UDC 004.02
IRSTI 20.53.15

OPERATIONAL CALCULUS OF MODIFIED SUBSET CONSTRUCTION

Mirzakhmet Syzdykov

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

Abstract We present the continuation of studying Extended Regular Expression (ERE) on the view of modified subset construction within the overridden operators like intersection, subtraction, and re-written complement. As before we have stated that in this case the complexity has a decreasing nature and tendency. We will give the strict definition of the operational part of this modified subset construction which is due to Rabin and Scott. The complexity of algorithm remains a magnitude less than NP-hard problems for which we have given the strict proof of equivalence in the prior work, so this work continues the studying of the comparable proof for a variety of problems to be computationally complex, however, explainable in terms of unified approach like operational calculus. In this calculus the general points of research are given to the representation of modified subset construction with at least two operands which are to be computed by subset construction and in terms of complexity of the effective algorithm they are computed using modified subset construction.

Keywords: subset construction, extended regular expressions, modification, operations, calculus.

Introduction

The subset or Rabin-Scott construction which was full described in [1] represents conservative system of choosing between determinism and non-determinism in both aspects, however, lacks the efficiency of complexity in case of deterministic machine operating on the finite set of states, thus, it's obvious that it will lead the number of states as well as number of operations to grow exponentially in time of $O(2^n)$.

The latter case isn't limited to the usage of the classical Thompson algorithm [2], which is less complex and requires asymptotic explosion of complexity in $O(m \cdot n)$, where m is the number of symbols in sought or input string and n is the number of elements in matched regular expression. To the present time Thompson's constructions weren't used for extended regular expression matching.

Samuel C. Hsieh showed a more effective algorithm for intersection operator [3], however, it's still NP-hard as its complexity can be denoted by $O(n^t)$, where n is the average size of length of operands in ERE and t is the number of &-operators.

We have shown that ERE for intersection problem can be computed on both deterministic and non-deterministic finite automata, NFA and DFA respectively [4]. We have also introduced the sliced model of computation for our algorithm which tends to be in magnitude faster by applying operational calculus [5].

Our algorithm for NFA or DFA supersedes previous results [6, 7] which operate on the cross-product construction of the DFA by applying the operational calculus in the form of the operational logic for the set of operands to be performed in-time and in-memory. The non-emptiness intersection problem was shown to be NP-hard for sparse set of automata [7], however, we give another argument towards our conjecture of equivalence of complexity classes.

Aho-Corasick trees [8] and Lempel-Ziv-Welch [9] streams for regular expression matching are also discussed in this article as well as the "P versus NP" conjecture [10] for common case of the problem of deciding whether the intersection of the given languages is empty or not.

Modified Subset Construction

This construction differs from the usual approach by Rabin-Scott in implementation of additional operators for the closure function which is defined as follows:

$$\varepsilon - closure(S) = \{t : f^{i}(s, \varepsilon) = t, \forall s \in S, \forall i = 1..n, n \in N\}. \qquad (1)$$

Where in (1), f(x, y) is the reaching state function in the NFA and S is the set of states. We extend this construction with the additional operator in our defined calculus as it's given in prior work [4]:

$$\varepsilon - modified - closure(S) = \{t : t \in \varepsilon - closure(S) \cup t : g(t) = 0\}. \qquad (2)$$

Where in (2) the g(t) is the base function which is computed during the approximation of the algorithm to the given point. By this point we define the artificial states which are implied for the subset construction with modifications the complexity of which is $2^{o(n)}$. The summary for this function can be found in [4].

Thus, for intersection operator g(t) is defined as follows:

$$g(t) = deg^{+}(t) - |visited(t)|. \qquad (3)$$

Whereas in (3) $deg^{+}(t)$ denotes the number of incoming edges for the given state t which is artificial by the definition as it wasn't implemented or introduced in prior works [1, 2, 3], visited(t) is the function denoting the number of visited edges during the closure computation process – we conclude that at each step this function is evaluated to its default value of zero.

The function g(t) for subtraction operator is defined as the logical gate consisting of binary input:

$$g(t) = \begin{cases} 0, visited(t) = \{L\} \\ -1 \end{cases}. \qquad (4)$$

Where in (4) L is the left operand to be visited and right operand is omitted if it wasn't visited before, otherwise, the logical expression defined by function g(t) in (4) is evaluated to false and no further calculation is permitted.

For the complement the function g(t) is defined within the modified construction and re-writing of this operator within the same expression (4).

On P versus NP

Since it was shown that non-emptiness intersection problem can be decided in non-polynomial time for specific cases, our algorithm decides it in time O(PQI) for any case [5].

Thus, we conclude that there's another argument towards the proof of P = NP. As the function g(t) in (2) is invariant and is to be computed for the implied states to model the existence of primarily intersection operator, it's still well-known that it was extended for the case of subtraction operator and re-writing of this operator for complement.

According to functional hypothesis there could be a set of positive transformations leading the algorithmic logic to be reduced to polynomial memory and time complexity, however, this question remains open as per our prior works we based our proof on the observation of the proved NP-hard problem to be solved using the full view of the input parameters in problem in polynomial time. PQI-operator [5] was introduced before to represent the exact computational complexity of the process in subset construction with function modifiers. These modifiers represent artificially implemented structures in the graph of automaton to be translated into the semaphore or any other logic gate so that the latter statement holds true and non-feasible subsets of automaton states aren't acceptable when traversing it through closure functions when passing it through the filter function g(t) defined in (3) for emptiness problem and in (4) for the common logical case.

The "P versus NP" theorem which wasn't still explicitly reviewed remains as a closed question as the author of the scientific work proposing the computational models for better evaluation of algorithm complexity had a better understood theoretical experience which leads to the question of the relation between polynomial (P) and non-polynomial (NP) classes to be open. However, still we have the facts which show that using subset construction in its modified full form can lead to the appearance of the more effective algorithms for non-emptiness intersection problem as well as to other problems where the redundant logical accepting states can be implemented as it's shown in the subset construction for extended operators in ERE like intersection, subtraction and complement.

Aho-Corasick Trees and Intersection Operator

As we have defined the intersection operator within the common Thompson's construction at first and then Rabin-Scott subset construction, it's possible to get the point for Aho-Corasick trees which denote the finite set of words and can be seen as a DFA.

For further purpose we can use the intersection state in NFA and get the construction for each accepting state in Aho-Corasick automata by implementing the state transition from accepting states to the finishing state of the pre-defined regular expression pattern, thus, giving the possibility to decide what words belong to the regular set.

The observation above leads to the minimization of accepting states by applying subset construction backwards from accepting states in Aho-Corasick tree [8]. This leads to the application of matching algorithmic constructions of using the mixed stream for both Aho-Corasick tree or compressed entity like Lempel-Ziv-Welch (LZW) stream.

The main conjecture is that Aho-Corasick trees optimized backwards by using Rabin-Scott backwards construction lead to the imminent minimization of this tree. This can be proved by the fact that each of the accepting and ending states in this tree conforms the decreasing function opt(x) which is defined as follows:

$$opt(x) = \{x \in subset - construction(A), \exists\, f(x,a), f(y,a) : a \in A\}. \qquad (5)$$

The definition (5) gives us the observation that trees are given from the starting single point and cannot be optimized further as they represent the optimized tree during online construction of this tree within the additional string to be added or which is already included in the tree. Thus, we can conclude that the state minimization process is to be started from the accepting states in backward direction.

Intersection operator can be applied to compressed or non-compressed trees within the observable time frame. This operator is for deciding the more complex and efficient algorithm for matching the regular expression pattern against the set of words rather than a single word or input stream of single source – in contrary, Aho-Corasick trees are of multiple sources and as we have shown can be minimized also.

For LZW input streams [9], it's defined that the ending mark from the encoded input can be used further when constructing tree itself during the invocation process – this is a linear process not requiring additional resources like memory and time and, thus, we conclude that these streams are unary.

Containerization of NFA and DFA for ERE and Stop Marks

This step of process includes the experimentation with the non-deterministic finite automata (NFA) conversion to deterministic finite automata (DFA) along the Extended Regular Expressions (ERE) within the aimed operators like intersection, subtraction and complement. The practice shows that in common case this is the best practice for implementation of composite NFA and DFA via subset construction.

In common sense, this is a good approach for developing analyzing tools in biomedicine for processing big amount of DNA sequences.

For the question of P = NP via O-operator proof, we can conclude that a single case is quite clear to conclude that this is a way of solving NP-hard problems laying outside NP-complexity class.

Matching algorithm with stop marks

We define the matching according to the non-trivial symbol in the sequence of concatenation, whereas the fully connected clique of states for empty transitions lets the exact word to be matched and consequent cliques are matched according to this stop mark.

The cliques are defined as the strongly connected components in which any word can be defined in the final set, thus, allowing the mark to be matched before the actual matching starts – this technique was used before to prove the equivalence of P and NP classes.

Conclusion

We have defined the necessary relations between operational calculus and ERE constructions and evaluation on either NFA or DFA through Modified Subset Construction (MSC). This calculus gives a broader observation of how our model is to be represented in operational logic and applied mathematics.

This relation is to introduce the solution to regular language non-emptiness intersection problem within the time $2^{o[f(n)]}$.

We have also shown that P versus NP conjecture for automata non-emptiness intersection problem can be considered decidable in polynomial time, thus giving the assumption that P equals NP.

The practical solutions to the minimization of Aho-Corasick tree and the usage of LZW input streams is also given as we have shown that these trees can be efficiently optimized using backward propagation method of the closure computation.

We can conclude more that operational calculus can be used in approximate regular expression matching, however, this is a well-studied question and doesn't require more attention as the definition of the new algebraic structure. This structure remains open for extended operators like intersection, subtraction and complement in ERE.

Acknowledgements

The author expresses gratitude to all the members of ResearchGate™ for their valuable comments on the publication of this work.

References

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

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

[3] Hsieh S. C. Product construction of finite-state machines. Proc. of the World Congress on Engineering and Computer Science. 2010. 141-143.

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

[5] Syzdykov M. et al. Introducing pqi-operator in theory of computational complexity. Advanced technologies and computer science. 2022. 3. 4-9.

[6] de Oliveira Oliveira M., Wehar M. On the fine grained complexity of finite automata non-emptiness of intersection. Developments in Language Theory. Proceed. of the 24th International Conference. Tampa, FL, USA, May 11–15, 2020. Cham: Springer International Publishing, 2020. 69-82.

[7] Fernau H., Hoffmann S., Wehar M. Finite automata intersection non-emptiness: Parameterized complexity revisited. arXiv preprint arXiv:2108.05244. 2021.

[8] Zha X., Sahni S. Highly compressed Aho-Corasick automata for efficient intrusion detection. 2008 IEEE Symposium on Computers and Communications. IEEE, 2008. 298-303.

[9] Bille P., Fagerberg R., Gørtz I. L. Improved approximate string matching and regular expression matching on Ziv-Lempel compressed texts. ACM Transactions on Algorithms. 2009. 6(1). 1-14.

[10] Cook S. The importance of the P versus NP question. Journal of the ACM (JACM). 2003. 50(1). 27-29.

МОДИФИКАЦИЯЛАНҒАН ІШКІ ЖИЫНДЫ ҚҰРУДЫҢ ОПЕРАЦИЯЛЫҚ ЕСЕБІ

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

Андатпа. Біз қиылысу, алу және қайта жазылған толықтауыш сияқты қайта анықталған операторлар ішіндегі өзгертілген ішкі жиын құрылысының көрінісі бойынша кеңейтілген тұрақты өрнекті зерттеудің жалғасын ұсынамыз. Бұрынғыдай біз бұл жағдайда күрделіліктің төмендеу сипаты мен тенденциясы бар екенін айттық. Біз Рабин мен Скоттқа байланысты осы өзгертілген ішкі жиынтық конструкцияның операциялық бөлігіне қатаң анықтама береміз. Алгоритмнің күрделілігі біз жұмыста эквиваленттіліктің қатаң дәлелін берген NP-қиын есептерден аз шама болып қала береді, сондықтан бұл жұмыс әртүрлі есептердің күрделі болуы үшін салыстырмалы дәлелдемелерді зерттеуді жалғастырады, алайда, операциялық есептеу сияқты біртұтас көзқарас тұрғысынан түсіндіріледі. Бұл есептеуде зерттеудің жалпы тармақтары ішкі жиынды құру арқылы есептелуі тиіс кемінде екі операндтары бар түрлендірілген ішкі жиын құрылысын ұсынуға берілген және тиімді алгоритмнің күрделілігі тұрғысынан олар өзгертілген ішкі жиын құрылысы арқылы есептеледі.

Кілттік сөздер: ішкі жиын құрылысы, кеңейтілген тұрақты тіркестер, модификация, амалдар, есептеу.

ОПЕРАЦИОННОЕ ИСЧИСЛЕНИЕ ПОСТРОЕНИЯ МОДИФИЦИРОВАННОГО ПОДМНОЖЕСТВА

Сыздыков Мирзахмет
КазНИТУ им. К.И. Сатпаева, Алматы, Казахстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Аннотация. Мы представляем продолжение изучения расширенных регулярных выражений с точки зрения построения модифицированного подмножества внутри переопределенных операторов, таких как пересечение, вычитание и перезаписанное дополнение. Как и прежде, мы утверждали, что в этом случае сложность имеет убывающий характер и тенденцию. Мы дадим строгое определение операционной части этой модифицированной конструкции подмножества, принадлежащей Рабину и Скотту. Сложность алгоритма остается на величину меньше, чем у NP-сложных задач, для которых мы дали строгое доказательство эквивалентности в предыдущей работе, поэтому эта работа продолжает изучение сравнимого доказательства для множества задач, которые, однако, являются вычислительно сложными. объяснимо с точки зрения единого подхода, такого как операционное исчисление. В этом исчислении основные точки исследования отданы представлению модифицированной конструкции подмножества с не менее чем двумя операндами, которые должны быть вычислены путем построения подмножества, и с точки зрения сложности эффективного алгоритма они вычисляются с использованием модифицированной конструкции подмножества.

Ключевые слова: построение подмножества, расширенные регулярные выражения, модификация, операции, исчисление.

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