# Algorytmiczne i kombinatoryczne problemy związane ze zliczaniem powtórzeń w słowach

autoreferat rozprawy doktorskiej

Jakub Radoszewski
Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

Powtórzenia w słowach stanowią jedno z fundamentalnych zagadnień w kombinatoryce tekstów. Prace prekursora tej dziedziny, A. Thuego [42], pochodzą z początków ubiegłego wieku. Problematyka powtórzeń w słowach pojawia się w różnych dziedzinach matematyki i informatyki, w szczególności w algorytmach wyszukiwania wzorca w tekście, teorii automatów, teorii języków formalnych, metodach kompresji tekstów, biologii obliczeniowej itd., patrz też [2, 15, 33]. Bardziej szczegółowy opis zastosowań można znaleźć np. w pracy przeglądowej [8].

Najważniejszymi typami powtórzeń w słowach są kwadraty, wyższe potęgi oraz maksymalne powtórzenia. Analizę powtórzeń prowadzi się m.in. w kontekście kombinatorycznym – wyznaczanie oszacowań na maksymalną liczbę powtórzeń danego typu w słowie zadanej długości – oraz w kontekście algorytmicznym – projektowanie algorytmów służących do efektywnego wyszukiwania podsłów odpowiadających poszczególnym typom powtórzeń. Niniejsza rozprawa stanowi przyczynek do tej dziedziny.

Szczególny nacisk kładziemy na analizę struktury powtórzeń silnie okresowych: podsłów będących wyższymi potęgami oraz maksymalnych powtórzeń sześciennych. Uzyskujemy łatwiejsze i dokładniejsze oszacowania niż w przypadku powtórzeń o długich okresach. Otrzymane narzędzia pozwalają odkryć nowe własności ogólnych maksymalnych powtórzeń. Wskazujemy także nowe zastosowania algorytmiczne maksymalnych powtórzeń.

## Definicje

Rozważamy słowa będące skończonymi ciągami symboli należących do skończonego zbioru zwanego alfabetem. Dla danego słowa $u = u_1 u_2 \dots u_n$, przez $u[i\,..\,j]$ oznaczamy podsłowo równe $u_i \dots u_j$ (w szczególności $u[i] = u[i\,..\,i]$). Podsłowa typu $u[1\,..\,i]$ nazywamy prefiksami słowa $u$, a podsłowa typu $u[i\,..\,n]$ – sufiksami słowa $u$.

*Potęgą* słowa $w$ o wykładniku $k$, dla $k \geqslant 0$ całkowitego, nazywamy słowo $u = w^k$, czyli $u = ww \dots w$ ($k$-krotne sklejenie). *Kwadratem* nazywamy drugą potęgę (niepustego) słowa, podobnie *sześcianem* nazywamy trzecią potęgę (niepustego) słowa. Dla danego słowa $u$, przez $\mathsf{squares}(u)$ oznaczamy liczbę różnych kwadratów będących podsłowami słowa $u$, analogicznie przez $\mathsf{cubes}(u)$ oznaczamy liczbę różnych sześcianów w słowie $u$. Przez $\mathsf{squares}(n)$ oraz $\mathsf{cubes}(n)$ oznaczamy maksymalną wartość funkcji $\mathsf{squares}(u)$ i $\mathsf{cubes}(u)$ po wszystkich słowach $u$ długości $n$. Przykładowo, $\mathsf{squares}(10) = 6$, a słowem realizującym to maksimum jest np. aababbabba, które zawiera sześć różnych kwadratów: aa, bb, abab, abbabb, babbab i bbabba (należy podkreślić, że kwadrat bb jest liczony tylko raz). Podobnie, zachodzi $\mathsf{cubes}(10) = 3$, czemu odpowiada przykładowe słowo aaaabababa zawierające następujące sześciany: aaa, ababab i bababa.

Powiemy, że liczba $p$ jest (najkrótszym) *okresem* słowa $u = u_1u_2 \dots u_n$, jeżeli $p$ jest najmniejszą liczbą całkowitą dodatnią spełniającą $u_i = u_{i+p}$ dla każdego $1 \leqslant i \leqslant n - p$. *Maksymalnym powtórzeniem* (ang. *run*) w słowie $u$ nazywamy przedział $[i, j]$, taki że okres $p$ podsłowa $u[i \mathinner{.\,.} j]$ mieści się co najmniej dwukrotnie w tym podsłowie (tj. zachodzi warunek $2p \leqslant j - i + 1$), a ponadto przedział $[i, j]$ wyznacza maksymalne podsłowo o okresie $p$, tzn. $u[i - 1] \neq u[i + p - 1]$ oraz $u[j - p + 1] \neq u[j + 1]$, jeśli w ogóle litery $u[i - 1]$ i $u[j + 1]$ istnieją. *Wykładnikiem* maksymalnego powtórzenia nazywamy ułamek $(j - i + 1)/p$. Maksymalne powtórzenie o wykładniku nie mniejszym niż 3 nazywamy *maksymalnym powtórzeniem sześciennym*. Przykładowo, jednym z maksymalnych powtórzeń w słowie baabaababaababaababab jest abaababaababaababa o okresie 5 i wykładniku $3\frac{3}{5}$; jest to maksymalne powtórzenie sześcienne.

Przez $\mathsf{runs}(u)$ i $\mathsf{cubic\text{-}runs}(u)$ oznaczamy liczbę maksymalnych powtórzeń i, odpowiednio, liczbę maksymalnych powtórzeń sześciennych w słowie $u$. Definiujemy też funkcje $\mathsf{exp\text{-}runs}(u)$ i $\mathsf{exp\text{-}cubic\text{-}runs}(u)$ jako sumę wykładników maksymalnych powtórzeń oraz maksymalnych powtórzeń sześciennych w $u$. Dla $n$ będącego dodatnią liczbą całkowitą, przez $\mathsf{runs}(n)$, $\mathsf{cubic\text{-}runs}(n)$, $\mathsf{exp\text{-}runs}(n)$ oraz $\mathsf{exp\text{-}cubic\text{-}runs}(n)$ oznaczamy maksimum z odpowiedniej funkcji po wszystkich słowach długości $n$. Mamy, na przykład, $\mathsf{runs}(10) = 6$, a jednym ze słów dających taką liczbę maksymalnych powtórzeń jest aababaabab: są to, mianowicie, aa (występujące dwukrotnie), ababa, abab, abaaba i całe słowo aababaabab. To samo słowo akurat realizuje maksimum $\mathsf{exp\text{-}runs}(10) = 12{,}5$, ale w ogólnym przypadku taka zbieżność nie musi zachodzić.

# Wyniki rozprawy na tle dziedziny

Najważniejsze znane oszacowania dotyczące potęg, przedstawione w pracy [20], są związane z funkcją $\mathsf{squares}(n)$: $n - o(n) \leqslant \mathsf{squares}(n) \leqslant 2n$. Od tego czasu (13 lat) wielokrotnie podejmowano próby zmniejszenia luki pomiędzy ograniczeniem dolnym a górnym, patrz np. [17, 19], ale jedynym faktycznym usprawnieniem, jakie udało się osiągnąć, było oszacowanie $\mathsf{squares}(n) \leqslant 2n - \Theta(\log n)$ zawarte w pracy [26]. Wyniki eksperymentalne sugerują, że $\mathsf{squares}(n) < n$. Dokładne oszacowania udało się uzyskać w przypadku słów Fibonacciego (zdefiniowanych rekurencyjnie jako: $F_0 = \mathsf{a}$, $F_1 = \mathsf{ab}$, $F_n = F_{n-1}F_{n-2}$ dla $n \geqslant 2$), patrz [21, 27], oraz ogólniejszych słów Sturma, patrz [16, 35, 36].

W tej rozprawie przedstawiamy dowody następujących oszacowań dla podobnie zdefiniowanej funkcji $\mathsf{cubes}(n)$:

$$0{,}5\,n - 2\sqrt{n} \leqslant \mathsf{cubes}(n) \leqslant 0{,}8\,n.$$

W ten sposób osiągamy ponadtrzykrotnie mniejszą lukę między ograniczeniami niż w przypadku funkcji $\mathsf{squares}(n)$. Dowód dolnego ograniczenia polega na analizie wystąpień sześcianów w pewnej nieskończonej sekwencji słów binarnych. Natomiast górne ograniczenie wynika z analizy własności okresowych skrajnie prawych wystąpień sześcianów w słowie. Wyniki dotyczące sześcianów zostały przedstawione w **rozdziale 3** rozprawy.

W zakresie kombinatoryki maksymalnych powtórzeń w słowach kluczowym faktem jest to, że $\mathsf{runs}(n) = O(n)$. Ten wynik pochodzi z prac [28, 29], w których pokazano jedynie taką zależność asymptotyczną, natomiast nie udało się wskazać żadnego konkretnego współczynnika w tej zależności. W wyniku prac szeregu autorów, patrz [6, 7, 9, 23, 24, 32, 37, 38, 40, 41], udało się uzyskać dużo dokładniejsze ograniczenia związane z tą funkcją: $0{,}944575712\,n \leqslant \mathsf{runs}(n) \leqslant 1{,}029\,n$. Trzeba w tym miejscu nadmienić, że wszystkie dowody faktu liniowego rzędu funkcji $\mathsf{runs}(n)$ są stosunkowo skomplikowane i wymagają rozpatrzenia licznych przypadków. Znów, eksperymenty numeryczne sugerują hipotezę, że $\mathsf{runs}(n) < n$ (patrz [28, 29]). Podobnie jak w przypadku kwadratów, także tutaj były możliwe dokładne oszacowania w przypadku klasycznych rodzin słów Fibonacciego [27, 28, 29, 39] i Sturma [1, 22, 35].

Silniejszą własnością kombinatoryczną maksymalnych powtórzeń w słowach jest fakt, że $\mathsf{exp\text{-}runs}(n) = O(n)$, po raz pierwszy udowodniony w pracach [28, 29]. W przypadku tej funkcji dotychczas jedynym konkretnym oszacowaniem było ograniczenie górne $5{,}6\,n$ pochodzące z pracy [7]. W tej samej pracy znajdowała się, poparta analizą komputerową, hipoteza, że $\mathsf{exp\text{-}runs}(n) \leqslant 2{,}9\,n$, natomiast znaną hipotezą, pochodzącą jeszcze z przełomu XX i XXI wieku, było, że $\mathsf{exp\text{-}runs}(n) < 2n$ (patrz [29, 30]).

W **rozdziale 4** rozprawy przeprowadzamy analizę kombinatoryczną maksymalnych powtórzeń sześciennych. W jej wyniku udaje nam się uzyskać następujące oszacowania:

$$0{,}41\,n < \mathsf{cubic\text{-}runs}(n) < 0{,}5\,n.$$

Dowód górnego ograniczenia, a zatem faktu, że $\mathsf{cubic\text{-}runs}(n) = O(n)$, wykorzystuje jedynie podstawowe własności kombinatoryczne słów Lyndona (czyli minimalnych lub maksymalnych leksykograficznie słów pierwotnych) i jest stosunkowo krótki. Z kolei dolne ograniczenie zostało skonstruowane z użyciem odpowiednio zmodyfikowanych słów Fibonacciego. Przy tym dla standardowych słów Fibonacciego otrzymaliśmy oszacowanie $\mathsf{cubic\text{-}runs}(F_n) \approx 0{,}2361\,n$, a wspomniana modyfikacja tych słów została wyznaczona metodami heurystycznymi bazującymi na algorytmach genetycznych. Dodatkowo, w przypadku słów binarnych pokazujemy lepsze górne ograniczenie $0{,}48\,n$.

Korzystając z narzędzi opartych na własnościach słów Lyndona, w **rozdziale 5** poprawiamy dotychczas znane oszacowania na funkcję $\mathsf{exp\text{-}runs}(n)$, otrzymując:

$$2{,}035\,n < \mathsf{exp\text{-}runs}(n) < 4{,}1\,n.$$

W szczególności, obalamy hipotezę Kolpakova i Kucherova [29, 30], zgodnie z którą miało zachodzić $\mathsf{exp\text{-}runs}(n) < 2n$. Wskazujemy mianowicie rodzinę słów binarnych, również wygenerowaną metodami heurystycznymi, w której suma wykładników maksymalnych powtórzeń przekracza $2{,}035\,n$. Uzyskujemy także lepsze ograniczenie $\mathsf{exp\text{-}cubic\text{-}runs}(n) < 2{,}5\,n$.

Znana jest cała gama algorytmów zliczania i wyznaczania powtórzeń w słowach. Liniową złożoność czasową mają algorytmy sprawdzania, czy słowo zawiera jakikolwiek kwadrat, pochodzące z klasycznych już prac [5, 34] oraz względnie nowej pracy [4] rozwiązującej ten problem w wersji *on-line*, znajdowania wszystkich kwadratów w słowie [25], a także wyznaczania wszystkich maksymalnych powtórzeń w słowie [28, 29, 30]. Niestety, liniowe wyznaczanie kwadratów w słowie wymagało dotychczas (algorytm z pracy [25]) bardzo skomplikowanych analiz, a ponadto wykorzystywało drzewa sufiksowe, które są obarczone dużą stałą czasową i pamięciową. Natomiast znane są bardziej praktyczne niż w [28, 29, 30] algorytmy wyznaczające wszystkie maksymalne powtórzenia, w szczególności, używające tablic sufiksowych zamiast drzew sufiksowych, patrz praca [3].

W **rozdziale 6** wskazujemy związki między obydwoma wymienionymi problemami, dzięki którym uzyskujemy liniowy algorytm wyznaczania wszystkich kwadratów w słowie i, ogólniej, dowolnych potęg o ustalonym wykładniku, na podstawie maksymalnych powtórzeń w słowie. Oprócz prostoty, przewagą naszego algorytmu nad tym przedstawionym w pracy [25] jest wykorzystanie spośród tekstowych struktur danych jedynie tablic sufiksowych. Używając tego samego podejścia, otrzymujemy zwarty wzór na liczbę wszystkich wystąpień potęg w słowie, jako funkcję maksymalnych powtórzeń.

Mimo że zbiór maksymalnych powtórzeń stanowi wygodną, liniową reprezentację wszystkich powtórzeń w słowie, dotychczas znanych było bardzo niewiele wyników wykorzystujących ten fakt (najprostsze z nich, dotyczące wystąpień potęg, zajmują zaledwie stronę w pracach [28, 29]). Z tego względu przedstawiamy jeszcze jedno zastosowanie tego wyniku: liniowy algorytm wyznaczania wszystkich okresów lokalnych w słowie. *Okresem lokalnym* na pozycji $i$ w słowie $u$ nazywamy długość okresu najkrótszego kwadratu wyśrodkowanego na pozycji $i$. Przy tym mówimy, że kwadrat $ww$ jest wyśrodkowany na pozycji $i$ słowa $u$, jeśli dla $x = u[1\mathinner{.\,.} i]$ oraz $y = u[i+1\mathinner{.\,.} n]$ zachodzą dwa warunki: (a) $w$ jest sufiksem słowa $x$ lub $x$ jest sufiksem słowa $w$; (b) $w$ jest prefiksem słowa $y$ lub $y$ jest prefiksem słowa $w$.

Okresy lokalne mają związek z tzw. faktoryzacją krytyczną słowa, więcej na ten temat można przeczytać np. w książce [15]. Wcześniej znany był liniowy algorytm wyznaczania wszystkich okresów lokalnych [18], w którym zastosowano wiele różnych technik algorytmicznych z nietrywialnymi modyfikacjami. Liniowy algorytm przedstawiony w tej rozprawie wykorzystuje, oprócz maksymalnych powtórzeń, liniowe rozwiązanie klasycznego problemu zarysu wieżowców na Manhattanie oraz dobrze znaną w algorytmice tekstów funkcję prefiksową z algorytmu Knutha-Morrisa-Pratta.

# Podsumowanie

W rozprawie przedstawiamy nowe wyniki kombinatoryczne i algorytmiczne dotyczące powtórzeń w słowach. Koncentrujemy się na strukturze powtórzeń silnie okresowych (sześcianów i maksymalnych powtórzeń sześciennych), dla których ustanawiamy dokładniejsze oszacowania niż w przypadku powtórzeń o długich okresach. Na tej podstawie udaje nam się polepszyć dotychczas znane ograniczenia na maksymalną sumę wykładników maksymalnych powtórzeń w słowie. Przedstawiamy także nowe liniowe algorytmy: zliczania kwadratów i wyższych potęg, znajdowania wszystkich potęg i wyznaczania okresów lokalnych w słowie, w których kluczową rolę odgrywa struktura maksymalnych powtórzeń w słowie. Przedstawione wyniki zostały wcześniej opublikowane w pracach [10, 11, 12, 13, 14, 31].

# Literatura


[1] P. Baturo, M. Piatkowski, and W. Rytter. The number of runs in Sturmian words. In O. H. Ibarra and B. Ravikumar, editors, *CIAA*, volume 5148 of *Lecture Notes in Computer Science*, pages 252–261. Springer, 2008.

[2] J. Berstel and J. Karhumäki. Combinatorics on words: a tutorial. *Bulletin of the EATCS*, 79:178–228, 2003.

[3] G. Chen, S. J. Puglisi, and W. F. Smyth. Fast and practical algorithms for computing all the runs in a string. In B. Ma and K. Zhang, editors, *CPM*, volume 4580 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 2007.

[4] G.-H. Chen, J.-J. Hong, and H.-I. Lu. An optimal algorithm for online square detection. In A. Apostolico, M. Crochemore, and K. Park, editors, *CPM*, volume 3537 of *Lecture Notes in Computer Science*, pages 280–287. Springer, 2005.

[5] M. Crochemore. Transducers and repetitions. *Theor. Comput. Sci.*, 45(1):63–86, 1986.

[6] M. Crochemore and L. Ilie. Analysis of maximal repetitions in strings. In L. Kucera and A. Kucera, editors, *MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 465–476. Springer, 2007.

[7] M. Crochemore and L. Ilie. Maximal repetitions in strings. *J. Comput. Syst. Sci.*, 74(5):796–807, 2008.

[8] M. Crochemore, L. Ilie, and W. Rytter. Repetitions in strings: Algorithms and combinatorics. *Theor. Comput. Sci.*, 410(50):5227–5235, 2009.

[9] M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the "runs" conjecture. In P. Ferragina and G. M. Landau, editors, *CPM*, volume 5029 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2008.

[10] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, and T. Walen. Extracting powers and periods in a string from its runs structure. In E. Chávez and S. Lonardi, editors, *SPIRE*, volume 6393 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2010.

[11] 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. H. Dediu, H. Fernau, and C. Martín-Vide, editors, *LATA*, volume 6031 of *Lecture Notes in Computer Science*, pages 227–238. Springer, 2010.

[12] 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. System Sci.*, 2011, doi: 10.1016/j.jcss.2011.12.005.

[13] M. Crochemore, M. Kubica, J. Radoszewski, W. Rytter, and T. Walen. On the maximal sum of exponents of runs in a string. In C. S. Iliopoulos and W. F. Smyth, editors, *IWOCA*, volume 6460 of *Lecture Notes in Computer Science*, pages 10–19. Springer, 2010.

[14] M. Crochemore, M. Kubica, J. Radoszewski, W. Rytter, and T. Walen. On the maximal sum of exponents of runs in a string. *Journal of Discrete Algorithms*, 2011, doi: 10.1016/j.jda.2011.12.016.

[15] M. Crochemore and W. Rytter. *Jewels of Stringology.* World Scientific, 2003.

[16] D. Damanik and D. Lenz. Powers in Sturmian sequences. *Eur. J. Comb.*, 24(4):377–390, 2003.

[17] A. Deza, F. Franek, and M. Jiang. A *d*-step approach for distinct squares in strings. In R. Giancarlo and G. Manzini, editors, *CPM*, volume 6661 of *Lecture Notes in Computer Science*, pages 77–89. Springer, 2011.

[18] J.-P. 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.

[19] K. Fan, S. J. Puglisi, W. F. Smyth, and A. Turpin. A new periodicity lemma. *SIAM J. Discrete Math.*, 20(3):656–668, 2006.

[20] A. S. Fraenkel and J. Simpson. How many squares can a string contain? *J. of Combinatorial Theory Series A*, 82:112–120, 1998.

[21] A. S. Fraenkel and J. Simpson. The exact number of squares in Fibonacci words. *Theor. Comput. Sci.*, 218(1):95–106, 1999.

[22] F. Franek, A. Karaman, and W. Smyth. Repetitions in Sturmian strings. *Theoretical Computer Science*, 249(2):289 – 303, 2000.

[23] F. Franek and Q. Yang. An asymptotic lower bound for the maximal number of runs in a string. *Int. J. Found. Comput. Sci.*, 19(1):195–203, 2008.

[24] M. Giraud. Not so many runs in strings. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *LATA*, volume 5196 of *Lecture Notes in Computer Science*, pages 232–239. Springer, 2008.

[25] 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.

[26] L. Ilie. A note on the number of squares in a word. *Theoretical Computer Science*, 380:373–376, 2007.

[27] 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.

[28] R. M. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 596–604, 1999.

[29] R. M. Kolpakov and G. Kucherov. On maximal repetitions in words. *Journal of Discrete Algorithms*, 1:159–186, 1999.

[30] R. M. Kolpakov and G. Kucherov. On the sum of exponents of maximal repetitions in a word. *Tech. Report 99-R-034, LORIA*, 1999.

[31] M. Kubica, J. Radoszewski, W. Rytter, and T. Walen. On the maximal number of cubic subwords in a string. In J. Fiala, J. Kratochvíl, and M. Miller, editors, *IWOCA*, volume 5874 of *Lecture Notes in Computer Science*, pages 345–355. Springer, 2009.

[32] K. Kusano, W. Matsubara, A. Ishino, H. Bannai, and A. Shinohara. New lower bounds for the maximum number of runs in a string. *CoRR*, abs/0804.1214, 2008.

[33] M. Lothaire. *Combinatorics on Words*. Addison-Wesley, Reading, MA., U.S.A., 1983.

[34] M. G. Main and R. J. Lorentz. Linear time recognition of squarefree strings. In A. Apostolico and Z. Galil, editors, *Combinatorial Algorithms on Words*, volume F12 of *NATO ASI Series*, pages 271–278. Springer-Verlag, 1985.

[35] M. Piatkowski. *Efficient algorithms related to combinatorial structure of words*. PhD thesis, Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, 2011.

[36] M. Piatkowski and W. Rytter. Asymptotic behaviour of the maximal number of squares in standard Sturmian words. In *Proceedings of the Prague Stringology Conference, PSC*, pages 237–248, 2009.

[37] 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.

[38] W. Rytter. The number of runs in a string: Improved analysis of the linear upper bound. In B. Durand and W. Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 184–195. Springer, 2006.

[39] W. Rytter. The structure of subword graphs and suffix trees in Fibonacci words. *Theor. Comput. Sci.*, 363(2):211–223, 2006.

[40] W. Rytter. The number of runs in a string. *Inf. Comput.*, 205(9):1459–1469, 2007.

[41] J. Simpson. Modified Padovan words and the maximum number of runs in a word. *Australasian J. of Comb.*, 46:129–145, 2010.

[42] A. Thue. Uber unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I Math-Nat.*, 7:1–22, 1906.