Bounds on the Number of Longest Common Subsequences

Ronald I. Greenberg
Loyola University of Chicago
rig@cs.luc.edu

- Will define LCS soon for those unfamiliar.
- For those familiar, a few words of where we'll be going.
  - Attack questions like "What is the maximum possible number of different LCSs in two input strings of t total characters?"
  - Has ramifications for running time as a function of input size for algorithms that generate all LCSs.

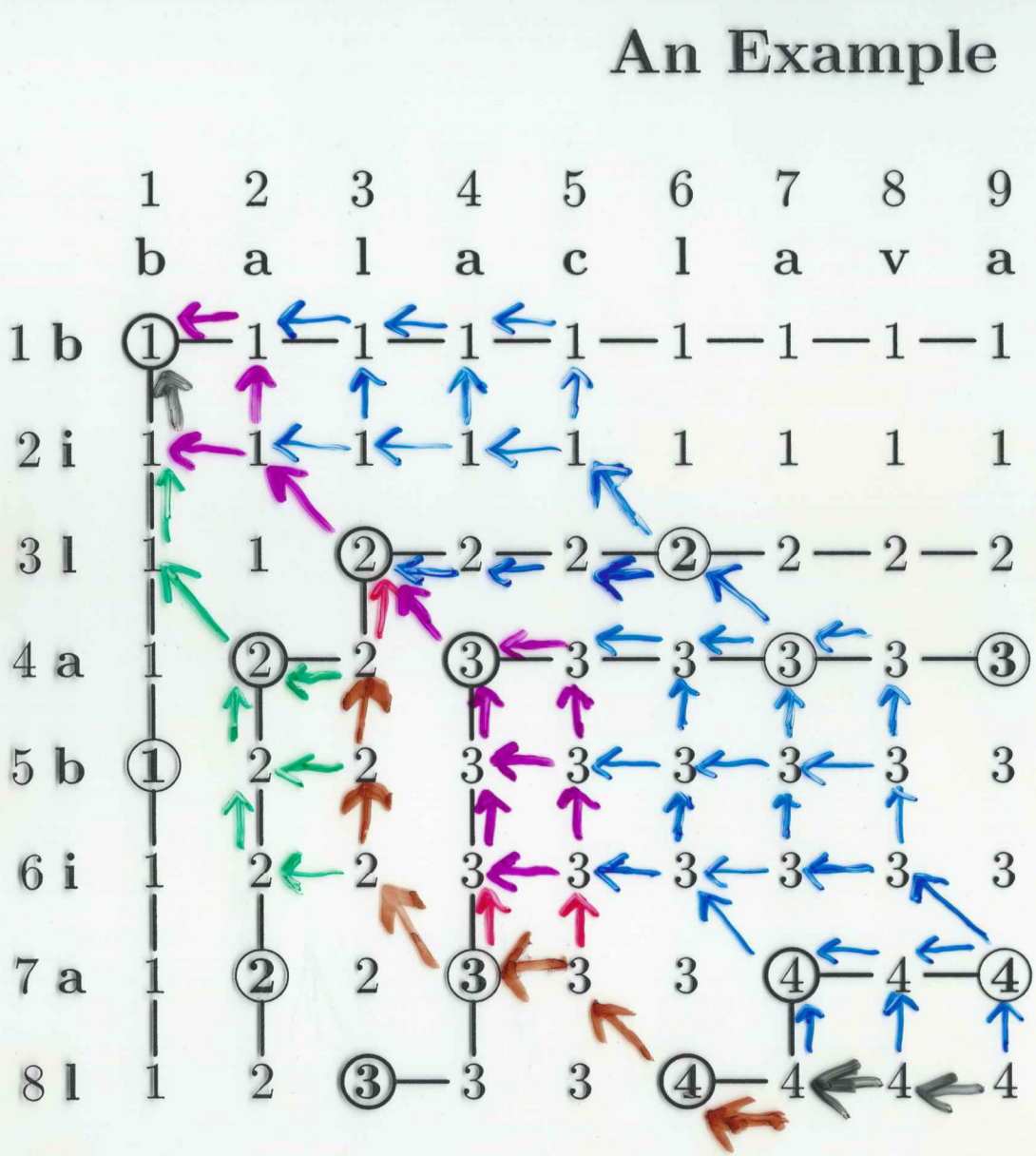
Outline

- Background and Terminologies
- Summary of Results
- Bounding the Maximum Number of Distinct LCSs
- The Maximum Number of LCS Embeddings
- How Inefficient is it to Generate All LCSs Naively?
- Conclusion

Terminologies

- Input: Two sequences $A = a_1 a_2 \ldots a_m$ and $B = b_1 b_2 \ldots b_n$ $(m \leq n)$ over an alphabet Σ.
- *common subsequence* of A and B: a sequence that can be obtained from either A or B by deleting some symbols.
- *longest common subsequence (LCS)*: a common subsequence of greatest possible length.
- A pair of sequences may have many different LCSs. In addition, a single LCS may have many different *embeddings*, i.e., positions in the two strings to which the characters of the LCS correspond.

An Example



LCS embeddings

$b_1^1 a_2^4 a_4^7 l_6^8$

$b_1^1 l_3^3 a_4^7 l_6^8$

$b_1^1 l_3^3 a_4^4 l_6^8$

$b_1^1 l_3^3 a_4^4 a_7^7$

$b_1^1 l_3^3 a_4^4 a_9^7$

$b_1^1 l_3^3 a_7^4 a_9^7$

$b_1^1 l_6^3 a_7^4 a_9^7$

distinct LCSs

baal

blal

blaa

Ink in different colors to show embeddings of different LCSs.

- Seven different embeddings and three distinct LCSs for the strings $A = \texttt{bilabial}$ and $B = \texttt{balaclava}$.
- The matches are circled and contours indicating rank boundaries are shown by connecting lines.
- In the matrix, the $[i, j]$ entry shows the rank $L[i, j]$ as per

$$L[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ L[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } a_i = b_j \\ \max\{L[i-1, j], L[i, j-1]\} & \text{otherwise} \end{cases}$$

- For the purposes of this talk, ignore the uses of bold for certain circles and ranks.
- The naive method of generating all LCSs for this pair of strings would produce a list of length 100, because there would be many duplications. (From following all paths indicated by arrows; more on this later.)

Motivation

- Much prior work on finding one LCS. (Applications such as DNA sequence comparison, genome mapping, UNIX `diff`.)
  - Simple $O(mn)$ dynamic programming solution.
  - Other results improving time and/or space usage.
- Also works on finding all LCSs (e.g., Greenberg 2002, Rick 2000, Gotoh 1990, Altschul & Erickson 1986).
  - Time proportional to output size (plus preprocessing of $O(mn)$ or less).
  - No nontrivial bound on time as a function of *input* size.

Partial Summary of Results

Let $D(t)$ be the maximum possible number of distinct LCSs and $E(t)$ be the maximum possible number of LCS embeddings, each for two input sequences of total length t. For simplicity, assume $6 \mid t$ here.

- $1.2^t < 3^{t/6} \leq D(t) \leq 4^{t/5} < 1.32^t$.
- $D(t) = 3^{t/6}$ if no repeated characters in either input seq.
- $E(t) = \begin{pmatrix} \left\lfloor \left(5t + 3 + \sqrt{5(t+1)^2 + 4}\right)/10 \right\rfloor \\ \left\lceil \left(5t - 3 - \sqrt{5(t+1)^2 + 4}\right)/10 \right\rceil \end{pmatrix}$.
- $\lim_{t\to\infty} E(t) \approx .965(1.62)^t/\sqrt{t}$.
- The time to naively generate all LCS embeddings or all LCSs may exceed the output size by a factor of $\Theta(2^t/\sqrt{t})$.

- Everything works with $6 \not| t$, but the $D(t)$ expressions get a little more complicated.

- The limit result doesn't really merit listing as a main result in its own right since it follows from the line above by using Stirling's approximation to the factorial, but this gives an easier to digets notion of the magnitude of $E(t)$.

- There are some other results as well, e.g., determination of the maximum possible number of LCS embeddings in two input strings that each contain n characters. Smaller than maximum possible number of embeddings in two input strings of total length $2n$.

The Maximum Number of Distinct LCSs — Lower Bound

Theorem 1 *$D(t) > 3^{t/6} > 1.2^t$ for $t \mid 6$.*

Proof. Let the inputs be of the forms

`abcdefghijkl...`

and

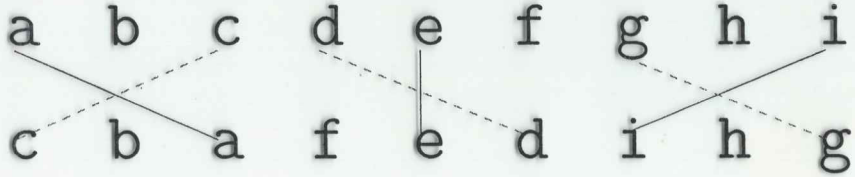
`cbafedihglkj...`

■

The Maximum Number of Distinct LCSs — Upper Bound

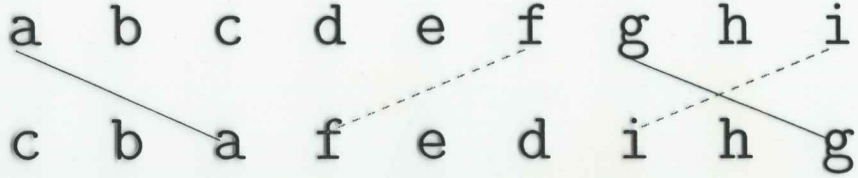
Lemma 2 *Consider any embedding of each of two LCSs starting with different characters. The embeddings of the initial characters of these two LCSs must "cross".*

Ex.: `aei` & `cdg` in



Proof. If the embeddings of the initial characters didn't cross, one of the initial characters could be prepended to the other sequence to yield a longer common subseq., e.g.:

`ag` & `fi` in



■

The Maximum Number of Distinct LCSs — Upper Bound

Theorem 3 $D(t) \leq 4^{t/5} < 1.32^t$.

Proof. Follows by induction once we show $D(t) \leq kD(t-(k+1))$ for some k. The inequality follows from letting k be the number of choices for the first character in an LCS of the input strings. Once the first character is chosen, Lemma 2 tells us that $k+1$ characters of the input strings are removed from consideration for construction of the remainder of the LCS.

Ink in illustration of k crossing initial characters.

The Maximum Number of Distinct LCSs — No Repeated Characters

Theorem 4 *$D(t) = 3^{t/6}$ for $t \mid 6$ if there are no repeated characters in either input sequence.*

Proof. Lower bound done earlier.

Upper bound like previous proof, except that when we make one of k choices for the first character of the LCS, we eliminate $2k$ characters from possible use in the remainder of the LCS, so $D(t) \leq kD(t-2k)$ for some k.

Can eliminate all characters that don't appear in both sequences from the start. Then when we make one of the k choices for the initial character we eliminate from consideration both copies of each of the k characters involved.

Again, draw.

The Maximum Number of LCS Embeddings – Relation to Maximum Number of Embeddings of a Single LCS

- *Claim*: The total number of LCS embeddings is maximized when there is just one LCS.
- Rigorous argument deferred.
- Based on claim, focus henceforth on maximum number of embeddings of a single LCS.

The Maximum Number of Embeddings of an LCS of Specified Length

Lemma 5 *The maximum possible number of embeddings $E(n, m, l)$ of a single LCS of length l in two input sequences of lengths m and n is*

$$E(n, m, l) = \max_{y \le l} \binom{m-y}{l-y} \binom{n+y-l}{y} .$$

Proof. Lower bound from embeddings of $\mathtt{a}^{l-y}\mathtt{b}^{y}$ in $\mathtt{a}^{m-y}\mathtt{b}^{y}$ and $\mathtt{a}^{l-y}\mathtt{b}^{n+y-l}$. Upper bound follows since each character of any LCS must have a fixed embedding in at least one of the two input strings; y is no. of characters with a fixed embedding in the first input string.

The Maximum Number of Embeddings of an LCS of Specified Length — Choice of y

Lemma 6 *The value of y maximizing $E(n, m, l)$ is* $y^* = \left\lceil \frac{l(n-l)+l-m}{m+n-2l} \right\rceil$.

Lemma 7 *The maximum possible number of embeddings $E(n, n, l)$ of a single LCS of length l in two input sequences of length n is*

$$E(n, n, l) = \binom{n - \lfloor l/2 \rfloor}{n-l} \binom{n - \lceil l/2 \rceil}{n-l} .$$

The Maximum Number of Embeddings of an LCS

Theorem 8 *The maximum possible number of embeddings of a single LCS in two input sequences of length n is*

$$\binom{\left\lfloor \frac{1}{2}\left(1+1/\sqrt{5}\right)(n+1)\right\rfloor}{\left\lceil \left(5n-1-\sqrt{5(n+1)^2-4}\right)/10\right\rceil} * \binom{\left\lfloor \left(5n+1+\sqrt{5(n+1)^2-4}\right)/10\right\rfloor}{\left\lfloor \frac{1}{2}\left(1-1/\sqrt{5}\right)(n+1)\right\rfloor}.$$

The Maximum Number of Embeddings of an LCS - Limit

Corollary 9 *The limit as n goes to infinity of the maximum possible number of embeddings of a single LCS in two input sequences of length n is*

$$\frac{\phi^2\sqrt{5}}{2\pi}\left(\phi^2\right)^n \Big/ n \approx .932(2.62)^n/n \ ,$$

where $\phi = (1+\sqrt{5})/2$ (the golden ratio).

Proof. Use Stirling's approximation to the factorial:

$$n! = \sqrt{2\pi n}(n/e)^n(1+\Theta(1/n)) \ .$$

For this form of Stirling's approximation, see Knuth1973 p. 111.

The Maximum Number of Embeddings of an LCS — Only Total No. of Input Characters Constrained

Lemma 10 *With LCS length l, $E(t) = \binom{t-l}{l}$.*

Theorem 11

$$E(t) = \begin{pmatrix} \left\lfloor \left(5t+3+\sqrt{5(t+1)^2+4}\right)/10 \right\rfloor \\ \left\lceil \left(5t-3-\sqrt{5(t+1)^2+4}\right)/10 \right\rceil \end{pmatrix} .$$

Corollary 12

$\lim_{t\to\infty} E(t) = \phi\sqrt{\sqrt{5}/(2\pi)}\,\phi^t/\sqrt{t} \approx .965(1.62)^t/\sqrt{t}$.

Initial lemma here follows from the earlier lemma for inputs of length m and n.

Final corollary with $t = 2n$ does not make $E(t)$ as small as on previous slide. Difference is $\approx .682/\sqrt{n}$ versus $.932/n$, i.e., bigger by a factor of about $.732\sqrt{n}$.

How Bad is it to Generate All LCSs Naively?

The method:

- Compute $\forall\ i$ & j "bottom-up" ranks $L[i,j] =$

$$\begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ L[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } a_i = b_j \\ \max\{L[i-1, j], L[i, j-1]\} & \text{otherwise} \end{cases}$$

- Backtrace from position $[m, n]$. At each stage, if $[i, j]$ is a match, can add a character to LCS and go to $[i-1, j-1]$. Also, an option to not add a character to LCS and move to $[i-1, j]$ or $[i, j-1]$ if rank there equals $L[i, j]$.

Flip back to bilabial/balaclava table.

How Bad is it to Generate All LCSs Naively?

Theorem 13 *The naive method of generating all LCS embeddings (or all LCSs) may require time exceeding the output size by a factor of* $\Theta(\binom{n+m}{m})$ *in the worst case.*

Proof sketch: For lower bound, can just consider a pair of sequences with no matches, but this comes from operating extremely naively. Even if we reduce the naivety by printing outputs whenever we hit a node of rank 0, we can still construct examples with $\Omega(\binom{n+m}{m})$ overhead. For upper bound, a proof by induction on $n + m$; additional details too much to include here.

Some Recap & Conclusions

- $1.2^t < D(t) < 1.32^t$. Would be nice to close the gap.
- $\lim_{t\to\infty} E(t) \approx .965(1.62)^t/\sqrt{t}$.
- Time to naively generate all LCS embeddings or all LCSs may exceed the output size by a factor of $\Theta(2^t/\sqrt{t})$.
- Any algorithm that finds all distinct LCSs by generating all embeddings and removing duplicates has a worst-case inefficiency factor exponential in the input size.
- Using the naive method to generate all embeddings and removing duplicate embeddings or LCSs is inefficient by an even larger exponential factor in the worst case.
- Some results have been obtained in context of specified input lengths m and n; would be nice to extend all.