Multi Suffix Tree Dictionary in Optimal Space

Stefan Edelkamp

Institut für Informatik
Albert-Ludwigs-Universität
Am Flughafen 17, D-79110 Freiburg

edelkamp@informatik.uni-freiburg.de

The talk is about a dictionary data structure D for matching multiple pattern. If the input alphabet Σ is bounded and the patterns m in D are considered to be mutually substring free an optimal worst case bound of $O(|m|)$ for both insertion and deletion of a pattern $m \in \Sigma^*$ is achieved. Since the strings in D are not substrings of each other this is a special case of the *dynamic dictionary matching problem* (*DDMP*).

The structure D is realised as a multi suffix tree. Let M be the set of strings stored in the dictionary D resulting from an arbitrary sequence of *Insert* and *Delete* operations. The main result is that the space complexity of the augmented multi suffix tree representing D is $O(d)$ with $d = \sum_{m \in M} |m|$. This bound is optimal if we assume that each pattern in the dictionary uses at least linear space and has not been achieved before.

Further on, I present a new incremental *Search* operation to find one substring of x in time $O(|x|)$. In the general case of the *DDMP* I will show that searching all *tocc* (total number of occurences) substrings in x can be performed in almost $O((|x| + tocc) \log d)$ time without destroying the linear efficiency results of *Insert* and *Delete*.

Last but not least, I will talk about the use of the multi suffix tree dictionary in state space search of huge problem spaces that do not fit in the memory. Thus we may assume that the problem space is described implicitely by a bounded number of transitions and that transition sequences can be regarded as strings. The set of strings M can be used as a set of forbidden words, so-called duplicates, in the search. Duplicates are transition sequences that are more expensive than their counterparts. When a substring of the generated path according to a node in the search tree is encountered the tree is pruned since a shorter path has been or will be examined.

Literatur

[1] A. V. Aho and M. J. Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

[2] A. Amir, M. Farach, R. M. Idury, J. A. L. Poutré, and A. Schäffer. Improved dynamic dictionary matching. *Information and Computation*, 119(2):258–282, 1995.

[3] E. M. McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM*, 23(2):262–272, 1976.

[4] L. A. Taylor and R. E. Korf. Pruning duplicate nodes in depth-first search. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 756–761. AAAI Press, 1993.