Вероятностный протокол нахождения наибольшего общего начала

18 марта 2017 г.

Теорема 1 ([1])**.** *Пусть даны n и $\delta > 0$. Существует вероятностный протокол с общими случайными битами и коммуникацией $O(\log(n/\delta))$, который для любой пары слов (x, y) длины n с вероятностью ошибки не более δ находит длину наибольшего общего начала слов x, y.*

Доказательство. Алисе и Бобу надо найти номер первого бита, в котором их слова различаются. Это делается бинарным поиском. Для этого они применяют вероятностный протокол для равенства (RET) к словам x, y. Напомним, что этот протокол имеет параметр ε, передает $\log_2(1/\varepsilon)$ бит, не ошибается на входах, где $x = y$, и ошибается с вероятностью не более ε на всех остальных парах. Параметр ε подберем позднее. Если RET сообщит, что $x = y$, то протокол останавливается с результом n. Иначе, Алиса и Боб запускают протокол RET на первых половинах своих последовательностей. Если ответ результат будет положительным (первые половины совпадают, то различие надо искать во второй половине, а иначе в первой). После $\log n$ шагов Алиса и Боб найдут первое различие.

Теперь определим параметр ε. Мы запускаем $\log n$ раз RET и в каждом запуске вероятность ошибки может быть ε, поэтому общая вероятность ошибки оценивается сверху как $\varepsilon \log n$. Поэтому нам нужно положить $\varepsilon = \delta / \log n$ и протокол будет передавать $\log n \log(\log n/\delta)$ бит, что немного больше, чем было обещано.

Сократить количество бит можно следующим приемом. Модифицируем алгоритм бинарного поиска так, чтобы он приводил к правильному ответу при условии, что RET ошибался не более, чем примерно в $1/4$ запусков. Вспмоним, что на каждом шаге этого алгоритма мы имели пару некоторый подотрезок $[l, r]$ отрезка $[1, n+1]$, прендположительно содержащий первый различающий бит (мы считаем, что этот бит равен $n+1$, если $x = y$). Сначала $l = 1, r = n+1$ и за один шаг заменяли либо l на $(l+r)/2+1$, либо r на $(l+r)/2$. При этом правая граница всегда правильна, поскольку мы уменьшаем r до $(l+r)/2$, если RET сообщил, что первый половины различаются, а такое сообщение не может быть ошибочным.

В модифицированном алгоритме бинарного поиска мы повторяем N раз следующую последовательность действий.

> (1) Проверяем равенство $x_{<l} = y_{<l}$ с помощью теста RET с параметром $1/8$. Если он говорит, что равенство неверно, то делаем откат на одну вершину назад в дереве двоичного поиска.
> (2) Если RET говорит, что равенство верно, и при этом мы находимся не в листе дерева поиска (то есть, $r > l$), то действуем, как в обычном алгоритме двоичного поиска.

Предположим, что N достатночно велико, а количество K ошибочных ответов теста RET не слишком большое. Докажем, что в этом случае в конце мы будем в правильной вершине дерева поиска. Предположим, что последнее не верно. Заметим, что раз попав в нужный лист дерева $[l, l]$, мы никогда уже оттуда не уйдем. В самом деле, в этом случае $x_{<l} = y_{<l}$, а тест RET не ошибается в случае равенства. Значит мы ни разу не были в правильной вершине. Нетрудно доказать, что в этому случае (когда мы ни разу не побывали в целевой вершине) следующая величина не увеличивается в ходе каждого выполнения цикла:

$$
\begin{aligned}
&(\text{количество вызовов RET})\\
&+2(\text{расстояние в дереве бинарного поиска}\\
&\text{цели})\\
&-4(\text{количество ошибочных ответов теста RET}).
\end{aligned}
$$

В самом деле, если в ходе цикла хотя бы раз тест RET дал ошибочный овтет, то последние член уменьшился на 4, первое слагаемое увеличилось не более, чем на 2, и второе, не более, чем на 2. Если же в ходе цикла тест не давал ошибочных ответов, то мы обязательно делаем шаг в правильном направлении, поэтому второй член уменьшается на 2, первый увеличивается не более, чем на 2, а третий не изменяется.

Сначала указанная величина равна $2\log n$, поэтому она никогда не становится больше $2\log n$. При этом в конце протокола она не меньше $N+2-4K$. Поэтому мы получим противоречие, если $N-4K \geqslant 2\log n$, то есть количество ошибок не превосходит $N/4 - \log n/2$.

Теперь нам нужно подобрать N так, чтобы вероятность события $K \geqslant N/4 - \log n/2$ была меньше δ. Положим $N = \max\{8\log n, C\log(1/\delta)\}$. Если случилось событие $K \geqslant N/4 - \log n/2$, то K/N отклонилось от своего среднего значения $1/8$ не менее, чем на $(1/4-1/16)-1/8 = 1/16$. Если C достаточно велико, то по неравенству Чернова вероятность этого события не больше $e^{-2(1/16)^2 N} \leqslant e^{-2(1/16)^2 C\log(1/\delta)} \leqslant \delta$. $\square$

Список литературы

[1] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Computing with noisy information. SIAM Journal on Computing, 23(5):1001–1018, 1994.