The combinatorial algorithms in my practice

Valentin Bakoev

Faculty of Mathematics and Informatics,
"St. Cyril and St. Methodius" University of Veliko Tarnovo, Bulgaria

National Seminar on Mathematical Software
and Combinatorial Algorithms,
December 7-8, 2020

Outline of the talk

1. Combinatorial algorithms – main questions
2. Combinatorial algorithms – some main sources and tools
3. Teaching in combinatorial algorithms
4. Combinatorial algorithms in the author's research
5. An example
6. Instead of a conclusion

1. Main questions

When we talk about combinatorial algorithms, many questions arise, for example:

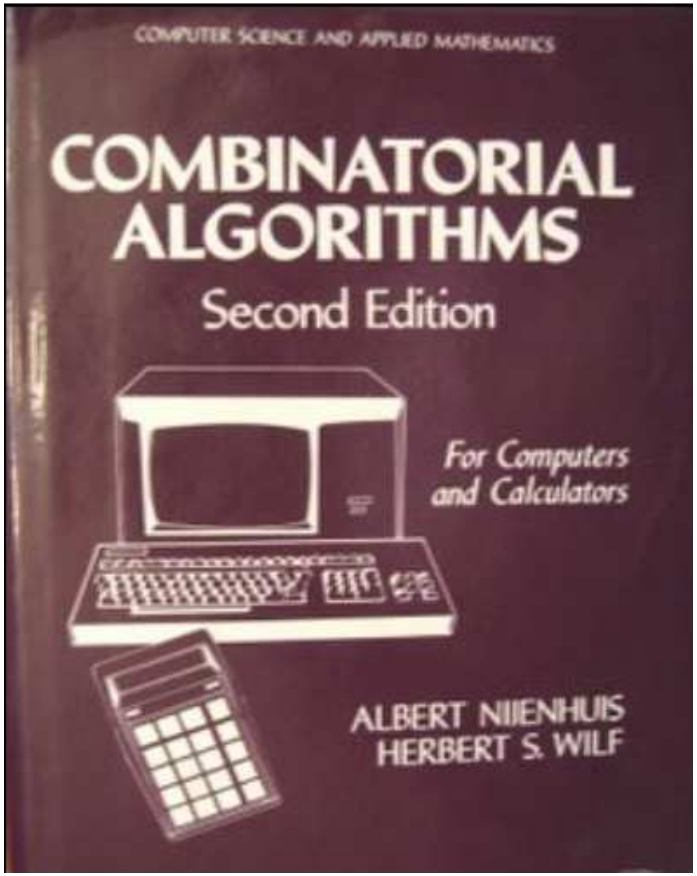
1. What are the combinatorial algorithms?
2. What areas do they concern, what is their scope?
3. What tools, techniques, tricks, etc. they use?
4. What mathematical software can be used to solve combinatorial problems?

In this report, the author aims to share his modest experience.

2. Some main sources

Nijenhuis A., Wilf H., Combinatorial Algorithms for Computers and Calculators, (First Ed.) 1975, (Second Ed., Academic Press) 1978.

The book considers combinatorial algorithms for constructing basic combinatorial objects such as permutations, subsets, and partitions; both randomly and sequentially.
Fortran programs for all of them are provided, as well as a discussion of the theory behind each of them.



2. Some main sources

Reingold E., Nievergeld J. and Deo N., Combinatorial algorithms, theory and practice, Prentice-Hall, 1977.

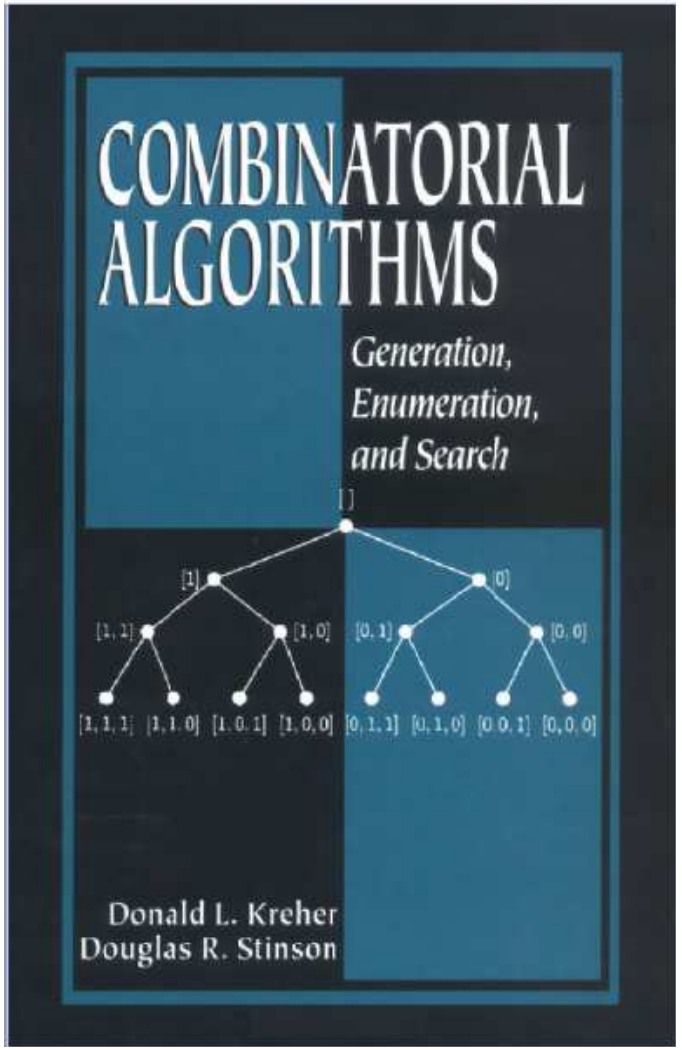
For the topics included, see CONTENTS, p. 474.

2. Some main sources

Kreher D., Stinson D., Combinatorial algorithms: generation, enumeration and search, CRC Press, 1999.

What has changed in about 20 years?

See CONTENTS, p. 8.

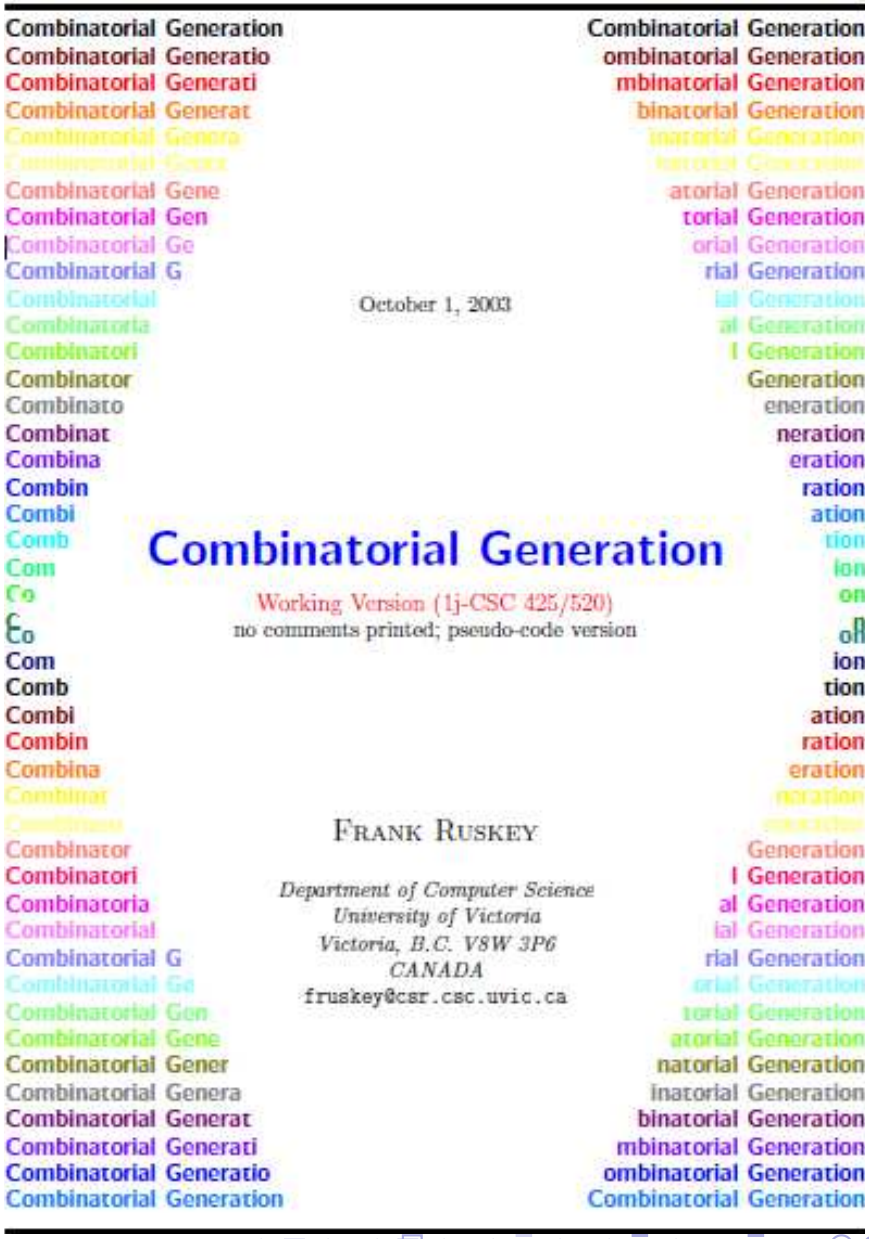


2. Some main sources

Ruskey Fr., Combinatorial generation, (Working version), 2003.

For the topics included, see CONTENTS, p. 10.

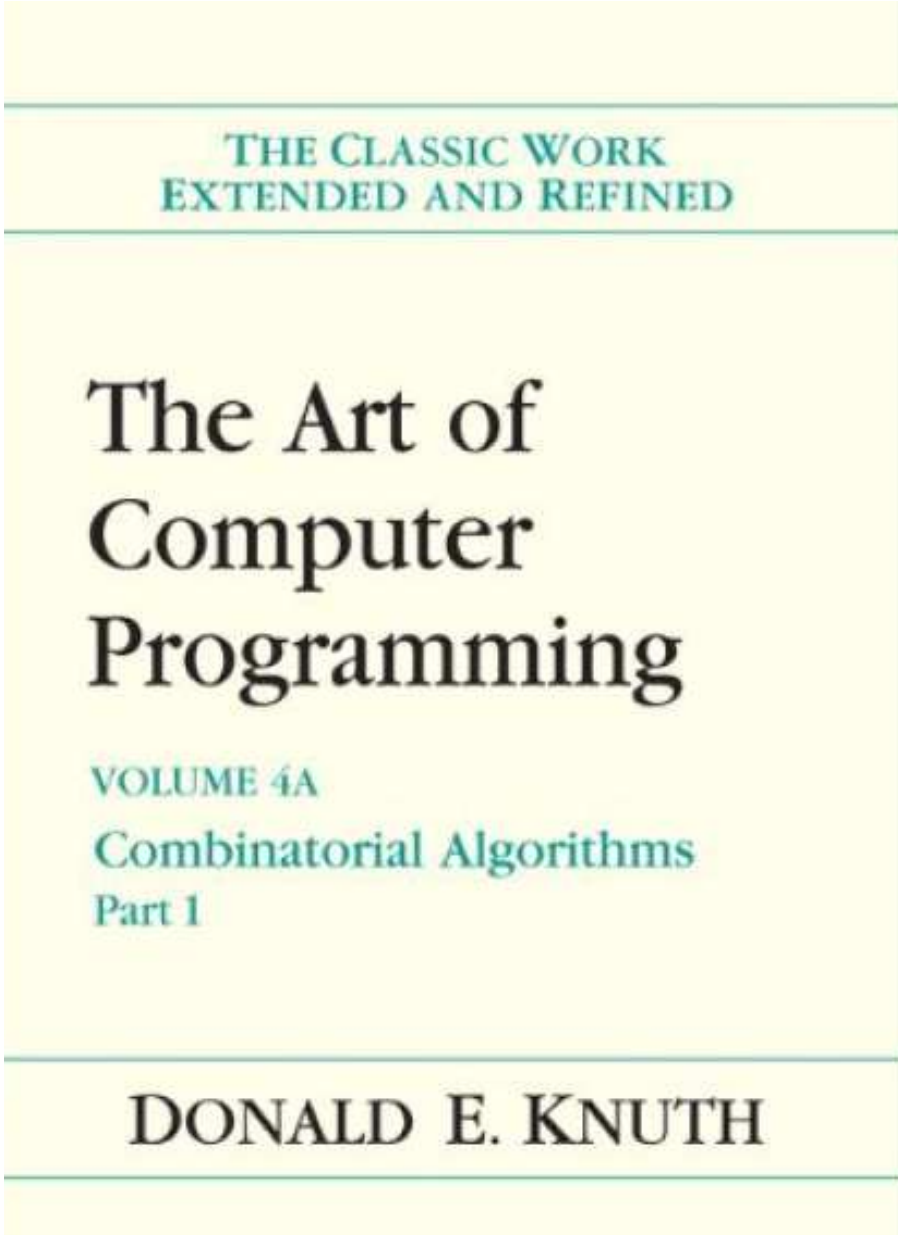
“The fastest known algorithms for generating certain objects are based on Gray codes...” (p.115)



2. Some main sources

Knuth D.,THE ART OF COMPUTER PROGRAMMING, Volume 4A / Combinatorial Algorithms, Part 1, Addison-Wesley, 2011.

See CONTENTS, p. XV.



2. Some main sources

Some citations from the Knuth's preface:

- "Combinatorial algorithms can be defined informally as techniques for the high-speed manipulation of combinatorial objects such as permutations or graphs."
- "OK, it's clear that the field of Combinatorial Algorithms is vast, and I can't cover it all."
- "The title of Volume 4 is Combinatorial Algorithms, and when I proposed it I was strongly inclined to add a subtitle: The Kind of Programming I Like Best."
- "Why is it that, for me, combinatorics arouses feelings of pure pleasure, yet for many others it evokes pure panic?"

2. Some main sources

- “Many combinatorial questions that I once thought would never be answered during my lifetime have now been resolved, and those breakthroughs have been due mainly to improvements in algorithms rather than to improvements in processor speeds.”

  In addition, “Most exponential time algorithms are merely variations of exhaustive search, whereas polynomial time algorithms generally are made possible only through the gain of some deeper insight into the structure of a problem”
  (Garey M., and Johnson D., Computers and Intractability. A Guide to the Theory of NP-Completeness, 1979, p. 8).

2. Some main sources

There are many other sources, for example:

- Many books/textbooks on Discrete Mathematics, Algorithms and Data structures, Cryptography, etc., contain combinatorial algorithms.
- Many techniques, tricks (bitwise and others), transformations, background theory, etc., can be found in:

Joux A., Algorithmic Cryptanalysis. Chapman & Hall/CRC Cryptography and Network Security, 2012.

Arndt J., Matters Computational. Ideas, Algorithms, Source Code, Springer, 2011.
(Free accessible at https://www.jjj.de/fxt/fxtbook.pdf)

2. Some main sources and tools

- Many papers, reports, dissertations, etc. consider combinatorial algorithms of different types.
- Combinatorial algorithms are available in Computer Algebra Systems (CASs) such as Mathematica, Maple, Matlab, Maxima, Sage, and others. Most of them contain extensions like "Combinatorica" in the Mathematica CAS.

  Pemmaraju S., Skiena S. Computational Discrete Mathematics. Combinatorics and Graph Theory with Mathematica. Cambridge (UK): Cambridge University Press; 2003.

  Furthermore, every CAS has its own programming language.
- Databases: https://oeis.org; http://codetables.markus-grassl.de (incl. Brouwer's tables), DIMACS, etc.

2. Some main sources and tools

- Specially created CASs and packages as Magma, GAP, Q-Extension and QextNewEdition, QLC, etc.

Powerful tools for accelerating many existing combinatorial algorithms/programs and in creating new ones are:

1. Bitwise data representation, bitwise operations and algorithms.
2. Parallel algorithms.
3. Low-level programming, use of fast processor instruction set (SIMD vector instructions that operate on multiple values contained in one large register at the same time), GNU Compiler Collection (GCC) vector support and built-in (intrinsic) to the compiler functions, etc.
4. Mathematical libraries, such as Math, Boost, etc. in C++.

3. Teaching in combinatorial algorithms

Teaching in combinatorial algorithms – my teachers and our joint work:

1. Prof. Krassimir Manev – I was his assistant in the course "Discrete Mathematics" (1990 – 1997). His course contained several combinatorial algorithms (on Boolean functions, graphs, etc.). He was a supervisor of my dissertation (Ph.D.) entitled "Algorithms for studying combinatorial objects". Our joint research with Krassimir Manev is aimed at:
   - The "3-SAT problem" – we have done classification (with respect to several equivalences) of all Boolean functions of n variables, whose conjunctive normal forms contain up to 3 variables. Some necessary and sufficient conditions for the "3-SAT problem" in these cases were derived.
   - The Zhegalkin transform – we derived the transformation matrix in two ways, independently of the other authors. These results were used in creating fast bitwise algorithms for computing the ANF of Boolean functions.

3. Teaching in combinatorial algorithms

2. Prof. Stoyan Kapralov – I was his assistant in the course "Mathematical foundations of Informatics" (1991 – 1997). This course contained several lectures on the generation of basic combinatorial objects. In the period 2016–2018, we investigated and classified all closed knight's pats of length 4, 6, 8 and 10.
3. Prof. Iliya Bouyukliev – while we worked together. Our joint work was aimed at efficiently computing the number of codewords of fixed weights in linear codes, as well as at efficient computing of some vector operations over GF(3) and GF(4).

3. Teaching in combinatorial algorithms

4. The author is grateful to Prof. Stefan Dodunekov, as well as to Prof. S. Bouyuklieva, Prof. T. Baicheva and many other colleagues for their advice, support, help, and more.

Figure 1: Stefan Dodunekov (5.09.1945 – 5.08.2012)

3. Teaching in combinatorial algorithms

Teaching in combinatorial algorithms – the author's experience:

- In the **bachelor's** programs: in the courses "Graph algorithms", "Algorithms and Data structures", "Competitive programming" (including the Programming Competitions at the University of Veliko Tarnovo), as well as in some lectures of the course Discrete mathematics/Discrete structures.
- In the **master's** programs: the course "Discrete Mathematics" contains several topics on generating some basic combinatorial objects. Many combinatorial and cryptographic algorithms are considered in the courses "Algorithmic strategies and models" and "Information protection".

4. Combinatorial algorithms and the author's research

The author notes:

- Almost all of his research is in the field of discrete mathematics and combinatorial algorithms. Its results are in 2 main directions: solving enumeration problems and creating new algorithms or improving existing ones.
- In his research, he first seeks to study the combinatorial objects, to derive and prove their important combinatorial, structural and algorithmic properties. On this basis, he aims to develop new efficient algorithms or improve the efficiency of existing algorithms.

4. Combinatorial algorithms and the author's research

The author's research experience includes the usage of:

1. The algorithmic strategies: Brute force (exhaustive search, generating, etc.), Backtracking, Divide and conquer, Dynamic programming, Greedy algorithms.
2. Generating algorithms for: basic combinatorial objects; Boolean functions: up to 3 variables in CNF, monotone, randomly generated for test examples; codewords; partitons of numbers and sets; sequences; random graphs with certain parameters; paths in graphs; use of ranking and unranking functions, etc.
3. Generating in a certain order: lexicographic, reverse lexicographic, in a Gray code (or with a minimal change), partial orders for POSets.
4. Searching algorithms: exhaustive search, backtracking, binary search, WLO search, etc.

4. Combinatorial algorithms and the author's research

5. Sorting algorithms (incl. Counting sort, Bucket sort), Graph algorithms (incl. with Priority queues).
6. Algorithms performing equivalence checks and for counting the number of equivalence classes.
7. Algorithms for checking theoretical results or results obtained from my other algorithms.
8. Precomputing of: weights of binary vectors; sequences of numbers; discrete structures.
9. CAS: Mathematica, Maxima.
10. Databases: OEIS, DIMACS.
11. Bitwise representations, operations and algorithms.

Example

Usually, the *n-dimensional Boolean cube* is defined as $\{0,1\}^n = \{(x_1, x_2, \ldots, x_n) | \, x_i \in \{0,1\}$, for $i = 1, 2, \ldots, n\}$, which is the set of all n-dimensional binary vectors.

Definition

1) The set $\{0,1\} = \{(0),(1)\}$ is called *one-dimensional Boolean cube* and its elements (0) and (1) are its *one-dimensional binary vectors*.
2) Let $\{0,1\}^{n-1} = \{\alpha_0, \alpha_1, \ldots, \alpha_{2^{n-1}-1}\}$ be the $(n-1)$*-dimensional Boolean cube* and $\alpha_0, \alpha_1, \ldots, \alpha_{2^{n-1}-1}$ be its $(n-1)$*-dimensional binary vectors*.
3) The *n-dimensional Boolean cube* $\{0,1\}^n$ is built by taking the vectors of $\{0,1\}^{n-1}$ twice: firstly, each vector of $\{0,1\}^{n-1}$ is prefixed by zero, and thereafter each vector of $\{0,1\}^{n-1}$ is prefixed by one, i.e.,

$$\{0,1\}^n = \{(0,\alpha_0),(0,\alpha_1),\ldots,(0,\alpha_{2^{n-1}-1}),$$
$$(1,\alpha_0),(1,\alpha_1),\ldots,(1,\alpha_{2^{n-1}-1})\}.$$

Example

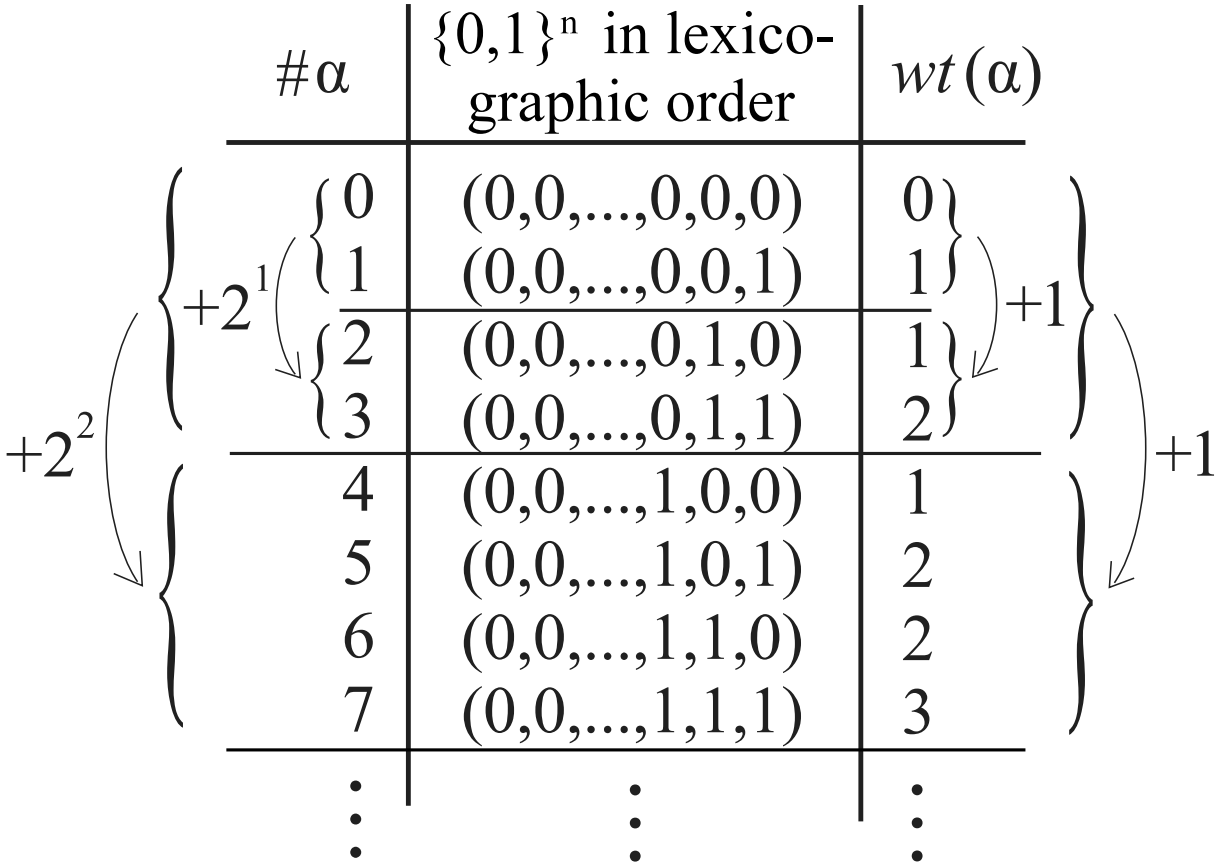


Figure 2: The relationship between the lexicographic order of the vectors and their serial numbers and weights

Example

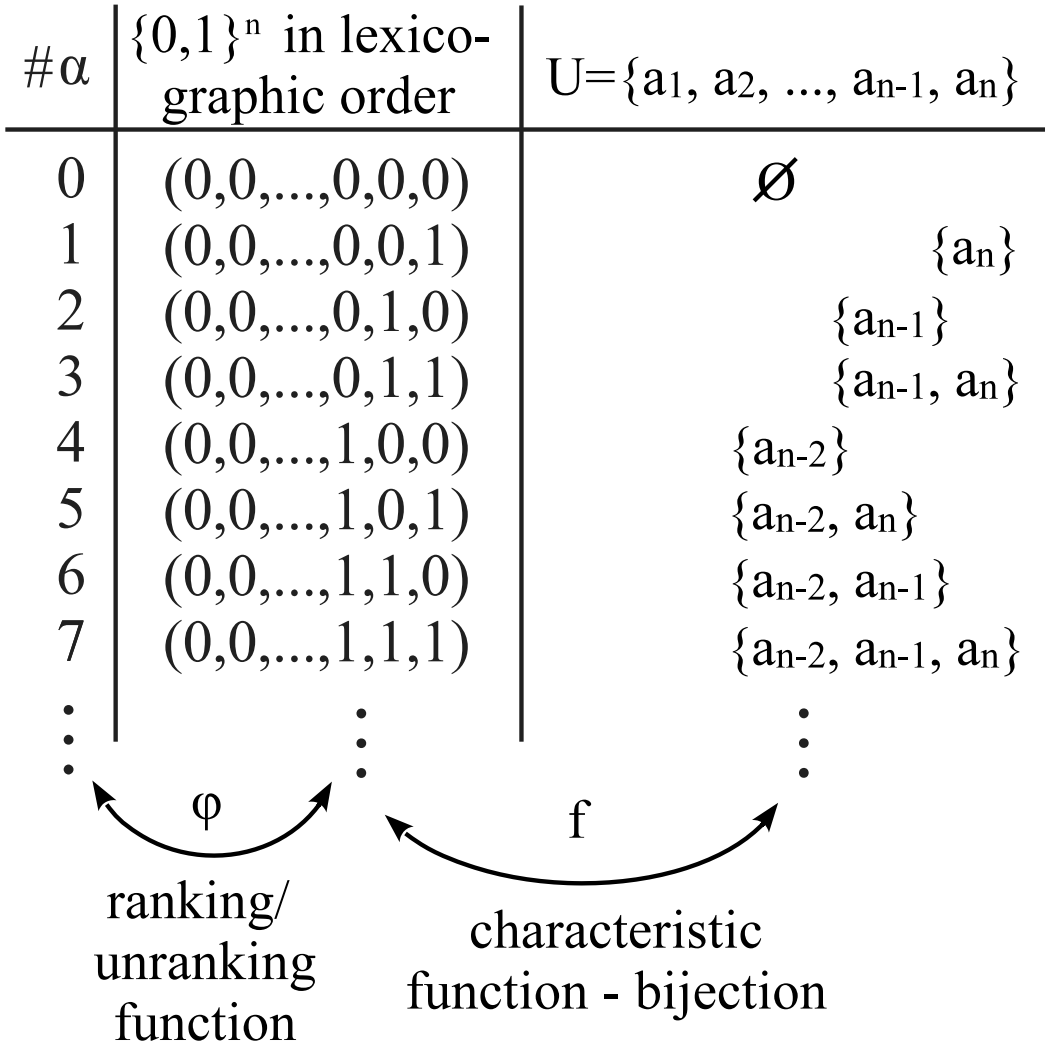


Figure 3: Characteristic function, ranking and unranking functions of the lexicographic order

Example

Definition

1) The vectors (0) and (1) of $\{0,1\}^1$ are ordered in a Reflected Cyclic Gray Code (RCGC).
2) Let $\beta_0, \beta_1, \ldots, \beta_{2^{n-1}-1}$ be the sequence of all vectors of $\{0,1\}^{n-1}$, ordered in a RCGC.
3) We perform two basic steps to obtain the vectors of $\{0,1\}^n$ in a RCGC. Firstly, we take the sequence of vectors $\beta_0, \beta_1, \ldots, \beta_{2^{n-1}-1}$ of $\{0,1\}^{n-1}$ in a RCGC and we add zero at the beginning of each of its vector. Secondly, we take the same sequence in a reversed order, i.e., $\beta_{2^{n-1}-1}, \ldots, \beta_1, \beta_0$ – we say that it is obtained by *reflection* of the original sequence. Afterward we add one at the beginning of each vector of the reflected sequence. The obtained sequence

$$(0,\beta_0),(0,\beta_1),\ldots,(0,\beta_{2^{n-1}-1}),(1,\beta_{2^{n-1}-1}),\ldots,(1,\beta_1),(1,\beta_0)$$

contains all vectors of $\{0,1\}^n$ ordered in a *reflected cyclic Gray code*.

Example

#α	$\{0,1\}^n$ in a Gray code n,n-1, ...,3, 2, 1 – № of coord.	T(n)
0	(0,0, ... ,0,0,0)	
1	(0,0, ... ,0,0,1)	1
3	(0,0, ... ,0,1,1)	2
2	(0,0, ... ,0,1,0)	1
6	(0,0, ... ,1,1,0)	3
7	(0,0, ... ,1,1,1)	1
5	(0,0, ... ,1,0,1)	2
4	(0,0, ... ,1,0,0)	1
⋮	⋮	4
		⋮

#α	$\{0,1\}^n$ in lexico-graphic order	wt(α)	$d(\alpha_i,\alpha_{i+1})$
0	(0,0,...,0,0,0)	0	
1	(0,0,...,0,0,1)	1	1
2	(0,0,...,0,1,0)	1	2
3	(0,0,...,0,1,1)	2	1
4	(0,0,...,1,0,0)	1	3
5	(0,0,...,1,0,1)	2	1
6	(0,0,...,1,1,0)	2	2
7	(0,0,...,1,1,1)	3	1
⋮	⋮	⋮	4
			⋮

Transition sequence: $T(1)= 1$; $T(n)= T(n-1), n, T(n-1)= d(\alpha_i,\alpha_{i+1})$, for $i= 0, 1, ..., 2^n-2$

OEIS - A001511

Figure 4: The order in an RCGC and the transition sequence

Example

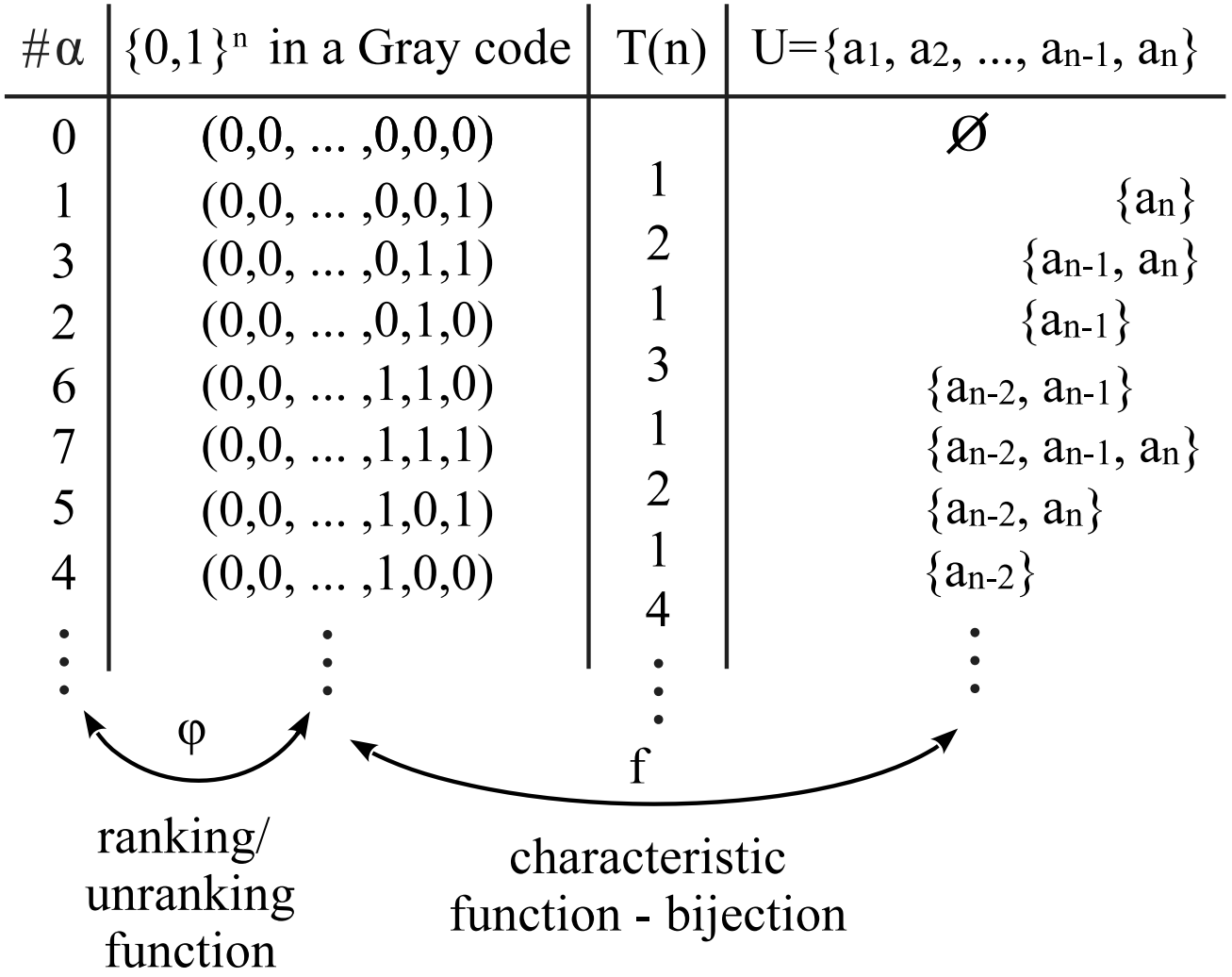


Figure 5: Characteristic function, ranking and unranking functions of the order in an RCGC

Instead of a conclusion

We believe that each participant in the seminar will draw their own conclusions and find benefits for themselves.

Thank you for your attention!