1995

A Greedy Algorithm for the Shortest Common Superstring is Asymptotically Optimal

Alan Frieze

Wojciech Szpankowski
Purdue University, spa@cs.purdue.edu

Report Number:
95-052
A GREEDY ALGORITHM FOR THE
SHORTEST COMMON SUPERSTRING
IS ASYMPTOTICALLY OPTIMAL

Alan Frieze
Wojciech Szpankowski

CSD-TR-95-052
August 1995
A GREEDY ALGORITHM FOR THE SHORTEST COMMON SUPERSTRING IS ASYMPTOTICALLY OPTIMAL

July 14, 1995

Alan Frieze*  Wojciech Szpankowski†
Department of Mathematics  Department of Computer Science
Carnegie Mellon University  Purdue University
Pittsburgh, Pennsylvania 15213  W. Lafayette, IN 47907
U.S.A.  U.S.A.
afrp@andrew.cmu.edu  spa@cs.purdue.edu

Abstract

There has recently been a resurgence of interest in the shortest common superstring problem due to important applications in molecular biology (e.g., recombination of DNA) and data compression. The problem is NP-hard, but it has been known for some time that greedy algorithms work well for this problem. More precisely, it was proved in a recent sequence of papers that in the worst case a greedy algorithm produces a superstring that is at most \( \beta \) times \((2 \leq \beta \leq 3)\) worse than optimal. We analyze the problem in a probabilistic framework, and consider the total overlaps \( O_{\text{opt}} \) and \( O_{\text{greedy}} \) produced respectively by the optimal algorithm and a greedy one which turn out to be asymptotically equivalent. More precisely, we show that with high probability \( \lim_{n \to \infty} \frac{O_{\text{opt}}}{n \log n} = \lim_{n \to \infty} \frac{O_{\text{greedy}}}{n \log n} = \frac{1}{H} \) where \( n \) is the number of original strings, and \( H \) is the entropy of the underlying alphabet. Our result holds under a condition that the lengths of all strings are not too short. Finally, we provide several generalizations and extensions of our basic result.

1 Introduction

Various versions of the shortest common superstring (in short: SCS) problem play important roles in data compression and DNA sequencing. In fact, in laboratories DNA sequencing (cf. [2, 5, 10, 13]) is routinely done by randomly sequencing large amounts of relatively short fragments and then heuristically finding the shortest common superstring. The problem can be formulated as follows: given a collection of strings, say \( x^1, x^2, \ldots, x^n \) over an alphabet \( \Sigma \), find the shortest string \( z \) such that each of \( x_i \) appears as a substring (a consecutive block) of \( z \). In DNA sequencing, another formulation of the problem might be of even greater interest. We call it an approximate SCS and one asks for finding a superstring that contains

---

*This work was supported by CCR-9225008.
†This research was supported in part by NSF Grants CCR-9201978 and NCR-9206315.
approximately (e.g., in the Hamming distance sense) the original strings $x^1, x^2, \ldots, x^n$ as substrings.

It is known that computing the shortest common superstring is NP-hard since it is as hard as finding a maximum hamiltonian path in a special graph [18]. Thus constructing a good approximation to SCS is of prime interest. Only recently, the open problem of how to approximate the shortest superstring in a polynomial time was solved (cf. [3, 11, 18]). In particular, it was proved that a greedy algorithm can compute in $O(n \log n)$ time a superstring that in the worst case is only $\beta$ times (where $2 \leq \beta \leq 3$) longer than the shortest common superstring ([3, 11]).

Our results are also about greedy approximations of the shortest common superstring but in a probabilistic framework. We shall prove that a greedy algorithm of the SCS problem is asymptotically optimal in the sense that it produces a total overlap of SCS that differs from the optimal (maximum) overlap by a quantity that is order of magnitude smaller than the leading term of the overlap. More precisely, let $n$ be the number of (long) strings. We assume that the lengths of all strings are $\Omega(\log n)$ (see below for more precise formulation and relaxation of this assumption; cf. also [1]). Let also $O^n_{\text{greedy}}$ and $O^n_{\text{opt}}$ denote respectively the total overlap produced by greedy and optimal algorithms of SCS. We prove that with high probability (in short whp) $O^n_{\text{greedy}} \sim \frac{1}{H} n \log n$ and $O^n_{\text{opt}} \sim \frac{1}{H} n \log n$ for large $n$ where $H$ is the entropy of the alphabet. Thus, the relative error of the greedy and optimal overlaps tends to zero in probability as $n \to \infty$.

In this preliminary version we consider only the Bernoulli model in which symbols of the alphabet $\Sigma$ are generated independently with unequal probability of symbol generation. We deal with the Bernoulli model for simplicity but the results can be extended to more general models such as the mixing model that includes the Markovian model.

Furthermore, we only consider one greedy algorithm (which seems to be the weakest one) that works as follows: It starts with an arbitrary string and attach to it another string with the longest mutual overlap among all strings not yet used in building the superstring.

The literature on worst-case analysis of SCS is quite impressive with [3, 11, 18] obtaining important results (cf. also [7]). Probabilistic analysis of SCS is very scarce. Only very recently Alexander [1] proved that the average optimal overlap in the Bernoulli model is $EO^n_{\text{opt}} \sim \frac{1}{H} n \log n$. To the best of our knowledge (based on a correspondence of one of us with K. Alexander) the method used in [1] cannot be easily extended to a greedy approximation which is our main contribution. We also prove that $O^n_{\text{opt}} \sim \frac{1}{H} n \log n$ whp.

2 Main Results

Before we present our main results, we introduce some notation, and describe precisely our greedy algorithm.

Suppose $x = x_1 x_2 \ldots x_r$ and $y = y_1 y_2 \ldots y_s$ are strings over the same finite alphabet $\Sigma = \{\omega_1, \omega_2, \ldots, \omega_V\}$ where $V = |\Sigma|$ is the size of the alphabet. We also write $|x|$ for the
length of $x$. We define their overlap $o(x,y)$ by

$$o(x,y) = \max\{j : y_i = x_{r-i+1}, 1 \leq i \leq j\}.$$ 

If $k = o(x,y)$ then

$$x \oplus y = x_1x_2 \ldots x_ry_{k+1}y_{k+2} \ldots y_s.$$

Throughout the paper, all logarithms are to the base $e$ unless explicitly stated otherwise.

We study the following algorithm: its input is the $n$ strings $x^1, x^2, \ldots, x^n$ over $\Sigma$. It outputs a string $z$ which is a superstring of the input.

**Algorithm G1**

1. begin
2. $z \leftarrow x^1$; $I \leftarrow \{2,3,\ldots,n\}$;
3. while $I \neq \emptyset$ do
4. begin
5. $o(z,x^i) = \max\{o(z,x^s) ; s \in I\}$;
6. $z \leftarrow z \oplus x^i$;
7. $I \leftarrow I \setminus \{t\}$
8. end
9. output $z$
10. end

We will assume that the input vectors are independently generated. Each $x = x^j = x_1x_2 \ldots x_t$ is of the same length $t$ and $x_i$ is generated independently of $x_1, x_2, \ldots, x_{i-1}$ (the Bernoulli Model). Furthermore $P(x_i = \omega_j) = p_j > 0$ for $1 \leq j \leq V$. Let

$$H = -\sum_{i=1}^{n} p_i \log p_i$$

be the associated entropy.

Our interest lies in estimating the total overlap $O_n$ which we define next. Let $S$ be a set of superstrings, and we write $z_{gr}^t$ for the length of a partial superstring found up to the $t$th step of G1. The following two quantities are of interest to us:

$$O_{n}^{opt} = \sum_{i=1}^{n} |x^i| - \min_{x \in S} |x|,$$   \hspace{1cm} (1)

$$O_{n}^{gr} = \sum_{t=1}^{n} o(z_{gr}^t, x^t)$$   \hspace{1cm} (2)
where $o(x^i, x^j)$ is the maximum overlap found in Step 5 of the algorithm $G_1$.

Now we ready to formulate our main result:

**Theorem.** Consider the Shortest Common Superstring problem under the Bernoulli model. Let $P = \sum_{i=1}^{m} p_i^2$. Then, with high probability (whp)

$$\lim_{n \to \infty} \frac{O_n^{opt}}{n \log n} = \frac{1}{H} \quad \text{(pr.)} \quad \lim_{n \to \infty} \frac{O_n^{gr}}{n \log n} = \frac{1}{H}$$

provided

$$|x^i| > -\frac{d}{\log p} \log n$$

for all $1 \leq i \leq n$, where (pr.) means convergence in probability.

**Remarks and Extensions**

(i) **Not Equal Length Strings.** The assumption regarding equal length strings is not relevant as long as there are enough long strings satisfying (4). A precise formulation of the proportion of short and long strings such that the above Theorem still holds can be found in Alexander [1].

(ii) **Mixing Model.** We can relax our assumption concerning the Bernoulli model, and our results hold for a larger class of models known as the *mixing model* which includes the Markovian model as a special case. In the mixing model one assumes that two events, say $E_1$ and $E_2$ separated by $d$ symbols are almost independent as $d \to \infty$. More precisely, there exist a function $\psi_d$ such that

$$(1 - \psi_d)P(E_1)P(E_2) \leq P(E_1E_2) \leq (1 + \psi_d)P(E_1)P(E_2)$$

provided $\psi_d \to 0$ as $d \to \infty$. The reader is referred to [17] for more details regarding mixing models for problems on strings.

(iii) **Other Greedy Algorithms.** Certainly, there are other natural greedy algorithms for the SCS problem. The best known is the one which we call $G_2$ and can be phrased as: *repeatedly choose a pair of strings with maximum overlap and merge them.* We do not at this point have a detailed but we conjecture that our algorithm $G_1$ is stochastically dominated by $G_2$. This would enable us to prove a result similar to Theorem above.

(iv) **SCS Does Not Compress Optimally.** The SCS can be used to compress strings. Indeed, instead of storing all strings of total length $n\ell$ we can store the Shortest Common Superstring and $n$ pointers indicating the beginning of an original string. But, this does not provide optimal compression (which is known to be the entropy $H$ [4]). To see this, let us compute the compression ratio $C_n$ which becomes

$$C_n = \frac{n\ell - \frac{1}{H} n \log n + n \log_2(n\ell - \frac{1}{H} n \log n)}{n\ell} \to 1$$

while for the optimal compression we should have $C_n \to H < 1$. No wonder that SCS is *not* optimal: in the construction of SCS we do not use all available redundancy of all strings but only the one contained in suffixes/prefixes of the original strings.
(v) **Approximate SCS.** Let us define a distance between two strings, say \( x \) and \( y \) as the relative Hamming distance, that is, 
\[
d_n(x, y) = n^{-1} \sum_{i=1}^{n} d_1(x_i, y_i)
\]
where \( d_1(x, y) = 0 \) for \( x = y \) and 1 otherwise where \( x, y \in \Sigma \) and \( |x| = |y| = n \). For a given \( D < 1 \), we introduce an approximate SCS as follows: Construct the shortest common superstring of strings \( x^1, x^2, \ldots, x^n \) such that every string \( x^i \) is within Hamming distance \( D \) of a substring of the superstring. More precisely, the **Approximate (Lossy) Shortest Common Superstring** is a string of shortest length such that there exists a substring, say \( z^j \), of \( z \) such that 
\[
d(x^i, z^j) \leq D \quad \text{for all} \quad 1 \leq i \leq n.
\]
Based on our analysis and recent result of Łuczak and Szpankowski [12] we conjecture that also in this problem the optimal and greedy overlaps are asymptotically equivalent. However, the constant in front of \( n \log n \) is not any longer the entropy \( H \) but rather a generalized Rényi entropy introduced in [12]. To be more precise, let \( B_D(y) \) be a ball of radius \( D \) of sequences of length \( \ell \) centered at \( y \), that is, 
\[
B_D(y) = \{ x : d(x, y) \leq D \}.
\]
Then, defined the generalized Rényi entropy \( r_0(D) \) as in [12], i.e.,
\[
r_0(D) = \lim_{K \to \infty} -\frac{\ell}{k} \log \frac{P(\mathcal{B}_D(X^k))}{k}.
\]
It is proved in [12] that the limit above exists in the mixing model. With this definition in mind, we conjecture that the main result of this paper in the approximate case becomes
\[
\lim_{n \to \infty} \frac{O_n^{opt}}{n \log n} = \frac{1}{r_0(D)} \quad \text{(pr.)} \quad \lim_{n \to \infty} \frac{O_n^{Gr}}{n \log n} = \frac{1}{r_0(D)}.
\]
for large \( n \), provided \( \ell > \frac{1}{r_1(D)} \log n \), where \( r_1(D) \) is another constant defined in [12]. The upper bound is easy and can be proved as follows: First of all, we note that using the subadditive ergodic theorem we can replace the limit of the expected value in (5) by the almost sure limit (cf. [12]). This can be translated into the following generalization of the Asymptotic Equipartition Property (AEP): The set of all strings \( w_n \) of length \( n \) can be partitioned into two subsets \( \mathcal{B}_n \) ("bad set") and \( \mathcal{G}_n \) ("good set") such that for any \( \varepsilon > 0 \) and sufficiently large \( n \) we have 
\[
P(\mathcal{B}_n) \leq \varepsilon, \quad \text{and} \quad 2^{-n(r_0(D)+\varepsilon)} \leq P(\mathcal{B}_D(w_n)) \leq 2^{-n(r_0(D)-\varepsilon)} \quad \text{for} \quad w_n \in \mathcal{G}_n.
\]
By (6), it suffices to prove that 
\[
P(M_n > k) \leq O(1/n^\varepsilon) \quad \text{for} \quad k = r_0^{-1}(D) \log n.
\]
But 
\[
P\{M_n \leq k\} \leq n P(B_D(w_k)) + P(\mathcal{B}_k),
\]
and using the above AEP property, we immediately establish the upper bound. \( \square \)

### 3 Analysis

In this section we prove our Theorem. We observe that \( O_n^{Gr} \leq O_n^{opt} \), thus in a subsection below we first derive an upper bound on \( O_n^{opt} \), and then deal with a lower bound for \( O_n^{Gr} \) which will finish the proof.
3.1 Upper Bound on $O_n^{opt}$

Define $C_{ij}$ as the length of the longest suffix of $x^i$ that is equal to the prefix of $x^j$. Let

$$M_n(i) = \max_{1 \leq j \leq n, j \neq i} \{C_{ij}\}.$$ 

We write $M_n$ for a generic random variable distributed as $M_n(i)$ (since $M_n \overset{d}{=} M_n(i)$ for all $i$, where $\overset{d}{=}$ means “equal in distribution”). Observe that

$$O_n^{opt} \leq \sum_{i=1}^n M_n(i). \quad (6)$$

Thus, we need a probabilistic analysis of $M_n$ to obtain an upper bound on $O_n^{opt}$.

The following lemma summarizes our knowledge of $M_n$ and suffices to prove an upper bound on $O_n^{opt}$. We point out that $M_n$ has been studied before in several papers devoted to tries (e.g., [6, 8, 14]). For the proof of our Theorem, we need only part (i) of the lemma below. But, probabilistic behavior of $M_n$ is of its own interest, and find many other application in algorithms on strings. Therefore, we present below an extended lemma.

Lemma. (i) For any $\varepsilon > 0$

$$\lim_{n \to \infty} P\{(1 - \varepsilon) \frac{1}{H} \log n \leq M_n \leq (1 + \varepsilon) \frac{1}{H} \log n\} = 1 - O(1/n^\varepsilon) \quad (7)$$

where $H$ is the entropy defined in above. Furthermore, for almost all strings that are sufficiently long all but $\varepsilon n$ of the numbers $M_n/\log n$ are within $\varepsilon$ of $1/H$.

(ii) For large $n$ we have

$$EM_n = \frac{1}{H} \log n + \frac{\gamma}{H} + \frac{h_2}{2H} - P_1(\log n) + O(1/n) \quad (8)$$

$$Var M_n = \frac{h_2 - H^2}{H^3} \log n + C + P_2(\log n) + O(1/n) \quad (9)$$

where $h_2 = \sum_{i=1}^n p_i \log^2 p_i$, $\gamma = 0.577\ldots$ is the Euler constant, $P_1(x)$ and $P_2(x)$ are fluctuating function with small amplitude.

(iii) The following is true for $p \neq q$

$$\frac{M_n - EM_n}{\sqrt{Var M_n}} \overset{d}{\to} N(0, 1) \quad (10)$$

where $N(0, 1)$ is the standard normal distribution. The rate of convergence is $O(1/\sqrt{\log n})$, and the convergence also holds in moments.

Proof. Part (i) follows from (ii). The second part of (i) was proved by Shields [15] (cf. [12] for a more general proof for an approximate matching). So, we concentrate on the remaining parts, however, a short proof of an upper bound in (7) can be derived as follows.
Let \( w_k \) be a typical and given string of length \( k \), that is, by Shannon-McMillan-Breiman [4] theorem \( P(w_k) \leq e^{-k(H-\varepsilon)} \) for any \( \varepsilon > 0 \). This is true for all strings of length \( k \) except for those belonging to a set \( B_k \) such that \( P(B_k) \leq \varepsilon \). Then,

\[
P(M_n > k) \leq nP(w_k) + P(B_k)
\]

and the result follows immediately after substituting \( k = (1 + \varepsilon)H^{-1} \log n \).

Now we proceed to prove parts (ii) and (iii). For simplicity of presentation we now work on a binary alphabet with \( p_1 = p \) and \( p_2 = q = 1 - p \). From the inclusion-exclusion rule we have

\[
P\{\bigcup_{j=1}^r [C_j \geq k]\} = \sum_{r=1}^n (-1)^{r+1} \binom{n}{r} P\{C_1 \geq k, \ldots, C_r \geq k\}
\]

where the last equality is a consequence of

\[
P\{C_1 \geq k, \ldots, C_r \geq k\} = (p^{r+1} + q^{r+1})^k .
\]

Let now \( G_n(z) \) be the probability generating function of \( M_n \), and \( \hat{G}_n(z) = \sum_{k \geq 0} z^k P\{M_n \geq k\} \) (clearly, \( G_n(z) = (1 - \hat{G}_n(z))/1 - z \)). Thus,

\[
\hat{G}_n(z) = -\sum_{r=1}^n (-1)^r \binom{n}{r} \frac{1}{1 - z(p^{r+1} + q^{r+1})} .
\]

Observe that \( EM_n = \hat{G}_n(1) \) and \( EM_n(M_n - 1) = 2\hat{G}_n'(1) \). In both cases we have to deal with alternating sums, that is,

\[
EM_n = -\sum_{r=1}^n (-1)^r \binom{n}{r} \frac{1}{1 - (p^{r+1} + q^{r+1})}
\]

\[
EM_n(M_n - 1) = -\sum_{r=1}^n (-1)^r \binom{n}{r} \frac{p^{r+1} + q^{r+1}}{(1 - z(p^{r+1} + q^{r+1}))^2} .
\]

Observe also that (12) has a form of an alternating sum, too.

To deal efficiently with such sums we use Mellin-like approach (cf. [8, 16]). In particular, for all sequence \( f_k \) that do not grow too fast at infinity we have

\[
\sum_{r=1}^n (-1)^r \binom{n}{r} f_r = \left(1 + O\left(\frac{1}{n}\right)\right) \frac{1}{2\pi i} \int_{1/2-i\infty}^{1/2+i\infty} n^{-s} \Gamma(s) f(-s) ds .
\]

Then part (ii) is a direct consequence of the above and the Cauchy residue theorem. Part (iii) follows from the above and Goncharov's theorem (cf. [8]) which states that \( M_n \) are normally distributed if for a complex \( \theta \) (cf. [6])

\[
\lim_{n \to \infty} e^{-\theta \mu_n/\sigma_n} G_n(e^{\theta/\sigma_n}) = e^{\frac{1}{2} \theta^2}
\]

where \( \mu_n = EM_n \) and \( \sigma_n = \sqrt{\text{Var} M_n} \).
3.2 Lower Bound on $O_n^{gr}$

In this subsection we prove a lower bound on $O_n^{gr}$, thus proving our Theorem.

We now further assume that $\ell$ is sufficiently large so that it is unlikely for there to be a pair $i, j$ such that $o(x_i, x_j) \geq \ell/2$. Put $P = p_1^2 + p_2^2 + \cdots + p_r^2$ and let $E$ denote the event that there is no such pair. If $\ell = K \log n$ for some constant $K > 0$, then

$$
P(-E) \leq \left(\frac{n}{2}\right) \sum_{k=\ell/2}^{\ell} P^k = O(n^{2+K\log P}/2) = \sigma(1),
$$

provided

$$K > \frac{4}{\log P}. \quad (14)$$

We now analyse the performance of $G_1$ under the assumption that $(14)$ holds.

Given $(14)$ we write $x^t = a^t b^t$ where $a^t$ (resp. $b^t$) is the $\ell/2$-prefix (resp. suffix) of $x^t$. If $E$ occurs then the string $z$ produced by $G_1$ is unchanged if we replace $5.$ by

5a. $o(b, a^t) = \max\{o(b, a^s) ; s \in I\}$;

where $b$ is the $\ell/2$ suffix of $z$. Let $G_1a$ denote $G_1$ with 5. replaced by 5a.

We now do a probabilistic analysis of algorithm $G_1a$. The first observation is that the strings $b^s$, $s \in I$ have no influence on the choice of $t$ in 5a. Indeed we could delay generating $b^t$ until after $a^t$ has been chosen. This idea has been labelled the method of deferred decisions by Knuth, Motwani and Pittel [9]. Thus at the end of an execution of 5a:

**Proposition 1** The $\ell/2$ suffix of $z$ is random and independent of the previous history of the algorithm.

We continue by examining the likely shape of the strings $a^1, a^2, \ldots, a^n$. For $1 \leq k \leq \ell/2$ and $a \in \Sigma^{\ell/2}$ let $\rho_t = \rho_t(a, k)$ be defined by

$$\rho_t = |\{1 \leq i \leq k : a_i = \omega_t\}|.$$

Now for each $t, i, k, \rho_t$ is distributed as the binomial $B(k, \rho_t)$. For $\epsilon > 0$ and integer $k$ let

$$\Omega(k, \epsilon) = \{a \in \Sigma^k : \rho_t(a, k) \leq (1 + \epsilon)k\rho_t, 1 \leq t \leq m\}.$$

Let $a^{i,k}$ denote the $k$-prefix of $a^i$. Applying the Chernoff bounds we obtain

$$\mathbb{P}(a^{i,k} \notin \Omega(k, \epsilon)) \leq \sum_{t=1}^{m} e^{-\epsilon^2 k\rho_t/3}. \quad (15)$$

Our choice of $\epsilon, k$ for the remainder of this section is

$$\epsilon = (\log n)^{-1/3} \text{ and } k = \lfloor (1 - 2\epsilon)\frac{1}{H} \log n \rfloor.$$
So $e^2k \to \infty$ with $n$ and whp almost every $a^{i,k} \in \Omega(k,\epsilon)$. Next let $M(k,\epsilon) = \{|i: a^{i,k} \notin \Omega(k,\epsilon)|\}$. If $\theta = \theta(k,\epsilon)$ denotes the RHS of (15) then $M(k,\epsilon)$ is stochastically dominated by $B(n,\theta)$. So whp

$$M(k,\epsilon) = o(\epsilon n).$$  \hspace{1cm} (16)$$

Now consider a fixed $a \in \Omega(k,\epsilon)$. Then for each $1 \leq i \leq n$ we have

$$P(a^{i,k} = a) = \prod_{i=1}^{m} p^k_{i} = \xi(a)$$

$$\geq \prod_{i=1}^{m} p_{i}^{k(1+\epsilon)}$$

$$= \left( \prod_{i=1}^{m} p_{i}^{k} \right)^{k(1+\epsilon)}$$

$$= e^{-k(1+\epsilon)H}. \hspace{1cm} (18)$$

Let $N(a) = \{|i: a^{i,k} = a|\}$. Clearly $N(a)$ is distributed as $B(n,\xi(a))$ where $\xi(a)$ is the RHS of (17). With our definition of $k,\epsilon$ we see from (18) that $n\xi(a) \geq n^\epsilon$. Hence,

$$P(\exists a \in \Omega(k,\epsilon): N(a) \leq (1 + \epsilon)n\xi(a)) \leq |\Omega(k,\epsilon)| e^{-c N(a)/3}$$

$$\leq |\Omega(k,\epsilon)| e^{-c n^\epsilon/3}$$

$$\leq m^k e^{-c n^\epsilon/3}$$

$$= o(1). \hspace{1cm} (19)$$

Our useful knowledge of the shape of $a^1, a^2, \ldots, a^n$ is summarised in (16) and (19).

We now consider a tree process that mimics GLa. Let $T$ denote an infinite rooted $V$-ary tree. The $V$ edges leading down from each vertex are labelled with $\omega_1, \omega_2, \ldots, \omega_V$. The child $w$ of vertex $v$ for which edge $(v, w)$ is called the $\omega_i$ child of $v$. A vertex $v$ of $T$ at depth $d$ is identified with a string $s_d s_{d-1} \ldots s_1$ and is labelled with an integer $\nu(v)$. Here the edges of the path from the root of $T$ to $v$ has labels $s_1, s_2, \ldots, s_d$ and $\nu(v)$ is the number of $i$ such that the $d$-prefix of $a^i$ is $s_d s_{d-1} \ldots s_1$. Thus $T$ is defined by the strings $a^i$ and is independent of the strings $b^j$.

We model the progress of GLa in the following way: A particle $Z$ starts at the root. When at a vertex $v$ it moves to $v$'s $\omega_j$ descendent with probability $p_j$. The particle stops at depth $\ell/2$. Let $w = s_{\kappa} s_{\kappa-1} \ldots s_1$ be the lowest vertex on the path traversed that has a non-zero $\nu$ value. This process models the computation of the largest suffix $s_{\kappa} s_{\kappa-1} \ldots s_1$ of $z$ which can be merged with a prefix of an $a^i$ i.e. $a^i$.

We then model the deletion of $a^i = a_1 a_2 \ldots a_{\ell/2}$ which had the prefix $a_1 a_2 \ldots a_{\kappa}$. Let $w_i = a_1 a_2 \ldots a_i$. Put $\nu(w_i) = \max\{0, \nu(w_i) - 1\}$ for $1 \leq i \leq \ell/2$.

We repeat the above process $n$ times achieving values $\kappa_1, \kappa_2, \ldots, \kappa_n$ of $\kappa$. We will show that whp

$$\kappa_1 + \kappa_2 + \cdots + \kappa_n \geq (1 - 5c) \frac{1}{H} n \log n. \hspace{1cm} (20)$$
The final argument goes as follows. We want to show that w.h.p. we will have \( \kappa_t \geq k \) for \( 1 \leq t \leq n_0 = \lceil (1 - 3\epsilon)n \rceil \). Now most of the time the \( k \)-suffix \( z^k \) of \( z \) lies in \( \Omega(k, \epsilon) \). Indeed, the probability it doesn't is at most \( \theta \). This follows by calculation (15) and because \( s_1s_2 \ldots \) is a random string. If \( z^k \in \Omega(k, \epsilon) \) and

\[
\nu(a) \neq 0 \text{ for all } a \in \Omega(k, \epsilon)
\]

then \( \kappa \geq k \). We argue next that w.h.p. (21) holds up to \( n_0 = \lceil (1 - 3\epsilon)n \rceil \). If we consider a fixed \( a \in \Omega(k, \epsilon) \) then at this point the number of decrements \( r(a) \) in \( \nu(a) \) is distributed as \( B(n_0, \xi(a)) \). Hence, using \( n_0 \xi(a) \geq (1 - 3\epsilon)n^\epsilon \),

\[
P(\exists a \in \Omega(k, \epsilon) : r(a) \geq (1 + \epsilon)n_0 \xi(a)) \leq 2|\Omega(k, \epsilon)|e^{-\left(1 - 3\epsilon\right)n^\epsilon/3} = o(1).
\]

So w.h.p. at this point \( \nu(a) \geq n(1 - \epsilon)\xi(a) - (n_0 + cn)\xi(a) > 0 \) for every \( a \in \Omega(k, \epsilon) \). Thus, (20) follows immediately.

References


