Weekplan: Lempel-Ziv full-text indexing

Nicola Prezza

References and Reading

[1] Navarro, G., and Mäkinen, V. (2007). Compressed full-text indexes. ACM Computing Surveys (CSUR), 39(1)

[2] Prezza, N. (2016) Compressed Computation for Text Indexing. PhD thesis

[3] Kärkkäinen, J., and Ukkonen, E. (1996). Lempel-Ziv parsing and sublinear-size index structures for string matching. In Proc. 3rd South American Workshop on String Processing (WSP'96).

[4] Kreft, S., and Navarro, G. (2013). On compressing and indexing repetitive sequences. Theoretical Computer Science, 483, 115-133.

Notes: [1] and references therein is an excellent and comprehensive survey covering the subject of compressed text indexing. [2] is my PhD thesis: here you find all the material covered in this lesson (and much more) down to all details. [3] describes the first compressed index (LZ78), while [4] The first LZ77 index.

Exercises

1 LZ77 trie The *LZ77 trie* is the trie of all LZ77 phrases. Solve the following exercises:

1.1 Draw the LZ77 trie of $T = ACGCGACACACACGGTGGGT\$$

1.2 Assuming you have access to the text T, design a data structure taking $\Theta(z)$ words of space representing the LZ77 trie of T. The structure should support fast child operations (you can assume constant-size alphabet)

2 LZ77 sparse suffix tree The *LZ77 sparse suffix tree* is the path-compressed trie of all suffixes of $\overleftarrow{T}$ (T reversed) that start at a LZ77 phrase boundaries (w.r.t. the LZ77 factorization of T).

2.1 Draw the LZ77 sparse suffix tree of $T = ACGCGACACACACGGTGGGT\$$

2.2 Write on each explicit tree node N the lexicographic range of the suffixes under N

3 LZ search algorithm completeness Prove the following properties of the LZ77/78 parsings:

3.1 Every string appearing in the text has at least one primary occurrence

3.2 Let $S = T[i, \ldots, i+m-1]$ be a secondary occurrence. Prove that, following backward the chain of copies starting from S (i.e. source of the phrase containing S, source of the source, ...), we end up in a *primary* occurrence $T[i', \ldots, i'+m-1] = S$ (with $i' < i$). Prove moreover that this occurrence is unique.

4 LZ77 text extraction Recall that the LZ77 variant with *self-references* is the one where we allow the source of any phrase Z to (partially) overlap Z itself. Let h be the parse height of the LZ77 parse.

4.1 How big is h in the worst case if we allow self-references? and if we do not allow them?

4.2 Describe a data structure of $\Theta(z)$ words of space that permits to extract any text character in $O(h \log\log n)$ time. Show how to achieve the same task in $O(h \log z)$ time (this is faster if $z \ll \log n$).

5 LZ78 self-index Show how to obtain a LZ78 self-index taking $\Theta(z \log n)$ words of space and supporting `locate` in $O(m(m + \log z) + occ \log n)$ time (i.e. the index must be as fast as the LZ78 full-text index)