New Features of Structator v1.02

Benjamin Albrecht
January 2012

Structator v1.02 allows the user to perform a search without an index containing the *aflk* tables. The advantage of omitting these tables is that there is less file traffic and, thus, the system produces less page faults during a search after a particular sequence-structure pattern. However, omitting these tables each affix link has to be computed via a certain search method (see Sec. 2) which take some extra time. In general, one can say that a search with an index containing the *aflk* tables provides the better runtime if its size does not exceed the system's memory.

1 afconstruct

There are two new options for the index construction:

- `-lcpTree`
  Constructs a table of 1 Byte per entry for the *forward* sequences.
- `-lcprTree`
  Constructs a table of 1 Byte per entry for the *reverse* sequences.

These two options allow the user to generate a table of 1 Byte per entry which accelerates the search after an affix link. Providing the tables *aflk* an affix link is computed in constant time. Omitting these tables an affix link has to be located somewhere in the suffix array. Therefore, Structator v1.02 now offers two kind of search methods which are described in the following section.

2 afsearch

There are two new possibilities to perform a search after a particular sequence-structure pattern:

- A search with the precomputed tables *suf* and *lcp*.
  With only 10 Bytes per entry, this search method requires the smallest precomputed index of all other methods. A mapping of the tables *suf* and *lcp* can be achieved by applying the options `-lcp` and `-lcpr`.
- A search with the precomputed tables *suf*, *lcp*, and *lcpTree*.
  With 12 Bytes per entry this search method requires a larger precomputed index than the first method. With the help of the table *lcpTree*, however, a search can be performed a bit faster and the index is still smaller than a search with the table *aflk* (18 per entry). The mapping of the tables *suf*, *lcp*, and *lcpTree* can be achieved by applying the options `-lcpTree` and `-lcprTree`.