# Homework6 Solution

洪方舟
2016013259
Email: hongfz16@163.com

2018 年 4 月 6 日

## Ex 32.4-8

下面先证明一个引理：如果 $q = m$ 或 $P[q+1] \neq a$，则 $\delta(q,a) = \delta(\pi[q],a))$；
因为 $\pi[q] < q$，则 $\delta(q,a) \geq \delta(\pi[q],a)$；不妨令 $x = \delta(q,a), y = \delta(\pi[q],a)$，根据后缀函数的定义有 $P_x \sqsupset P_q a$，也即 $P_{x-1} \sqsupset P_q$，又根据前缀函数的定义，有 $x - 1 \leq \pi[q]$，且有 $P_x \sqsupset P_{\pi[q]} a$, 根据后缀函数的定义有 $y$ 是最长的前缀长度，于是有 $x \leq y$，也即 $\delta(q,a) \leq \delta(\pi[q],a)$，因此有 $\delta(q,a) = \delta(\pi[q],a)$；
引理证明完毕，下面给出相应算法；

COMPUTE-TRANSITION-FUNCTION$(P, \Sigma)$
1 let $\pi$ =COMPUTE-PREFIX-FUNCTION$(P)$
2 **for** $q = 0$ to $m$
3 **do for** each character $a \in \Sigma$
4 **do if** $q == m$ or $P[q+1] \neq a$
5 **then if** $q == 0$
6 **then** $\delta(q,a) = 0$
7 **else** $\delta(q,a) = \delta(\pi[q],a)$
8 **else** $\delta(q,a) = q + 1$
9 RETURN $\delta$

## Problem 32-1

### a.

首先证明引理：对于 $\pi[i] = i - k$，如果 $k|i$ 则有最大重复因子 $\frac{i}{k}$，否则最大重复因子为 1
对于第一种情况 $k|i$，此时根据前缀函数的定义有 $\frac{i}{k}$ 是 $P_i$ 的一个重复因子，如果有一个更大的重复因子 $x > \frac{i}{k}$，则有 $\frac{i}{x} < k$，并且满足 $P_{i-\frac{i}{x}} \sqsupset P_i$，这与前缀函数的定义矛盾！因此 $\frac{i}{k}$ 就是 $P_i$ 的最大重复因子；
对于第二种情况 $k$ 不能整除 $i$，假设 $P_i$ 有最大重复因子 $x > 1$，也即 $y^x = P_i$，则应当有 $\pi[i] \geq |y^{x-1}|$，否则就不满足前缀函数的定义，又因为此时 $k$ 不能整除 $i$，则有 $\pi[i] > |y^{x-1}|$，那么令 $y = ab$，且满足 $P_{\pi[i]} = y^{x-1}a$，此时由前缀函数的定义有 $(ab)^{x-1}a = (ba)^{x-1}b$，也即 $ab = ba$，那么一定存在 $\omega \in \Sigma^*, z > 1$ 使得 $y = \omega^z$，这与 $x$ 是最大重复因子矛盾！因此 $x = 1$，也即此中情况下最大重复因子为 1；
综上，引理证毕！下面给出计算最大重复因子的算法；

COMPUTE-MAX-REPETITION($P$)

```
1  let π =COMPUTE-PREFIX-FUNCTION(P)
2  for i = 1 to m
3        do let k = i − π[i]
4           if i%k == 0
5              then ρ[i] = i/k
6              else  ρ[i] = 1
7  RETURN ρ
```

## b.

对于长度为 $i$ 的串，如果有最大重复因子 $r$，则只需要确定前 $\frac{i}{r}$ 个字母就可以确定整个字符串，那么对于长度为 $i$ 的串，最大重复因子为 $r|i$ 的概率为 $\frac{2^{i/r}}{2^i}$，那么对于长度为 $i$ 的串，最大重复因子不小于 $r$ 的概率为

$$\begin{aligned}\sum_{r'\geq r,r'|i}\frac{1}{2^{i(r'-1)/r'}} &= \frac{1}{2^i}\sum_{r'\geq r,r'|i}2^{i/r'}\\ &= \frac{1}{2^i}\sum_{j=1}^{\lfloor i/r\rfloor}2^j\\ &\leq 2^{i/r-i+1}\end{aligned}$$

对于长度为 $m$ 的串，$\rho^*(P)$ 的期望值由下面的式子决定其上界

$$\begin{aligned}E(\rho^*(P)) &\leq \sum_{i=1}^{m}2^{i/r-i+1}i\\ &= 2\sum_{i=1}^{m}(2^{\frac{1}{r}-1})^i\\ &= 2\frac{a+a^{m+1}m(a-1)-a^{m+1}}{(1-a)^2}, where\ a=2^{\frac{1}{r}-1}\\ &\leq \frac{2}{1-2^{1/r-1}}\end{aligned}$$

由于最终表达式中没有出现 $m$，则可知 $E(\rho^*(P)) = O(1)$；

## c.

时间复杂度说明：假设 $s$ 一共进行了 $x$ 次循环，每一次循环中 $q$ 增加了 $q_i$ 次，也即总的操作数为 $\sum_{i=1}^{x}q_i$，又有 $\sum_{i=1}^{x}\frac{q_i}{k} = n-m$，则总的操作数为 $\sum_{i=1}^{n}q_i = (1+\rho^*(P))(n-m)$，又因为字符串预处理时间复杂度为 $O(m)$，所以该算法的时间复杂度为 $O(\rho^*(P)n+m)$；

正确性说明：本算法的主要思想是匹配模式串中重复出现的单元，该方法与 $Brute-Force$ 方法唯一不同的地方就在于变量 $s$ 变化时增加的数量 $\Delta = \lceil q/k\rceil$，$\Delta$ 是该字符串中可能重复出现单元的最小长度，每次跳过这个值是安全的，因为最大重复因子保证了被跳过的子串一定是最小单元的重复子串的前缀，如果在被匹配串上连最小的重复子串都无法匹配，则匹配失败的子串的后缀一定无法匹配，因此该算法的正确性可以得到保证。