Bar-Ilan University
אוניברסיטת בר-אילן

Department of Computer Science
Professor Amihood Amir

המחלקה למדעי המחשב
פרופסור עמיהוד אמיר

ב״ה, לי שבט תשע״ט
Feb. 5, 2019

Professor Pawel Strzelecki, Dean
Faculty of Mathematics, Informatics and Mechanics
University of Warsae
Poland

Dear Professor Strzelecki,

Re: Review of the Ph.D. thesis of T. Kociumaka: *Efficient Data Structures for Internal Queries in Texts*

Attached is my review of the Ph.D. thesis of Tomasz Kociumaka.
The thesis is excellent and I recommend granting him the Ph.D. with an honorary distinction. The reason for my decision appears in the review.

I am at your service if any additional information is required.

With best regards,

Amihood Amir
Professor of Computer Science

Tel: 03 531 8770 :טל • Fax: 03 7384056 :פקס • amir@cs.biu.ac.il • www.cs.biu.ac.il/~amir
Bar-Ilan University (RA), Ramat Gan 52900, Israel • **www.biu.ac.il** • אוניברסיטת בר-אילן (ע"ר), רמת גן 52900, ישראל

Review of PhD Dissertation:

Efficient Data Structures for Internal Queries in Texts

By *Tomasz Kociumaka*

Context:

Handling texts has become a crucial task in modern times due to the explosion of digital data. There are a number of models that answer various needs. For example, the *world wide web* requires indexing efficient in both time and space. However, there are applications that require analysis and study of huge bodies of data. Examples are: astrophysical data, economic data, or genomic data. These are extremely large bodies of data that require internal tools for in-depth analysis. One may want to answer questions regarding repetitions of data, extensions of substrings, structure of substrings, etc. This thesis answers questions in this arena – local internal queries in a given extremely large dataset.

Queries Considered:

One of the main tools in pattern matching algorithms is the *longest common prefix* (LCP), also known as the *longest common extension* (LCE). This query requires that, given two positions in a text, one should answer what is the longest common substring that starts at these two positions. The idea is to preprocess the text as efficiently as possible, and answer the queries as fast as possible. Already in the early 70's a magical algorithm, using suffix trees and lowest common ancestor queries, was proposed. This allowed preprocessing in *linear time* (for finite fixed alphabets) and subsequent queries answered in *constant time*. I remember that when I first saw that algorithm I could not believe it – it seemed like magic! That algorithm is one of the cornerstones of pattern matching and is used as a tool in many important algorithms. Nevertheless, with the huge amount of data availabe, some of which having very small alphabets (for example, the genome has an alphabet of size 4, requiring only two bits to encode) there arose the need for an algorithm achieving those times in a **bit-complexity** sense, rather than the original word-based algorithm. Such an algorithm would reduce the space necessary for an LCE algorithm on the genome to 3% of the word-based algorithm. This is a huge reduction in space (and an added reduction in time). This is actually achieved in this thesis.

Another important query is *periodicity*. Periodicity is a repeated concatenation of a substring in the original text. Such repetitions are indications to various interesting phenomena in the genome, for example. However, the genome's size (more than $3x10^9$ base pairs) requires the answer to such queries to be super fast. In this thesis, an answer to these queries is achieved in constant time following a linear text preprocessing.

Another important aspect considered in this thesis is answering various internal queries in a compressed text. Text compression was originally defined in order to save space. However, it is clear that if algorithms can be run on the compressed text, time will be saved, proportional to the compression ratio. This is easy enough to achieve

for simple compressions, such as run-length (used by fax transmission). But more sophisticated compression schemes are adaptive. This adds a new level of complexity. Previous results dealt with the well-known Lempel-Ziv compression. However, to my knowledge, no work was done in the more recent Burrows-Wheeler transform, that is being used in more and more compressions. This thesis gives the first results in answering internal queries on the BWT-compressed text.

Methodology:

The thesis defines some elementary types of query that, if answered efficiently can be used as building blocks for the important internal queries we had described. To solve these queries, the thesis makes sophisticated use of some of the most modern techniques available. Some of these methods involve building tree structures. The height of such trees is logarithmic, which would have introduced a logarithmic factor to many of the proposed solution. It was thus necessary to do more than just ingeniously use sophisticated techniques. It was necessary to create some new techniques, which have the potential of serving future efficient algorithms in the field. An example is the ability to extend a periodic substring to maximal repetition.

Publications:

The results in these thesis appeared as three papers. Two in the *Symposium of Discrete Algorithms* (SODA) – the best international conference on Algorithms, and one in *Combinatorial Pattern Matching* (CPM) – the best international conference in the string processing area. The results already achieved international recognition and are sure to promote further research.

Exposition:

The thesis is clear and well-written. Although the material is quite hard technically, the writing is clear and convenient to follow. There is also adequate use of figures and examples.

Conclusion:

This thesis is definitely worthy of awarding a Ph.D. In addition, because of the novelty of the techniques and importance of the problem, as well as the distinguished venues it appeared in, I recommend **awarding the Ph.D. with an honorary distinction**.