Минимизация автомата Ахо-Корасик. Приложение к вычислению *P*-значения вхождений образцов

Фурлетова Е.И.

Институт математических проблем биологии РАН, Пущино, Московская область, Россия

evgfurletova@yandex.ru

Автомат Ахо-Корасик широко используется в компьютерных науках при анализе вхождений слов-образцов в текстовых последовательностях. В частности, он применяется при вычислении вероятности (*P*-значения) обнаружения слов из заданного набора в случайной последовательности. В биоинформатике *P*-значение используют для оценки статистической значимости кластеров функционально-значимых фрагментов биологических последовательностей. На практике автомат Ахо-Корасик может иметь огромное число состояний, что усложняет его использование, поэтому разработка эффективного алгоритма минимизации автомата Ахо-Корасик является актуальной задачей. В данной работе предложен алгоритм построения псевдо-минимального автомата Ахо-Корасик, основанный на специальном отношении эквивалентности на его состояниях. Сложности по времени и по памяти алгоритма линейны по числу состояний изначального автомата. В ряде случаев, в частности, для набора, состоящего из слов одинаковой длины, построенный автомат совпадает с минимальным. Для набора, заданного словом-шаблоном, алгоритм строит минимальный автомат напрямую без построения изначального автомата Ахо-Красик. Псевдо-минимальный автомат использован в алгоритмах нахождения точного и приближенного *P*-значений обнаружения слов в последовательности.

Ключевые слова: *автомат Ахо-Корасик, минимальный автомат, P-значение.*

Aho-Corasick automaton minimization. Application to the computation of pattern occurrences *P*-value

Furletova E.I.

Institute of Mathematical Problems of Biology RAS, Pushchino, Russia

Aho-Corasick automaton is widely used in computer sciences for the analysis of occurrences of patterns in text strings. In particular, it is applied for computing of the probability (*P*-value) to find words from a given set in a random sequence. In bioinformatics, *P*-value is used to estimate the statistical significance of clusters of functionally-significant fragments of biological sequences. In practice, the Aho-Corasick automaton can have a huge number of states that complicates its use. Development of an efficient algorithm of its minimization is an important task. This paper presents an algorithm for constructing of a pseudo-minimal Aho-Corasick automaton that is based on a special equivalence relation on its states. The time and space complexities of the algorithm are linear on the number of states of the initial automaton. In some cases, in particular, for a set consisting of words of a same length, the constructed automaton is minimal one. For a set described by a template word, the algorithm builds the minimal automaton directly without constructing the initial one. The pseudo-minimal automaton is used for computing the exact and approximate *P*-values of patterns occurrences.

Key words: Aho-Corasick automaton, minimal automaton, P-value.

1. Введение

Автомат Ахо-Корасик был впервые предложен в 1975 г. Альфредом Ахо и Маргарет Корасик для поиска вхождений набора образцов (слов) в тексте [1]. Пусть дан алфавит Σ и множество слов H в этом алфавите. Автомат Ахо-Корасик распознает язык $\Sigma \cdot H$, т.е. множество всех строк, заканчивающихся словом из H. Последовательно получая на вход символы текстовой строки, автомат переходит в соответствующие состояния. Если достигнуто допускающее состояние, то имеет место вхождение образца из H. Состояниями автомата Ахо-Корасик являются префиксы слов из H, допускающими состояниями – слова из H. Из состояния q по символу a автомат переходит в состояние $\delta(q,a)$, соответствующее максимальному суффиксу qa,

который является префиксом некоторого слова из H. Данный автомат является полным и детерминированным.

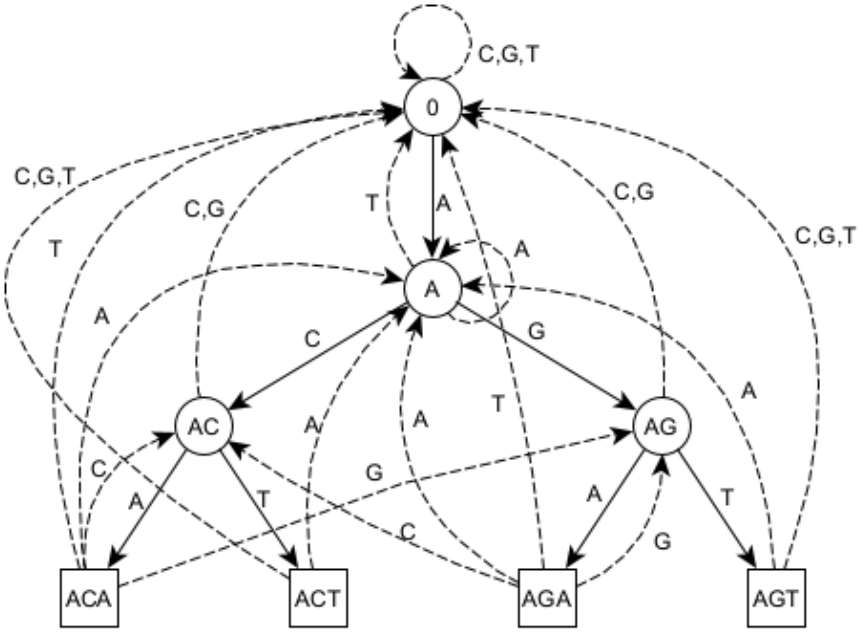

Рис. 1. Автомат Ахо-Корасик для набора, заданного шаблоном H = A{C,G}{A,T}. Допускающие состояния отмечены прямоугольниками, остальные – окружностями. Прямые переходы отмечены прямыми сплошными линиями, обратные – пунктирными дугами.

Автомат Ахо-Корасик возникает при решении задач лингвистики, компьютерной безопасности, биоинформатики и в других областях компьютерных наук [2]. В частности, при оценке достоверности обнаружения функционально-значимых фрагментов в биологических последовательностях [3, 4]. Например, факторов регуляции транскрипции.

На практике автомат Ахо-Корасик может иметь огромное число состояний. Например, семейство белков GAS_VESICLE_C из базы данных PROSITE [5] задается шаблоном (мотивом) FLXXTXXXRXXXAXXQXXXLXXF, где символы, кроме X, обозначают определенную аминокислоту, X – любую аминокислоту. Тогда соответствующий набор состоит из $\approx 10^{29}$ слов, а автомат Ахо-Корасик содержит $\approx 10^{30}$ состояний. Построение, хранение и использование такого автомата практически затруднительно. Минимальный автомат для рассмотренного набора имеет всего лишь 1480 состояний.

Отсюда возникает необходимость построения минимального автомата. Наиболее эффективный из универсальных алгоритмов минимизации конечных автоматов, алгоритм Хопкрофта [6], имеет временную сложность $O(|\Sigma|\cdot s\cdot log(s))$, где s – число состояний изначального автомата. Однако для многих прикладных задач, когда автомат имеет огромное число состояний, этот алгоритм недостаточно эффективен, поэтому построение специального алгоритма минимизации автомата Ахо-Корасик является актуальной задачей.

В работе [7] предложен алгоритм построения минимального автомата Ахо-Корасик, временная сложность которого линейна по числу состояний s изначального автомата. Однако алгоритм имеет квадратичную сложность по памяти. Другим подходом является построение псевдо-минимального автомата, размер которого лежит в промежутке между размером изначального и минимального автоматов. В работе [8] предложен линейный по времени и по памяти алгоритм построения псевдо-минимального автомата, который основан на специальном отношении эквивалентности на состояниях автомата Ахо-Корасик. Однако размер такого псевдо-минимального автомата может значительно превосходить размер минимального.

2. Основные результаты

2.1. Основные определения

Автомат Ахо-Корасик – это пятерка $AC\_aut(H) = (Q, \Sigma, \delta, q_0, F)$, где Q – множество состояний, состоящее из всех префиксов слов из H; $q_0 = \varepsilon$ – начальное состояние; $F = Q \cap \Sigma^*\cdot H$ – множество допускающих состояний; δ: $Q\times\Sigma\to Q$ – функция переходов. Имеем $\delta(q, a) = qa$, если $qa \in Q$, иначе $\delta(q, a) = sl(qa) = \delta(sl(q), a)$, где $sl(x)$ (суффиксная

ссылка) – максимальный суффикс x, который является префиксом некоторого слова из H. В первом случае будем говорить, что $\delta(q, a)$ – префиксный переход, а во втором случае – суффиксный.

Определим отношение $\sim^R$-эквивалентности на множестве состояний автомата Ахо-Корасик. Два состояния q и g являются $\sim^R$-эквивалентными ($q \sim^R g$), если $q = g$ или выполняются следующие условия:

1. $qt \in F \Leftrightarrow gt \in F$;
2. $sl(q) \sim^R sl(g)$.

Если для состояний q и g выполняется первое условие $\sim^R$-эквивалентности, то будем говорить, что q и g префиксно-эквивалентны. Таким образом, два состояния автомата Ахо-Корасик $\sim^R$-эквивалентны, если они префиксно-эквивалентны и их суффиксные ссылки $\sim^R$-эквивалентны. $\sim^R$-минимальным будем называть автомат, который не содержит различных $\sim^R$-эквивалентных состояний.

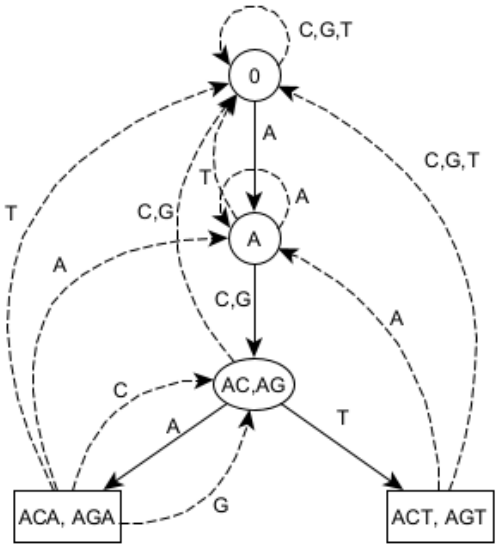

Рис. 2. Минимальный автомат Ахо-Корасик для набора, заданного шаблоном H = A{C,G}{A,T}.

2.2. $\sim^R$-эквивалентность и эквивалентность Нероуда

Напомним, что два состояния q и g конечного автомата являются эквивалентными в классическом смысле (эквивалентными по определению Нероуда), если для любого слова $t \in \Sigma^*$ имеем $\delta(q,t) \in F \Leftrightarrow \delta(g,t) \in F$, см. [9]. Автомат является минимальным, если он не содержит двух различных эквивалентных состояний. Показано, что $\sim^R$-эквивалентные состояния являются эквивалентными в классическом смысле, то есть число состояний $\sim^R$-минимального автомата лежит в промежутке между числом состояний минимального и изначального автоматов. В общем случае обратное неверно. Однако в некоторых случаях, в частности, в случае набора, состоящего из слов одинаковой длины, $\sim^R$-эквивалентность тождественна эквивалентности Нероуда.

2.3. Построение $\sim^R$-минимального автомата

$\sim^R$ – минимальный автомат может быть построен в два этапа. На первом этапе состояния разбиваются на классы префиксной эквивалентности. Если в автомате Ахо-Корасик удалить суффиксные переходы, то получится автомат, соответствующий граф которого не содержит циклы. Эквивалентность Нероуда для такого автомата тождественна префиксной эквивалентности изначального автомата Ахо-Корасик. Таким образом, нахождение классов префиксной эквивалентности аналогично минимизации этого автомата. Минимизация ациклических автоматов широко изучена в литературе [10]. Алгоритмы минимизации можно условно разделить на два типа. Для алгоритмов первого типа требуется построение изначального автомата, который они минимизируют. Алгоритмы второго типа для заданного набора слов строят напрямую минимальный автомат без построения изначального. Сложности по времени и по памяти наилучших из алгоритмов первого типа линейны по числу состояний изначального автомата.

На втором этапе построенные классы префиксной эквивалентности разбиваются на классы $\sim^R$-эквивалентности. Этот этап также имеет временную сложность, линейную по числу состояний автомата Ахо-Корасик.

2.4. Автомат Ахо-Корасик для наборов, состоящих из слов одинаковой длины

Как было указано ранее, для наборов слов одинаковой длины $\sim^R$-эквивалентность тождественна эквивалентности Нероуда. Такие наборы широко встречаются в компьютерных науках. В частности, в биоинформатике для задания мотивов функционально-значимых фрагментов биологических последовательностей используются матрицы позиционных весов (position weight matrix, PWM) и специальные слова-шаблоны (consensus, degenerate pattern, indeterminate pattern), при которых слова в мотивах (наборах) имеют одинаковую длину, см. [11].

Отдельно рассмотрим случай, когда набор задан шаблоном – словом в алфавите 2^Σ. Пусть задан шаблон $P = P_1...P_m$, где $P_i \in 2^\Sigma$. Слово $h = h_1...h_m$ принадлежит набору H, если $h_i \in P_i$. При таком задании набора H все состояния одинаковой длины

автомата Ахо-Корасик являются префиксно-эквивалентными, на основании чего был разработан алгоритм, который по заданному слову-шаблону напрямую строит минимальный автомат без построения промежуточных структур. Сложности по времени и по памяти такого алгоритма линейны по размеру итогового минимального автомата.

3. Оценка достоверности кластеров функционально-значимых фрагментов в биологических последовательностях

Функционально-значимые участки в биологических последовательностей, как правило, отличаются повышенным содержанием определенных слов [12]. Классической мерой перепредставленности слов из данного набора на участке последовательности является *P*-значение [13], т.е. вероятность обнаружить слова из данного набора в случайной последовательности заданной длины не менее заданного количества раз. Созданию эффективного алгоритма нахождения *P*-значения посвящено большое число работ [14]. Ряд разработанных алгоритмов использует специальный конечный автомат для вычислений, в частности, автомат Ахо-Корасик и производные от него структуры [3, 4]. Минимизация используемых структур данных позволяет значительно повысить эффективность алгоритмов.

В частности, в работе [4] нами был разработан алгоритм SufPref для нахождения указанного *P*-значения. Алгоритм использует для вычислений структуру данных – граф перекрытий, которая получается из автомата Ахо-Корасик путем удаления ряда вершин. Граф перекрытий можно также минимизировать, разбив его вершины на классы $\sim^R$-эквивалентности.

Также в работе [15] нами был предложен минимальный автомат для вычисления вероятностей клампов (строк, состоящих из перекрывающихся вхождений слов из набора) для наборов, заданных словами-шаблонами. По данным вероятностям можно найти приближенное *P*-значение с заранее заданной точностью, см. [16]. Этот автомат также строится на основе минимального автомата Ахо-Корасик.

4. Благодарности

Работа выполнена при поддержке грантов РФФИ 14-04-32220-mol-a и 14-01-93106-NCNILa. А так же при поддержке стипендии имени Мечникова от Campus France.

5. Список литературы

1. Aho A.V., Corasick M.J. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*. 1975. V. 18. P. 6.
2. Hasib S., Motwani M., Saxena A. Importance of Aho-Corasick string matching algorithm in real world applications. *IJCSIT*. 2013. V. 4. № 3. P. 467–469.
3. Boeva V., Clément J., Régnier M., et all. Exact p-value calculation for heterotypicclusters of regulatory motifs and its application in computational annotation of cisregulatory modules. *Algorithms Mol. Biol*. 2007. V. 2. P. 13. doi: 10.1186/1748-7188-2-13
4. Régnier M., Furletova E., Yakovlev V., Roytberg M. Analysis of pattern overlaps and exact computation of *P*-values of pattern occurrences numbers: Case of Hidden Markov Models. *Algorithms Mol. Biol.* 2014. V. 9. № 25. doi: 10.1186/s13015-014-0025-1
5. Sigrist C.J., de Castro E., Cerutti L., et all. New and continuing developments at PROSITE. *Nucleic Acids Res.* 2013. V. 41. D344-7. doi: 10.1093/nar/gks1067
6. Hopcroft J. An n log n algorithm for minimizing states in a finite automaton. *Theory of Machines and Computations.* 1971. P. 189–196. doi: 10.1016/B978-0-12-417750-5.50022-1
7. AitMous O., Bassino F., Nicaud C. Building the minimal automaton of A^*X in linear time, when X is of bounded cardinality. *CPM 2010. Lecture Notes in Computer Science*. V. 6129. doi: 10.1007/978-3-642-13509-5_25
8. AitMous O., Bassino F., Nicaud C. An efficient linear pseudo-minimization algorithm for Aho-Corasick automata. *CPM 2012. Lecture Notes in Computer Science.* V. 7353. P. 110–123. doi: 10.1007/978-3-642-31265-6_9
9. Fleck A. C. A Simplified view of Nerode equivalence. *Computing Letters*. 2000. V. 1. № 3. P. 93–96.
10. Bubenzer J. Minimization of acyclic DFAs. In: *Proceedings of the Prague stringology conference*. 2011. P. 132–146.
11. Stormo G.D. DNA binding sites: representation and discovery. *Bioinformatics*. 2000. V. 16. № 1. P. 16–23. doi: 10.1093/bioinformatics/16.1.16
12. Leung M.Y., Marsh G. M., Speed T.P. Over and underrepresentation of short DNA words in Herpesvirus genomes. *J. Comput. Biol*. 1996. V 3. P. 345–360. doi: 10.1089/cmb.1996.3.345
13. Zhang J., Jiang B., Li M., Tromp J., Zhang X., Zhang M. Computing exact p-values for DNA motifs. *Bioinformatics*. 2006. V. 23. P. 531–537. doi: 10.1093/bioinformatics/btl662
14. Lothaire. M. Statistics on words with applications to biological sequence. *Applied combinatorics on*

words. (Encyclopedia of Mathematics and its Applications). Cambridge: Cambridge University Press. 2005. V. 105. P. 268–352. doi: 10.1017/CBO9781107341005

15. Furletova E., Holub J., Regnier M. Minimized compact automaton for clumps over degenerate patterns. 2019. URL: https://hal.inria.fr/hal-01940837 (accessed 01.10.2022).
16. Regnier M., Fang B., Iakovishina D. Clump combinatorics, automata, and word asymptotics. *ANALCO*. 2014. P. 62–73. doi: 10.1137/1.9781611973204.6.