Секция «Дискретная математика и математическая кибернетика»

Порядок сложности поиска вхождений подслова

Перпер Евгений Михайлович

Аспирант

Московский государственный университет имени М.В.Ломоносова,
Механико-математический факультет, Кафедра математической теории
интеллектуальных систем, Москва, Россия

E-mail: e_m_perper@mail.ru

В работе рассматривается следующая задача: для произвольного подслова требуется перечислить все вхождения этого подслова в слова из заданного множества. Будем считать, что длина всех слов в этом множестве равна n, длина каждого запроса не превышает n, а число букв в алфавите равно k, причём $k \geq 2$. Похожая задача рассматривалась в [3,4], однако там требовалось перечислить не сами вхождения, а слова, содержащие эти вхождения.

Для того, чтобы формализовать задачу, дадим несколько определений, как это было сделано в [3]. Слово длины s, где $s \leq n$, представим набором длины n, первые s элементов которого принадлежат множеству $N_k \stackrel{def}{=} \{1, 2, \ldots, k\}$, и если $s < n$, то каждый из оставшихся $n - s$ элементов равен $k + 1$. Множество всех слов длины не более n обозначим через W_n^k. Для каждого слова $w \in W_n^k$ его длину будем обозначать через $l(w)$; i-й буквой слова $w \in W_n^k$, где $i \in \{1, \ldots, n\}$, назовём i-й элемент соответствующего слову w набора. В частности, если $i > l(w)$, то i-я буква слова w равна $k+1$. Будем обозначать i-ю букву слова w через $w[i]$. Через $w[a..b]$, где $a, b \in \mathbb{N}, 1 \leq a \leq b \leq l(w)$, будем обозначать такое слово $v \in W_n^k$, что $l(v) = b - a + 1$ и $v[i] = w[a + i - 1]$ для всех натуральных i, не превышающих $l(v)$. Будем называть слово $w[a..b]$ подсловом слова w, начинающимся с его a-й буквы и заканчивающимся его b-й буквой. Будем говорить, что слова $w \in W_n^k$ и $v \in W_n^k$ равны или совпадают (и писать $w = v$) , если $l(v) = l(w)$ и $v[1] = w[1], v[2] = w[2], \ldots, v[l(v)] = w[l(w)]$. Пару (w, i), где $w \in W_n^k$ и $i \in \mathbb{N}, i \leq l(w)$, назовём вхождением в слово w. Множество всех вхождений в слова из некоторого множества M будем обозначать как $Oc(M)$.

Введём бинарное отношение ρ, которое позволит устанавливать, когда вхождение $(v, i) \in Oc(N_k^n)$ удовлетворяет запросу $x \in W_n^k$ (отношение поиска):

$$x \ \rho \ (v, i) \Leftrightarrow x = v[i..l(x) + i - 1], \ x \in W_n^k, \ v \in N_k.$$

Рассмотрим задачу поиска вхождений (ЗПВ) $I =< W_n^k, V, \rho >$, где $V = \{v_1, v_2, \ldots, v_p\}$, $V \subseteq N_k^n$. Будем считать, что задача $I =< W_n^k, V, \rho >$ состоит в перечислении для произвольно взятого запроса $x \in W_n^k$ всех тех и только тех вхождений $\omega \in Oc(V)$, для которых выполнено $x \ \rho \ \omega$. Множество V назовём библиотекой. Так как множество запросов и отношение поиска фиксированы, ЗПВ I полностью определяется библиотекой.

Для решения ЗПВ I воспользуемся информационно-графовой моделью данных [1,2]. Введём понятие информационного графа (ИГ) в соответствии с [1,2], с тем отличием, что с помощью построенного ИГ будут перечисляться элементы множества $Oc(V)$, а не V. В определении понятия ИГ используются W_n^k и $Oc(V)$, а также множество F предикатов, заданных на множестве W_n^k (предикаты — это функции, которые могут принимать только два значения: 0 или 1), и множество G переключателей, заданных на множестве W_n^k (переключатели – это функции, область значений которых является начальным отрезком натурального ряда). Пару $\mathbb{F} =< F, G >$ будем называть базовым множеством.

Рассмотрим произвольный ориентированный граф. Выделим в нём одну вершину. Назовём её корнем. Выделим в графе какие-либо другие вершины. Назовём их листьями. Сопоставим каждому листу некоторое вхождение из множества $Oc(V)$. Это соответствие

назовем нагрузкой листьев. Выделим в графе некоторые вершины (это могут быть в том числе корень и листья) и назовем их точками переключения. Если β — вершина графа, то через ψ_β обозначим полустепень исхода вершины β. Каждой точке переключения β сопоставим какой-либо символ из G. Это соответствие назовем нагрузкой точек переключения. Для каждой точки переключения β ребрам, из нее исходящим, поставим во взаимно однозначное соответствие числа из множества $\{1, 2, \ldots, \psi_\beta\}$. Эти ребра назовем переключательными, а это соответствие — нагрузкой переключательных ребер. Ребра, не являющиеся переключательными, назовем предикатными. Каждому предикатному ребру графа сопоставим некоторый символ из множества F. Это соответствие назовем нагрузкой предикатных ребер. Полученный нагруженный граф назовем информационным графом над базовым множеством $\mathbb{F} =< F, G >$.

Определим функционирование ИГ. Скажем, что предикатное ребро проводит запрос $x \in W_n^k$, если предикат, приписанный этому ребру, принимает значение 1 на запросе x; переключательное ребро, которому приписан номер r, проводит запрос $x \in W_n^k$, если переключатель, приписанный началу этого ребра, принимает значение r на запросе x; ориентированная цепь ребер проводит запрос $x \in W_n^k$, если каждое ребро цепи проводит запрос x; запрос $x \in W_n^k$ проходит в вершину β ИГ, если существует ориентированная цепь, ведущая из корня в вершину β, которая проводит запрос x; запись y, приписанная листу α, включается в ответ ИГ на запрос $x \in W_n^k$, если запрос x проходит в лист α. Ответом ИГ u на запрос x назовем множество вхождений, попавших в ответ ИГ на запрос x, и обозначим его $J_u(x)$. Эту функцию $J_u(x)$ будем считать результатом функционирования ИГ u и называть функцией ответа ИГ u. Теперь понятие ИГ полностью определено.

Скажем, что ИГ u решает ЗПВ $I = \{W_n^k, V, \rho\}$, если для любого запроса $x \in W_n^k$ ответ на этот запрос содержит все те и только те вхождения из $Oc(V)$, которые удовлетворяют запросу x, то есть $J_u(x) = \{\omega \in Oc(V) : x\rho\omega\} \stackrel{def}{=} J_I(x)$.

Определим понятие сложности информационного графа на запросе. Сделаем это в более общем виде, чем в [3]: будем считать, что время вычисления функции f от запроса x зависит не только от f, но и от x. Обозначим через $Pr(u, x)$ множество всех предикатных вершин ИГ u, в которые проходит запрос x, а через $Sw(u, x)$ — множество всех переключательных вершин ИГ u, в которые проходит запрос x. Обозначим через $E(\beta)$ множество всех рёбер, выходящих из вершины β. Тогда определим сложность вычисления ИГ u на запросе x следующим образом:

$$T(u, x) = \sum_{\beta \in Sw(u,x)} t(g_\beta, x) + \sum_{\beta \in Pr(u,x)} \sum_{e \in E(\beta)} t(f_e, x),$$

где g_β — переключатель, сопоставленный вершине β, f_e — предикат, сопоставленный ребру e, $t(f, x)$ — время вычисления функции f от запроса x. $T(u, x)$ характеризует время обработки запроса x.

Объемом $Q(u)$ ИГ u назовем число ребер в ИГ u. $Q(u)$ соответствует объёму памяти, используемой информационным графом.

В качестве базового множества $\mathbb{F}$ рассмотрим $< F, G >$, $F = \{f(x)\}$, $G = \{g_i(x), i \in \mathbb{N}, i \le n\} \bigcup \{h_{v,i}(x), v \in W_{n-1}^k, i \in \mathbb{N}, i \le n\}$. Эти функции определяются следующим образом: $f(x) \equiv 1$; $g_i(x) = x[i]$;

$$h_{v,i}(x) = \begin{cases} 1, \text{ если } l(x) - i + 1 < l(v) \text{ и } x[i..l(x)] = v[1..l(x) - i + 1], \\ 2, \text{ если } l(x) - i + 1 \ge l(v) \text{ и } x[i..i + l(v) - 1] = v, \\ 3, \text{ во всех остальных случаях.} \end{cases}$$

С помощью предиката, тождественно равного 1, моделируется перечисление ответа на запрос. Вычисление этого предиката соответствует переходу по ссылке (например, к слову

из ответа), в то время как вычисление любого переключателя $g_i(x)$ включает в себя выделение i-й буквы слова x и переход по ссылке. По этой причине сложность вычисления предиката, тождественно равного 1, считается положительной, но меньшей, чем сложность вычисления переключателя $g_i(x)$. Будем считать, что сложность вычисления любого переключателя $g_i(x)$ равна 1, а сложность вычисления предиката $f(x)$ равна $t, 0 < t < 1$. Алгоритм вычисления переключателя $h_{v,i}(x)$ можно реализовать в виде цикла проверки на равенство букв $x[i+j-1]$ и $v[j]$ для $j = 1, 2, ..,$ который завершается, когда равенство нарушается либо слово v заканчивается. В каждом шаге этого цикла происходит два сравнения, а после завершения цикла осуществляется выбор одной из ссылок 1, 2 и 3 и переход по одной из выбранных ссылок. Положив сложность шага цикла равной 2, а суммарную сложность всех операций, осуществляемых вне цикла, равной 1, получим, что

$$t(h_{v,i}, x) = \begin{cases} 1 + 2 * (\max(l(x), i-1) - i + 2), \text{ если } h_{v,i}(x) = 1\ , \\ 1 + 2 * l(v), \text{ если } h_{v,i}(x) = 2, \\ 3 + 2 * \max\limits_{j: x[i..i+j-1]=v[1..j]} j, \text{ если } h_{v,i}(x) = 3. \end{cases}$$

Пусть $U(I, \mathbb{F})$ — множество всех ИГ над базовым множеством $\mathbb{F}$, решающих ЗПВ $I =< W_n^k, V, \rho >$; $e(x) = \min(l(x) + 1, n)$; $R(I, \mathbb{F}, x) = e(x) + t \cdot (|J_I(x)| - 1)$.

Теорема 1. *Для каждой задачи $I =< W_n^k, V, \rho >$, для любых $u \in U(I, \mathbb{F})$ и $x \in W_n^k$ выполняется следующее условие: если $|J_I(x)| \geq 1$, то $T(u, x) \geq R(I, \mathbb{F}, x)$.*

Пусть $U_{fast}(I, \mathbb{F}) = \{u \in U(I, \mathbb{F}) : \forall x \in W_n^k\ T(u, x) \leq 3R(I, \mathbb{F}, x)\}$, т.е. это множество информационных графов, решающих ЗПВ I, в которых для каждого запроса ответ находится за время, не превышающее $3R(I, \mathbb{F}, x)$. Для запросов, которым удовлетворяет хотя бы одно вхождение, это время превышает минимально возможное время поиска не более чем в 3 раза. Обозначим $Q_{fast}(p, n) = \max\limits_{I: I=<W_n^k, V, \rho>, |V|=p} \min\limits_{u \in U_{fast}(I, \mathbb{F})} Q(u)$.

Теорема 2. *При любых натуральных n и p выполнено $pn \leq Q_{fast}(p, n) \leq (2k + 11)pn$.*

Другими словами, теорема 2 утверждает, что для любой ЗПВ I найдётся информационный граф из множества $U_{fast}(I, \mathbb{F})$, чей объём превышает минимально возможный (теоретически) объём графа из $U_{fast}(I, \mathbb{F})$ не более чем в $2k + 11$ раз.

Источники и литература

1) Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. М., Физматлит, 2002.

2) Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С. Введение в теорию интеллектуальных систем. М., Издательский отдел факультета ВМиК МГУ. С. 94-117.

3) Перпер Е. М. Нижние оценки временной и объёмной сложности задачи поиска подслова. // Дискрет. матем. 2014. Т. 26, вып. 2. С. 58–70

4) Перпер Е. М. О функциональной сложности поиска подстроки. // Интеллектуальные системы. 2012. Т. 16, вып. 1-4. С. 299-320

Слова благодарности

Автор выражает благодарность проф. Э. Э. Гасанову за научное руководство и постановку задачи.