

Bibliography

- [1] Z. Adamczyk and W. Rytter. A note on a simple computation of the maximal suffix of a string. *J. Discrete Algorithms*, 20:61–64, 2013.
- [2] D. Adjeroh, T. Bell and A. Mukherjee. *The Burrows-Wheeler Transform*. Springer, 2008.
- [3] A. V. Aho and M. J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, 1975.
- [4] A. V. Aho, J. E. Hopcroft and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [5] C. Allauzen, M. Crochemore and M. Raffinot. Factor oracle: A new structure for pattern matching. In J. Pavelka, G. Tel and M. Bartosek, eds., *SOFSEM '99, Theory and Practice of Informatics, 26th Conference on Current Trends in Theory and Practice of Informatics*, Milovy, Czech Republic, 27 November–4 December 1999, Lecture Notes in Computer Science, vol. 1725, pp. 295–310. Springer, 1999.
- [6] J. Allouche and J. O. Shallit. The ubiquitous Prouhet-Thue-Morse sequence. In C. Ding, T. Helleseht and H. Niederreiter, eds., *Sequences and Their Applications, proceedings SETA 98*, pp. 1–16. Springer-Verlag, 1999.
- [7] J. Allouche and J. O. Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, 2003.
- [8] J. Almeida, A. Costa, R. Kyriakoglou and D. Perrin. Profinite semigroups and symbolic dynamics, volume 2274 of *Lecture Notes in Mathematics*, Springer, 2020.
- [9] M. Alzamel, M. Crochemore, C. S. Iliopoulos, et al. How much different are two words with different shortest periods? In L. S. Iliadis, I. Maglogiannis and V. P. Plagianakos, eds., *Artificial Intelligence Applications and Innovations AIAI 2018 IFIP WG 12.5 International Workshops, SEDSEAL, 5G-PINE, MHDW, and HEALTHIOT*, Rhodes, Greece, 25–27 May 2018, IFIP Advances in Information and Communication Technology, vol. 520, pp. 168–178. Springer, 2018.
- [10] A. Amir and M. Farach. Efficient matching of nonrectangular shapes. *Ann. Math. Artif. Intell.*, 4:211–224, 1991.
- [11] A. Amir, M. Farach and S. Muthukrishnan. Alphabet dependence in parameterized matching. *Inf. Process. Lett.*, 49(3):111–115, 1994.

- [12] A. Amir, C. S. Iliopoulos and J. Radoszewski. Two strings at Hamming distance 1 cannot be both quasiperiodic. *Inf. Process. Lett.*, 128:54–57, 2017.
- [13] S. Amoroso and G. Cooper. Tessellation structures for reproduction of arbitrary patterns. *J. Comput. Syst. Sci.*, 5(5):455–464, 1971.
- [14] A. Apostolico and M. Crochemore. Optimal canonization of all substrings of a string. *Inf. Comput.*, 95(1):76–95, 1991.
- [15] A. Apostolico and R. Giancarlo. Pattern matching machine implementation of a fast test for unique decipherability. *Inf. Process. Lett.*, 18(3):155–158, 1984.
- [16] A. Apostolico and R. Giancarlo. The Boyer-Moore-Galil string searching strategies revisited. *SIAM J. Comput.*, 15(1):98–105, 1986.
- [17] G. Assayag and S. Dubnov. Using factor oracles for machine improvisation. *Soft Comput.*, 8(9):604–610, 2004.
- [18] G. Badkobeh. Infinite words containing the minimal number of repetitions. *J. Discrete Algorithms*, 20:38–42, 2013.
- [19] G. Badkobeh and M. Crochemore. Fewest repetitions in infinite binary words. *RAIRO Theor. Inf. Applic.*, 46(1):17–31, 2012.
- [20] G. Badkobeh, G. Fici and S. J. Puglisi. Algorithms for anti-powers in strings. *Inf. Process. Lett.*, 137:57–60, 2018.
- [21] R. A. Baeza-Yates. Searching subsequences. *Theor. Comput. Sci.*, 78(2):363–376, 1991.
- [22] H. Bai, A. Deza and F. Franek. On a lemma of Crochemore and Rytter. *J. Discrete Algorithms*, 34:18–22, 2015.
- [23] B. S. Baker. A theory of parameterized pattern matching: Algorithms and applications. In S. R. Kosaraju, D. S. Johnson and A. Aggarwal, eds., *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, 16–18 May 1993, San Diego, CA, pp. 71–80. ACM, 1993.
- [24] B. S. Baker. Parameterized pattern matching: Algorithms and applications. *J. Comput. Syst. Sci.*, 52(1):28–42, 1996.
- [25] H. Bannai, M. Giraud, K. Kusano, W. Matsubara, A. Shinohara and J. Simpson. The number of runs in a ternary word. In J. Holub and J. Zdárek, eds., *Proceedings of the Prague Stringology Conference 2010*, Prague, Czech Republic, 30 August–1 September 2010, pp. 178–181. Prague Stringology Club, Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, 2010.
- [26] H. Bannai, I. Tomohiro, S. Inenaga, Y. Nakashima, M. Takeda and K. Tsuruta. The “runs” theorem. *SIAM J. Comput.*, 46(5):1501–1514, 2017.
- [27] P. Baturo, M. Piatkowski and W. Rytter. Usefulness of directed acyclic subword graphs in problems related to standard sturmian words. *Int. J. Found. Comput. Sci.*, 20(6):1005–1023, 2009.
- [28] M. Béal, F. Mignosi and A. Restivo. Minimal forbidden words and symbolic dynamics. In C. Puech and R. Reischuk, eds., *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science*, Grenoble, France, 22–24 February 1996, Lecture Notes in Computer Science, vol. 1046, pp. 555–566. Springer, 1996.

- [29] M. Béal and D. Perrin. Synchronised automata. In V. Berthé and M. Rigo, eds., *Combinatorics, Words and Symbolic Dynamics. Encyclopedia of Mathematics and Its Applications*, pp. 213–240. Cambridge University Press, 2016.
- [30] M.-P. Béal, M. Crochemore, F. Mignosi, A. Restivo and M. Sciortino. Forbidden words of regular languages. *Fundam. Inform.*, 56(1,2):121–135, 2003.
- [31] J. L. Bentley and R. Sedgewick. Fast algorithms for sorting and searching strings. In M. E. Saks, ed., *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, LA, 5–7 January 1997, New Orleans, pp. 360–369. ACM/SIAM, 1997.
- [32] J. Berstel. Langford strings are square free. *Bull. EATCS*, 37:127–128, 1989.
- [33] J. Berstel and A. de Luca. Sturmian words, Lyndon words and trees. *Theor. Comput. Sci.*, 178(1–2):171–203, 1997.
- [34] J. Berstel and J. Karhumäki. Combinatorics on words: A tutorial. *EATCS*, 79:178, 2003.
- [35] J. Berstel, A. Lauve, C. Reutenauer and F. Saliola. *Combinatorics on Words*, CRM Monograph Series, vol. 27. Université de Montréal et American Mathematical Society, 2008.
- [36] J. Berstel and D. Perrin. *Theory of Codes*. Academic Press, 1985.
- [37] J. Berstel and A. Savelli. Crochemore factorization of sturmian and other infinite words. In *Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS 2006*, Stará Lesná, Slovakia, 28 August–1 September 2006, Lecture Notes in Computer Science, vol. 4162, pp. 157–166. Springer, 2006.
- [38] A. Blumer, J. Blumer, D. Haussler, A. Ehrenfeucht, M. T. Chen and J. I. Seiferas. The smallest automaton recognizing the subwords of a text. *Theor. Comput. Sci.*, 40:31–55, 1985.
- [39] K. S. Booth. Lexicographically least circular substrings. *Inf. Process. Lett.*, 10(4/5):240–242, 1980.
- [40] J. Bourdon and I. Rusu. Statistical properties of factor oracles. *J. Discrete Algorithms*, 9(1):57–66, 2011.
- [41] R. S. Boyer and J. S. Moore. A fast string searching algorithm. *Commun. ACM*, 20(10):762–772, 1977.
- [42] F. Brandenburg. Uniformly growing k -th power-free homomorphisms. *Theor. Comput. Sci.*, 23:69–82, 1983.
- [43] D. Breslauer. An on-line string superprimitivity test. *Inf. Process. Lett.*, 44(6):345–347, 1992.
- [44] D. Breslauer, L. Colussi and L. Toniolo. Tight comparison bounds for the string prefix-matching problem. *Inf. Process. Lett.*, 47(1):51–57, 1993.
- [45] D. Breslauer, R. Grossi and F. Mignosi. Simple real-time constant-space string matching. *Theor. Comput. Sci.*, 483:2–9, 2013.
- [46] S. Brlek, D. Jamet and G. Paquin. Smooth words on 2-letter alphabets having same parity. *Theor. Comput. Sci.*, 393(1–3):166–181, 2008.
- [47] S. Burkhardt and J. Kärkkäinen. Fast lightweight suffix array construction and checking. In R. A. Baeza-Yates, E. Chávez and M. Crochemore, eds., *Combinatorial Pattern Matching, CPM 2003*, Lecture Notes in Computer Science, vol. 2676, pp. 55–69. Springer, 2003.

- [48] A. Carayol and S. Göller. On long words avoiding Zimin patterns. In H. Vollmer and B. Vallée, eds., *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017*, 8–11 March, 2017, Hannover, Germany, vol. 66 of LIPIcs, pp. 19:1–19:13. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
- [49] Y. Césari and M. Vincent. Une caractérisation des mots périodiques. *C. R. Acad. Sci.*, 286:1175, 1978.
- [50] S. Chairungsee and M. Crochemore. Efficient computing of longest previous reverse factors. In Y. Shoukourian, ed., *Seventh International Conference on Computer Science and Information Technologies, CSIT 2009*, pp. 27–30. The National Academy of Sciences of Armenia Publishers, Yerevan, Armenia, 2009.
- [51] S. Chairungsee and M. Crochemore. Using minimal absent words to build phylogeny. *Theor. Comput. Sci.*, 450(1):109–116, 2012.
- [52] S. Chairungsee and M. Crochemore. Longest previous non-overlapping factors table computation. In X. Gao, H. D. and M. Han, eds., *Combinatorial Optimization and Applications - 11th International Conference, COCOA 2017*, Shanghai, China, 10–18 December, 2017, Proceedings, Part II, vol. 10628 *Lecture Notes in Computer Science*, pp. 483–491. Springer, 2017.
- [53] J. Champarnaud, G. Hansel and D. Perrin. Unavoidable sets of constant length. *IJAC*, 14(2):241–251, 2004.
- [54] S. Cho, J. C. Na, K. Park and J. S. Sim. A fast algorithm for order-preserving pattern matching. *Inf. Process. Lett.*, 115(2):397–402, 2015.
- [55] J. G. Cleary, W. J. Teahan and I. H. Witten. Unbounded length contexts for PPM. In J. A. Storer and M. Cohn, eds., *Proceedings of the IEEE Data Compression Conference, DCC 1995*, Snowbird, UT, 28–30 March, 1995, pp. 52–61. IEEE Computer Society, 1995.
- [56] J. G. Cleary and I. H. Witten. A comparison of enumerative and adaptive codes. *IEEE Trans. Inf. Theory*, 30(2):306–315, 1984.
- [57] J. Clément, P. Flajolet and B. Vallée. The analysis of hybrid trie structures. In H. J. Karloff, ed., *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 25–27 January 1998, San Francisco, 25–27 January 1998, pp. 531–539. ACM/SIAM, 1998.
- [58] P. Clifford and R. Clifford. Simple deterministic wildcard matching. *Inf. Process. Lett.*, 101(2):53–54, 2007.
- [59] R. Cole. Tight bounds on the complexity of the Boyer-Moore string matching algorithm. *SIAM J. Comput.*, 23(5):1075–1091, 1994.
- [60] R. Cole and R. Hariharan. Tighter upper bounds on the exact complexity of string matching. *SIAM J. Comput.*, 26(3):803–856, 1997.
- [61] R. Cori and D. Perrin. Automates et commutations partielles. *ITA*, 19(1):21–32, 1985.
- [62] G. V. Cormack and R. N. Horspool. Algorithms for adaptive Huffman codes. *Inf. Process. Lett.*, 18(3):159–165, 1984.
- [63] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, 3rd Edition. MIT Press, 2009.
- [64] M. Crochemore. An optimal algorithm for computing the repetitions in a word. *Inf. Process. Lett.*, 12(5):244–250, 1981.

- [65] M. Crochemore. Sharp characterization of square-free morphisms. *Theor. Comput. Sci.*, 18(2):221–226, 1982.
- [66] M. Crochemore. *Régularités évitables*. Thèse d'état, Université de Haute-Normandie, 1983.
- [67] M. Crochemore. Transducers and repetitions. *Theor. Comput. Sci.*, 45(1):63–86, 1986.
- [68] M. Crochemore. Longest common factor of two words. In H. Ehrig, R. A. Kowalski, G. Levi and U. Montanari, eds., *TAPSOFT'87: Proceedings of the International Joint Conference on Theory and Practice of Software Development*, Pisa, Italy, 23–27 March, 1987, Volume 1: Advanced Seminar on Foundations of Innovative Software Development I and Colloquium on Trees in Algebra and Programming (CAAP'87), vol. 249, Lecture Notes in Computer Science, pp. 26–36. Springer, 1987.
- [69] M. Crochemore. String-matching on ordered alphabets. *Theor. Comput. Sci.*, 92(1):33–47, 1992.
- [70] M. Crochemore, A. Czumaj, L. Gasieniec, S. Jarominek, T. Lecroq, W. Plandowski and W. Rytter. Speeding up two string-matching algorithms. *Algorithmica*, 12(4/5):247–267, 1994.
- [71] M. Crochemore, C. Epifanio, R. Grossi and F. Mignosi. Linear-size suffix tries. *Theor. Comput. Sci.*, 638:171–178, 2016.
- [72] M. Crochemore, S. Z. Fazekas, C. S. Iliopoulos and I. Jayasekera. Number of occurrences of powers in strings. *Int. J. Found. Comput. Sci.*, 21(4):535–547, 2010.
- [73] M. Crochemore, R. Grossi, J. Kärkkäinen and G. M. Landau. Computing the Burrows-Wheeler transform in place and in small space. *J. Discrete Algorithms*, 32:44–52, 2015.
- [74] M. Crochemore, C. Hancart and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007.
- [75] M. Crochemore, A. Héliou, G. Kucherov, L. Mouchard, S. P. Pissis and Y. Ramusat. Absent words in a sliding window with applications. *Inf. Comput.*, 270, 2020.
- [76] M. Crochemore and L. Ilie. Computing longest previous factors in linear time and applications. *Inf. Process. Lett.*, 106(2):75–80, 2008.
- [77] M. Crochemore and L. Ilie. Maximal repetitions in strings. *J. Comput. Syst. Sci.*, 74(5):796–807, 2008.
- [78] M. Crochemore, L. Ilie, C. S. Iliopoulos, M. Kubica, W. Rytter and T. Waleń. Computing the longest previous factor. *Eur. J. Comb.*, 34(1):15–26, 2013.
- [79] M. Crochemore, L. Ilie and E. Seid-Hilmi. The structure of factor oracles. *Int. J. Found. Comput. Sci.*, 18(4):781–797, 2007.
- [80] M. Crochemore, L. Ilie and L. Tinta. The “runs” conjecture. *Theor. Comput. Sci.*, 412(27):2931–2941, 2011.
- [81] M. Crochemore, C. S. Iliopoulos, T. Kociumaka et al. Order-preserving indexing. *Theor. Comput. Sci.*, 638:122–135, 2016.
- [82] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, et al. The maximum number of squares in a tree. In J. Kärkkäinen and J. Stoye, eds., *Combinatorial Pattern Matching: 23rd Annual Symposium, CPM 2012*, Helsinki, Finland, 3–5 July,

2012. Proceedings, Lecture Notes in Computer Science, vol. 7354, pp. 27–40. Springer, 2012.
- [83] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, et al. Near-optimal computation of runs over general alphabet via non-crossing LCE queries. In S. Inenaga, K. Sadakane and T. Sakai, eds., *String Processing and Information Retrieval - 23rd International Symposium, SPIRE 2016*, Beppu, Japan, 18–20 October, 2016, Proceedings, vol. 9954, *Lecture Notes in Computer Science*, pp. 22–34, 2016.
- [84] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, K. Stencel and T. Walen. New simple efficient algorithms computing powers and runs in strings. *Discrete Appl. Math.*, 163:258–267, 2014.
- [85] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter and T. Walen. On the maximal number of cubic runs in a string. In A. Dediu, H. Fernau, and C. Martín-Vide, eds., *Language and Automata Theory and Applications, 4th International Conference, LATA 2010*, Trier, Germany, 24–28 May, 2010. Proceedings, vol. 6031, *Lecture Notes in Computer Science*, pp. 227–238. Springer, 2010.
- [86] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter and T. Walen. The maximal number of cubic runs in a word. *J. Comput. Syst. Sci.*, 78(6):1828–1836, 2012.
- [87] M. Crochemore, C. S. Iliopoulos, M. Kubica, W. Rytter and T. Waleń. Efficient algorithms for three variants of the LPF table. *J. Discrete Algorithms*, 11:51–61, 2012.
- [88] M. Crochemore, G. M. Landau and M. Ziv-Ukelson. A subquadratic sequence alignment algorithm for unrestricted scoring matrices. *SIAM J. Comput.*, 32(6):1654–1673, 2003.
- [89] M. Crochemore and T. Lecroq. Tight bounds on the complexity of the Apostolico-Giancarlo algorithm. *Inf. Process. Lett.*, 63(4):195–203, 1997.
- [90] M. Crochemore, M. Lerest and P. Wender. An optimal test on finite unavoidable sets of words. *Inf. Process. Lett.*, 16(4):179–180, 1983.
- [91] M. Crochemore and R. Mercas. On the density of Lyndon roots in factors. *Theor. Comput. Sci.*, 656:234–240, 2016.
- [92] M. Crochemore, F. Mignosi and A. Restivo. Automata and forbidden words. *Inf. Process. Lett.*, 67(3):111–117, 1998.
- [93] M. Crochemore, F. Mignosi, A. Restivo and S. Salemi. Text compression using antidictionaries. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, eds., *International Conference on Automata, Languages and Programming* (Prague, 1999), Lecture Notes in Computer Science, pp. 261–270. Springer-Verlag, 1999. Rapport I.G.M. 98-10, Université de Marne-la-Vallée.
- [94] M. Crochemore and D. Perrin. Two-way string-matching. *J. ACM*, 38(3):651–675, 1991.
- [95] M. Crochemore and E. Porat. Fast computation of a longest increasing subsequence and application. *Inf. Comput.*, 208(9):1054–1059, 2010.
- [96] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.
- [97] M. Crochemore and W. Rytter. Squares, cubes, and time-space efficient string searching. *Algorithmica*, 13(5):405–425, 1995.

- [98] M. Crochemore and W. Rytter. *Jewels of Stringology*. World Scientific Publishing, Hong-Kong, 2002.
- [99] M. Crochemore and G. Tischler. Computing longest previous non-overlapping factors. *Inf. Process. Lett.*, 111(6):291–295, 2011.
- [100] M. Crochemore and Z. Troníček. On the size of DASG for multiple texts. In A. H. F. Laender and A. L. Oliveira, eds., *String Processing and Information Retrieval, 9th International Symposium, SPIRE 2002*, Lisbon, Portugal, 11–13 September, 2002, Proceedings, vol. 2476, *Lecture Notes in Computer Science*, pp. 58–64. Springer, 2002.
- [101] M. Crochemore and R. Vérin. On compact directed acyclic word graphs. In J. Mycielski, G. Rozenberg, and A. Salomaa, eds., *Structures in Logic and Computer Science, A Selection of Essays in Honor of Andrzej Ehrenfeucht*, vol. 1261 *Lecture Notes in Computer Science*, pp. 192–211. Springer, 1997.
- [102] A. Deza and F. Franek. A d -step approach to the maximum number of distinct squares and runs in strings. *Discrete Appl. Math.*, 163(3):268–274, 2014.
- [103] A. Deza, F. Franek and A. Thierry. How many double squares can a string contain? *Discrete Appl. Math.*, 180:52–69, 2015.
- [104] F. Durand and J. Leroy. The constant of recognizability is computable for primitive morphisms. *CoRR*, abs/1610.05577, 2016.
- [105] J. Duval. Factorizing words over an ordered alphabet. *J. Algorithms*, 4(4):363–381, 1983.
- [106] J. Duval, R. Kolpakov, G. Kucherov, T. Lecroq and A. Lefebvre. Linear-time computation of local periods. *Theor. Comput. Sci.*, 326(1-3):229–240, 2004.
- [107] M. Effros. PPM performance with BWT complexity: A new method for lossless data compression. In *Data Compression Conference, DCC 2000*, Snowbird, UT, 28–30 March, 2000, pp. 203–212. IEEE Computer Society, 2000.
- [108] N. Faller. An adaptive system for data compression. In *Record of the 7th Asilomar Conference on Circuits, Systems, and Computers*, pp. 593–597, 1973.
- [109] H. Fan, N. Yao and H. Ma. Fast variants of the Backward-Oracle-Marching algorithm. In *ICICSE '09, Fourth International Conference on Internet Computing for Science and Engineering*, pp. 56–59. IEEE Computer Society, 2009.
- [110] M. Farach. Optimal suffix tree construction with large alphabets. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97*, Miami Beach, FL, 19–22 October, 1997, pp. 137–143. IEEE Computer Society, 1997.
- [111] S. Faro and T. Lecroq. Efficient variants of the Backward-Oracle-Matching algorithm. *Int. J. Found. Comput. Sci.*, 20(6):967–984, 2009.
- [112] P. Ferragina and G. Manzini. Indexing compressed text. *J. ACM*, 52(4):552–581, 2005.
- [113] G. Fici, A. Restivo, M. Silva and L. Q. Zamboni. Anti-powers in infinite words. *J. Comb. Theory, Ser. A*, 157:109–119, 2018.
- [114] N. J. Fine. Binomial coefficients modulo a prime. *Am. Math. Mon.*, 54(10, Part 1):589–592, December 1947.
- [115] J. Fischer and V. Heun. Theoretical and practical improvements on the RMQ-problem, with applications to LCA and LCE. In M. Lewenstein and G. Valiente, eds., *Combinatorial Pattern Matching, 17th Annual Symposium, CPM 2006*,

- Barcelona, Spain, 5–7 July, 2006, Proceedings, vol. 4009 *Lecture Notes in Computer Science*, pp. 36–48. Springer, 2006.
- [116] J. Fischer, Š. Holub, T. I and M. Lewenstein. Beyond the runs theorem. In 22nd *SPIRE*, Lecture Notes in Computer Science, vol. 9309, pp. 272–281, 2015.
 - [117] A. S. Fraenkel and J. Simpson. How many squares must a binary sequence contain? *Electr. J. Comb.*, 2, 1995.
 - [118] A. S. Fraenkel and J. Simpson. How many squares can a string contain? *J. Comb. Theory, Ser. A*, 82(1):112–120, 1998.
 - [119] F. Franek, A. S. M. S. Islam, M. S. Rahman and W. F. Smyth. Algorithms to compute the Lyndon array. *CoRR*, abs/1605.08935, 2016.
 - [120] H. Fredricksen and J. Maiorana. Necklaces of beads in k colors and k-ary de bruijn sequences. *Discrete Math.*, 23(3):207–210, 1978.
 - [121] A. Fruchtmann, Y. Gross, S. T. Klein and D. Shapira. Weighted adaptive Huffman coding. In A. Bilgin, M. W. Marcellin, J. Serra-Sagrista and J. A. Storer, eds., Data Compression Conference, DCC 2020, Snowbird, UT, 24–27 March 2020, pp. 368–385. IEEE, 2020. <http://arxiv.org/abs/2005.08232v1>.
 - [122] Z. Galil. On improving the worse case running time of the Boyer-Moore string matching algorithm. *Commun. ACM*, 22(9):505–508, 1979.
 - [123] Z. Galil and R. Giancarlo. On the exact complexity of string matching: Upper bounds. *SIAM J. Comput.*, 21(3):407–437, 1992.
 - [124] Z. Galil and J. I. Seiferas. Time-space-optimal string matching. *J. Comput. Syst. Sci.*, 26(3):280–294, 1983.
 - [125] R. G. Gallager. Variations on a theme by Huffman. *IEEE Trans. Inf. Theory*, 24(6):668–674, 1978.
 - [126] J. Gallant, D. Maier and J. A. Storer. On finding minimal length superstrings. *J. Comput. Syst. Sci.*, 20(1):50–58, 1980.
 - [127] P. Gawrychowski, T. Kociumaka, J. Radoszewski, W. Rytter and T. Walen. Universal reconstruction of a string. In F. Dehne, J. Sack and U. Stege, eds., *Algorithms and Data Structures: 14th International Symposium, WADS 2015*, Victoria, BC, Canada, 5–7 August, 2015. Proceedings, vol. 9214, *Lecture Notes in Computer Science*, pp. 386–397. Springer, 2015.
 - [128] P. Gawrychowski, T. Kociumaka, W. Rytter and T. Walen. Faster longest common extension queries in strings over general alphabets. In R. Grossi and M. Lewenstein, eds., *27th Annual Symposium on Combinatorial Pattern Matching, CPM 2016*, 27–29 June, 2016, Tel Aviv, Israel, vol. 54, LIPIcs, pp. 5:1–5:13. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
 - [129] P. Gawrychowski and P. Uznanski. Order-preserving pattern matching with k mismatches. In A. S. Kulikov, S. O. Kuznetsov and P. A. Pevzner, eds., *Combinatorial Pattern Matching: 25th Annual Symposium, CPM 2014*, Moscow, Russia, 16–18 June, 2014. Proceedings, vol. 8486, *Lecture Notes in Computer Science*, pp. 130–139. Springer, 2014.
 - [130] A. Glen, J. Justin, S. Widmer and L. Q. Zamboni. Palindromic richness. *Eur. J. Comb.*, 30(2):510–531, 2009.
 - [131] S. W. Golomb. *Shift Register Sequences* 3rd rev. ed. World Scientific, 2017.
 - [132] J. Grytczuk, K. Kosinski and M. Zmarz. How to play Thue games. *Theor. Comput. Sci.*, 582:83–88, 2015.

- [133] C. Guo, J. Shallit and A. M. Shur. On the combinatorics of palindromes and antipalindromes. *CoRR*, abs/1503.09112, 2015.
- [134] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [135] D. Gusfield and J. Stoye. Linear time algorithms for finding and representing all the tandem repeats in a string. *J. Comput. Syst. Sci.*, 69(4):525–546, 2004.
- [136] C. Hancart. On Simon’s string searching algorithm. *Inf. Process. Lett.*, 47(2):95–99, 1993.
- [137] T. Harju and T. Kärki. On the number of frames in binary words. *Theor. Comput. Sci.*, 412(39):5276–5284, 2011.
- [138] A. Hartman and M. Rodeh. Optimal parsing of strings. In A. Apostolico and Z. Galil, eds., *Combinatorial Algorithms on Words*, vol. 12, NATO ASI Series F: Computer and System Sciences, pp. 155–167, Springer, 1985.
- [139] M. M. Hasan, A. S. M. S. Islam, M. S. Rahman and M. S. Rahman. Order preserving pattern matching revisited. *Pattern Recogn. Lett.*, 55:15–21, 2015.
- [140] D. Hendrian, T. Takagi and S. Inenaga. Online algorithms for constructing linear-size suffix trie. *CoRR*, abs/1901.10045, 2019.
- [141] D. Hickerson. There are at most $2n$ distinct twins in any finite string of length n . Personal communication by D. Gusfield, 2003.
- [142] C. Hohlweg and C. Reutenauer. Lyndon words, permutations and trees. *Theor. Comput. Sci.*, 307(1):173–178, 2003.
- [143] R. Houston. Tackling the minimal superpermutation problem. *CoRR*, abs/1408.5108, 2014.
- [144] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proc. I.R.E.*, 40:1098–1101, 1951.
- [145] R. M. Idury and A. A. Schäffer. Multiple matching of parameterized patterns. *Theor. Comput. Sci.*, 154(2):203–224, 1996.
- [146] L. Ilie. A simple proof that a word of length n has at most $2n$ distinct squares. *J. Comb. Theory, Ser. A*, 112(1):163–164, 2005.
- [147] L. Ilie. A note on the number of squares in a word. *Theor. Comput. Sci.*, 380(3):373–376, 2007.
- [148] L. Ilie and W. F. Smyth. Minimum unique substrings and maximum repeats. *Fundam. Inform.*, 110(1-4):183–195, 2011.
- [149] C. S. Iliopoulos, D. Moore and W. F. Smyth. A characterization of the squares in a Fibonacci string. *Theor. Comput. Sci.*, 172(1-2):281–291, 1997.
- [150] S. Inenaga, H. Hoshino, A. Shinohara, M. Takeda, S. Arikawa, G. Mauri and G. Pavesi. On-line construction of compact directed acyclic word graphs. In A. Amir and G. M. Landau, eds., *Combinatorial Pattern Matching, 12th Annual Symposium, CPM 2001*, Jerusalem, Israel, 1-4 July 2001, Proceedings, vol. 2089, *Lecture Notes in Computer Science*, pp. 169–180. Springer, 2001.
- [151] B. W. Jackson. Universal cycles of k -subsets and k -permutations. *Discrete Math.*, 117(1-3):141–150, 1993.
- [152] N. Johnston. All minimal superpermutations on five symbols have been found. www.njohnston.ca/2014/08/all-minimal-superpermutations-on-five-symbols-have-been-found/, 2014.

- [153] J. Kärkkäinen and P. Sanders. Simple linear work suffix array construction. In J. C. M. Baeten, J. K. Lenstra, J. Parrow and G. J. Woeginger, eds., *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003*, Eindhoven, The Netherlands, 30–4 June, 2003. Proceedings, vol. 2719, *Lecture Notes in Computer Science*, pp. 943–955. Springer, 2003.
- [154] J. Kärkkäinen P. Sanders and S. Burkhardt. Linear work suffix array construction. *J. ACM*, 53(6):918–936, 2006.
- [155] T. Kasai, G. Lee, H. Arimura, S. Arikawa and K. Park. Linear-time longest-common-prefix computation in suffix arrays and its applications. In A. Amir and G. M. Landau, eds., *Combinatorial Pattern Matching, 12th Annual Symposium, CPM 2001*, Jerusalem, Israel, 1–4 July 2001, Proceedings vol. 2089, *Lecture Notes in Computer Science*, pp. 181–192. Springer, 2001.
- [156] J. Katajainen, A. Moffat and A. Turpin. A fast and space-economical algorithm for length-limited coding. In J. Staples, P. Eades, N. Katoh and A. Moffat, eds., *Algorithms and Computation, 6th International Symposium, ISAAC '95*, Cairns, Australia, 4–6 December 1995, Proceedings vol. 1004, *Lecture Notes in Computer Science*, pp. 12–21. Springer, 1995.
- [157] D. Kempa, A. Policriti, N. Prezza and E. Rotenberg. String attractors: Verification and optimization. *CoRR*, abs/1803.01695, 2018.
- [158] A. J. Kfoury. A linear-time algorithm to decide whether A binary word contains an overlap. *ITA*, 22(2):135–145, 1988.
- [159] D. K. Kim, J. S. Sim, H. Park and K. Park. Constructing suffix arrays in linear time. *J. Discrete Algorithms*, 3(2-4):126–142, 2005.
- [160] J. Kim, P. Eades, R. Fleischer, S. Hong, C. S. Iliopoulos, K. Park, S. J. Puglisi and T. Tokuyama. Order-preserving matching. *Theor. Comput. Sci.*, 525:68–79, 2014.
- [161] D. E. Knuth. Dynamic Huffman coding. *J. Algorithms*, 6(2):163–180, 1985.
- [162] D. E. Knuth, J. H. Morris Jr. and V. R. Pratt. Fast pattern matching in strings. *SIAM J. Comput.*, 6(2):323–350, 1977.
- [163] P. Ko and S. Aluru. Space efficient linear time construction of suffix arrays. *J. Discrete Algorithms*, 3(2-4):143–156, 2005.
- [164] T. Kociumaka, J. Pachocki, J. Radoszewski, W. Rytter and T. Walen. Efficient counting of square substrings in a tree. *Theor. Comput. Sci.*, 544:60–73, 2014.
- [165] T. Kociumaka, J. Radoszewski, W. Rytter, J. Straszynski, T. Walen and W. Zuba. Efficient representation and counting of antipower factors in words. In C. Martín-Vide, A. Okhotin and D. Shapira, eds., *Language and Automata Theory and Applications: 13th International Conference, LATA 2019*, St. Petersburg, Russia, 26–29 March, 2019, Proceedings, vol. 11417, *Lecture Notes in Computer Science*, pp. 421–433. Springer, 2019.
- [166] W. Kolakoski. Problem 5304. *Am. Math. Mon.*, 72(674), 1965.
- [167] R. M. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99*, 17–18 October 1999, New York, pp. 596–604. IEEE Computer Society, 1999.
- [168] R. M. Kolpakov and G. Kucherov. Finding approximate repetitions under Hamming distance. In F. Meyer auf der Heide, ed., *Algorithms - ESA 2001*,

- 9th Annual European Symposium, Aarhus, Denmark, 28–31 August 2001, Proceedings, vol. 2161, *Lecture Notes in Computer Science*, pp. 170–181. Springer, 2001.
- [169] K. Kosinski, R. Mercas and D. Nowotka. Corrigendum to ‘a note on Thue games’ [*Inf. Process. Lett.* 118 (2017) 75–77]. *Inf. Process. Lett.*, 130:63–65, 2018.
 - [170] M. Kubica, T. Kulczynski, J. Radoszewski, W. Rytter and T. Walen. A linear time algorithm for consecutive permutation pattern matching. *Inf. Process. Lett.*, 113(12):430–433, 2013.
 - [171] P. Kurka. *Topological and Symbolic Dynamics*. Société Mathématique de France, 2003.
 - [172] L. L. Larmore and D. S. Hirschberg. A fast algorithm for optimal length-limited Huffman codes. *J. ACM*, 37(3):464–473, 1990.
 - [173] A. Lefebvre and T. Lecroq. Compror: On-line lossless data compression with a factor oracle. *Inf. Process. Lett.*, 83(1):1–6, 2002.
 - [174] A. Lempel. On a homomorphism of the de Bruijn graph and its applications to the design of feedback shift registers. *IEEE Trans. Comput.*, 19(12):1204–1209, 1970.
 - [175] M. Lothaire. *Combinatorics on Words*. Addison-Wesley, 1983. Reprinted in 1997.
 - [176] M. Lothaire. *Algebraic Combinatorics on Words. Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2002.
 - [177] M. Lothaire. *Applied Combinatorics on Words*. Cambridge University Press, 2005.
 - [178] M. Maekawa. A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems. *ACM Trans. Comput. Syst.*, 3(2):145–159, 1985.
 - [179] M. G. Main and R. J. Lorentz. An $O(n \log n)$ algorithm for recognizing repetition. Report CS-79-056, Washington State University, Pullman, 1979.
 - [180] M. G. Main and R. J. Lorentz. Linear time recognition of square-free strings. In A. Apostolico and Z. Galil, eds., *Combinatorial Algorithms on Words*, vol. 12, *Series F: Computer and System Sciences*, pp. 271–278. Springer, 1985.
 - [181] G. S. Makanin. The problem of solvability of equations in a free semi-group. *Math. Sb.*, 103(2):147–236, 1977. In Russian. English translation in: *Math. USSR-Sb.*, 32, 129–198, 1977.
 - [182] A. Mancheron and C. Moan. Combinatorial characterization of the language recognized by factor and suffix oracles. *Int. J. Found. Comput. Sci.*, 16(6):1179–1191, 2005.
 - [183] S. Mantaci, A. Restivo, G. Romana G. Rosone and M. Sciortino. String attractors and combinatorics on words. *CoRR*, abs/1907.04660, 2019.
 - [184] S. Mantaci, A. Restivo, G. Rosone and M. Sciortino. Suffix array and Lyndon factorization of a text. *J. Discrete Algorithms*, 28:2–8, 2014.
 - [185] S. Mantaci, A. Restivo and M. Sciortino. Burrows-Wheeler transform and Sturmian words. *Inf. Process. Lett.*, 86(5):241–246, 2003.
 - [186] W. J. Masek and M. Paterson. A faster algorithm computing string edit distances. *J. Comput. Syst. Sci.*, 20(1):18–31, 1980.
 - [187] E. M. McCreight. A space-economical suffix tree construction algorithm. *J. ACM*, 23(2):262–272, 1976.

- [188] J. Mendivelso and Y. Pinzón. Parameterized matching: Solutions and extensions. In J. Holub and J. Žďárek, eds., *Proceedings of the Prague Stringology Conference 2015*, pp. 118–131, Czech Technical University in Prague, Czech Republic, 2015.
- [189] T. Mieno, Y. Kuhara, T. Akagi, et al. Minimal unique substrings and minimal absent words in a sliding window, *International Conference on Current Trends in Theory and Practice of Informatics*, pp. 148–160, Springer, 2019.
- [190] A. Moffat. Implementing the PPM data compression scheme. *IEEE Trans. Commun.*, 38(11):1917–1921, 1990.
- [191] S. P. Mohanty. Shortest string containing all permutations. *Discrete Math.*, 31:91–95, 1980.
- [192] E. Moreno and D. Perrin. Corrigendum to ‘on the theorem of Fredricksen and Maiorana about de Bruijn sequences’. *Adv. Appl. Math.*, 33(2):413–415, 2004.
- [193] G. Navarro and N. Prezza. Universal compressed text indexing. *Theor. Comput. Sci.*, 762:41–50, 2019.
- [194] G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings: Practical On-line Search Algorithms for Texts and Biological Sequences*. Cambridge University Press, 2002.
- [195] J. Nilsson. Letter frequencies in the Kolakoski sequence. *Acta Phys. Pol. A*, 126(2):549–552, 2014.
- [196] E. Ohlebusch. *Bioinformatics Algorithms*. Oldenbusch Verlag, 2013.
- [197] R. Oldenburger. Exponent trajectories in symbolic dynamics. *Trans. AMS*, 46:453–466, 1939.
- [198] T. Ota, H. Fukae and H. Morita. Dynamic construction of an antidictionary with linear complexity. *Theor. Comput. Sci.*, 526:108–119, 2014.
- [199] T. Ota and H. Morita. On a universal antidictionary coding for stationary ergodic sources with finite alphabet. In *International Symposium on Information Theory and Its Applications, ISITA 2014*, Melbourne, Australia, 26–29 October 2014, pp. 294–298. IEEE, 2014.
- [200] T. Ota and H. Morita. A compact tree representation of an antidictionary. *IEICE Trans.*, 100-A(9):1973–1984, 2017.
- [201] J. Pansiot. Decidability of periodicity for infinite words. *ITA*, 20(1):43–46, 1986.
- [202] N. Prezza. String attractors. *CoRR*, abs/1709.05314, 2017.
- [203] S. J. Puglisi, J. Simpson and W. F. Smyth. How many runs can a string contain? *Theor. Comput. Sci.*, 401(1-3):165–171, 2008.
- [204] J. Radoszewski and W. Rytter. On the structure of compacted subword graphs of Thue-Morse words and their applications. *J. Discrete Algorithms*, 11:15–24, 2012.
- [205] N. Rampersad, J. Shallit and M. Wang. Avoiding large squares in infinite binary words. *Theor. Comput. Sci.*, 339(1):19–34, 2005.
- [206] D. Repke and W. Rytter. On semi-perfect de Bruijn words. *Theor. Comput. Sci.*, 720:55–63, 2018.
- [207] A. Restivo and S. Salemi. Overlap-free words on two symbols. In M. Nivat and D. Perrin, eds., *Automata on Infinite Words*, Ecole de Printemps d’Informatique Théorique, Le Mont Dore, 14–18 May, 1984, vol. 192, *Lecture Notes in Computer Science*, pp. 198–206. Springer, 1985.

- [208] C. Reutenauer. *From Christoffel Words to Markov Numbers*. Oxford University Press, 2018.
- [209] G. Rozenberg and A. Salomaa. *The Mathematical Theory of L Systems*. Academic Press, 1980.
- [210] M. Rubinchik and A. M. Shur. *Eertree: An efficient data structure for processing palindromes in strings*. CoRR, abs/1506.04862, 2015.
- [211] F. Ruskey and A. Williams. An explicit universal cycle for the $(n-1)$ -permutations of an n -set. *ACM Trans. Algorithms*, 6(3):45:1–45:12, 2010.
- [212] W. Rytter. A correct preprocessing algorithm for Boyer-Moore string-searching. *SIAM J. Comput.*, 9(3):509–512, 1980.
- [213] W. Rytter. The structure of subword graphs and suffix trees of Fibonacci words. *Theor. Comput. Sci.*, 363(2):211–223, 2006.
- [214] W. Rytter. The number of runs in a string. *Informat. Comput.*, 205(9):1459–1469, 2007.
- [215] W. Rytter. Two fast constructions of compact representations of binary words with given set of periods. *Theor. Comput. Sci.*, 656:180–187, 2016.
- [216] W. Rytter. Computing the k -th letter of Fibonacci word. Personal communication, 2017.
- [217] A. A. Sardinas and G. W. Patterson. A necessary and sufficient condition for the unique decomposition of coded messages. Research Division Report 50-27, Moore School of Electrical Engineering, University of Pennsylvania, 1950.
- [218] J. Sawada and P. Hartman. Finding the largest fixed-density necklace and Lyndon word. *Inf. Process. Lett.*, 125:15–19, 2017.
- [219] J. Sawada, A. Williams and D. Wong. A surprisingly simple de Bruijn sequence construction. *Discrete Math.*, 339(1):127–131, 2016.
- [220] B. Schieber. Computing a minimum weight-link path in graphs with the concave Monge property. *J. Algorithms*, 29(2):204–222, 1998.
- [221] M. Sciortino and L. Q. Zamboni. Suffix automata and standard Sturmian words. In T. Harju, J. Karhumäki and A. Lepistö, eds., *Developments in Language Theory, 11th International Conference, DLT 2007, Turku, Finland, 3–6 July, 2007*, Proceedings, vol. 4588, *Lecture Notes in Computer Science*, pp. 382–398. Springer, 2007.
- [222] J. Shallit. On the maximum number of distinct factors of a binary string. *Graphs Combinat.*, 9(2–4):197–200, 1993.
- [223] Y. Shiloach. A fast equivalence-checking algorithm for circular lists. *Inf. Process. Lett.*, 8(5):236–238, 1979.
- [224] R. M. Silva, D. Pratas, L. Castro, A. J. Pinho and P. J. S. G. Ferreira. Three minimal sequences found in ebola virus genomes and absent from human DNA. *Bioinformatics*, 31(15):2421–2425, 2015.
- [225] I. Simon. String matching algorithms and automata. In R. Baeza-Yates and N. Ziviani, eds., *Proceedings of the 1st South American Workshop on String Processing*, pp. 151–157, Belo Horizonte, Brasil, 1993. Universidade Federal de Minas Gerais.
- [226] S. S. Skiena. *The Algorithm Design Manual*. 2nd ed., Springer, 2008.
- [227] T. Skolem. On certain distributions of integers in pairs with given differences. *Math. Scand.*, 5:57–68, 1957.

- [228] B. Smyth. *Computing Patterns in Strings*. Pearson Education Limited, 2003.
- [229] M. Szykula. Improving the upper bound on the length of the shortest reset word. In R. Niedermeier and B. Vallée, eds., *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018*, 28 February – 3 March 2018, Caen, France, vol. 96, LIPIcs, pp. 56:1–56:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [230] J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theor. Comput. Sci.*, 57:131–145, 1988.
- [231] Z. Troníček and B. Melichar. Directed acyclic subsequence graph. In J. Holub and M. Simánek, eds., *Proceedings of the Prague Stringology Club Workshop 1998*, Prague, Czech Republic, 3–4 September, 1998, pp. 107–118. Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University, 1998.
- [232] Z. Troníček and A. Shinohara. The size of subsequence automaton. *Theor. Comput. Sci.*, 341(1–3):379–384, 2005.
- [233] K. Tsuruta, S. Inenaga, H. Bannai and M. Takeda. Shortest unique substrings queries in optimal time. In V. Geffert, B. Preneel, B. Rován, J. Stuller and A. M. Tjoa, eds., *SOFSEM 2014: Theory and Practice of Computer Science: 40th International Conference on Current Trends in Theory and Practice of Computer Science*, Nový Smokovec, Slovakia, 26–29 January 2014, Proceedings, vol. 8327, Lecture Notes in Computer Science, pp. 503–513. Springer, 2014.
- [234] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.
- [235] J. van Leeuwen. On the construction of Huffman trees. In *ICALP*, pp. 382–410, 1976.
- [236] J. S. Vitter. Design and analysis of dynamic Huffman codes. *J. ACM*, 34(4):825–845, 1987.
- [237] N. Vörös. On the complexity measures of symbol-sequences. In A. Iványi, ed., *Conference of Young Programmers and Mathematicians*, pp. 43–50, Budapest, 1984. Faculty of Sciences, Eötvös Loránd University.
- [238] B. Walczak. A simple representation of subwords of the Fibonacci word. *Inf. Process. Lett.*, 110(21):956–960, 2010.
- [239] T. A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.
- [240] W. A. Wythoff. A modification of the game of Nim. *Nieuw Arch. Wisk.*, 8:199–202, 1907/1909.
- [241] I.-H. Yang, C.-P. Huang and K.-M. Chao. A fast algorithm for computing a longest increasing subsequence. *Inf. Process. Lett.*, 93(5):249–253, 2005.
- [242] E. Zalinescu. Shorter strings containing all k-element permutations. *Inf. Process. Lett.*, 111(12):605–608, 2011.
- [243] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory*, 23(3):337–343, 1977.