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: 9

1 DecodeChar

The $DecodeChar(i)$ Problem can be solved in $O(n)$ space and $O(n)$ preprocessing time by using dynamic programming to store a constant amount of information per SLP rule about the size of the rule. The query time can be solved in $O(h)$ time by traversing the directed acyclic graph of SLP rules using the size of the sub rules to determine which node to visit next.

1.1 Analysis of datastructure

The data structure consists of n nodes in an acyclic graph from the SLP. By finding the size of each node, I will be able to solve the query problem. The size is illustrated with an example:
$X_1 = a$
$X_2 = b$
$X_3 = X_1X_2$
$X_4 = X_2X_3$
$X_1.size = 1$
$X_2.size = 1 = 1$
$X_3.size = 1 + 1 = 2$
$X_4.size = 1 + 2 = 3$
Cost-wise, a constant amount of information(a single integer) will be stored in every node, thus $O(1) * O(n) = O(1 * n) = O(n)$ space.
Further information on whether it is a "right" or "left" node is necessary. E.g. following my example above; $X_4.leftnode = X_2$ and $X_4.rightnode = X_3$.

1.2 Analysis of construction of data

As the graph of the $O(n)$ nodes is directed and acyclic, it is possible using dynamic programming to compute the size of each rule $X_i, i\epsilon 1, .., n$ in the degree of the nodes times the amount. As the SLP rules are defined by either having 2 outgoing (non-terminal) or 1 outgoing (terminal) (Cording, Notes on Compression Schemes), thus the degrees of the nodes can be considered constant. Seeing as the algorithm for preprocessing is dependant on the amount of nodes $O(n)$ times the outgoing degree $O(1)$ the total preprocessing time is thus $O(n)$. On a more specific algorithmic level, the dynamical programming works by computing the length of $X_1 = a$ then storing this value such that write and look-up time is constant, then at the next iteration when computing e.g. from the example; $X_3 = X_1X_2$ the value for X_1 and X_2 can be extracted in constant time.(in a more generic statement, $X_n = X_{n-1}X_{n-2}$ is a constant operation when the dynamic programming algorithm has reached $i = n, i\epsilon 1, .., n$.)

1.3 Analysis of supporting the query

As the top rule contains every letter in the string, and the left and right child node contains a non-intersecting prefix and suffix of that string. It is possible to decide which child node the char being searched for is placed. e.g. if $X_n = X_{n-1}X_{n-2}$ and $i < X_{n-1}.size$ then we know that i needs to be inside X_{n-1}. This approach is then applied recursively until a terminal node has been found, which is then returned.
The run-time of this algorithm is dependant on how many children the algorithm passes on the way to the terminal node. As the SLP is a directed acyclic graph it can be viewed of as a tree, and thus the maximal amount of nodes it can pass depends on the maximal height of that tree, thus $O(h)$.