, *QSAR Comb. Sci.* **25** (2006) 474–483.

[27] F. Zhou, H. Liu, K. Wang, Z. Wen, G. Chen, Preparation of simple imidazoles in continuous-flow microreactor, *Huagong Xuebao (Chin. Ed.)* **69** (2018) 2481–2487.

[28] S. Kula, A. Szlapa-Kula, S. Kotowicz, M. Filapek, K. Bujak, M. Siwy, H. Janeczek, S. Maćkowski, E. Schab-Balcerzak, Phenanthro[9,10-d]imidazole with thiophene rings toward OLEDs application, *Dyes Pigm.* **159** (2018) 646–654.

[29] Z. Jingjing, G. Qinghe, W. Xia, G. Xiao, W. Yan-Dong, W. Anxin, Dual roles of methyl ketones in radziszewski-type reaction: formal [2 + 1 + 1 + 1] synthesis of 1,2,5-trisubstituted imidazoles, *Org. Lett.* **18** (2016) 1686–1689.