Approximate string matching as an algebraic computation

Alexander Tiskin
DIMAP and Department of Computer Science, University of Warwick, United Kingdom

ABSTRACT

Approximate string matching has a long history and employs a wide variety of methods (see e.g. the survey [2]). We consider a variant of approximate matching that compares a fixed pattern string to every substring in the text string by a rational-weighted edit distance (e.g. the indel distance, defined as the number of character insertions and deletions, or the indelsub/Levenshtein distance, where character substitutions are also allowed). By a simple transformation of the pattern and the text, the problem can be reduced to computing the longest common subsequence between the pattern and every substring of the text. This generic form of approximate matching captures many different problems that have been considered in the past. For such problems, ad-hoc dynamic programming algorithms have typically been designed; a recent example, motivated by modern genome sequencing technologies, is given by [1].

We show that many of these specialised solutions can be unified, and often improved or generalised, by expressing the approximate matching problem in the language of abstract semigroup algebra. Our approach creates a powerful alternative to standard dynamic programming, allowing the computation to be performed independently and simultaneously on different parts of the pattern and the text. As a result, our method provides efficient solutions in situations where this extra independence can be exploited: in particular, approximate matching on compressed strings, parallel string comparison, and local comparison of genome sequences. This paper references our recent work [3].

BODY

Matching a pattern approximately to every substring in a text = computing in the classical braid group, where crossings are made idempotent.

REFERENCES

[1] C. S. Iliopoulos, L. Mouchard, and S. P. Pissis. A parallel algorithm for the fixed-length approximate string matching problem for high throughput sequencing technologies. In *Parallel Computing: From Multicores and GPU's to Petascale*, volume 19 of *Advances in Parallel Computing*, pages 150–157. IOS Press, 2010.

[2] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[3] A. Tiskin. Semi-local string comparison: Algorithmic techniques and applications. Technical Report 0707.3619, arXiv.

Volume 1 of Tiny Transactions on Computer Science