2001–2010: Ten Years of Exact String Matching Algorithms

Simone Faro[1] and Thierry Lecroq[2]

[1] Università di Catania, Dipartimento di Matematica e Informatica, Viale Andrea Doria 6, I-95125 Catania, Italy, faro@dmi.unict.it
[2] University of Rouen, LITIS EA 4108, 76821 Mont-Saint-Aignan Cedex, France, Thierry.Lecroq@univ-rouen.fr

The *online exact string matching problem* consists in finding *all* occurrences of a given pattern p in a text t. It is an extensively studied problem in computer science, mainly due to its direct applications to such diverse areas as text, image and signal processing, speech analysis and recognition, information retrieval, data compression, computational biology and chemistry.

In the last decade more than 50 new algorithms have been proposed for the problem, which add up to a wide set of (almost 40) algorithms presented before 2000 [1]. We will review the most efficient string matching algorithms presented in the last decade in order to bring order among the dozens of articles published in this area.

We performed comparisons between 85 exact string matching algorithms with 12 texts of different types [4]. We divide the patterns into four classes according to their length m: very short ($m \leq 4$), short ($4 < m \leq 32$), long ($32 < m \leq 256$) and very long ($m > 256$). We proceed in the same way for the alphabets according to their size σ: very small ($\sigma < 4$), small ($4 \leq \sigma < 32$), large ($32 \leq \sigma < 128$) and very large ($\sigma > 128$). According to our experimental results (see Figure 1), we conclude that the following algorithms are the most efficient in the following situations:

- SA [11]: very short patterns and very small alphabets.
- TVSBS [10]: very short patterns and small alphabets, and long patterns and large alphabets.
- FJS [5]: very short patterns and large and very large alphabets.
- EBOM [3]: short patterns and large and very large alphabets.
- SBNDM-BMH and BMH-SBNDM [6]: short patterns and very large alphabets.
- HASHq [8]: short and large patterns and small alphabets.
- FSBNDM [3]: long patterns and large and very large alphabets.
- SBNDMq [2]: long pattern and small alphabets.
- LBNDM [9]: very long patterns and very large alphabets.
- SSEF [7]: very long patterns.

Among these algorithms all but one (the SA algorithm) have been designed during the last decade, four of them are based on comparison of characters, one of them is based on automata while six of them are bit-parallel algorithms.

In order to ease further works for developing fast exact string matching algorithms, we developed smart (string matching algorithms research tool, http://www.dmi.unict.it/~faro/smart/) which is a tool that provides a standard framework for researchers in string matching. It helps users to test, design, evaluate and understand existing solutions for the exact string matching problem. Moreover it provides the implementation of (almost) all string matching algorithms and a wide corpus of text buffers.

Simone Faro, Thierry Lecroq: 2001–2010: Ten Years of Exact String Matching Algorithms, pp. 1–2.
Proceedings of PSC 2011, Jan Holub and Jan Žďárek (Eds.), ISBN 978-80-01-04870-2

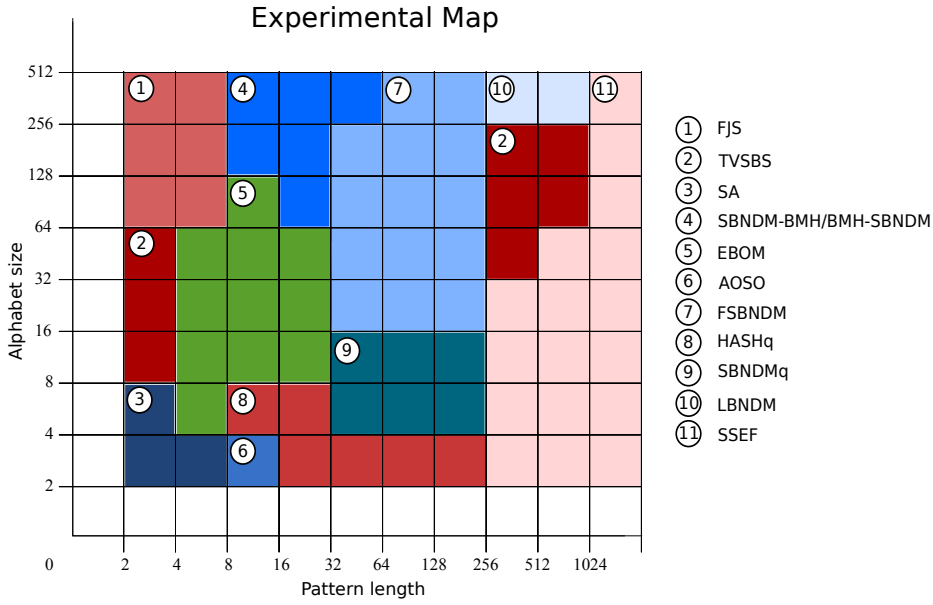

Figure 1. Experimental map of the best results obtained in our evaluation. Comparison based algorithms are presented in red gradations, automata based algorithms are presented in green gradations and bit parallel algorithms are presented in blue gradations.

References

1. C. Charras and T. Lecroq: *Handbook of exact string matching algorithms*, King's College Publications, 2004.
2. B. Durian, J. Holub, H. Peltola, and J. Tarhio: *Tuning BNDM with q-grams*, in Proceedings of the Workshop on Algorithm Engineering and Experiments, ALENEX 2009, I. Finocchi and J. Hershberger, eds., New York, New York, USA, 2009, SIAM, pp. 29–37.
3. S. Faro and T. Lecroq: *Efficient variants of the Backward-Oracle-Matching algorithm*, in Proceedings of the Prague Stringology Conference 2008, J. Holub and J. Žďárek, eds., Czech Technical University in Prague, Czech Republic, 2008, pp. 146–160.
4. S. Faro and T. Lecroq: *The exact string matching problem: a comprehensive experimental evaluation*, Report arXiv:1012.2547, Computing Research Repository, 2010.
5. F. Franek, C. G. Jennings, and W. F. Smyth: *A simple fast hybrid pattern-matching algorithm.* J. Discret. Algorithms, 5(4) 2007, pp. 682–695.
6. J. Holub and B. Durian: *Talk: Fast variants of bit parallel approach to suffix automata*, in The Second Haifa Annual International Stringology Research Workshop of the Israeli Science Foundation, `http://www.cri.haifa.ac.il/events/2005/string/presentations/Holub.pdf`, 2005.
7. M. O. Külekci: *Filter based fast matching of long patterns by using simd instructions*, in Proceedings of the Prague Stringology Conference 2009, J. Holub and J. Žďárek, eds., Czech Technical University in Prague, Czech Republic, 2009, pp. 118–128.
8. T. Lecroq: *Fast exact string matching algorithms.* Inf. Process. Lett., 102(6) 2007, pp. 229–235.
9. H. Peltola and J. Tarhio: *Alternative algorithms for bit-parallel string matching*, in Proceedings of the 10th International Symposium on String Processing and Information Retrieval SPIRE'03, M. A. Nascimento, E. S. de Moura, and A. L. Oliveira, eds., vol. 2857 of Lecture Notes in Computer Science, Manaus, Brazil, 2003, Springer-Verlag, Berlin, pp. 80–94.
10. R. Thathoo, A. Virmani, S. S. Lakshmi, N. Balakrishnan, and K. Sekar: *TVSBS: A fast exact pattern matching algorithm for biological sequences.* J. Indian Acad. Sci., Current Sci., 91(1) 2006, pp. 47–53.
11. S. Wu and U. Manber: *Fast text searching allowing errors.* Commun. ACM, 35(10) 1992, pp. 83–91.