Combinatorics, Probability and Computing (1998) 7, 365–373. Printed in the United Kingdom
© 1998 Cambridge University Press

Common Subsequences and Supersequences and their Expected Length

VLADO DANČÍK†

Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovakia
(e-mail: dancik@turing.upjs.sk)

Received 1 October 1996; revised 9 March 1998

Let $f(n,k,l)$ be the expected length of a longest common subsequence of l sequences of length n over an alphabet of size k. It is known that there are constants $\gamma_k^{(l)}$ such that $f(n,k,l) \to \gamma_k^{(l)} n$ as $n \to \infty$, and we show that $\gamma_k^{(l)} = \Theta(k^{1/l-1})$ as $k \to \infty$. Bounds for the corresponding constants for the expected length of a shortest common supersequence are also presented.

1. Introduction and preliminaries

The problem of finding longest common subsequences arises in various situations. As typical we can mention approximate string matching and text comparisons (*e.g.*, the *diff* function in UNIX) [1, 12]. Another important area where the longest common subsequence problem appears is molecular biology. The longest common subsequence problem is a special case of the more general sequence alignment problem [16].

The expected length of a longest common subsequence of two random sequences is a well-studied open problem (see [4], [6], [11], [10], [5], [2]). In this paper we shall concentrate on longest common subsequences of several sequences. We show that this expected length for l sequences of length n over an alphabet of size k is $\Theta(\frac{n}{k^{1-1/l}})$ for, roughly speaking, $n \gg k \gg l$; the exact meaning is given by Theorem 2.1 and Corollary 2.1. We also consider a dual case, the expected length of a shortest common supersequence.

Let $\Sigma = \{0, 1, \ldots, k-1\}$ be a fixed alphabet of size k. Let Σ^* be the set of all strings over Σ, including the empty string λ. The length of string u is denoted by $|u|$. The set of all sequences of length n is denoted by Σ^n. We shall work over the set $\Pi_n^l = (\Sigma^n)^l$ of all l-tuples of sequences of length n. We denote the set of all l-tuples of sequences by

† Supported by VEGA grant 1/4034/97. Most of the work was done while the author was a postgraduate student at Warwick University, England. Current address: Millennium Pharmaceuticals, 40 Erie Street, Cambridge, MA 02139–4254, USA (e-mail: dancik@mpi.com)

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

$\Pi^l = (\Sigma^*)^l$. The total length of the l-tuple $\mathbf{u} = (u_1, \ldots, u_l) \in \Pi^l$ is the sum of lengths of all sequences from the l-tuple, that is, $|\mathbf{u}| = l(u_1, \ldots, u_l) = |u_1| + \cdots + |u_l|$. Concatenation of two l-tuples $\mathbf{u} = (u_1, \ldots, u_l)$ and $\mathbf{v} = (v_1, \ldots, v_l)$ is defined by $\mathrm{cat}(\mathbf{u}, \mathbf{v}) = (u_1 v_1, \ldots, u_l v_l)$. Concatenation of more than two l-tuples is defined analogously.

A sequence u is a *subsequence* of a sequence v if u can be obtained from v by deletion of some symbols. A sequence v is a *common subsequence* of $\mathbf{u} = (u_1, \ldots, u_l) \in \Pi^l$ if, for every $i = 1, \ldots, l$, sequence v is a subsequence of sequence u_i. A *longest common subsequence* is a common subsequence of maximal possible length. We shall denote the length of a longest common subsequence of sequences $\mathbf{u} = (u_1, \ldots, u_l)$ by $\mathbf{L}(\mathbf{u}) = \mathbf{L}(u_1, \ldots, u_l)$.

2. Longest common subsequences

The expected length $\mathbf{EL}_n^{(l)}$ of a longest common subsequence is given by

$$\mathbf{EL}_n^{(l)} = \frac{1}{k^{ln}} \sum_{\mathbf{u} \in \Pi_n^l} \mathbf{L}(\mathbf{u}).$$

It is known (Chvátal and Sankoff [4]) that for every $k \geqslant 2$ and every $l \geqslant 2$ there are constants

$$\gamma_k^{(l)} = \lim_{n \to \infty} \frac{\mathbf{EL}_n^{(l)}}{n} = \sup_n \frac{\mathbf{EL}_n^{(l)}}{n}.$$

The existence of the constants γ_k follows from the fact that $\mathbf{EL}_n^{(l)}$ is superadditive. To get lower bounds for the expected length we design an algorithm producing any common subsequence (not necessarily the longest one). The expected length of such a common subsequence will be a lower bound for $\mathbf{EL}_n^{(l)}$.

We can use the following algorithm to obtain lower bounds for $\gamma_k^{(l)}$ for sufficiently large k. We scan the first input tape until $k^{1-1/l}$ different symbols are found. Let K_1 be the set of these symbols. Then we scan the second tape until $k^{1-2/l}$ different symbols from K_1 are read. We denote the set of symbols from K_1 scanned on the second tape by K_2. We shall continue in this way. Suppose we have determined the set K_{i-1}. We scan the ith tape until $k^{1-i/l}$ different symbols from K_{i-1} are found. These symbols form the set K_i. Having specified K_{l-1}, we search for some symbol $a \in K_{l-1}$ on the last tape. For convenience we put $K_0 = \Sigma$. Since

$$a \in K_{l-1} \subseteq K_{l-2} \subseteq \cdots \subseteq K_1,$$

we can put a into a common subsequence produced by the algorithm. Computing the average progress on each tape yields following lower bounds.

Theorem 2.1. *For every $l \geqslant 2$ we have*

$$\liminf_{k \to \infty} k^{1-1/l} \gamma_k^{(l)} \geqslant 1.$$

Proof. Let us consider the ith tape. We have constructed the set K_{i-1} of $k^{1-(i-1)/l}$ symbols. We are reading the ith tape until $k^{1-i/l}$ symbols from K_{i-1} are found. Let $\mathbf{EK}_i$ be the expected number of symbols read to achieve this. Since $\lim_{n \to \infty} \frac{\mathbf{EL}_n}{n} \geqslant \frac{1}{\max\{\mathbf{EK}_i : 1 \leqslant i \leqslant l\}}$, to

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

prove the theorem we have to show that for every i

$$\mathbf{EK}_i \leqslant k^{1-1/l}(1+o(1)).$$

The expected waiting time to collect one of j symbols is $1+(1-\frac{j}{k})+(1-\frac{j}{k})^2+\cdots=\frac{k}{j}$; therefore

$$\mathbf{EK}_i = \sum_{j=|K_{i-1}|-|K_i|+1}^{|K_{i-1}|} \frac{k}{j}.$$

There are $k^{1-i/l}$ summands and the largest one is $\frac{k}{k^{1-(i-1)/l}-k^{1-i/l}+1} \leqslant \frac{k}{k^{1-(i-1)/l}-k^{1-i/l}}$; hence

$$\begin{aligned}\mathbf{EK}_i &\leqslant k^{1-i/l}\frac{k}{k^{1-(i-1)/l}-k^{1-i/l}}\\ &\leqslant k^{1-1/l}\frac{1}{1-k^{-1/l}}\\ &\leqslant k^{1-1/l}(1+o(1)).\end{aligned}$$

□

Now we show how to get upper bounds that match the lower bounds up to a constant factor. Let $\mathscr{F}(i,n)$ be the set of l-tuples from Π_n^l with a longest common subsequence of length i, that is,

$$\mathscr{F}(i,n) = \{\mathbf{u} \in \Pi_n^l : \mathbf{L}(\mathbf{u}) = i\}.$$

The number of l-tuples in $\mathscr{F}(i,n)$ is denoted by $F(i,n)$. Let $H(i,n)$ be any upper bound for $F(i,n)$. If y is such that $\sum_{i=\lceil yn\rceil}^{n} H(i,n)$ is very small compared with k^{ln}, then l-tuples with a longest common subsequence larger than yn do not contribute much to the average and therefore the expected length must be no greater than yn. This is the basic idea of the first upper bound for pairs of sequences given by Chvátal and Sankoff [4] and also the basic idea of all further upper bounds.

Lemma 2.1. *If y is such that $\sum_{i=\lceil yn\rceil}^{n} H(i,n) = o(k^{ln})$, then $\gamma_k^{(l)} \leqslant y$.*

Proof. We note that

$$\mathbf{EL}_n^{(l)} = \frac{1}{k^{ln}}\sum_{i=0}^{n} iF(i,n), \quad \text{and} \quad \frac{1}{k^{ln}}\sum_{i=0}^{n} F(i,n) = 1.$$

Hence

$$\begin{aligned}\frac{\mathbf{EL}_n^{(l)}}{n} &= \frac{1}{nk^{ln}}\left(\sum_{i=0}^{\lceil yn\rceil-1} iF(i,n) + \sum_{i=\lceil yn\rceil}^{n} iF(i,n)\right)\\ &\leqslant \frac{1}{nk^{ln}}\left(yn\sum_{i=0}^{n} F(i,n) + n\sum_{i=\lceil yn\rceil}^{n} H(i,n)\right) = y + o(1)\end{aligned}$$

□

An l-tuple of sequences $(u_1,\ldots,u_l) \in \Pi$ is a match if $u_1,\ldots,u_l$ have the same last symbol. A collation of order i is a sequence of i matches followed by an l-tuple. We say that

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

collation $\mathbf{u}_1, \ldots, \mathbf{u}_{i+1}$ generates the l-tuple $\mathrm{cat}(\mathbf{u}_1, \ldots, \mathbf{u}_{i+1})$. For example, collation

$$\begin{pmatrix} 10 \\ 110 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 01 \\ 01 \end{pmatrix} \begin{pmatrix} 10 \\ 0 \\ 10 \end{pmatrix} \begin{pmatrix} 1 \\ \lambda \\ 10 \end{pmatrix}$$

of order 3 generates the triple

$$\begin{pmatrix} 101101 \\ 110010 \\ 0011010 \end{pmatrix}.$$

Let $H(i,m)$ be the number of all collations of order i generating an l-tuple of total length m. For every l-tuple $\mathbf{u}$ with a longest common subsequence of length i there is at least one collation of order i generating $\mathbf{u}$, and therefore $F(i,n) \leqslant H(i,ln)$. To get upper bounds for $H(i,ln)$ we shall follow the framework given in [5].

Theorem 2.2. *Let* $h(z) = \frac{kz^l}{(1-kz)^l}$. *Let real number* z_0 *be such that* $h(z_0) < 1$. *Then*

$$\gamma_k^{(l)} \leqslant \frac{\log(k^l z_0^l)}{\log h(z_0)}.$$

Proof. To get an upper bound for $F(i,n)$ we have to evaluate $H(i,m)$. We shall use generating functions to achieve this. There are k possibilities for a matching symbol and its contribution is kz^l. The contribution of every sequence, not counting the matching symbol, is $\frac{1}{1-kz}$, and therefore the generating function for the number of matches is

$$h(z) = \frac{kz^l}{(1-kz)^l}.$$

The generating function $\sum_{m=0}^{\infty} H(i,m)z^m$ for the number of all collations of order i is then given by $s(z)(h(z))^i$, where $s(z)$ is independent of i. To bound coefficient A_m of a generating function $a(x) = \sum_{j=0}^{\infty} A_j x^j$, $A_j \geqslant 0$, we can use the following inequality:

$$A_m \leqslant \inf_x \left\{ \frac{a(x)}{x^m} : a(x) \text{ converges} \right\}. \tag{2.1}$$

Let Z be a set of all $z \in (0,1)$ such that $h(z) < 1$. Now we can bound $H(i,n)$:

$$\begin{aligned} \sum_{i=\lceil yn \rceil}^{n} H(i,ln) &\leqslant \sum_{i=\lceil yn \rceil}^{n} \inf_{z \in Z} \frac{s(z)(h(z))^i}{z^{ln}} \\ &\leqslant \inf_{z \in Z} \sum_{i=\lceil yn \rceil}^{n} \frac{s(z)(h(z))^i}{z^{ln}} \\ &\leqslant \inf_{z \in Z} ns(z) \left(\frac{(h(z))^y}{z^l} \right)^n \\ &\leqslant ns(z_0) \left(\frac{(h(z_0))^y}{z_0^l} \right)^n. \end{aligned}$$

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

For every $y > \frac{\log(k^l z_0^l)}{\log h(z_0)}$ we have $\frac{(h(z_0))^y}{z_0^l} < k^l$ and therefore $\sum_{i=\lceil yn \rceil}^{n} H(i, ln) = o(k^{ln})$. Using Lemma 2.1 we can conclude that $\gamma_k^{(l)} \leqslant y$. □

For $l = 2$ the upper bound from Theorem 2.2 corresponds to the upper bound of Chvátal and Sankoff [4]. Now we analyse the speed of convergence of the upper bounds (2.2).

Corollary 2.1. *For every $l \geqslant 2$ we have*

$$\limsup_{k\to\infty} k^{1-1/l}\gamma_k^{(l)} \leqslant e.$$

Proof. Let $z_0 = \frac{k^{1-1/l}-e}{k^{2-1/l}}$, then $kz_0 = 1 - ek^{1/l-1}$. For $h(z_0)$ we have

$$h(z_0) = \left(\frac{k^{1/l-1}kz_0}{1-kz_0}\right)^l = \left(\frac{k^{1/l-1}(1-ek^{1/l-1})}{ek^{1/l-1}}\right)^l = \frac{(1-ek^{1/l-1})^l}{e^l}.$$

From (2.2) we have

$$k^{1-1/l}\gamma_k^{(l)} \leqslant k^{1-1/l}\frac{\log(1-ek^{1/l-1})}{\log(1-ek^{1/l-1}) - \log e}.$$

Since $\lim_{x\to 0^+} \frac{\log(1+ax)}{x} = a \log e$, we get

$$\begin{aligned}\limsup_{k\to\infty} k^{1-1/l}\gamma_k^{(l)} &\leqslant \limsup_{k\to\infty}\left(\frac{\log(1-ek^{1/l-1})}{k^{1/l-1}}\cdot\frac{-1}{\log e - \log(1-ek^{1/l-1})}\right)\\ &\leqslant (-e\log e)(-1/\log e) = e.\end{aligned}$$

□

In [13] Steele conjectures that $\gamma_k^{(l)} = (\gamma_k^{(2)})^{l-1}$ for $l > 2$. As we can see from Corollary 2.1 and Theorem 2.1 this is not the case. Sankoff and Mainville [11] conjecture that $\lim_{k\to\infty} \sqrt{k}\gamma_k^{(2)} = 2$. A natural extension of their conjecture is that $\lim_{k\to\infty} k^{1-1/l}\gamma_k^{(l)} = 2$. Jiang and Li [9] have considered the case when $l = n$. Using Kolmogorov complexity, they have shown that $|\mathbf{EL}_n^{(n)} - \frac{n}{k}| < n^{\frac{1}{2}+\varepsilon}$ for every $\varepsilon > 0$ and large enough n. For any constant l, their algorithm gives the lower bound $\gamma_k^{(l)} \geqslant \frac{1}{k}$. A slight modification of the proof of Corollary 2.1 gives us $\limsup_{l\to\infty} k\gamma_k^{(l)} \leqslant e$.

3. Shortest common supersequences

In this section we shall consider a 'dual' notion, namely shortest common supersequences. Sequence u is a *supersequence* of v if v is a subsequence of u. We say that $u \in \Sigma^*$ is a *common supersequence* of $v_1, \ldots, v_l$ if, for every $i = 1, \ldots, l$, sequence u is a supersequence of v_i. We say that u is a *shortest common supersequence* if u is a common supersequence of minimal possible length. The length of a shortest common supersequence is denoted by $\mathbf{S}(v_1, \ldots, v_l)$. Creating shortest common supersequences is natural when merging sequences. This is useful for some types of compression [14] or for efficient planning [8]. Longest common subsequences and shortest common supersequences of two sequences are dual, and $\mathbf{S}(u, v) + \mathbf{L}(u, v) = |u| + |v|$. However, if we have more than two sequences and we

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

know only their lengths and the length of their longest common subsequence, then we are not able to compute the length of a shortest common supersequence.

We can define the expected length $\mathbf{ES}_n^{(l)}$ of a shortest common supersequence of l sequences of length n analogously to $\mathbf{EL}_n^{(l)}$. Properties of $\mathbf{ES}_n^{(l)}$ are similar and the constants $\sigma_k^{(l)} = \lim_{n\to\infty} \frac{\mathbf{ES}_n^{(l)}}{n}$ exist. For $l = 2$ we have $\sigma_k^{(2)} = 2 - \gamma_k^{(2)}$. To obtain bounds on $\sigma_k^{(l)}$, $l > 2$, we shall use methods similar to the methods from the previous section.

To obtain upper bounds for the expected length of a shortest common supersequence of l sequences we shall consider algorithms that produce a common supersequence of its input. The expected length of a common supersequence produced by the algorithm will be an upper bound for $\sigma_k^{(l)}$.

We can use a tournament-style algorithm to produce a common supersequence of l sequences $u_1, \ldots, u_l$. In the first round u_1 and u_2 produce a common supersequence $v_{1,1}$, while u_3 and u_4 produce a common supersequence $v_{1,2}$, and so on. In the second round we repeat the process with $v_{1,1}, \ldots, v_{1,\lceil l/2\rceil}$. After $\lceil \log l \rceil$ steps we get a 'winner' $v_{\lceil \log l\rceil,1}$ – a common supersequence of $u_1, \ldots, u_l$.

Timkovskii [15] has asked a question about how good an approximation is the tournament algorithm. Bradford and Jenkyns [3] have found an example of three sequences of length 12 over the alphabet of size nine such that no tournament-like algorithm can compute their shortest common supersequence.

We describe a different algorithm that is easy to analyse. Let $T^{(l)}(i)$ be the set of symbols that appear at the ith positions, $T^{(l)}(i) = \{u_1[i], \ldots, u_l[i]\} \subseteq \Sigma$. Let v_i, $i = 1, \ldots, n$ be a sequence of length $\mathbf{T}^{(l)}(i) = |T^{(l)}(i)|$ consisting of all symbols from $T^{(l)}(i)$. The algorithm in the ith step, $i = 1, \ldots, n$, finds $T^{(l)}(i)$ and appends v_i to the output. The sequence $v_1, \ldots, v_n$ produced by the algorithm is a common supersequence of $u_1, \ldots, u_l$ and $\mathbf{S}(u_1, \ldots, u_l) = \sum_{i=1}^{n} \mathbf{T}^{(l)}(i)$. The probability that $\mathbf{T}^{(l)}$ equals m is $\binom{k}{m} \sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} \left(\frac{j}{k}\right)^l$ [7, Section IV.2]. Since $\mathbf{ES}_n^{(l)} = n\mathbf{ET}^{(l)}$, we get the following upper bound.

Theorem 3.1. *For every $l \geqslant 3$ and $k \geqslant 2$ we have*

$$\sigma_k^{(l)} \leqslant \sum_{m=1}^{k} m \binom{k}{m} \sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} \left(\frac{j}{k}\right)^l. \qquad \square$$

To get lower bounds we shall use the following lemma corresponding to Lemma 2.1.

Lemma 3.1. *Let $F(i,n) = |\{\mathbf{u} \in \Pi_n^l : \mathbf{S}(\mathbf{u}) = i\}|$ and let $H(i,n)$ be any upper bound for $F(i,n)$. If y is such that $\sum_{i=0}^{\lfloor yn\rfloor} H(i,n) = o(k^{ln})$ then $y \leqslant \sigma_k^{(l)}$.* $\square$

For common supersequences the role of matches will be played by 'placements' and a 'distribution' will be the notion analogous to a collation.

An l-tuple $u_1, \ldots, u_l$ is a *placement* if $u_1, \ldots, u_l \in \{\lambda, a\}$ for some $a \in \Sigma$ and not all of $u_1, \ldots, u_l$ are empty sequences. A sequence of i placements will be called a *distribution* of order i. The distribution $\mathbf{d}_1, \ldots, \mathbf{d}_i$ *generates* l-tuple $\mathbf{u}$ if $\mathrm{cat}(\mathbf{d}_1, \ldots, \mathbf{d}_i) = \mathbf{u}$.

Let $\mathscr{D}(i)$ be the set of all distributions of order i. Let $\mathscr{G}(m)$ be the set of all distributions

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

Table 1 Upper and lower bounds for the expected length of a shortest common supersequence of l sequences over an alphabet of size k

k	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=8$	$l=16$
2	1.50000	1.75000	1.87500	1.93750	1.96875	1.99219	1.99997
	1.09321	1.14253	1.17267	1.19287	1.20731	1.22656	1.25822
3	1.66667	2.11112	2.40741	2.60494	2.73663	2.88295	2.99544
	1.16377	1.25982	1.32366	1.36923	1.40336	1.45107	1.53569
4	1.75000	2.31250	2.73438	3.05079	3.28809	3.59955	3.95991
	1.21514	1.34913	1.44278	1.51244	1.56642	1.64476	1.79311
5	1.80000	2.44000	2.95200	3.36160	3.68928	4.16114	4.85927
	1.25447	1.41951	1.53883	1.63017	1.70270	1.81106	2.02791
6	1.83334	2.52778	3.10649	3.58874	3.99062	4.60460	5.67548
	1.28583	1.47679	1.61830	1.72893	1.81841	1.95512	2.24164
7	1.85715	2.59184	3.22158	3.76135	4.22402	4.96050	6.40578
	1.31162	1.52463	1.68549	1.81331	1.91818	2.08131	2.43661
8	1.87500	2.64063	3.31055	3.89673	4.40964	5.25113	7.05547
	1.33333	1.56541	1.74333	1.88656	2.00542	2.19301	2.61512
9	1.88889	2.67903	3.38135	4.00564	4.56057	5.49231	7.63290
	1.35196	1.60076	1.79387	1.95098	2.08260	2.29285	2.77923
10	1.90000	2.71000	3.43900	4.09510	4.68559	5.69533	8.14698
	1.36819	1.63183	1.83857	2.00828	2.15158	2.38282	2.93073
11	1.90910	2.73554	3.48686	4.16987	4.79079	5.86842	8.60608
	1.38251	1.65943	1.87851	2.05972	2.21377	2.46452	3.07115
12	1.91667	2.75695	3.52720	4.23327	4.88050	6.01764	9.01762
	1.39527	1.68418	1.91451	2.10627	2.27025	2.53919	3.20180
13	1.92308	2.77515	3.56168	4.28771	4.95788	6.14755	9.38799
	1.40675	1.70657	1.94721	2.14871	2.32189	2.60782	3.32378
14	1.92858	2.79082	3.59148	4.33494	5.02531	6.26162	9.72267
	1.41715	1.72696	1.97710	2.18762	2.36938	2.67122	3.43804
15	1.93334	2.80445	3.61749	4.37632	5.08457	6.36256	10.0263
	1.42664	1.74564	2.00458	2.22349	2.41326	2.73007	3.54540

generating an l-tuple of total length m. Since for every l-tuple $\mathbf{u} \in \Pi_n^l$ there exists a distribution of order $\mathbf{S}(\mathbf{u})$ that generates $\mathbf{u}$, we have

$$F(i,n) \leqslant |\mathscr{G}(ln) \cap \mathscr{D}(i)| = G(i,n).$$

Every placement of total length m corresponds to a nonempty subset of $\{1,\ldots,l\}$ of size m, and thus the generating function for the number of placements is $k((1+z)^l - 1)$. The generating function $\sum_{j=0}^{\infty} |\mathscr{G}(j) \cap \mathscr{D}(i)| x^j$ for the number of distributions in the $\mathscr{D}(i)$ is then $k^i((1+z)^l-1)^i$. From (2.1) for $z_0 > 2^{1/l} - 1$ and $y < \frac{l \log(z_0 k)}{\log(k(1+z_0)^l - k)}$ we have

$$\begin{aligned}
\sum_{i=0}^{\lfloor yn \rfloor} G(i,n) &\leqslant \sum_{i=0}^{\lfloor yn \rfloor} \inf_{z \in Z} \frac{k^i((1+z)^l-1)^i}{z^{ln}} \leqslant \inf_{z \in Z} \sum_{i=0}^{\lfloor yn \rfloor} \frac{k^i((1+z)^l-1)^i}{z^{ln}} \\
&\leqslant \sum_{i=0}^{\lfloor yn \rfloor} \frac{k^i((1+z_0)^l-1)^i}{z_0^{ln}} \leqslant yn \left(\frac{k^y((1+z_0)^l-1)^y}{z_0^l} \right)^n = o(k^{ln}).
\end{aligned}$$

Thus we have just proved the following theorem.

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

Theorem 3.2. *For every $l \geqslant 2$ and $k \geqslant 2$ we have*

$$\sigma_k^{(l)} \geqslant \frac{l \log z_0 k}{\log k((1+z_0)^l - 1)},$$

where real number z_0 is such that $z_0 > 2^{1/l} - 1$.

Upper and lower bounds for $k = 2, \ldots, 15$ and $l = 2, 3, 4, 5, 6, 8, 16$ obtained using Theorems 3.1 and 3.2 can be found in Table 1. Better bounds for $l = 2$ follow from bounds for longest common subsequences using duality properties. Unfortunately, lower bounds from Theorem 3.2 do not appear to match upper bounds from Theorem 3.1.

Acknowledgement

The author would like to thank Professor M. Paterson for his help, many valuable comments and suggestions. Also, comments of an anonymous referee led to improvements to the paper.

References

[1] Aho, A. V. (1990) Algorithms for finding patterns in strings. In *Handbook of Theoretical Computer Science* (J. van Leeuwen, ed.), Elsevier, Amsterdam, chapter 5, pp. 255–300.
[2] Baeza-Yates, R. A., Gavaldà, R. and Navarro, G. (1996) Bounding the expected length of longest common subsequences and forests. In *Third South American Workshop on String Processing, Proceedings* (N. Ziviani, R. Baeza-Yates, and K. Guimarães, eds), Carleton University Press, pp. 1–15.
[3] Bradford, J. H. and Jenkyns, T. A. (1991) On the inadequacy of tournament algorithms for the *N*-SCS problem. *Inform. Process. Letters* **38** 169–171.
[4] Chvátal, V. and Sankoff, D. (1975) Longest common subsequence of two random sequences. *J. Appl. Probab.* **12** 306–315.
[5] Dančík, V. and Paterson, M. (1995) Upper bounds for the expected length of a longest common subsequence of two binary sequences. *Random Struct. Alg.* **6** 449–458.
[6] Deken, J. G. (1979) Some limit results for longest common subsequences. *Discrete Math* **26** 17–31.
[7] Feller, W. (1968) *An Introduction to Probability Theory and its Applications*, Vol. I, third edition, Wiley, New York.
[8] Foulser, D. E., Li, M. and Yang, Q. (1992) Theory and algorithms for plan merging. *Artificial Intelligence* **57** 143–181.
[9] Jiang, T. and Li, M. (1995) On the approximation of shortest common supersequences and longest common subsequences. *SIAM J. Comput.* **24** 1122–1139.
[10] Paterson, M. and Dančík, V. (1994) Longest common subsequences. In *19th International Symposium Mathematical Foundations of Computer Science, Proceedings* (I. Prívara, B. Rovan, and P. Ružička, eds), Vol. 841 of *Lecture Notes in Computer Science*, Springer, pp. 127–142.
[11] Sankoff, D. and Kruskal, J. B. (1983) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, MA.
[12] Simon, I. (1989) Sequence comparison: Some theory and some practice. In *Electronic Dictionaries and Automata in Computational Linguistics, Proceedings* (M. Gross and D. Perrin, eds), Vol. 377 of *Lecture Notes in Computer Science*, Springer, pp. 79–92.
[13] Steele, J. M. (1986) An Efron–Stein inequality for nonsymmetric statistics. *Ann. Statist.* **14** 753–758.

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press

[14] Storer, J. A. (1988) *Data Compression: Methods and Theory*, Computer Science Press, Rockville, Maryland.
[15] Timkovskii, V. G. (1990) Complexity of common subsequence and supersequence problems and related problems. *Cybernetics* **25** 565–580.
[16] Waterman, M. S. (1995) *Introduction to Computational Biology: Maps, Sequences and Genomes*, Chapman & Hall, London.

https://doi.org/10.1017/S096354839800368X Published online by Cambridge University Press