

On Equivalence of Tractable and Non-polynomial Classes of Complexity

Mirzakhmet Syzdykov*

Independent Researcher, Kazakhstan

*Corresponding Author

Mirzakhmet Syzdykov, Independent Researcher, Kazakhstan.

Submitted: 2025, Sep 10; Accepted: 2025, Oct 20; Published: 2025, Oct 28

Citation: Syzdykov, M. (2025). On Equivalence of Tractable and Non-polynomial Classes of Complexity. *J Electrical Electron Eng*, 4(5), 01-03.

Abstract

In this preamble we give the full diversification towards our methods applied to the universality of non-deterministic finite automats with respect to the question of equivalence of complexity classes like tractable, or polynomial, and non-tractable, or non-polynomial – the study goes deep into what wasn't reconsidered before according to the pattern matching within extended operators like intersection, subtraction and complement: the latter gives the full power of our automaton construction and method of validation which, in turn, leads to the "Time hierarchy theorem" collapse.

Keywords: P Versus NP, EXPTIME, P, NP, Equivalence, Proof

1. Introduction

The long-standing question and theorem is due to Stephen Cook which states if the solution for non-polynomial problem can be verified and solved in both polynomial times, the past research also showed that P-class problem cannot be within the exponential one simultaneously, EXPTIME – we will show that this is a common assumption without the existence of the correctable methodology for extended regular expression matching with subtraction which requires determinization in exponential time $O(2^n)$, where n is the length of the automaton or corresponding regular expression [1-3].

The remained question was the rigor proof of the correctness of our solution towards language or automaton complementation:

$(\sim L(r) \mid A)$, as we have used the tagging approach or even event-driven simulation we have also broadened the question towards the existence of the linear solution in subset construction within the example of 'state explosion' [4,5].

2. Statement & Proof

The statement of the problem is whether there exist polynomial solution to the problem of constructing the valid non-deterministic automaton (NFA) for the complement of the language $L(r)$ or any other automaton A – as we know this problem is in EXPTIME, however, we have already presented a rigor proof based upon the event-driven model and tagging conditions thus allowing this method to be linear [3,4].

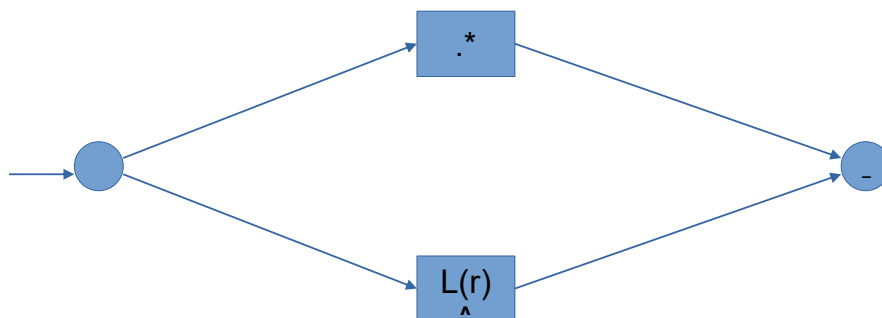


Figure 1: The Construction of Minus Event-Driven Model for Complement of Language $L(r)$

On the figure above we see the valid construction for the ‘minus-operator’ and its correctness according to event driven model of reaching the star language “.*” and not-reaching the language defined by $L(r)$ or any other automaton defined by A [3]. As we can see it’s linear in size and operation $O(n = |L(r)|, |A|)$, while the traditional construction is exponentially explosive and leads to the determinization with worst complexity $O(2^n)$.

We also make a remark that our event-driven construction is correct for any other arbitrary automaton, not only that which lay within P-domain of regular languages.

3. Discussion

We have shown in short, our approach of defining the complementary language or automaton, while preserving the linear, or polynomial, balance between the functionality and definition – all these leads us to the strong understanding of non-polynomial complexity as ‘what can be simulated with modified model’.

To go beyond the presented complement operation, we know that both subtraction and intersection of non-deterministic finite automata (NFA) require also exponential and factorial number of steps which lays in classes of complexity EXPTIME and NP as well, while our constructions produce linear results with magnitude $O(l)$, these constructions are better illustrated in [6].

Subtraction and complement of arbitrary automata are in P and NP, additionally to our prior arguments towards our proof, we are to complete the final note to be more rigor: yes, indeed, $P = NP$ as any arbitrary automaton which can be isomorphic to graph structures or to be of any topological order can be constructed and simulated using our methodology, which is worth to study in our first works.

Our method is linear in size of automata and relates to such operations like subtraction and complement based upon the event-driven model.

3.1 AI as a Delegate or Helper

Since the last time the artificial intelligence (AI) became a real trend and had faced many applications as well as an ethical side assumption from reasoning point. Obviously the first statement is that AI can replace the real reasoning by which human or even human labor can gradually supersede. The second statement in recent press as well as in many other relevant scientific articles discusses the assumption of the reflection of AI on real life.

We will show further that underlying question remains still open. Yes, indeed, human technology has delegated the helper function on AI in which it recently within the huge wave replaces many specialties, say, like an operator of call center. On this occasion we see that it’s helper, however, the ongoing question is how far we will delegate the role of AI nowadays and generally, whether it will be able to make decisions. For the past time we have seen the ambient reduction to this state, once AI is adopted and can be

considered in one of the shaped human roles.

However, it’s obvious that to delegate the decision-making onto AI requires at least the 100% sure that it can replace the real person or even figure in order to accomplish this task or even human-related mission.

Yes, AI is a crucial trend nowadays which cannot be overpraised, however, the main question of whether we can delegate it important decision-making ability is still vitally overwhelmed by its praises. From both sides, surely it can be considered as positive or negative, the positive side is that it’s a good application of AI who cannot make a mistake, however, from the negative side AI, which is trained on some set of data in order to build the interaction model, it’s still questioned whether it can replace the critical thinking which is characteristic only to human with hundred percent probability.

Thus, while there’s a debate on ethical questions related to AI, it’s still not disputed whether AI can be delegated the higher or even the highest role in human labor, law and life.

3.2 NP-Complete Problems Blindly

The statement was given before by Stephen Cook in his seminal work, where there’s a clear definition of oracle with ‘certificate’ and the problem solution itself defined as an automation.

Prior to this ‘milestone work’, we have worked on narratives of experimental proof and showed that the equivalence function exists, this is, however, not a definitive answer with the full composition of the problem according to oracle and problem languages.

Before we have shown that intersection of languages can be simulated on finite automaton with the ‘witness state’.

Let’s define the language proposed by our solution as $L_{\text{solution}}(A)$, where A is the alphabet and this language defines the certificates for the oracle.

We are to show that P equals NP and, thus, the problem can be solved and verified in any polynomial time.

Let’s assume that P not equals NP and thus, we have:

$$t\{L_{\text{solution}}(A)\} \gg |w|^{O(1)}.$$

The above statement means as originally that solution language cannot be defined by any polynomial automaton, however, there exists Aho-Corasick automaton which can encode any of the string of the fixed length $|w|$ within the polynomial complexity.

For our proof we are assuming that the solution automaton is polynomial in size and simulation and, thus, any set of solutions can be found on this suffix tree automaton due to Aho and Corasick.

Thus, it follows that complexity classes are equal on case of language $L_{\text{solution}}(A)$:

$$P=NP.$$

Generally speaking, we have used the ‘postpone’ technique in our proof, showing that the solution set of words is fixed and polynomial, thus, solution can be found in any polynomial time, since the verification of our word is polynomial, then what if we got the set of solutions, now we can construct the automaton on which the solution can be found by reverse approach from the states which are already verified.

As before we have shown that the function $f(NP) = P$ may exist, here we go with the same assumption, so that $g(f(NP)) = P$, even if $f(NP) = NP$, where $g(x)$ is an Aho-Corasick automaton function on the set of solutions $f(NP)$. Thus, more theoretical approach is more recommended, rather than pointing onto the question of finding the function $f(x)$.

Here in discussion section, we will give the proof by contradiction assuming $P \neq NP$:

$$P \neq NP \Rightarrow f(NP) \neq f(P) \Rightarrow g(f(NP)) \neq [g(f(P))=P].$$

The above statement is a contradiction, thus $P = NP$ as there exist the second function by which we can simply represent our solution tree as an Aho-Corasick automaton.

We could say more about our previous functional hypothesis, however, there is the time to make concluding remarks as for this time, this is all what could be made to the present time.

We have shown additional the proof by ‘blind technique’ in which we propose the existence of the set of solutions and the corresponding automaton, on which we find this solution by reverse search from the verified finishing states or leafs or Aho-Corasick tree, then going to the starting state from the parent node onward.

Thus, if solution exists for the related language of certificates, then it can be effectively encoded and found by the reverse technique. All these give the proof by not finding exact algorithm, but rather the theoretical approach.

Finishing all the above, we can make an outlet which goes as “we can verify the problem as well as construct the solution decision tree in the same polynomial time asserting that solution exists” – this is the preamble of our proof method.

4. Conclusion

We have demonstrated the underlying power and existence of the non-deterministic finite automaton for recognizing complementation of the given language or for constructing complement or subtraction for given automaton, which was exponential before, and this is a total contrary towards hierarchy theorem and many other misbeliefs like the intractability of the problem.

Thus, we have: $P = NP$, and our outlet goes as “**the problem can be verified and solved in polynomial time by using extensible approach**”.

Acknowledgment

We appreciate the rigor of the scientific journals in order to produce the prior works, which gave significant results in complexity theory.

References

1. Cook, S. (2003). The importance of the P versus NP question. *Journal of the ACM (JACM)*, 50(1), 27-29.
2. Hartmanis, J., & Stearns, R. E. (1965). On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117, 285-306.
3. Syzdykov, M. (2017). Methodology to Produce Deterministic Automaton for Extended Operators in Regular Expression. *International Journal of Scientific & Engineering Research*, 8, 1497-1500.
4. Syzdykov, M. (2022). Extended Regular Expressions in Finite Automata Revisited. *Advanced Technologies and Computer Science*, (1), 4-7.
5. Syzdykov, M. (2022). Equivalence of Complexity Classes via Finite Automata Derivatives. *Advanced Technologies and Computer Science*, 1(4), 9-14.
6. Syzdykov, M. (2017). Deterministic automata for extended regular expressions. *Open Computer Science*, 7(1), 24-28.

Copyright: ©2025 Mirzakhmet Syzdykov. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.