**UDC 004.02**
**IRSTI 20.53.15**

# EQUIVALENCE OF COMPLEXITY CLASSES VIA FINITE AUTOMATA DERIVATIVES

**Mirzakhmet Syzdykov**

al-Farabi Kazakh National University, Almaty, Kazakhstan
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

**Abstract.** In this article practical, experimental and theoretical results of the conducted research are presented, these results refer to the main question in complexity theory like the "P versus NP" theorem, first proposed by Stephen Cook in his seminal paper, we give concise question of the equivalence of these classes from the state of conversion algorithm of non-deterministic finite automata (NFA) to deterministic finite automata (DFA), we do it by considering the canonical regular expression form presented by Schneider Klaus and do the derivative processing from the Berry-Sethi algorithm, the result gives P-complete algorithm along the canonical form of regular expressions when applying subset construction for specific regular expressions or NFA, DFA in this case remains applicable even for Turing tape machines, however, due to the past statement of the "P versus NP" theorem given by author in previous work this leads to the conclusion of the equivalence of complexity classes like P and NP as in canonical form the subset construction produces exponential growth of the number of states of DFA.

**Keywords:** subset construction, regular expression, P versus NP, derivatives.

## Кіріспе

As we stated before the conducted research proves the equivalence of polynomial (P) and non-polynomial (NP) classes of complexity known as "P versus NP" theorem [1] on the sample of subset construction algorithm [2] by the derivative algorithm [3] on the canonical regular expression and NFA [4].

Previously we stated that the main theorem by Cook can be solved if for the given non-polynomial class of complexity in subset construction, which belongs to the exponential classes like EXPTIME and EXPSPACE, there exists the method to achieve it in polynomial time.

According to conducted research many scientists regard this notion as final and almost impossible to prove, however, we follow the same strategy presented before [5].

We do our statement on the regular expression experimental results [6]. These results lead to the observation that for canonical form of regular expression, leading subset construction algorithm to the exponential blow up of number of states in final DFA, there exist a practical and natural way of obtaining the rule which describes the problematic canonical form of regular expression in more practical way like polynomial or P-classes of computational complexity.

The "P versus NP" is a classical example of unsolved theorems from the past works: it states that there exist no polynomial algorithm to solve the NP-complete problem, which were classified in past work [5].

Derivatives were first presented in [7, 8] in order to describe the algebraic properties of regular languages – this was, however, overcome by Berry and Sethi who presented an algorithmic approach in construction DFA directly from regular expression.

The Knapsack problem which is known to be co-NP complete is also presented in this article as it can be solved using Dynamic Programming (DP) [9].

Subset Construction on Single Fire Automata (SFA). As we have defined before in [6] that there exist the rules for SFA to apply DeMorgan law, in inverse, we can apply this rule to the not-starred logical "OR" operator, the changes aren't made for the operator which is closed under Kleene closure.

In general, we do this for our experimental results on the one-step automata classes like SFA, the concept and design of which was first presented by the author of this work.

The subset construction on SFA gives the experimentation environment for building the DFA for canonical expression by Schneider [4], which is defined as:

$$(a + b)^*b(a + b)(a + b). \qquad (1)$$

In the next section we will present the order of simplifying the regular expression (1) using the derivative algorithm by Berry-Sethi [3].

Generally, derivatives were also studied by Janusz Brzozowski [7] and Valentin Antimirov [8], however, definitive algorithm is due to Gerard Berry and Ravi Sethi.

As derivatives were first proposed by above authors, they still remain the point of interest of the modern research according to the Berry-Sethi algorithm.

We use this approach for our canonical form to prove the existence of polynomial algorithms to solve NP-complete problem.

Schneider proves that canonical regular expression (1) for subset construction is NP-complete, however, our research shows that it can be solved by extending the automata with feasible rule set which can be applied to the parameters like input and the canonical form.

The number of post OR-operators after marked symbol in (1) is defined as a parameter t in this work.

Schneider proves [4] that the algorithm to convert NFA to DFA has exponential complexity of O (2t+1).

Thus, the canonical form (1) lies in EXPSPACE and EXPTIME class which is more that class NP. By proving the P-completeness of the overall question we can make a decision that P-complete rule for NP-complete problem by Schneider solves the question as NP-class lies inside EXPTIME-class of the computational complexity.

## Application of Derivatives to the Canonical Form

The regular expression (1) can be approximated by the doubled or invariant derivatives, the main idea is based on the past research [3].

Thus, we get the following rules to be applied to the canonical exponential form:

$$[Canonical\ Form]\ da\ db. \qquad (2)$$

We apply the rule (2) until we reach the mark in (1) defined by single arbitrary symbol on the binary alphabet.

Thus, we get the result which leads to the equation of the mark to be matched in any case, meanwhile the pre- and post-expressions of canonical form (1) represent the fully defined set over the closed alphabet under binary notation.

Thus, the resulting rule will be defined as:

$$I\ [n - t] = \text{"b"}, \qquad (3)$$

where I is an input string, n is a current position on given input and t is the number of post-repetitions of the fully defined OR-operator on the canonical form (1).

Thus, we have defined the final rule which is applicable even for Turing tape machines and other type of automata due to the presence of the mark symbol in the canonical regular expression (1).

## Experimental Results on Extended Regular Expressions

As it follows from the result in previous section, we have also conducted the research on the SFA with OR-rewriting rule as it was presented in [6] with respect to the Kleene closure under which the OR-remains not re-written.

DeMorgan's law is extended in this article for OR-operators as well as this leads to the composition of SFA in general, giving the possibility to the states to be fired once at a time as in this case we can avoid the state explosion effect or, simply, "blow up" which leads to the steady exponential increase of the number of states for the canonical form – in the original paper [4] t = 2, however, the author makes proof towards any value of the free parameter t.

Bellman's Dynamic Programming [9] doesn't approximate to the P-class too as it was shown in another publication of the author of this article. This fact is due to the recurrence relation in the common case of Bellman's equation as the number of steps required to solve NP-complete problem grows faster than the recurrent function and, thus, the problem cannot be solved in the minimal polynomial time. We have seen this effect before when classical Knapsack problem cannot be solved for the arbitrary values of weights in the given input data.

The experimentation has also showed that SFA are more compact for subset construction rather than Berry-Sethi algorithm which produces the same exponential number of states for canonical regular expression.

We obtain the following resulting diagram of DFA obtained from SFA on the Figure 1.

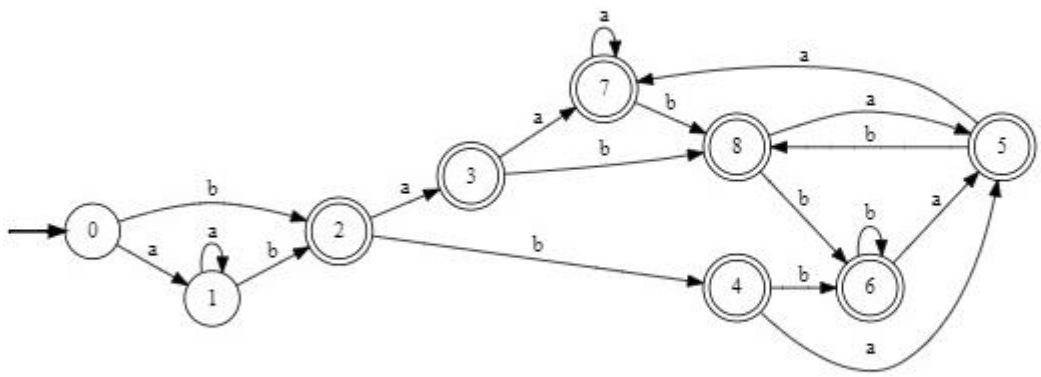


Figure 1 – DFA from SFA of the canonical regular expression

After minimization the obtained DFA is defined as:

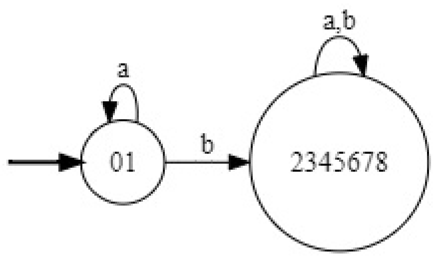


Figure 2 – Sample DFA after minimization of number of states

Obviously, the DFA on Figure 2 can be re-written in the regular expression form as:

$$a^*b(a + b)^*. \quad (4)$$

The experimentation was made on the "Regex+" software package that demonstrates high feasibility not only for extended regular expressions as well as for a canonical form – the result from above figures (1) and (2) is the result obtained by the program.

On the Figure 2 the last state is final accepting the language (4).

## Conclusion

We have demonstrated the steps required to simplify the canonical regular expression which was proved to be NP-complete along the subset construction.

Our research gives the result of the existence of the Fixed Input Automata (FIA), which can describe Schneider's canonical form.

The complexity of the presented sample FIA is O(n), where n is the number of input string to be matched against canonical regular expression – which, in case, is polynomial and belongs the P-class of complexity.

Thus, P equals NP according to the derivative application for simplification of the canonical form.

We make the final step in the discussion of the "P versus NP" theorem as the modified subset construction leads to the experimental evaluation of the canonical rule for the regular expression which was presented in this article.

By the term "proof" we mean the statement which cannot be argued due to the recent results on the question of relation between complexity classes, as NP is less than EXPTIME and the solving algorithm solves the problem in the EXPTIME, we conclude that P and NP classes are equal due to the fact that NP-class is bigger than P in order of magnitude as it was previously classified.

Thus, the classes of computational complexity like P and NP are equal due to the existence of the derivative solution for NP-complete problem of getting finite automaton for the canonical regular expression.

As we have defined the final point of the "P versus NP" theorem by giving not arguable argument towards the case when they're equal, we can conclude that algorithmic part of this question was studied less and rarely to give the algorithm which can solve the NP-complete problem in polynomial time.

We have reached the final result of the theorem and now it's time to solve other NP-complete problems by the experience of the research presented in this article.

## Acknowledgements

The author expresses gratitude to the ResearchGate™ community for collaboration in presenting this work as well as for other scientists who contributed to the "P versus NP" theorem.

## Funding

The work was fully supported by an educational grant of the Ministry of Education and Sciences of Republic of Kazakhstan during author studying at Satbayev University and during studying at the Institute of Problems in Informatics and Control.

## References

[1] Cook, Stephen. The importance of the P versus NP question. Journal of the ACM (JACM) 50.1. 2003. 27-29.

[2] Rabin, Michael O., Dana Scott. Finite automata and their decision problems. IBM journal of research and development 3.2. 1959. 114-125.

[3] Berry, Gerard, Ravi Sethi. From regular expressions to deterministic automata. Theoretical computer science. 1986. 48. 117-126.

[4] Schneider, Klaus, Jimmy Shabolt, John G. Taylor. Verification of reactive systems: formal methods and algorithms. Heidelberg: Springer, 2004.

[5] Syzdykov, Mirzakhmet. Functional hypothesis of complexity classes. advanced technologies and computer science. 2022. 3. 4-9.

[6] Syzdykov, Mirzakhmet. Membership Problem in Non-deterministic Finite Automata for Extended Regular Expressions in Linear Polynomial Time. Advanced technologies and computer science. 2021. 4. 14-17.

[7] Brzozowski, Janusz A. Derivatives of regular expressions. Journal of the ACM (JACM). 1964. 11.4. 481-494.

[8] Antimirov, Valentin. Partial derivatives of regular expressions and finite automaton constructions. Theoretical Computer Science. 1996. 155.2. 291-319.

[9] Bellman, Richard. The theory of dynamic programming. Bulletin of the American Mathematical Society. 1954. 60.6. 503-515.

## АҚЫРЛЫ АВТОМАТТАРДЫҢ ТУЫНДЫЛАРЫ АРҚЫЛЫ КҮРДЕЛІЛІК КЛАСТАРЫНЫҢ ЭКВИВАЛЕНТТІЛІГІ

Мырзахмет Сыздықов
әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

**Андатпа.** Бұл мақалада жүргізілген зерттеудің практикалық, эксперименттік және теориялық нәтижелері келтірілген. Бұл тұжырымдар күрделілік теориясындағы негізгі сұраққа сілтеме жасайды. Мысалы, Стивен Кук өзінің негізгі мақаласында алғаш рет ұсынған "P және NP" теоремасы, біз осы сыныптардың эквиваленттілігі туралы қысқаша сұрақ қоямыз. түрлендіру күйі анықталмаған ақырлы автоматтар алгоритмі (NFA) үшін детерминирленген ақырлы автоматтар (DFA), біз мұны Шнайдер Клаус ұсынған тұрақты өрнектің канондық формасын қарастыра отырып жасаймыз және Берри-Сети алгоритмінен алынған өңдеуді орындаймыз, нәтиже p-complete алгоритмін тұрақты өрнектердің канондық формасы бойынша белгілі бір тұрақты өрнектерге немесе NFA-ға ішкі жиынның құрылысын қолданған кезде береді, бұл жағдайда DFA тіпті Тьюринг таспа машиналарына да қатысты болып қалады, алайда автор алдыңғы жұмыста келтірген "P vs NP" теоремасының бұрынғы тұжырымына байланысты бұл эквиваленттілік туралы қорытындыға әкеледі. P және NP сияқты күрделілік кластары үшін канондық түрде Ішкі жиынды құру DFA күйлерінің санының экспоненциалды өсуіне әкеледі.

**Кілттік сөздер:** Ішкі жиынды құру, тұрақты өрнек, NP-мен салыстырғанда P, туындылар.

## ЭКВИВАЛЕНТНОСТЬ КЛАССОВ СЛОЖНОСТИ ЧЕРЕЗ ПРОИЗВОДНЫЕ КОНЕЧНЫХ АВТОМАТОВ

Мырзахмет Сыздыков
Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

**Аннотация.** В этой статье представлены практические, экспериментальные и теоретические результаты проведенного исследования, эти результаты относятся к основному вопросу в теории сложности, такому как теорема “P против NP”, впервые предложенная Стивеном Куком в его основополагающей статье, мы кратко задаем вопрос об эквивалентности этих классов из состояния преобразования алгоритм недетерминированных конечных автоматов (NFA) для детерминированных конечных автоматов (DFA), рассматривая каноническую форму регулярного выражения, представленную Шнайдером Клаусом, и выполняем производную обработку из алгоритма Берри-Сетхи, результат дает алгоритм P-complete вдоль канонической формы регулярных выражений при применении построения подмножества для конкретных регулярных выражений или NFA, DFA в этом случае остается применимым даже для ленточных машин Тьюринга, однако, из-за прошлой формулировки теоремы “P против NP”, приведенной автором в предыдущей работе, это приводит к выводу об эквивалентности классов сложности, таких как P и NP, поскольку в канонической форме построение подмножества приводит к экспоненциальному росту числа состояний DFA.

**Ключевые слова:** построение подмножества, регулярное выражение, P в сравнении с NP, производные.

*Сведения об авторе:*
*Анг.: Syzdykov Mirzakhmet - al-Farabi Kazakh National University, Almaty, Kazakhstan*
*Каз.: Сыздықов Мырзахмет- әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан.*
*Рус.: Сыздыков Мырзахмет- Казахский национальный университет имени аль-Фараби, Алматы, Казахстан.*