

CS161 Programming Project

Jason van der Merwe – Jasonvdm@stanford.edu – 05719899

Bridge Eimon – beimon@stanford.edu – 05716372

May 26, 2014

Theory

(a)

- (b) We will show that this modified algorithm still works because $\forall i, j \exists k \text{ LCS}(\text{cut}(A, i), \text{cut}(B, j)) \leq \text{LCS}(\text{cut}(A, k), \text{cut}(B, 0))$.

For $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j))$, we have three options. $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j))$ can be 0, 1, or greater than one. In the first case, if the longest common subsequence is 0, there are no common characters in the two strings, so $\forall k \text{ LCS}(\text{cut}(A, k), \text{cut}(B, 0)) = 0$. In the second case where the longest common subsequence between $\text{cut}(A, i)$ and $\text{cut}(B, j)$ is 1, there is at least one character in common between the two strings, so $\forall k \text{ LCS}(\text{cut}(A, k), \text{cut}(B, 0)) \geq 1$ since at least the one character the two strings share will be a common subsequence, and there could be more depending on the cut (i.e. the longest common subsequence between AB and BA is one, but the longest common subsequence between AB and AB is two).

If $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j)) > 1$, then there exist a sequence $\langle X_1, \dots, X_s \rangle$ in both $\text{cut}(A, i)$ and $\text{cut}(B, j)$ which is the longest common subsequence. If $\text{cut}(B, j)$ preserves the order of $\langle X_1, \dots, X_s \rangle$, then we know that $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j)) \leq \text{LCS}(\text{cut}(A, i), \text{cut}(B, 0))$ since the longest common subsequence $\langle X_1, \dots, X_s \rangle$ is in both $\text{cut}(A, i)$ and $\text{cut}(B, 0)$ in the same order. If $\text{cut}(B, j)$ does not preserve the order of $\langle X_1, \dots, X_s \rangle$, then since it is a different cut of B , we must have that $\text{cut}(B, 0)$ has the elements $\langle X_{\text{cut}+1}, \dots, X_s, X_1, \dots, X_{\text{cut}} \rangle$. In order to get a common subsequence in a cut of A of at least the size of $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j))$, we must ensure that elements $\langle X_1, \dots, X_s \rangle$ are also in the order $\langle X_{\text{cut}+1}, \dots, X_s, X_1, \dots, X_{\text{cut}} \rangle$. We can do this by finding the index of X_{cut} in $\text{cut}(A, i)$ and cutting there to ensure that $X_{\text{cut}+1}$ is the first element in the common subsequence in the new array. Thus, in the array $\text{cut}(\text{cut}(A, i), A.\text{index}(X_{\text{cut}}))$ the key elements are in order $\langle X_{\text{cut}+1}, \dots, X_s, X_1, \dots, X_{\text{cut}} \rangle$. We can rewrite $\text{cut}(\text{cut}(A, i), A.\text{index}(X_{\text{cut}})) = \text{cut}(A, i + C.\text{index}(X_{\text{cut}}))$ where $C = \text{cut}(A, i)$. Thus, there is a value for k such that $\langle X_{\text{cut}+1}, \dots, X_s, X_1, \dots, X_{\text{cut}} \rangle$ is a common subsequence, so $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j)) \leq \text{LCS}(\text{cut}(A, i + C.\text{index}(X_{\text{cut}})), \text{cut}(B, 0))$ since all the elements in $\text{LCS}(\text{cut}(A, i), \text{cut}(B, j))$ are in the same order in $\text{cut}(A, i + C.\text{index}(X_{\text{cut}}))$ and $\text{cut}(B, 0)$.

Lastly, we just need to prove that $\forall i, j \exists k \text{ } LCS(\textit{cut}(A, i), \textit{cut}(B, j)) \leq LCS(\textit{cut}(A, k), \textit{cut}(B, 0))$ is sufficient to prove that it is enough just to compare $LCS(\textit{cut}(A, k), \textit{cut}(B, 0)) \forall k$. Since we are only looking for the largest common subsequence, and we know that for every pair of $\textit{cut}(A, i)$ and $\textit{cut}(B, j)$ there is a pair $\textit{cut}(A, k)$ and $\textit{cut}(B, 0)$ which has a common subsequence of the same length or greater length. Thus, the longest common subsequence can be found from a $\textit{cut}(B, 0)$, so all we need to compare is $LCS(\textit{cut}(A, k), \textit{cut}(B, 0)) \forall k$.