Arbitrary-length analogs to de Bruijn sequences

0

01

011

0011

00111

001011

0010111

00010111

000101111

0001011101

00010111101

000110111001

0001101111001

00011010111001

000110101111001

0000110111100101

@AbhiNellore

work w/ Rachel Ward

CPM 2022, Prague, 2022-06-27

What to get out of this talk

- A different way to think about de Bruijn sequences that leads to a natural generalization

- Intuitive understanding of Lempel's lift (a useful tool in combinatorics on words) and 2 key tricks

Terminology

sequence: in this talk, a circular word on some alphabet of size *K*, generally [*K*] := {0, 1, ..., *K*-1}

substring: string of contiguous characters in a sequence

example sequence:

01101011

length-3 substrings:

011
110
101
010
101
011
110
101

Consider drawing each character of a length-L sequence uniformly at random from $[K]$.

Equally likely for any length-m string to be a substring at any position

$\implies \dfrac{L}{K^m} =$ expected number of occurrences of any length-m string as a substring

Can we write a random-like sequence?

Sure, almost!

Canonical example: de Bruijn sequence of order *n* on [*K*]

- length $L = K^n$
- every possible length-*n* string occurs exactly once as substring
- optimally short

But if every possible length-*n* string occurs exactly once as a substring...

then every length-*(n-1)* string occurs $K = L/K^{n-1}$ times as a substring,

every length-*(n-2)* string occurs $K^2 = L/K^{n-2}$ times as a substring,

.

.

.

, and every length-1 string occurs $K^{n-1} = L/K$ times as a substring.

de Bruijn sequence: binary example

$n = 3$

00010111

length-3 substrings in order:

000

001

010

101

011

111

110

100

Q: What's happening in an order-*n* de Bruijn sequence for substring sizes larger than *n*?

n = 3

00010111

length-4 strings

ins	outs
0001	0000
0010	0011
0101	0100
1011	1010
0111	0110
1110	1111
1100	1101
1000	1001

A: There's either 0 or 1 instance(s) of any given string.

So a de Bruijn sequence can't be perfectly "random-like"

$\dfrac{K^n}{K^m}$ isn't always an integer. But it's close enough...

that is

An order-*n* de Bruijn sequence on [*K*] is a sequence of length K^n s.t. any string on [*K*] of length $m \leq K^n$ occurs either $\lfloor K^n/K^m \rfloor$ or $\lceil K^n/K^m \rceil$ times as a substring.

Generalize this to arbitrary lengths

$$\frac{L}{K^m}$$ isn't always an integer,

so ask for the next-best thing

A $P_L^{(K)}$-sequence on [*K*] is a sequence of length L s.t. any string on [*K*] of length $m \leq L$ occurs either $\lfloor L/K^m \rfloor$ or $\lceil L/K^m \rceil$ times as a substring.

de Bruijn sequences exist for all possible combinations of *n* and *K* (proved first by explicit construction by Monroe Martin in 1934[1])

Do $P_L^{(K)}$-sequences exist for all possible combos of *K* and *L*?

Yes. We proved this by obtaining an $O(L)$-time, $O(L \log K)$-space general algorithm for construction.

[1] M. H. Martin, "A problem in arrangements," Bulletin of the American Mathematical Society, vol. 40, no. 12, pp. 859–864, 1934.

[2] A. Nellore and R. Ward, "Arbitrary-length analogs to de Bruijn sequences," arXiv:2108.07759, 2021. (This is the CPM 2022 paper I'm talking about now.)

How to de Bruijn sequence

- Greedy approaches
- Shift rules
- Concatenation
- Recursive

(Check out Joe Sawada's page debruijnsequence.org)

How to de Bruijn sequence

- Greedy approaches
- Shift rules
- Concatenation
- Recursive

Let's focus on one of these approaches

Tool: Lempel's lift (inverse of Lempel's D-morphism[1,2,3,4])

Gist: take an integral of a sequence (mod K)

[1] A. Lempel, "On a homomorphism of the de Bruijn graph and its applications to the design of feedback shift registers," IEEE Transactions on Computers, vol. 100, no. 12, pp. 1204–1209, 1970.

[2] C. Ronse, "Feedback shift registers," Lecture Notes in Computer Science, vol. 169, 1984.

[3] J. Tuliani, "De Bruijn sequences with efficient decoding algorithms," Discrete Mathematics, vol. 226, no. 1-3, pp. 313–336, 2001.

[4] A. Alhakim and M. Akinwande, "A recursive construction of nonbinary de Bruijn sequences," Designs, Codes and Cryptography, vol. 60, no. 2, pp. 155–169, 2011.

Tool: Lempel's lift (inverse of Lempel's D-morphism[1,2,3,4])

Given a sequence $s[j]$ on [*K*], write the integral

$$\mathscr{L}[r] := constant + \sum_{j=0}^{\ell} s[r] \pmod{K}$$

This effects an invertible 1-to-*K* map from the set of length-*m* strings into the set of length-*(m+1)* strings, for every *m*.

Binary case: 1-to-2 map of length-m strings into length-(m+1) strings

If I'm integrating a sequence, and I come upon a substring, my sum so far is either 1 or 0 (depending on the integration constant)

0 -> {00, 11}
010 -> {0011, 1100}
0110 -> {00100, 11011}

Can invert with derivative: take successive differences. "Lempel's D-morphism" or "Lempel's homomorphism": structure-preserving

$\implies$ we can make bigger de Bruijn sequences by integrating smaller de Bruijn sequences (mod K)!

Binary example

Integrate 1 1111
-> 0101

Order-1 de Bruijn!

Binary example

Integrate 01 01010101

-> 01100110

Order-2 de Bruijn!

Binary example

Integrate 0110

-> 0100
   1011

Two cycles. What now?

First trick: cycle joining[1]

Concatenate rotations of cycles whose prefixes are the same length-(n-1) string. Natural choice: alternation between 0s and 1s.

Rotate 0100, 1011 to get

0100 + 0111 = 01000111

Order-3 de Bruijn!

[1] S. W. Golomb, Shift Register Sequences: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Models. World Scientific, 2017.

Binary example

Integrate 01000111

-> 01111010
10000101

Rotate to alternating prefixes and join cycles

-> 10100111 + 10110000 = 1010011110110000

Order-4 de Bruijn!

General binary case

- When a cycle has an odd # of 1's, integral is unique (and self-dual)

- When a cycle has an even # of 1's, there are two integrals that are complements

- Total length of integrals always doubles the cycle length

How to $P_L^{(2)}$-sequence: second trick (binary case)

Insert a 1 in the longest string of 1's as necessary after integrating and joining cycles.

"as necessary": insert after *i*th integration iff (*N-i*)th place of binary rep of *L* is 1, as

$$L = \sum_{i=0}^{N} d_i 2^{N-i}$$

Say we want a $P^{(2)}_{13}$-sequence.

In binary, L = 1101

1) Integrate 1: 01
2) Insert 1 in longest string of 1s: 011
3) Integrate 011: 010, 101
4) Rotate and join cycles: 010011
5) Insert nada.
6) Integrate 011010: 010011101100
7) Insert 1 in longest string of 1s: 0100111101100

final sequence

Check: does 0100111101100 have $\lfloor 13/2^m \rfloor$ or $\lceil 13/2^m \rceil$ instances of every length-m binary string for $m \leq L$?

length-1:

6 0's
7 1's

✓

length-2:

3 00's
3 01's
3 10's
4 11's

✓

length-3:

1 000
2 001's
1 010
2 011's
2 100's
1 101
2 110's
2 111's

✓

length-4:

1 0010
1 0011
1 0110
1 0111
1 1000
1 1001
1 1100
1 1101
1 1110
1 1111

1 0001
1 0100
1 1011
0 0000
0 0101
0 1010

✓

Why does this method of generating $P_L^{(2)}$-sequences work?

- Integration preserves $P_L^{(2)}$-ness
  - Each length-*m* string occurrence gets mapped to 2 distinct length-(*m*+1) substrings while the total length of cycles is 2x'd (key to inductive proof)
- Joining/1-insertion preserves $P_L^{(2)}$-ness
  - Substring content at lengths $\leq \lceil \log_2 L \rceil$ is the same, except we may go from 0 instances of the length-$\lceil \log_2 L \rceil$ string 11...1 to 1 instance

How to $P_L^{(K)}$-sequence

- First trick: cycle joins can always happen at substrings that look like (*i*, *i*+1, *i*+2, ...) (mod *K*)
    - Or off by one from it, or of length -1 from usual
- Second trick: Base-2 rep is replaced by base-*K* rep; a digit d_i dictates a # of insertions performed after integration/joins, each in a longest string of the characters {1, 2, ..., d_i}

- Start integration with 12...d_0 for d_0 the most sig digit. There can be up to *K* cycles after integration

$P_{16}^{(3)}$-sequence example

In ternary, L = 121

1) Integrate 1: 012
2) Insert 1 (2) in a longest string of 1's (2's): 01122
3) Integrate 01122: 01210, 12021, 20102
4) Rotate and join cycles: 12100+12021, then 0120211210+01022 = 012021121001022
5) Insert 1 in longest string of 1s: 0120211121001022

final sequence

Why does this method of generating $P_L^{(K)}$-sequences work?

- Integration preserves $P_L^{(K)}$-ness
    - Each length-*m* string occurrence gets mapped to *K* length-(*m*+1) string occurrences while the total length of cycles is *K* x'd (key to inductive proof)
- Joining/insertion preserves $P_L^{(K)}$-ness
    - Substring content at lengths $\leq \lceil \log_K L \rceil$ is the same, except we go from 0 instances of each of some length-$\lceil \log_K L \rceil$ strings that look like *cc*...*c* to 1 instance

tinyurl.com/pklseq for code

Open questions

- What else is integration+tricks good for? (Orientable sequence generation[1], at least.)
- Can we make $P_L^{(K)}$-sequences that are efficiently decodable?
- How many $P_L^{(K)}$-sequences are there at a given combination of K and L?
- Can we write a shift rule that generates a $P_L^{(K)}$ -sequence without taking up lots of space?

[1] C. J. Mitchell and P. B. Wild, "Constructing orientable sequences," arXiv preprint arXiv:2108.03069, 2021.

Acknowledgements

- Rachel Ward (collab) & Oden Institute @ U of Texas
- Shunsuke Inenaga
- Anonymous reviewers for CPM 2022
- Michael Didas
- Organizing committee: Jan Holub, Hideo Bannai + Jan Trávníček, Ondřej Guth, Tomáš Pecka, Eliška Šestáková

Looser restrictions that a $P_L^{(K)}$-sequence satisfies

- Lempel-Radchenko sequences (cut-down de Bruijns): sequence of length L such that each length-$\lceil \log_K L \rceil$ substring is distinct.[1,2]
- Generalized de Bruijn sequences: Lempel-Radchenko sequence such that there is at least one instance of every length-m string for $m < \log_K L$.[3]

[1] A. Radchenko, Code Rings and Their Use in Contactless Coding Devices. PhD thesis, University of Leningrad, USSR, 1958.

[2] A. Lempel, "m-ary closed sequences," Journal of Combinatorial Theory, Series A, vol. 10, no. 3, pp. 253–258, 1971.

[3] D. Gabric, Š. Holub, and J. Shallit, "Generalized de Bruijn words and the state complexity of conjugate sets," in International Conference on Descriptional Complexity of Formal Systems, pp. 137–146, Springer, 2019.

Related literature: "normal periodic systems"

- A normal number is an irrational for which, in base K, for which any length-*m* string of digits has density K^{-m}.
  - Infinite analog to what we called $P_L^{(K)}$-sequences
  - Most numbers normal[1], but is π?
- Turns out work in '60s, '70s by Korobov[2], Stoneham[3] asks for reduced proper fractions in base *K*, how far does the repetend deviate from "normality"?

[1] E. Borel. "Les probabilités dénombrables et leurs applications arithmétiques", Rendiconti del Circolo Matematico di Palermo, 27: 247–271, 1909.

[2] N.M. Korobov. On the distribution of digits in periodic fractions. *Mathematics of the USSR-Sbornik*, *18*(4), 659, 1972.

[3] R. Stoneham. Normal recurring decimals, normal periodic systems, (j, ε)-normality, and normal numbers. *Acta Arithmetica*, *28*(4), 349-361, 1976.

Related literature: "normal periodic systems"

- Different from our perspective/terminology: we fix *L*, and they accept whatever *L* they get. Also, they're generally willing to accept $O(1)$ deviations from normality for some substring lengths as quite normal indeed.

- Flexibility of their constructions are limited, but at special *K*, *L* they do prove $P_L^{(K)}$-ness happens!

Related literature: "normal periodic systems"

- For example, for *L+1* an odd prime and *K* a primitive root *mod* $(L+1)^2$, $1/(L+1)$ expressed in base *K* is a $P_L^{(K)}$-sequence.[1]

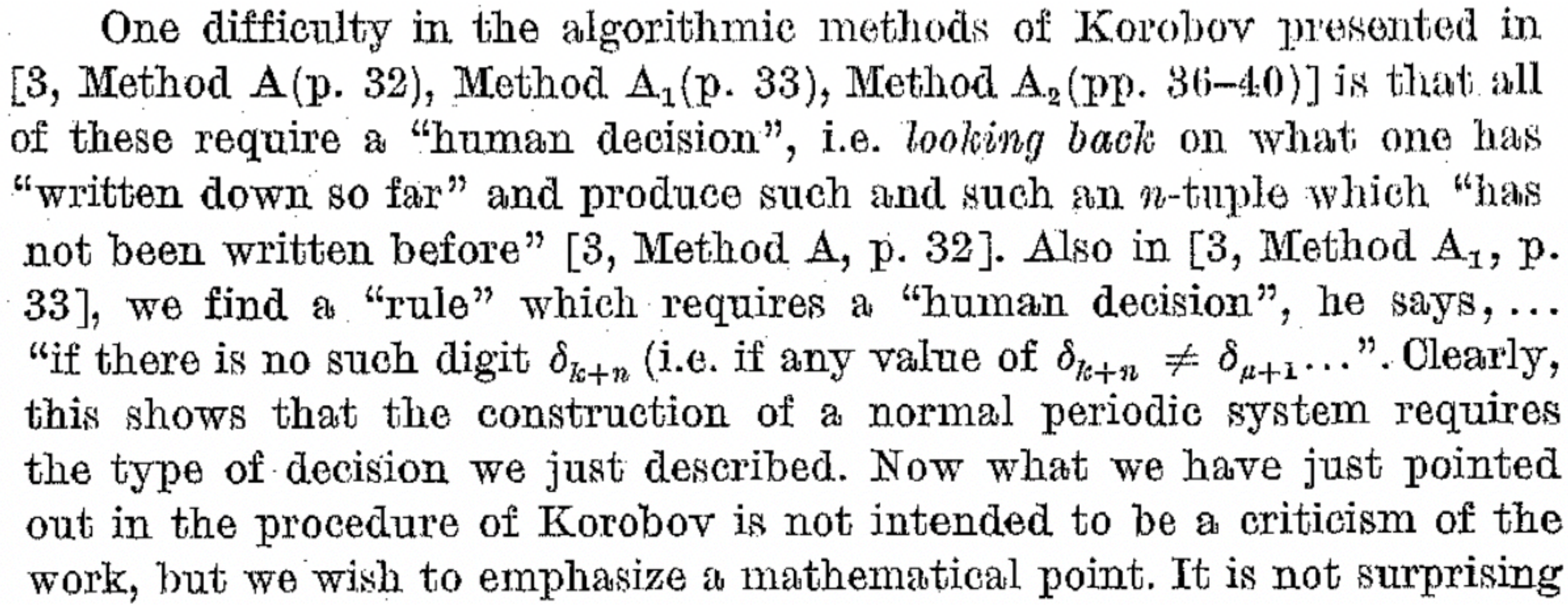
$$\frac{1}{19} = 0.\overline{000011010111100101}$$

$P_{18}^{(2)}$-sequence

[1] Stoneham, R.G. The reciprocals of integral powers of primes and normal numbers. *Proceedings of the American Mathematical Society*, *15*(2), 200-208, 1964.

Interesting read

R. Stoneham. Normal recurring decimals, normal periodic systems, (j, ε)-normality, and normal numbers. *Acta Arithmetica*, *28*(4), 349-361, 1976.

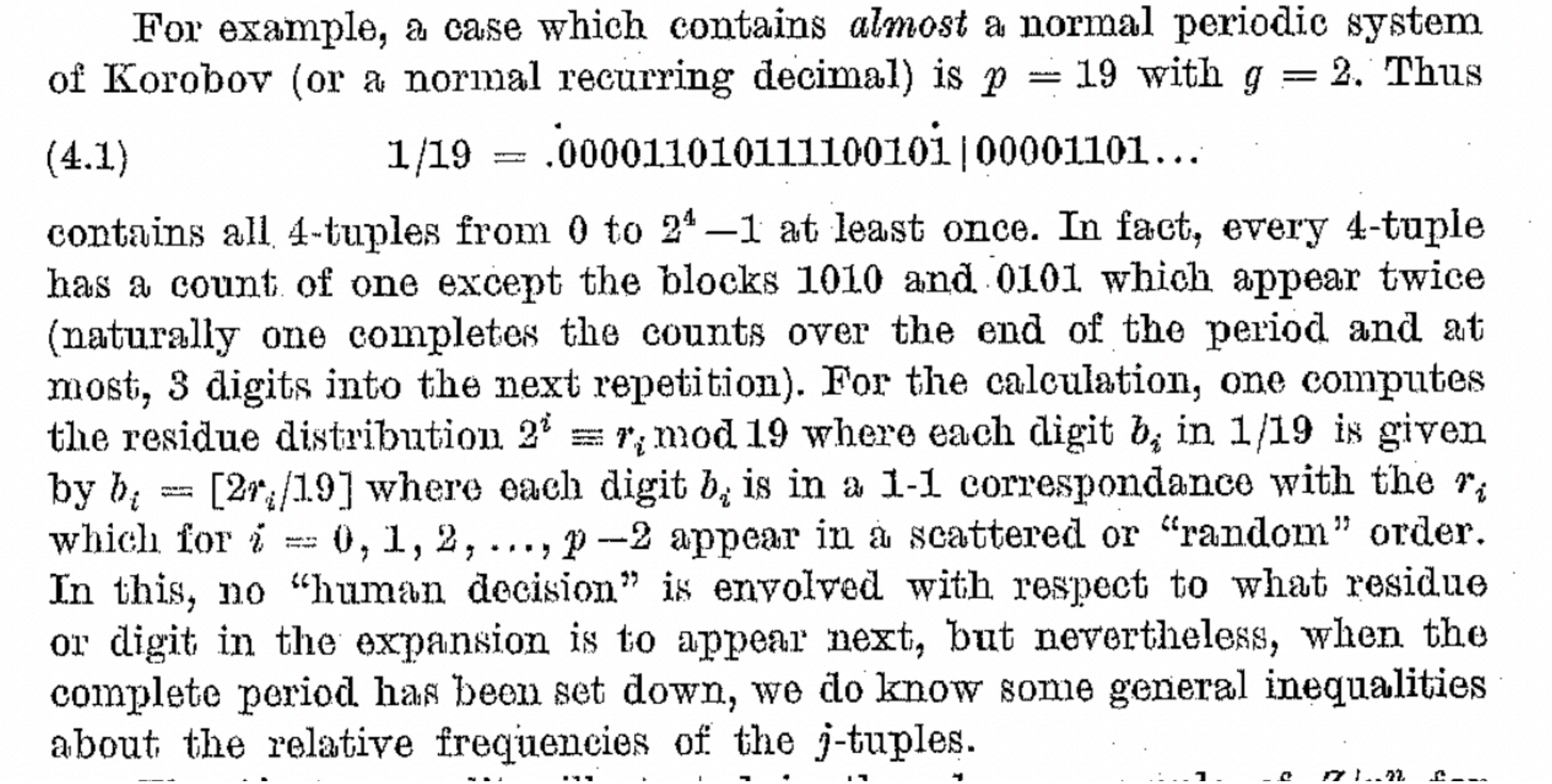


One difficulty in the algorithmic methods of Korobov presented in [3, Method A(p. 32), Method A_1(p. 33), Method A_2(pp. 36–40)] is that all of these require a "human decision", i.e. *looking back* on what one has "written down so far" and produce such and such an n-tuple which "has not been written before" [3, Method A, p. 32]. Also in [3, Method A_1, p. 33], we find a "rule" which requires a "human decision", he says, ... "if there is no such digit δ_{k+n} (i.e. if any value of $\delta_{k+n} \neq \delta_{\mu+1}$ ...". Clearly, this shows that the construction of a normal periodic system requires the type of decision we just described. Now what we have just pointed out in the procedure of Korobov is not intended to be a criticism of the work, but we wish to emphasize a mathematical point. It is not surprising

Korobov found something like Monroe Martin's greedy construction of a de Bruijn sequence, and Stoneham (maybe) doesn't like it.

Interesting read

R. Stoneham. Normal recurring decimals, normal periodic systems, (j, ε)-normality, and normal numbers. *Acta Arithmetica*, *28*(4), 349-361, 1976.



For example, a case which contains *almost* a normal periodic system of Korobov (or a normal recurring decimal) is $p = 19$ with $g = 2$. Thus

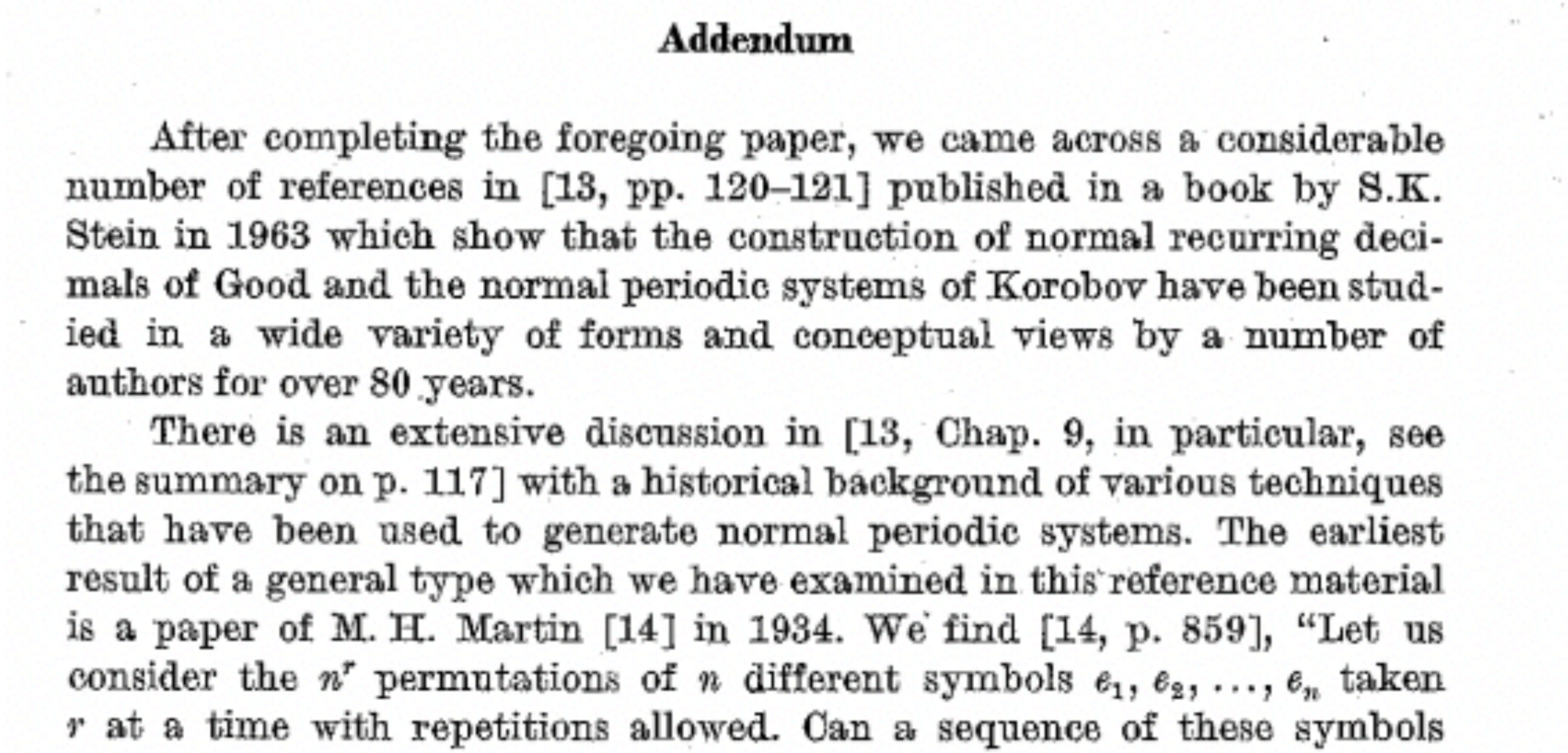
$$1/19 = .\dot{0}0001101011110010\dot{1}\,|\,00001101\ldots \tag{4.1}$$

contains all 4-tuples from 0 to 2^4-1 at least once. In fact, every 4-tuple has a count of one except the blocks 1010 and 0101 which appear twice (naturally one completes the counts over the end of the period and at most, 3 digits into the next repetition). For the calculation, one computes the residue distribution $2^i \equiv r_i \bmod 19$ where each digit b_i in $1/19$ is given by $b_i = [2r_i/19]$ where each digit b_i is in a 1-1 correspondance with the r_i which for $i = 0, 1, 2, \ldots, p-2$ appear in a scattered or "random" order. In this, no "human decision" is envolved with respect to what residue or digit in the expansion is to appear next, but nevertheless, when the complete period has been set down, we do know some general inequalities about the relative frequencies of the j-tuples.

Stoneham: "I can make almost a de Bruijn sequence without looking back! Beat that!"

Interesting read

R. Stoneham. Normal recurring decimals, normal periodic systems, (j, ε)-normality, and normal numbers. *Acta Arithmetica*, *28*(4), 349-361, 1976.



Addendum

After completing the foregoing paper, we came across a considerable number of references in [13, pp. 120–121] published in a book by S.K. Stein in 1963 which show that the construction of normal recurring decimals of Good and the normal periodic systems of Korobov have been studied in a wide variety of forms and conceptual views by a number of authors for over 80 years.

There is an extensive discussion in [13, Chap. 9, in particular, see the summary on p. 117] with a historical background of various techniques that have been used to generate normal periodic systems. The earliest result of a general type which we have examined in this reference material is a paper of M. H. Martin [14] in 1934. We find [14, p. 859], "Let us consider the n^r permutations of n different symbols $e_1, e_2, \ldots, e_n$ taken r at a time with repetitions allowed. Can a sequence of these symbols

Stoneham: "Wait! I had no idea there was a de Bruijn sequence literature outside the recurring decimal literature."

Irony is we had had no idea about the recurring decimal literature.

But it appears to us that to prove $P_L^{(K)}$-sequences can be constructed for arbitrary K, L easily, one has to manipulate the sequences directly---perhaps inelegantly, in Stoneham's eyes.