Секция 8

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ, АВТОМАТОВ И ГРАФОВ

УДК 519.713+519.766

О ПОСТРОЕНИИ МИНИМАЛЬНЫХ ДЕТЕРМИНИРОВАННЫХ КОНЕЧНЫХ АВТОМАТОВ, РАСПОЗНАЮЩИХ ПРЕФИКСНЫЙ КОД ЗАДАННОЙ МОЩНОСТИ

И. Р. Акишев, М. Э. Дворкин

В настоящее время префиксные коды находят широкое применение в различных областях информационных технологий. В связи с этим изучение их свойств представляет большой интерес.

Наиболее естественный подход к распознаванию префиксных кодов основан на применении конечных автоматов. Имеется ряд статей [1, 2], посвященных исследованию задачи минимизации числа состояний автомата, распознающего некоторый заданный префиксный язык.

В данной работе исследуется следующая задача: дано некоторое натуральное число n, необходимо построить детерминированный конечный автомат, принимающий некоторый префиксный код мощности n над алфавитом $\Sigma = \{0, 1\}$ и имеющий наименьшее возможное число состояний.

Имеют место следующие свойства искомого автомата.

Теорема 1. Пусть детерминированный конечный автомат A — минимальный автомат, принимающий некоторый непустой язык. Этот язык является префиксным тогда и только тогда, когда в автомате A ровно одно терминальное состояние, причем все переходы из него ведут в тупиковое состояние.

Теорема 2. Пусть детерминированный конечный автомат A — минимальный автомат, принимающий конечный префиксный язык. Тогда в этом автомате нет циклов, кроме петель, ведущих из тупикового состояния в само себя.

Теорема 3. В детерминированном конечном автомате, принимающем некоторый префиксный код заданной мощности n и имеющем минимальное число состояний, нет переходов в тупиковое состояние, кроме как из единственного терминального и тупикового состояний.

Так как искомый автомат не содержит циклов, можно выписать его состояния (кроме тупикового) в порядке обратной топологической сортировки:

$$q_0 = f, q_1, q_2, \ldots, q_{k-2} = s.$$

Рассматривается соответствующая последовательность мощностей правых контекстов для выписанных состояний:

$$a_0, a_1, a_2, \ldots, a_{k-2},$$

где $a_i = |R_{q_i}|$ (R_{q_i} — правый контекст состояния q_i).

Из равенств $R_{q_0} = R_f = \{\varepsilon\}$ следует, что $a_0 = 1$. Так как состояния автомата топологически отсортированы и из каждого выходит два перехода в нетупиковые состояния, то все последующие элементы последовательности a_i являются суммой каких-либо двух предыдущих. Последний элемент a_{k-2} соответствует R_s и равен мощности языка, принимаемого автоматом.

Определение 1. *Аддитивной цепочкой* [3] называется последовательность элементов a_i, такая, что

1) $a_0 = 1$;
2) $a_i = a_j + a_k$ для некоторых $j, k < i$ при всех $i > 0$.

Итак, искомому конечному автомату соответствует некоторая аддитивная цепочка. Из минимальной аддитивной цепочки посредством обратного построения можно получить конечный автомат, принимающий некоторый префиксный код заданной мощности. Длина аддитивной цепочки на 2 меньше числа состояний в соответствующем автомате. Отсюда вытекает следующая теорема.

Теорема 4. Задача нахождения детерминированного конечного автомата с минимальным числом состояний, принимающего некоторый префиксный код заданной мощности n, эквивалентна задаче построения кратчайшей аддитивной цепочки, заканчивающейся числом n.

Из теоремы 4 следует дополнительное свойство искомого префиксного кода.

Теорема 5. Префиксный код заданной мощности, соответствующий автомату с минимальным числом состояний, является полным.

Задача о нахождении кратчайшей аддитивной цепочки является классической задачей дискретной математики [3]. Наиболее известным ее применением является задача об оптимальном алгоритме возведения произвольного числа в заданную степень, представляющая интерес для криптографии [4].

Полиномиального решения данной задачи на данный момент неизвестно. Простым и достаточно эффективным приближенным решением является классический бинарный метод возведения в степень [3]. Активно применяются методы поиска приближенного ответа, в том числе ведутся исследования по применению генетических алгоритмов [5] и «муравьиных алгоритмов» [6]. В работе [7] показано, что задача нахождения кратчайшей аддитивной цепочки, которая содержит в качестве подпоследовательности данную последовательность $b_1, b_2, \ldots, b_k$, является NP-полной. То есть естественное обобщение рассматриваемой задачи не имеет полиномиального решения, если $P \neq NP$.

ЛИТЕРАТУРА

1. *Golin M. J., Na H.* Optimal prefix-free codes that end in a specified pattern and similar problems: The uniform probability case (extended abstract) // Data Compression Conference. 2001. P. 143–152.
2. *Han Y.-S., Salomaa K., Wood D.* State complexity of prefix-free regular languages // Proc. of the 8th Int. Workshop on Descriptional Complexity of Formal Systems. 2006. P. 165–176.
3. *Кнут Д. Э.* Искусство программирования. Т. 2. Получисленные алгоритмы. М.: Вильямс, 2004. 832 с.
4. *Bleichenbacher D.* Efficiency and security of cryptosystems based on number theory. Zürich, 1996.
5. *Cruz-Cortes N., Rodriguez-Henriquez F., Juarez-Morales R., Coello C. A.* Finding optimal addition chains using a genetic algorithm approach // LNCS. 2005. V. 3801. P. 208–215.

6. *Nedjah N., de Macedo M. L.* Finding minimal addition chains using ant colony // IDEAL / ed. by R. Y. Zheng, R. M. Everson, Y. Hujun. LNCS. 2004. V. 3177. P. 642–647.
7. *Downey P., Leong B., Sethi R.* Computing sequences with addition chains // SIAM J. Computing. 1981. V. 10. No. 3. P. 638–646.

УДК 519.171

О СООТНОШЕНИЯХ СТЕПЕНИ И ПЛОТНОСТИ НЕКОТОРЫХ ГРАФОВ

И. А. Бадеха, П. В. Ролдугин

В данной работе наиболее важным является понятие реберного покрытия графа кликами (РПК). РПК — это такой набор клик (полных подграфов) $K_1, ..., K_r$, что любое ребро графа G лежит хотя бы в одной из этих клик. В качестве клик, входящих в РПК, подразумеваются только максимальные по включению клики. Кроме того, будем отождествлять клику и множество ее вершин, то есть выражение «множество вершин R образует клику в графе G» означает, что множество вершин R порождает максимальный полный подграф в графе G. Назовем ребро e графа G собственным ребром клики K, если оно лежит в этой клике и не лежит ни в какой другой максимальной по включению клике графа G. Соответственно клику K, имеющую хотя бы одно собственное ребро, назовем зафиксированной.

Утверждение 1. Клика K входит в любое РПК графа G тогда и только тогда, когда она является зафиксированной.

Собственные ребра и соответственно зафиксированные клики допускают простую характеризацию, позволяющую легко распознать их в графе.

Утверждение 2. Ребро $e \in E(G)$ является собственным ребром некоторой клики K тогда и только тогда, когда множество вершин графа G, смежных одновременно с обоими концами ребра e, порождает полный подграф в G. Кроме того, этот полный подграф в объединении с концами ребра e образует клику K.

Из данного утверждения следует возможность нахождения всех зафиксированных клик графа за полиномиальное время (трудоемкость не более $O(n^4)$, где $n = |V(G)|$). Отсюда следует, что в определенном смысле графами, в которых сложно строить минимальное РПК, являются графы, не содержащие зафиксированных клик, или, что эквивалентно, собственных ребер. Далее такие графы, то есть графы, в которых каждое ребро лежит не менее чем в двух кликах, назовем графами, свободными от собственных ребер. Введем на множестве вершин графа G отношение эквивалентности. Две вершины x и y называются эквивалентными, если они смежны и их окружения совпадают, то есть $N(x) = N(y)$. Сжатым графом назовем граф, в котором все вершины попарно неэквивалентны.

Основное содержание работы отражает следующая теорема.

Теорема 1. Предположим, что G является связным сжатым графом, свободным от собственных ребер. Тогда

1) $\rho(G) \leqslant \Delta(G) - 1$;
2) если $\rho(G) = \Delta(G) - 1$, то $\Delta(G) = 4$, и граф G эквивалентен графу B;
3) если $\rho(G) = \Delta(G) - 2$, то в графе G существует не менее двух вершин степени $\Delta(G)$, либо граф G получается из графа B добавлением одной доминирующей вершины.