Algorithms for Massive Data Sets

Student name and id: Alexander R. Johansen (s145706)

Collaborator name(s) and id(s): Jonas (s142957)

Hand-in for week: 8

---

1 Patterns with many different symbols

The $PatternsWithManyDifferentSymbols(P, T, k)$ Problem can be solved in $O(n)$ time using a modification of the K-mismatch algorithm presented at the string matching lecture.

1.1 Analysis of data structure and algorithm

We are given the size of Σ and k to have the relation: $k \leq \Sigma/2$, which can be reformulated as: $2k \leq \Sigma$. This enables the use of the linear time algorithm proposed by Amir et al., 2000.

The algorithm has three stages, pre-processing of the Pattern P, filtering the text T and evaluating interesting positions.

The algorithm only pre-process the first $2k$ different letters in P creating the invariant that all of the preprocessed letters has to refer to the same starting position. As the preprocessed mass of letters only have one occurrence from each letter, filtering through T only takes one comparison at each step, resulting in $O(1) * n = O(n)$ work done at the filtering stage.

Further at the filtering stage "marks" are set at the preprocessed letters respective starting positions, if less than k marks are set at a given starting position it cannot be a starting position. Seeing as if less than k are set, more than k of the original string did not match with the given start position thus more than k mismatches has occurred. So with the invariant of having a min. of k marks at a starting position to make it interesting, and n marks being set. The amount of interesting positions can at max be n/k.

Given the k interesting positions, each position can be evaluated in $O(k)$ time using suffix trees over T and P (which can also be created in linear time). With $O(k)$ time per interesting position and $O(n/k)$ interesting positions the evaluation can be done in:
$O(k) * O(n/k) = O(n)$
Summation
Suffix tree: $O(n)$
Filtering: $O(n)$
Evaluation: $O(n)$
Total: $O(n)$

1.2 Weaker assumption: $k \leq \Sigma$

The placement of interesting positions is based on the invariant of being able to say that more than k-mismatches has occurred at a given position. If $k \leq \Sigma$ then it might not be possible to say that more than than k-mismatches has occurred, thus the evaluation stage will take $O(k) * O(n) = O(nk)$.