UDC 004.02
IRSTI 20.53.15

ON SUCCESSFUL P VERSUS NP WITHIN FINITE AUTOMATA AND REGULAR EXPRESSIONS

Mirzakhmet Syzdykov
Satbayev University, Almaty, Kazakhstan
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Abstract In this article the final proof of the equivalence of polynomial (P) and non-polynomial classes (NP) are provided within the given Klaus Schneider's regular expression forms which tend to be exponential in size and the problem in overall is seen to be in EXPTIME and EXPSPACE and, thus, NP-complete. We provide the full history of this proof with the experimentally obtained results. Our application of the obtained singular algorithm in terms of computational complexity responds to the famous theorem like "P versus NP" which wasn't solved before and now is resolved in this work. We will also discuss the future of the question from past as the novel proof is given.

Keywords: P versus NP, regular expressions, finite automata, proof.

Introduction

At first, we will give the definition of the problem for arguments like regular expressions [1] and finite automata which are constructed from these expressions using various types of algorithms like Thompson Construction [2], Berry-Sethi Method [3] and Rabin-Scott subset construction [4].

Regular expressions represent the modern tool of fast parsing and evaluation of textual data [5]. In this work they're defined as follows with respect to the L(r) function, which defines the set words in regular language:

$L(\varepsilon) = \{ \}$ – empty language;

$L(a) = \{ a \mid a \text{ in } A \}$, where A is an alphabet of the regular language;

$L(r_1 \mid r_2) = L(r_1) + L(r_2)$, union of two languages;

$L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$, concatenation of two language;

$L(r^*) = L(r)^*$, Kleene closure or star operator which defines infinite number of repeats.

In addition to the basic set which was standardized by POSIX and is supported by many regular expression flavors, we define the additional set of operators like intersection subtraction and complement [6].

Thompson's algorithm gives the broad evidence of the existence of non-deterministic finite automata (NFA) which imply the recursive construction for each type of operator. This algorithm is known to be practical in sense of the difficulty of implementation as it linearly depends on the size of the input, which is given by regular expression.

Berry-Sethi algorithm is a way of representing the derivative of the regular expression and construct DFA directly by applying necessary set of rules. There is an implementation based upon Abstract Syntax Trees (AST) [7]. In the original paper [3] the algorithm also supports extended operators.

Rabin and Scott present the subset or powerset construction to convert existing NFA to DFA [4]. This problem is NP-complete as it lies in EXPTIME and EXPSPACE class on the example given by Schneider Klaus [8].

We will further show the equivalence of contrary classes like P and NP due to the complexity – the problem by itself is stated in [9].

Schneider's canonical forms

As it was stated before in [8] there's a definition of the subset of regular expressions which lead to the effect known as state explosion when the number of states grows exponentially. We parametrize these expressions using t-constant which defines the number of repeating subgroups in Schneider's expressions which lead to the complexity $O(2^{t+1})$ over binary alphabet:

$(a \mid b)^* \cdot b \cdot (a \mid b) \cdot (a \mid b) \ldots (a \mid b) = (a \mid b)^* \cdot (a \mid b)^t$.

As the size of the DFA in subset construction for the above expression grows exponentially, it's obvious that this problem is NP-hard.

In this section we give the Turing tape automaton which depends on the parameter t and function f(r, t), this function is defined as follows:

f(r, t) = { r [|r| - t] = "b" }, where r is the regular expression input and matching string and t is a free parameter.

As we can see from the above function definition, it follows that the problem is solved in time and space O (1), rather than the proof by Schneider [8] that the overall task is exponential and thus empirically is impractical as many other NP-complete problems [10].

Pre-history of the obtained results

As we have proved the "P versus NP" theorem that P and NP classes are equivalent due to the problem which can be reduced to the NP-complete and the fact that there's minimal polynomial solution to this problem.

We have gone through the experimentation time which is described in [11, 12]. There we go through the definition of polynomial and non-polynomial methods of evaluation in concordance to the algorithm design and structure [11] for both classes. In [12] the rigorous proof is defined; however, it still wasn't shown that there exists the reducible function as it's defined in the previous section.

Present time fuzzy methods

The Ant Colony Optimization or, simply, ACO [13] is a well-defined heuristics method which is most known for the present time in order to solve NP-complete problems like Vertex Cover Problem (VCP) [14] or Travelling Salesman Problem (TSP).

Probably soon the quantum or alternative computing models will show more convenient methodology towards solving NP-complete problems for at least fuzzy threshold, meanwhile, ACO still remains very popular and easy to implement as its complexity converges to cubic with respect to the size of the input data.

The quantum computing still remains a modern trend in solving the algorithmic tasks, it's shown that with given amount of energy NP-complete problems can be solved in linear time [15].

Discussion on "P versus NP"

As to the Karp's 21 NP-complete problems [16], we have developed the stable polynomial solutions to the some of them which we met practically: for example, TSP which can be approximately using ant colony optimization is of graph theory and deals with shortest paths, which, in turn, can be vital in GIS-systems.

According to our recommendation the best way to represent the function in equation following in this section for solving the NP-*hard* problem (NP-complete) is the usage of already developed method by Richard Bellman presented in his famous work in [17]: this method is called dynamic programming, or simply DP. We will use DP in solving our NP-complete problems.

To this moment, the solutions presented by author in this work which are known to be NP-complete are as follows:

1. TSP uses weighted graph structure to represent the routes between pairs of cities on the map; the goal is to find the shortest path visiting all the cities around the built path.
2. Back-reference problem: the problem can be represented as a satisfiability or vertex color problem which are known to be NP-complete [18]; the problem is to find the referencing string of the previously captured group in the pattern: this back-references feature is often met in the modern programming languages like Perl, Ruby, Python, etc. [19]

3. Vertex Cover Problem (VCP): the problem is to cover all edges in graph using minimum or pre-defined number of vertexes; due to the [14] it's known to be NP-complete within the almost quadratic complexity measure.

Even though the "P versus NP" theorem was presented by Stephen Cook in his work [9] and was also included as one of the Millennium Theorems by the Clay Mathematics Institute [20], we have proved the equivalence of P- and NP-classes. The main question arises of how to devise the function F(x): to answer this question we will use Dynamic Programming as the main standard in solving the above NP-complete problems like TSP, Back-reference and VCP.

The solution of TSP is exact and can be found in the following dynamic recurrence relation:

$$F(x, n) = \min\{F(y, n-1) + d(x, y)\}.$$

Where in the above equation the *x* and *y* are variables for cities in the graph and *d(x, y)* is the distance function; *n* is the number of cities passed before as well.

This equation holds true for specific cases when we have a non-full graph: full graphs are usually represented by matrix.

The next problem is about back-referencing in regular expressions: we solve it by applying the same dynamic programming paradigm (DPP) – it can be noted that according to OLAP data cube construction in Business Intelligence systems, the same holds true within the dimension of incoming string with the position of the searched symbol during the matching process.

Thus, the following relation holds true:

$$F(backreference, position) = F(backreference, position + 1) \cdot string[position]$$

In the VCP we build the "concordance"-network of the vertexes in graph *G (V, E)* (*V* is the set of vertexes and *E* is a set of edges) within the cardinality between the pairs, or set in general, when there are common edges.

The function *F(x)* for the two vertexes, thus, can be represented as follows due to the set intersection theory:

$$F(u, v) = \deg(v) + \deg(u) - \deg(u, v),$$

Where in this equation the function deg(v) is the cardinality of the vertex in graph, which is usually represented by the set of adjacent edges.

Strict proof of P = NP

The strict proof relies on the fact that there's a unique transformation [12] of algorithm to be EXPTIME-complete which runs in minimal possible time O(1). This fact gives us the observation that any NP-complete problem can have a deterministic and minimal complexity solution on automata with marks. The algorithm self is defined in [11]. Thus, from all the transformations we choose one which has the minimal complexity of the source algorithm.

Acknowledgements

We respect the researchers from ResearchGate™ community who contributed to this important proof in the area of Computer Science and Applied Mathematics as to know that the problem can be solved efficiently means to solve it exactly and without any obstacles.

Conclusion

We have shown that there exists the solution in singular complexity O(1) for exponential or even NP-hard and NP-complete problem as it was presented by Klaus Schneider.

The notable fact is that this complexity is minimal possible and, thus, gives us the possibility not only to conclude that P equals NP, but also a knowledge of having different computational models to solve NP-complete problems.

As we have showed the equivalence of the minimal and maximal classes of complexity, we still have to make both ends meet and propose the future of research in order to obtain efficient and polynomial algorithms to the NP-hard problems.

References

[1] Karttunen L., Chanod, J. P., Grefenstette, G., & Schille, A. Regular expressions for language engineering. Natural Language Engineering. 1996. 2(4). 305-328.

[2] Thompson K. Programming techniques: Regular expression search algorithm. Communications of the ACM. 1968. 11(6). 419-422

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

[4] Rabin M. O., & Scott, D. Finite automata and their decision problems. IBM journal of research and development. 1959. 3(2). 114-125.

[5] Friedl J. E. Mastering regular expressions. " O'Reilly Media, Inc.". 2006.

[6] Syzdykov M. Deterministic automata for extended regular expressions. Open Computer Science. 2017. 7(1). 24-28.

[7] Borsotti A., Breveglieri L., Crespi Reghizzi, S., Morzenti, A. A deterministic parsing algorithm for ambiguous regular expressions. Acta Informatica. 2021. 58. 195-229.

[8] Schneider, K., Shabolt, J., & Taylor, J. G. Verification of reactive systems: formal methods and algorithms. Heidelberg: Springer. 2004. 210–212.

[9] Cook, S. The P versus NP problem. Clay Mathematics Institute. 2000. 2.

[10] Filar J. A., Haythorpe M., Taylor R. Linearly-growing reductions of Karp's 21 NP-complete problems. arXiv preprint arXiv:1902.10349. 2019.

[11] Syzdykov M. Functional hypothesis of complexity classes. Advanced technologies and computer science. 2021. 3. 4-9.

[12] Syzdykov M. Equivalence of Complexity Classes via Finite Automata Derivatives. Advanced technologies and computer science. 2022. 1(4). 9-14.

[13] Dorigo M., Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on evolutionary computation. 1997. 1(1). 53-66.

[14] Dinur I., Safra, S. On the hardness of approximating minimum vertex cover. Annals of mathematics. 2005. 439-485.

[15] Černý V. Quantum computers and intractable (NP-complete) computing problems. Physical Review A. 1993. 48(1). 116.

[16] Johnson D. S. The NP-completeness column: an ongoing guide. Journal of algorithms. 1985. 6(3). 434-451.

[17] Bellman R. Dynamic programming treatment of the travelling salesman problem. Journal of the ACM (JACM). 1962. 9(1). 61-63.

[18] Dominus M. J. Perl regular expression matching is np-hard. The Perl Journal. 2001.

[19] Stubblebine T. Regular Expression Pocket Reference: Regular Expressions for Perl, Ruby, PHP, Python, C, Java and. NET. " O'Reilly Media, Inc.". 2007.

[20] Clay Mathematics Institute. Millennium Problems. http://claymath.org/millennium-problems. [Electronic recurs: 04.01.2023].

АҚЫРҒЫ АВТОМАТТАР МЕН ТҰРАҚТЫ ӨРНЕКТЕР ШЕҢБЕРІНДЕ P ЖӘНЕ NP-ДІ СӘТТІ САЛЫСТЫРУ ТУРАЛЫ

Қ. И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті, Алматы, Қазақстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Аңдатпа. Бұл мақалада көлемі бойынша экспоненциалды болатын Клаус Шнайдердің тұрақты өрнектерінің берілген формалары шеңберіндегі көпмүшелік (P) және көпмүшелік емес сыныптардың (NP) эквиваленттілігінің нақты дәлелі келтірілген және мәселе жалпы EXPTIME және EXPSPACE-те қарастырылады және осылайша NP-толық болады. Эксперименттік нәтижелермен осы дәлелдің толық сипаттамасы келтірілді. Алынған сингулярлық алгоритмді есептеу күрделілігі тұрғысынан қолдануымыз бұрын шешілмеген, бірақ қазір осы жұмыста шешілгені белгілі және "P NP-ға қарсы" типті теоремаға сәйкес келеді. Сондай-ақ жаңа дәлел келтірілгендіктен, мәселенің болашағы өткеннен талқыланады.

Кілттік сөздер: P және NP, тұрақты өрнектер, ақырлы автоматтар, дәлел.

ОБ УСПЕШНОМ СРАВНЕНИИ Р И NP В РАМКАХ КОНЕЧНЫХ АВТОМАТОВ И РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Мырзахмет Сыздыков
Казахский национальный исследовательский технический университет имени К.И. Сатпаева, Алматы, Казахстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Аннотация. В этой статье приводится конечное доказательство эквивалентности полиномиальных (P) и неполиномиальных классов (NP) в рамках приведенных форм регулярных выражений Клауса Шнайдера, которые имеют тенденцию быть экспоненциальными по размеру, и проблема в целом рассматривается в EXPTIME и EXPSPACE и, таким образом является NP-полной. Мы приводим полную историю этого доказательства с экспериментально полученными результатами. Наше применение полученного сингулярного алгоритма с точки зрения вычислительной сложности соответствует известной теореме типа “P против NP”, которая ранее не была решена, а теперь решена в этой работе. Мы также обсудим будущее вопроса из прошлого по мере того, как будет приведено новое доказательство.

Ключевые слова: P против NP, регулярные выражения, конечные автоматы, доказательство.

Сведения об авторе:
Анг.: Syzdykov Mirzakhmet - Satbayev University, Almaty, Kazakhstan
Каз.: Сыздықов Мырзахмет- Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті, Алматы, Қазақстан.
Рус.: Сыздыков Мырзахмет- Казахский национальный исследовательский техический университет имени К.И. Сатпаева, Алматы, Казахстан.