NEWLY AVAILABLE SECTIONS OF THE CLASSIC WORK

The Art of Computer Programming

VOLUME 4
Constraint Satisfaction

FASCICLE 7

DONALD E. KNUTH

FREE SAMPLE CHAPTER

THE ART OF COMPUTER PROGRAMMING

VOLUME 4, FASCICLE 7

Constraint Satisfaction

DONALD E. KNUTH *Stanford University*

ADDISON–WESLEY
Hoboken, New Jersey

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Please contact us with concerns about any potential bias at `www.pearson.com/en-us/report-bias.html`.

Author websites are not owned or managed by Pearson.

Visit us on the Web: `informit.com/aw`

Library of Congress Control Number: 2024949448

Internet page `https://www-cs-faculty.stanford.edu/~knuth/taocp.html` contains current information about this book and related books.

See also `https://www-cs-faculty.stanford.edu/~knuth/sgb.html` for information about *The Stanford GraphBase*, including downloadable software for dealing with the graphs used in many of the examples.

And see `https://www-cs-faculty.stanford.edu/~knuth/mmix.html` for basic information about the `MMIX` computer.

Copyright © 2025 by Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit `www.pearson.com/global-permission-granting.html`.

ISBN-13 978-0-13-532824-8
ISBN-10 0-13-532824-1

First digital release, March 2025

PREFACE

> *There is quite a good deal of information in the book.*
> *I regret this very much; but really it could not be helped:*
> *information appears to stew out of me naturally,*
> *like the precious ottar of roses out of the otter.*
> *Sometimes it has seemed to me that I would give worlds*
> *if I could retain my facts; but it cannot be.*
> *The more I calk up the sources, and the tighter I get, the more I leak wisdom.*
> *Therefore, I can only claim indulgence at the hands of the reader, not justification.*
>
> — MARK TWAIN, *Roughing It* (1872)

FREEDOM is a wonderful thing. But we can also be thankful for the constraints that give structure to our lives and provide a focus for our creative juices. We experience moments of great satisfaction when challenges have been met. Combinatorial and musical patterns that fit together almost magically can be supremely satisfying.

This booklet is Fascicle 7 of *The Art of Computer Programming*, Volume 4: *Combinatorial Algorithms*. As explained in the preface to Fascicle 1 of Volume 1, I'm circulating the material in this preliminary form because I know that the task of completing Volume 4 will take many years; I can't wait for people to begin reading what I've written so far and to provide valuable feedback.

To put the material in context, this lengthy fascicle contains part of a long, long chapter on combinatorial searching. Chapter 7 will eventually fill at least four volumes (namely Volumes 4A, 4B, 4C, and 4D), assuming that I'm able to remain healthy. It begins in Volume 4A with a short review of graph theory and a longer discussion of "Zeros and Ones" (Section 7.1); that volume concludes with Section 7.2.1, "Generating Basic Combinatorial Patterns," which is the first part of Section 7.2, "Generating All Possibilities." Volume 4B resumes the story with Section 7.2.2, about backtracking in general. Section 7.2.2.1 discusses a family of methods called "dancing links," for updating data structures while backtracking, and shows that such methods are particularly successful when applied to XCC problems — "exact covering with colors." That sets the scene for Section 7.2.2.2, which fills the rest of Volume 4B; it is devoted to the important problem of Boolean satisfiability, aka 'SAT'.

Now comes Volume 4C, whose first third will eventually consist of the contents of the present fascicle. Our theme, "constraint satisfaction," is Section 7.2.2.3, and it takes the concepts of Sections 7.2.2.1 and 7.2.2.2 to a higher level. A fresh look at the basic notions leads to significant improvements and many individual topics of independent interest. Once again it has been a great pleasure for me to put these stories together, and I hope that I can convey the joy of what I've learned to as many people as possible.

Dozens of examples tie the subject matter of this fascicle to numerous other parts of computer science and mathematics. We'll see, for instance, applications to scene analysis (computer vision); we'll construct fascinating instances of "graceful graphs"; we'll discuss efficient algorithms that embed one graph as a subgraph of another; we'll learn new ways to look ahead when backtracking, and we'll investigate new heuristics for guiding a search that backtracks through a massive space of possibilities; we'll also learn how to avoid backtracking altogether, when possible. We'll study new sparse-set data structures that lead to "dancing cells" — a technique that often is even better than "dancing links"! Of course there are recreational topics galore, even including some new takes on the classic problem of a knight's tour. Puzzle connoisseurs will enjoy learning more about fillomino patterns. And so on.

As usual, more than half of this book is devoted to exercises and their answers. In fact, Section 7.2.2.3 has approximately 500 exercises, some of which are routine and some of which aren't yet solved; all of them have been designed for self-study. I've tried to make it easy for a reader to navigate through this maze of problems by presenting them in an order that roughly parallels the threads of the main text.

Look, for example, at a "random" page — page 42, say, which discusses a variety of ways to translate graph coloring problems into SAT clauses. On that page you'll see that exercises 221, 226, 224, 225, 228 are mentioned. So you can guess that the main exercises about SAT encoding probably have numbers near 225. The exercises have also been carefully indexed.

My goal as always has been to strive for a good balance between theory and practice (and fun). I try to discuss the theoretical discoveries that are most relevant to people who write programs in the "real world," and to present those theories without using advanced jargon. I also strive to encapsulate the history of the subject, so that readers can understand the "human dimension" and the process of discovery.

While writing this section I also wrote hundreds of programs for my own edification. (I usually can't understand things well until I've tried to explain them to a machine.) Most of those programs were quite short, of course; but several of them are rather substantial, and possibly of interest to others. Therefore I've made a selection available by listing some of them on the following webpage:

`https://cs.stanford.edu/~knuth/programs.html`

Prototypes of the main algorithms can be found there, together with a few other programs that are mentioned in the answers to certain exercises. If you want to see a program called FOO, look for FOO on that webpage. In particular, you can download the programs SSXCC0, SSXCC, SSXCC-BINARY, SSMCC, and XCCDC; those experimental versions of Algorithms C, C^+, B, F, and S were my constant companions while writing the middle portions of this fascicle.

Interested readers who wish to do their own experiments can find data files for the benchmark examples in the following two collections:

`https://cs.stanford.edu/~knuth/programs/xcc-benchmarks.tgz`
`https://cs.stanford.edu/~knuth/programs/mcc-benchmarks.tgz`

Incidentally, the illustrations in Section 7.2.2.3 are numbered beginning with Fig. 100, because the final illustration of Volume 4B is Fig. 99. The editor has decided to treat Chapter 7 as a single unit, even though it is being split across several physical volumes.

Special thanks are due to Christian Bessière, Víctor Dalmau, Ralph Freese, Daniel Horsley, Peter Jeavons, Ciaran McCreesh, Patrick Prosser, George Sicherman, Christine Solnon, Filip Stappers, Peter Stuckey, Kokichi Sugihara, James Trimble, Udo Wermuth, Ross Willard, and Dmitriy Zhuk, for their detailed comments on my early attempts at exposition, as well as to numerous other correspondents who have contributed crucial corrections. Thanks also to Stanford's InfoLab for providing extra computer power when my workstation was inadequate. And above all, I thank my wife for her constant support and for help with Figs. 100 and 101.

I happily offer a "finder's fee" of $2.56 for each error in this draft when it is first reported to me, whether that error be typographical, technical, or historical. The same reward holds for items that I forgot to put in the index. And valuable suggestions for improvements to the text are worth 32¢ each. (Furthermore, if you find a better solution to an exercise, I'll actually do my best to give you immortal glory, by publishing your name in the eventual book:–)

Cross references to yet-unwritten material sometimes appear as '00'; this impossible value is a placeholder for the actual numbers to be supplied later.

Happy reading!

Stanford, California D. E. K.
23 October 2024

P.S.: **A note on notations.** Some formulas in this booklet use the notation '$\nu\, x$' for the "sideways sum" or "population count" function, as well as the notation 'ρx' for the "ruler" function. Those functions, and other bitwise notations, are discussed extensively in Section 7.1.3 of Volume 4A.

Other formulas use the notation $\langle xyz \rangle$ for the median function, which is discussed extensively in Section 7.1.1. Hexadecimal constants are preceded by a number sign or hash mark: `#123` means $(123)_{16}$.

Furthermore, there's a special list of 'Notational conventions' in the index.

If you run across other notations that appear strange, please look at the Index to Notations (Appendix B) at the end of Volume 4A or 4B. Volume 4C will, of course, have its own Appendix B some day.

> *The field of combinatorial algorithms is too vast*
> *to cover in a single paper or even in a single book.*
> — ROBERT ENDRE TARJAN, *SIAM Review* (1978)

CONTENTS

Chapter 7 — Combinatorial Searching . . . . [4A.1]

7.2. Generating All Possibilities . . . . [4A.281]
  7.2.1. Generating Basic Combinatorial Patterns . . . . [4A.281]
  7.2.2. Backtrack Programming . . . . [4B.30]
    7.2.2.1. Dancing links . . . . [4B.65]
    7.2.2.2. Satisfiability . . . . [4B.185]
    7.2.2.3. Constraint satisfaction 1
      Related models 2
      *Statistical mechanics 4
      A simple example 6
      Automating automobiles 7
      Line labeling in computer vision 9
      Graph labeling 16
      *Graceful digraphs 24
      Graph embedding 28
      *Supplemental labels and graphs 35
      Special cases of subgraph isomorphism 37
      Solving a CSP 38
      Translating CSP to SAT 39
      SAT encodings of general relations 46
      Consistency 48
      Efficiency 54
      Representing the domains 56
      *Dancing cells 61
      *Dynamic variable ordering heuristics 65
      *Maintaining XCC supports 69
      Performance on benchmarks 73
      *Sparse-set methods for MCC problems 76
      Tractable families of CSPs 79
      Implicational constraints 80
      Max-closed constraints 82
      CRC constraints 84
      Polymorphisms 86
      The indicator problem 87
      NP-complete families of CSPs 90
      *Polarities and clones 92
      *The dichotomy theorem 96

*Exploiting a polymorphism 99
*Visiting all solutions 106
A brief history 107
For further reading 111
Exercises 113

Answers to Exercises 158

Index to Algorithms and Theorems 262

Answers to Puzzles in the Answers 263

Index and Glossary 264

I always thought Volume 4 was a myth,
like the missing part of the Dead Sea scrolls.

— BILL GASARCH (blog post, 10 January 2008)

> *His Lady sad to see his sore constraint,*
> *Cride out, Now now Sir knight, shew what ye bee.*
> — EDMUND SPENSER, *The Faerie Queene* (1590)

> *The work under our labour grows, Luxurious by restraint.*
> — JOHN MILTON, *Paradise Lost* (1667)

> *Liberty exists in proportion to wholesome restraint.*
> — DANIEL WEBSTER (1847)

> *It is impossible to be an artist and not care for laws and limits.*
> *Art is limitation; the essence of every picture is the frame.*
> — GILBERT K. CHESTERTON, *Orthodoxy* (1908)

> *I surround myself with obstacles.*
> *Whatever diminishes my discomfort diminishes my strength.*
> *The more constraints one imposes, the more one*
> *frees one's self of the chains that shackle the spirit.*
> — IGOR STRAVINSKY, *Poétique musicale sous forme de six leçons* (1939)

7.2.2.3. Constraint satisfaction. In Section 7.2.2.1 we solved numerous examples of XCC problems — exact covering with colors — which featured "items" and "options." Then in Section 7.2.2.2 we resolved lots of SAT problems — Boolean satisfiability — which featured "literals" and "clauses." All of these, and more, are instances of a combinatorial challenge that's more general yet, the *constraint satisfaction problem* — often called the CSP for short — which we will see is based on "variables," "domains," and "constraints."

The idea is simple: We're given a finite list of *variables* $(x_1, x_2, \ldots, x_n)$, to which we can assign values that belong to given finite *domains* $(D_1, D_2, \ldots, D_n)$. And we're also given a set of *constraints* $\{R_1, R_2, \ldots, R_m\}$, each of which specifies that a certain subset of the values $(x_1, x_2, \ldots, x_n)$ must be mutually compatible. Some combinations of values are "good"; the others are "nogood."

For example, let $n = 5$, and suppose that each domain is a set of letters:

$$D_1 = \{\mathtt{B},\mathtt{S}\},\;\; D_2 = \{\mathtt{C},\mathtt{L}\},\;\; D_3 = \{\mathtt{A},\mathtt{I},\mathtt{U}\},\;\; D_4 = \{\mathtt{E},\mathtt{O}\},\;\; D_5 = \{\mathtt{D},\mathtt{N}\}. \qquad (1)$$

Thus there are $2\times2\times3\times2\times2 = 48$ possible settings of $x_1x_2x_3x_4x_5$, from `BCAED` to `SLUON`. Let's also impose three constraints:

$$\begin{aligned} R_1(x_1,x_3,x_5) &= \text{`}x_1x_3x_5 \in \{\mathtt{BAN},\mathtt{BUD},\mathtt{SIN}\}\text{'};\\ R_2(x_1,x_4) &= \text{`}x_1x_4 \in \{\mathtt{BE},\mathtt{SE},\mathtt{SO}\}\text{'};\\ R_3(x_2,x_4,x_5) &= \text{`}x_2x_4x_5 \in \{\mathtt{COD},\mathtt{CON},\mathtt{LED}\}\text{'}. \end{aligned} \qquad (2)$$

This CSP has two solutions, easily found by hand (see exercise 1).

Every SAT problem is obviously a CSP in which all the domains are $\{0,1\}$. For example, problem $F = \{1\bar{2}, 23, \bar{1}\bar{3}, \bar{1}\bar{2}3\}$ in 7.2.2.2–(3) has four constraints,

$$\begin{aligned} &x_1x_2 \in \{00,10,11\};\quad x_2x_3 \in \{01,10,11\};\quad x_1x_3 \in \{00,01,10\};\\ &\qquad\qquad x_1x_2x_3 \in \{000,001,010,011,100,101,111\}. \end{aligned} \qquad (3)$$

Conversely, every CSP can be formalized as an equivalent SAT problem, by using several SAT variables to represent each CSP variable x whose domain size d exceeds 2. For example, if the domain is $\{0, 1, \ldots, d-1\}$, Section 7.2.2.2 discussed the "log encoding," with $l = \lceil \lg d \rceil$ Boolean variables meaning that $x = (x_{l-1} \ldots x_1 x_0)_2$. There's also the "direct encoding," with d variables $x_k = [x = k]$, as well as the "order encoding," which has $d-1$ variables $x^j = [x \geq j]$. We also discussed a variety of ways to represent arbitrary constraints, in the form of one or more clauses involving such Boolean variables. Each of those encodings has its own virtues and weaknesses, depending on the application.

Every XCC problem can, similarly, be regarded as a CSP. One way is to have a variable x_i for every primary item i, with domain D_i equal to the set of options that contain i. The constraints are that x_i and x_j cannot be options that conflict: If $x_i = o_i$ and $x_j = o_j$, where $o_i \neq o_j$, then o_i and o_j cannot have a common primary item, nor can they have a common secondary item that's colored differently in o_i and o_j. Conversely, exercise 7.2.2.1–100 presented one way to encode any CSP as an XCC problem.

Thus XCC, SAT, and CSP can each be reduced to the other two.

We've already learned how to construct excellent XCC solvers and excellent SAT solvers, so we might be tempted to stop there, regarding CSP as a problem that's already been well solved. But we shall see that careful consideration of the CSP not only clarifies XCC and SAT, it also teaches us important new methods.

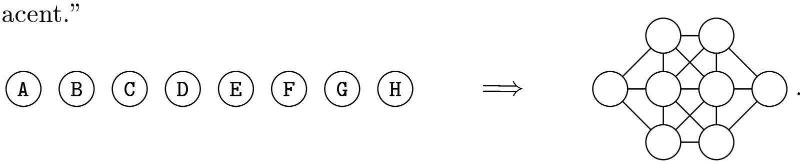
Related models. Many groups of researchers have independently adopted conceptual frameworks that are identical to or very similar to the notions of variables, domains, and constraints. For example, a theory of *relational structures* has been developed as part of the branch of mathematics called "model theory," which spawned "universal algebra." A relational structure is a set D together with a set $\{R_1, R_2, \ldots\}$ of relations or "predicates" defined on the elements of D. Each relation R_i depends on k elements, for some $k = k_i$, and it defines the k-tuples of elements for which that predicate is true. [See P. M. Cohn, *Universal Algebra* (1965), Chapter V.]

Let's be a little more precise. The *Cartesian product* of sets $(D_1, \ldots, D_n)$, denoted by $D_1 \times \cdots \times D_n$, is the set of all n-tuples $(x_1, \ldots, x_n)$ such that $x_i \in D_i$ for $1 \leq i \leq n$. Thus, $D_1 \times \cdots \times D_n$ is the set of all solutions to a CSP with domains $(D_1, \ldots, D_n)$, in the case when there are no constraints. An n-tuple such as $(x_1, \ldots, x_n)$ is often written simply as $x_1 \ldots x_n$, when commas aren't necessary. We also write $D \times \cdots \times D = D^n$ when the n domains are all identical.

A k-ary *relation* on sets $(D_1, \ldots, D_k)$ is a subset of $D_1 \times \cdots \times D_k$. We write either $R(x_1, \ldots, x_k)$ or $x_1 \ldots x_k \in R$ when we want to say that the k-tuple $(x_1, \ldots, x_k)$ satisfies relation R. The relation is called *binary* when $k = 2$, *ternary* when $k = 3$, *quaternary* when $k = 4$, and so on; it's *unary* when $k = 1$. (Strictly speaking, there also are *nullary* relations; see exercise 5.)

The simplest nontrivial relational structures arise where there's just a single binary relation. In fact, this case is so simple, we hardly ever think of it as a "relation structure" at all: We call it a *directed graph.* Indeed, we know well

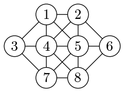
A simple example. To warm up, let's look at a little puzzle that appeared on a British TV show called *The Crystal Maze* in 1994. The task is simple—but you've got only two minutes to do it: "Place eight large disks, marked with the letters `A` through `H`, onto the eight circles shown; consecutive letters can't be adjacent."



(11)

We're actually facing two challenges here, namely (i) solve the puzzle; and (ii) express it as a constraint satisfaction problem, so that a computer can solve it for us. We'll tackle (ii), so as not to spoil the fun of (i). And we'll allow ourselves ten minutes, say, to accomplish goal (ii).

What are appropriate variables, domains, and constraints? We'd better label the vertices of the graph, so that we can readily describe what we want to define. One approach, based on the labeling shown, is to have eight variables $\{x_1, x_2, \ldots, x_8\}$, one for each vertex, each with domain $\{\texttt{A}, \texttt{B}, \ldots, \texttt{H}\}$. Then there are seventeen constraints, one for each edge of the graph; for example, the constraint for edge $1 — 2$ is



$$x_1x_2 \in \{\texttt{AC},\texttt{AD},\texttt{AE},\texttt{AF},\texttt{AG},\texttt{AH},\texttt{BD},\texttt{BE},\texttt{BF},\texttt{BG},\texttt{BH},\texttt{CA},\texttt{CE},\texttt{CF},\texttt{CG},\texttt{CH},\texttt{DA},\texttt{DB},\texttt{DF},\texttt{DG},\texttt{DH},\texttt{EA},\\ \texttt{EB},\texttt{EC},\texttt{EG},\texttt{EH},\texttt{FA},\texttt{FB},\texttt{FC},\texttt{FD},\texttt{FH},\texttt{GA},\texttt{GB},\texttt{GC},\texttt{GD},\texttt{GE},\texttt{HA},\texttt{HB},\texttt{HC},\texttt{HD},\texttt{HE},\texttt{HF}\}, \tag{12}$$

and the same relation is used for all of the other edges. It can be written much more succinctly, if we assume that the letters are represented by integer codes:

$$|x_1 - x_2| \;>\; 1. \tag{13}$$

OK, that took three minutes. Are we done? Well, no, actually; the seventeen constraints we've specified do not obviously rule out the possibility that $x_1 = x_8$. We're not allowed to put a disk on two different circles.

We could add eleven further constraints, namely $x_i \neq x_j$ for each of the yet-unconstrained pairs. But seasoned CSP solvers generally prefer to append a single *global* constraint instead, involving all of the variables at once:

$$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_8 \text{ are all different.} \tag{14}$$

Indeed, special methods have been devised for the "all-different" constraint, because it arises in so many different problems. With (14), we've satisfied (ii).

Five minutes to go. Is there a better way? Another possibility is to let the variables be $\{\texttt{A}, \texttt{B}, \ldots, \texttt{H}\}$, one for each *disk*, each with domain $\{1, 2, \ldots, 8\}$. Then only *seven* constraints are needed, one for each pair of consecutive letters; e.g.,

$$\texttt{AB} \in \{16,17,18,23,27,28,32,35,36,38,46,53,61,63,64,67,71,72,76,81,82,83\}. \tag{15}$$

And each of these constraints has only 22 tuples, compared to 42 in (12). It's a win! Of course we also need the global all-different constraint. (See exercise 13.)

If we only had more time, we could have discovered a completely different way to model problem (11) as a CSP, such as the approach in exercise 16.

Automating automobiles. We've already seen dozens and dozens of significant examples of constraint-based problems when we studied exact covering and SAT. But we certainly haven't exhausted all of the major applications, and several problems on our yet-unexamined list have been associated historically with the CSP. One of them, known as the *car sequencing problem*, is especially appropriate for us to study next, not only because its initials are "CSP" but also because it is problem 001 in CSPLIB — a noteworthy collection of benchmarks that was launched by I. P. Gent and T. Walsh in 1999 (see *LNCS* **1713** (1999), 480–481).

Consider the portion of an automobile assembly line where optional features are being installed on newly made vehicles. Some of the cars will be made with moonroofs; some will have heated seats; and so on. The assembly line is divided into work areas, one for each special feature. Work area w has space for q_w cars, where q_w is the number of time slots needed to install feature w as the conveyor belt moves the cars along. If at most p_w/q_w of the cars need that feature, p_w installers are on duty, one of whom will commence work when a car enters the area and walk with it until the installation is done. The car sequencing problem is the task of arranging a given set of cars into a sequence so that *no subsequence of* q_w *consecutive cars will include more than* p_w *that need feature* w.

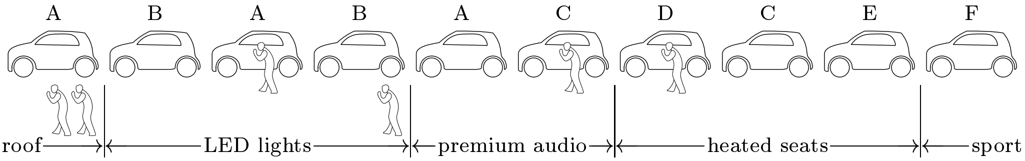


Fig. 100. Cars of models A, B, ... enter this assembly line at the far right, receiving optional features when they're in an appropriate work area. If this sequence has specifications (16), the final car (F) will be delayed in the LED area, because three cars in a row want that feature. The car sequencing problem tries to avoid such delays.

For example, there might be six models using the following subsets of five features:

Model	A	B	C	D	E	F	w	p_w	q_w
moonroof?	☐	☑	☐	☑	☑	☐	0	2	5
LED lights?	☑	☐	☑	☐	☑	☑	1	2	3
premium audio?	☐	☑	☑	☐	☐	☑	2	1	2
heated seats?	☐	☐	☐	☑	☐	☑	3	1	3
sport suspension?	☐	☐	☐	☐	☑	☐	4	1	5

(16)

Suppose ten cars of models $\{A, A, A, B, B, C, C, D, E, F\}$ are to be made. The sequence ABABACDCEF is almost correct, but it fails on the final car (see Fig. 100). Can you find a delay-free sequence? Notice that the left-right reflection of any solution is also a solution; we can rule out mirror images by requiring that model F, say, appears among the first five cars. Exercise 17 has the (unique) answer.

The car sequencing problem has boundary effects at the left and right that make it somewhat unrealistic. (Industrial assembly lines don't really start out empty every day!) Still, it's a nice clean problem, instructive to chew on.

An international competition was held in 2005, based on actual industrial data. It included additional constraints, such as the colors of paint to be used and the initial contents of the assembly line, and it inspired many creative solutions. [See C. Solnon, V. D. Cung, A. Nguyen, and C. Artigues, *EJOR* **191** (2008), 912–927.] The winning programs were based on local search methods analogous to WalkSAT, using “greedy” heuristics.

Line labeling in computer vision. Speaking of history, let’s turn now to some fascinating aspects of computer vision that influenced much of the early work on constraint processing. When a camera photographs a scene, it makes a two-dimensional image of three-dimensional reality; interesting problems arise when we try to reconstruct the reality from the image.

We’ll work with an extremely simplified yet powerful model, as the original researchers did: Our “reality” will be a world of special polyhedral objects, where exactly *three faces* meet at each of the vertices. For example, an ordinary cube or tetrahedron or  will qualify. But an octahedron will not, nor will an Egyptian-style pyramid, nor , because a vertex where four faces meet isn’t allowed. These three-faced concepts can be generalized, of course, but it’s helpful to start with a thorough understanding of the comparatively simple trihedral world.

More precisely, the 3D objects we shall deal with have no curved surfaces. They are defined by vertices, edges, and faces, where the vertices are “corners” at which edges and faces come together. All of the faces are “flat,” meaning that their points all lie on some plane. Each face is bounded by an exterior polygon, possibly with one or more interior polygons delimiting “holes” in the face. Each edge runs between two vertices and is part of the (infinite) line where the planes of two adjacent faces meet; it’s a segment of the polygonal boundaries of those faces. And significantly, *each vertex is the endpoint of exactly three edges.* We shall call such an object a *three-valent polyhedral object*, or 3VP for short. (See Fig. 101.)

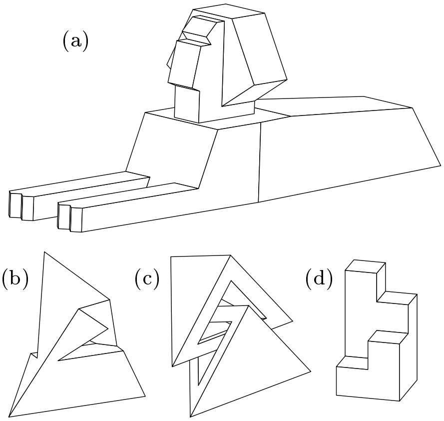


Fig. 101. Examples of 3VPs (three-valent polyhedra): (a) A stylized sphinx. [68 vertices, 102 edges, 38 faces.] (b) The Szilassi polyhedron, defined in exercise 26. Each of its seven faces is adjacent to all of the other six(!). [14 vertices, 21 edges, 7 faces.] (c) A clasp formed from two identical, interlocked objects, each of which is a tetrahedron from which a large triangular wedge has been hollowed out. [20 vertices, 30 edges, 14 faces.] (d) The histoscape for the matrix $\left(\begin{smallmatrix}4&3\\1&2\end{smallmatrix}\right)$, as defined in exercise 27. [20 vertices, 30 edges, 12 faces.] Many of the vertices, edges, and faces of these examples are invisible because they lie behind the parts that we *can* see.

The two-dimensional images shown here make sense to us, somehow, although significant depth information has been lost. In some mysterious way we’ve learned to rely on visual cues in order to understand what’s really present.

INDEX AND GLOSSARY

> *Index making has been held to be the driest*
> *as well as lowest species of writing.*
> *We shall not dispute the humbleness of it;*
> *but since we have had to make an Index ourselves,*
> *we have discovered that the task need not be so very dry.*
>
> — LEIGH HUNT, in *The Indicator* (1819)

When an index entry refers to a page containing a relevant exercise, see also the *answer* to that exercise for further information. An answer page is not indexed here unless it refers to a topic not included in the statement of the exercise.

$<$, $\leq$, $>$, $\geq$ relations, 87–88, 245.
$\perp$ state, 260.
$\Longleftrightarrow$: If and only if.

$0 \times n$ matrix, 241.
0-tuple, 113.
0/1/all constraints, 80.
0s and 1s, 107.
  matrices of, 80, 84, 148–149, 239–240.
2×2 matrices, 115, 118, 151.
2-regular graphs, 121.
2SAT problem, 46, 87, 239.
3-coloring a graph, 41, 91, 135–136.
3-cube, 116.
3-PARTITION problem, 194.
3-regular graphs, 44, 121.
3D objects projected to 2D, *see* Line labeling problem.
3DM (three-dimensional matching), 159.
3SAT problem, 46, 89, 90, 113, 130, 173.
3VP (3-valent polyhedral object), 9–16, 115–119.
4-cube, 201.
4-fold symmetry, 203, 214, 220–221.
4-letter words of English, 74.
4 queens problem, 139.
4-regular graph, 30.
4-vertex graphs, 134.
5-letter words of English, 29, 74, 138.
5-queens domination problem, 78.
5 queens problem, 139.
5-vertex graphs, 134.
8 queens problem, 51, 67.
13-colonies graph, 17, 121.
16 queens problem, 54, 65, 79, 139, 146.
90-degree rotational symmetry, 203, 214.
180-degree rotational symmetry, *see* Central symmetry.
$(132, 312)$-avoiding permutations, 219.
$(213, 231)$-avoiding permutations, 219.

α-graceful graphs, 123–124, 178, 183.
α-labeling, 182.
γ (Euler's constant), 188.
  as source of “random” data, 116.
λ (Laplace limit), 165.
μ (dual Laplace limit), 165.
νx (1s count, sideways sum), v, 58, 161, 193, 247.
π (circle ratio), *see* Pi.
π_j (projection to coordinate j), *see* Projection function.
$\Pi_\sigma R$ (projection of a relation), 48, 96, 99–101, 152, 241.
ϖ_n (Bell number), 64, 251.
ρx (ruler function, rightmost bit), v, 58, 183, 224.
ϕ (golden ratio), 163, 170.
  as source of “random” data, 116.

`A` (current age), 231.
a-code, *see* Asterisk codes for subcubes.
a.s.: Asymptotically almost surely, 130.
AAAI: American Association for Artificial Intelligence (founded in 1979); Association for the Advancement of Artificial Intelligence (since 2007).
Abrham, Jaromír, 175, 178.
AC3rm algorithm, 109.
AC6 algorithm, 54, 109.
AC2001 algorithm, 109.
Academic problems, 67, 106.
Acharya, Belmannu Devadas (ಬೆಳಮಣ್ಣು ದೇವದಾಸ ಆಚಾರ್ಯ), 184, 185.
Active items and options, 54, 62, 71.
`ACTIVE` pointer, 62, 71, 143, 231.
Active variables, 50, 138.
Acyclic digraph (dag), 82.
Acyclic orientations, 242.
Adamaszek, Michał Jan, 120, 175.
Adams, Peter, 190.
Adjacency matrices, 179.
Affine equivalence, 125–127.
`AGE(`o'`)`, 232.
Age of an option, 71–72, 144, 231.
Alekseev, Vladimir Evgenievich (Алексеев, Владимир Евгеньевич), 171.

Algebra of relations, 96, 152.
Algebraic number, 249.
All-different constraint, 6, 30, 34, 113, 131, 160, 164, 166, 213.
All-interval tone rows, 74.
All-ones matrices, 155, 241.
All shortest paths, 107.
All-zero matrices, 155, 241.
All-zero rows, 148.
Alon, Noga Mordechai (נוגה מרדכי אלון), 111, 204.
Alphabet blocks, 56, 74.
Amazing identity, 186.
Amazons, 157.
Analysis of algorithms, 22–23, 113, 114, 118–120, 122–125, 136–140, 142–144, 150.
Ancestor in a tree, 129.
AND operation, 142.
`ANDN` (and not), 201.
Anick, David Jay, 24, 122, 182.
Antiautomorphisms, 25, 125, 187.
Antisymmetric relations, 148.
Antisymmetries, 25, 125.
Appleton, Jenna Bou, 190.
Approximate GAD filtering, 133.
Arc consistency, 109, *see* Domain consistency.
Arcs, 3.
Arity, 3, 112, 139.
Artigues, Christian, 9.
Assignment of value to variable, 50, 54, 58, 106.
Associative law, 148.
Asterisk codes for subcubes, 129.
Asterisk in transition rule, 156.
Asymptotic methods, 113, 115, 122, 130, 140, 225, 243, 256–257.
At-least-one clauses, 39, 46.
At-most-one clauses, 39, 46, 137.
Atom in a molecule, 130.
Attributes in a database, 4.
Augmented Sierpiński simplex graph, 45.
Augmented Sierpiński tetrahedron, 45, 137.
Automaton, 112, 155–157.
Automorphisms, 17, 21, 23, 25, 32, 33, 38, 120, 121, 125, 128, 151, 153, 166, 178, 179, 181, 182, 187, 195, 199, 201.
Auxiliary variables, 46, 88–90, 212, 238, 239.
`AVAIL` list, 145, 240.
Average level, 143.
Axial symmetry, 203, 223.
Axioms, 148.
Aztec diamonds, 142, 184.
Aztec rectangles, 221–223.

Bacchus, Fahiem, 216.
Background plane, 172.
Backmarking, 38, 142.
Backtrack-free process, 80–85, 99–105.
Backtrack programming, iii, iv, 38, 65, 109–110, 120, 181, 192, 202, 219, 237.
Backtrack tree, 54, 136, 218.
Backward step, 142–143.
Bad strong component, 81.
Baker, Kirby Alan, 100, 252.
Balanced pentomino patterns, 78, 79.
Balanced ternary digits, 39, 85, 90.
Balls, two-dimensional, 78.
Bandwidth, 38, 113.
Barette, Olivier Franz René, 110.
Barto, Libor, 251, 254, 255.
Barycentric coordinates, 135, 172.
Basic constraints, 83.
Basis vectors, 244.
Baumert, Leonard Daniel, 109.
Baxter, Nicholas Edward, 74.
Bayramov, Rauf Abdulla oğlu, 110.
BDD: A reduced, ordered, binary decision diagram, 204, 244, 245.
Becket, Ralph William Allen Jude, 111.
Beek, Peter George van, 110, 111, 168, 216.
Beldiceanu, Nicolas, 112.
Bell, Alexander Graham, 44.
Bell, Eric Temple, numbers, 64, 251.
Beluhov, Nikolai Ivanov (Белухов, Николай Иванов), 186, 203, 204, 221, 222.
Benchmark problems, v, 7, 56, 64–65, 69, 74, 78–79, 111, 134, 211.
Bennaceur, Hachemi Ahmed, (هاشمي أحمد بن ناصر), 158.
Bent trominoes, 221.
Berge, Claude Jacques Roger, 198.
Bessière, Christian Paul André, v, 54, 109, 216.
Bi-degree of a vertex, 198.
Biaxial symmetry, 223.
Biconnected graph, 118.
Biere, Armin, 43, 211.
Bigraph, *see* Bipartite graph.
Binary branch, 50, 66, 77, 144, 218.
Binary constraints, 2, 53, 58, 85, 107, 109, 137, 139, 148, 149, 158, 215, 225.
  representing k-ary constraints, 47, 137, 139; *see also* Hidden variables.
Binary encoding, 40–42, 207–208.
Binary notation, 39.
Binary recurrence, 122.
Binary relations, *see* Binary constraints.
Binary representation, 135.
Binary search trees, 219.
Binary trees, 135.
Binding, 50, 142.
Binomial tree, 122.
Bipair, 236.
Bipartite graph, 33–34, 38, 47, 76, 121, 123, 124, 131, 132, 159, 191, 198, 215.

Bipartite matching, ix, 33–34, 37, 131–132, 135.
Birkhoff, Garrett, 92.
Birthday, 146.
Bit vectors, 58, 60, 142, 212.
Bitsets, 60, 142.
Bitwise operations, 54, 58, 108, 133, 181, 201.
  AND (&), 181, 193, 212, 224.
  complement, 174, 193.
  OR (|), 193, 224.
  XOR (⊕), 193.
Bitwise parity constraint, 104.
Bivariate exponential generating functions, 165, 241–242.
Bizarre graph, 64.
Blank (unmatchable) color, 237.
Blocked option, 71.
Bloom, Gary Stuart, 126, 188.
Blurred state, 174–175.
Bodnarchuk, Viktor Gavrilovich (Боднарчук, Виктор Гаврилович), 86, 110.
Boltzmann, Ludwig Eduard, 5.
book graphs, 134.
Boolean algebra, 43.
Boolean binary relations, 87, 149.
Boolean clones, 251.
Boolean functions, 89, 95, 204, 245.
  monotone, 87, 95, 240, 245, 246, 254.
  self-dual, 95, 150, 245, 254.
Boolean lattice, 24, 125, 126.
Boolean matrices, 58, 140, 215.
  multiplication of, 107.
Boolean operations, 151, 245.
Boolean satisfiability, iii, 1.
Boolean ternary relations, 87–90, 150, 246.
Boolean variables, 2, 8, 39, 113.
Bosák, Juraj, 22.
`BOUND` field, 77–78.
Boundary cycle, 13, 117–118, 163, 170.
Bounded degree, 130, 196.
Bounded permutations, 242.
Bounds consistency, 112.
Bousquet-Mélou, Mireille Françoise, 243.
Boussemart, Frédéric Claude, 67, 68.
Bowtie graph ($K_1 \mathrel{-\!\!-\!\!-} 2K_2$), 180.
Bowtie operation $\bowtie$ (join), 96, 241.
Bowtie relation between edges, 128–129.
Brady, Zarathustra Elessar (= Zeb), 111.
`BRAIN83` (brain graph), 30–31, 35, 131.
Branching option, choice of, 77–78.
Branching variable, choice of, 50, 66, 138.
Breadth-first search, ix, xi, 196, 200.
  versus depth-first search, 175.
Breaking symmetries, 21, 156, 176, 223.
Bridge in a graph, 170, 180.
Brinkmann, Gunnar, 179.
Bryant, Darryn Edward, 190.
Bshouty, Nader Hanna (نادر حنا بشوتي, נאדר חנא בשותי), 257.
Bucket sort, 133, 233.
Bulatov, Andrei Arnoldovich (Булатов, Андрей Арнольдович), 97, 98, 111, 254, 259.
Bull graph, 134.
Buratti, Marco, 127.
Burnside, William, lemma, 188, 191.

C_5 (5-cycle), 18, 22, 29, 67, 113, 128.
C_6 (6-cycle), 22, 191.
$C_{\vec{m}}$ (oriented m-cycle), 26, 126, 150, 159.
C_n (cycle graph of order n), 36, 123, 178, 180.
Cabon, Bertrand, 56.
Cache-friendly data structure, 60.
Cahit Arkut, İbrahim, 183.
Cancellation laws, 154.
Canonical forms, 21.
Canonical solutions, 157.
Car sequencing problem, 7–9, 38, 114.
Cardon, Stéphane Michel, 216.
Carletti, Vincenzo, 200.
Carlsson, Mats Hugo, 112.
Carmichael, Robert Daniel, 189.
Cartesian coordinates, 116–117, 162.
Cartesian product, 2, 113.
  of graphs ($G \mathbin{\square} H$), 29, 128, 129; *see also* Grid graphs.
  of relations ($R \times S$), 96, 152.
Catalan, Eugène Charles, numbers, 225.
Caterpillar nets, 24, 124, 185.
Cauchy, Augustin Louis, 165.
Central symmetry (180° symmetry), 10, 203, 223, 244, 261.
Centroid of a free tree, 221.
Certificate of correctness, xi, 261.
Certificate of unsatisfiability, 112, 157, 261.
Chain problem, 140.
Channelling constraints, 8.
Chatterjee, Sourav (সৌরভ চ্যাটার্জী), 130.
Chaucer, Geoffrey, 158.
Checkerboard pattern, 221.
Chemical compounds, 108, 130.
Chen, Hubert (= Hubie) (張胡比), 111.
Chen, Xinguang (= Albert) (陈新光), 216.
Chessboard, 64, 133, 147, 184.
Chesterton, Gilbert Keith, 1.
Choi, Chiu Wo (= Jeff) (蔡昭和), 112.
Chordless cycle, 191.
Chosen option, 71.
Christie Mallowan, Agatha Mary Clarissa Miller, 132, 157.
Chromatic numbers, 121.
Chunky-Octs®, 65, 142.
Chvátal, Václav (= Vašek), graph, 30–32, 35, 131, 132, 135.
Circuit analysis, 108.

Clauses, 1, 39.
  learning, 108, 261.
Clément de Saint-Marcq, Vianney Emmanuel Adrien Martin Marie Joseph, le, 110.
Clique hints, 44, 136, 137, 207–211.
Cliques, 3, 20, 29, 37, 65, 159, 204, 211.
Clone, 94–97, 151, 152, 251, 252.
Closed knight's tour, 75–76, 146.
Closed subsets, 93, 96, 150.
Clowes, Maxwell Bernard, 10, 11, 107.
`CLR` field, 61–64, 143.
Co-clones, *see* Relational clones.
Codd, Edgar Frank, 4.
Cohen, David Avram, 100, 110, 111, 241, 254.
Cohn, Paul Moritz, 2, 251.
Cohoon, David Kent, 160.
Collinear points, 237.
Color of an item, 77, 225.
Coloring a graph, iv, 41–45, 64, 135–138, 259.
Column convex matrix, 84.
Column sums, 242.
Combos, 94, 251.
Commas, 2.
Commutative law, 97, 148, 254.
Commuting matrices, 249.
Compact Maltsev stubs, 104, 154–155.
Compact representation of digraphs, 26, 189.
Compact stub for an NU polymorphism, 101–103, 111, 154–155.
Compact-Table data structure, 110, 142.
Comparison of multisets, 131.
Compatible options, 54, 69, 77, 144.
Compatible vertices, 130.
Competitions, 9, 111, 211–212.
Complement of a graph, 159, 160, 190, 200.
Complement of a labeling, 16, 19–21, 120, 122.
  edge-complement, 123.
Complement symmetry, 178.
Complete binary tree, 195, 209.
Complete bipartite graph ($K_{m,n}$), 123–124, 180, 188, 242.
  oriented, 126.
Complete graph (K_n), 3, 19, 108, 120–122, 124, 195; *see also* Cliques.
Complete relation $[0\,..\,d)^n$, 80, 84, 102, 152, 154, 163, 213, 245, 258.
Complex variable theory, 165.
Complexity of computation, 168, 173.
Components of a graph, *see* Connected components, Strong components.
Composite numbers, 76.
Composition of binary relations, 53, 139, 152, 248.
Composition of functions, 94, 153, 255.
"Compound" Boolean variables, 215.
Computer science versus mathematics, 239.
Computer vision, iv, 9, 158.
Concatenating relations, 96.
Concave edges, 10–12, 116, 166.
Conditional branches, 227.
Configurations, 4–5.
Conflict clauses, *see* Preclusion clauses.
Conflict-directed heuristic, 67.
Congruence enumeration, 188.
Conjugates of an algebraic number, 249.
Connected components, 117, 121, 122, 124.
Connected graphs, 129.
Connected patterns, 133.
Connected row convex constraints, 84–85, 110, 148–149.
Connectome, *see* `BRAIN83`.
CoNP-complete, 49.
Consecutive 1s, 169.
Conservative polymorphisms, 154, 244, 246, 253, 256.
Consistency, 48–53, 80, 109, 112, 156; *see also* Bounds consistency, Domain consistency, Forward consistency, Pairwise consistency, Path consistency, Singleton domain consistency.
Constant function, 253.
Constant relation, 3, 91.
Constraint, 1, *see* Relation.
Constraint network, 12, 79, 107.
Constraint satisfaction automaton, 112, 155–157.
Constraint satisfaction problem, 1–5, 115.
Contests, 9, 111, 211–212.
Contiguous United States of America, 18, 32–35.
Contraction of a graph, vi, 209.
Contributing to a compact stub, 102, 154, 256–258.
Converse of a digraph or binary relation, 126, 245.
Convex edge, 10–12, 116, 166.
Convex polyominoes, 243.
Coolsaet, Kris Karel, 179.
Cooper, Martin Charles, 100, 110, 150, 240, 241.
Coordinates, 116–117, 162.
Coplanar, 166.
Copy of a graph, *see* SIP.
Copying domains, 201.
Cordella, Luigi Pietro, 200.
Core of a set of constraints, 106, 153.
Correspondence relation, 80.
Counting, complexity of, 259.
cover(i), 54.
Covering the cycles, 135.
Covering the nogoods, 212.
Covering with disks, 146.
CRC (connected row convex) constraints, 84–85, 110, 148–149.

Creases, 166.
Critical block, *see* Hall set.
Crossword diagram, 79.
Crystal maze puzzle, 6, 113.
CSA: Constraint satisfaction automaton, 112, 155–157.
CSP: The constraint satisfaction problem, 1, 30, 205.
  represented as SAT, 2, 113.
  represented as XC, 135.
  represented as XCC, 2, 113, 135.
CSP solvers, 39, 112.
CSPLIB problems, 7.
Cube, subcubes of, 129.
Cubes, colored, 64.
Cubic (3-regular) graphs, 44, 121.
Cubies, 115, 167.
Cung, Vạn Đạt, 9.
Current domain, 50.
Cuts, Dedekind, 249.
Cutting edges, 196.
Cycle graph C_n, 22, 36, 67, 113, 124, 178.
  when α-graceful, 123.
  when graceful, 183.
  when rainbow graceful, 28.
Cyclic (v, k, λ)-difference set, 126, 127.
Cyclic shifts, 156.

D_v (domain of variable v), 52.
$d^+(v)$ and $d^-(v)$ (out-degree and in-degree), 126.
d-ary branching, 66, 75.
d-ary sequence: Sequence consisting of digits $\{0, 1, \ldots, d-1\}$, 55.
d-way branching, 66, 77, 109, 139, 144, 218, 230.
Dag: A directed acyclic graph.
Dalmau Lloret, Víctor Xavier, v, 111, 255.
Dancing cells, iv, 61–65, 77, 146, 227.
  versus dancing links, 64–65, 73, 78, 147.
Dancing links, iii, 55, 59, 69, 73, 75, 77, 79, 142, 146, 208, 211, 238.
Dart graph, 180.
Data compression, 99, 104.
Data structures, ix, 17, 20, 26, 55–62, 72, 102, 110, 112, 121, 133, 142, 145, 174, 182, 200, 214.
Databases, 4, 96, 216.
DC: Domain consistency, 50.
de Givry, Simon, 56.
deactivate(i), 235.
Dead state, 260.
Debruyne, Romuald Gérard Édouard, 216.
Dechter, Rina Kahana (רינה כהנא דכטר), 110, 111, 215, 225.
Decision problem, 96, 117, 119.
Decomposable constraints, 48, 100, 102, 137, 246.
Dedekind, Julius Wilhelm Richard, 87, 249.
Degenerate binary search trees, 219.
Del Fra, Alberto, 127.
Delayer, 218.
Deletion from sparse-set, 227.
Delorme, Charles Jean-Pierre, 126.
Demaine, Erik Dylan Anderson, 223.
Demeulenaere, Jordan Patrick Christian, 110.
Demirović, Emir, 112.
Density of d's, 141.
Dependence on arguments, 91.
Depth-first forest, vi–vii.
Depth-first search, vi–xi.
  versus breadth-first search, 175.
Derived operators, 94–95.
Determinant, 116, 122, 170, 182.
Deville, Yves Dominique Bernard, 110, 196.
Dhar, Vasant Kumar (वसन्त कुमार धर), 213.
Diaconis, Persi Warren, 130.
Diagonalization, *see* Equality selection relation.
Diameter 4, 24.
Dichotomy conjecture, 111.
Dichotomy theorem, 96–98, 111, 155, 259.
Dieker, Antonius Bernardus, 217.
Differential equation, 165.
Digits, extended hexadecimal, 188, 205.
Digraceful graph, 27, 126–127.
Digraphs (directed graphs), 2–3, 81–82, 125, 130, 132, 135, 152, 161, 195, 239, 248.
  graceful, 24–27, 125–127.
  representation of, 26, 125.
  small, 126.
Dinçbaş, Mehmet, 8.
Direct encoding, 2, 39, 40, 46, 47, 137, 207–212.
Direct product of graphs $(G \otimes H)$, 124.
Directed acyclic graphs, 38, 130, 242.
Directed cycles, 135.
Directed graphs, *see* Digraphs.
Discrete disks, 78, 146.
Discrete Fourier transform, 163.
Discriminator, 152, 244.
Disequality, *see* Nonequality relation.
Disney, Walter Elias, 159.
Distinct representatives, 197.
Distributive lattices, 197.
Dobbelaere, Bert Omer Rachel, 181.
Dodecahedron, 121, 131.
dom/wdeg heuristic, 68.
Domain consistency, 50–53, 69–73, 83–85, 109, 112, 138–140, 142, 145, 148, 174, 211, 221, 239.
  versus Forward consistency, 51–56, 74–76, 138, 140.
Domain filtering, 49–53, 139.
  bitwise, 201.
Domain versus range, 113.

Domains, 1, 8, 19, 70, 98, 113, 133, 147, 199.
  representation of, 56–60, 142, 147.
Domination, 78.
Dongen, Marc Remko Casper van, 111, 211, 217.
Double paw graph, 118, 180.
Double-snake puzzle, 64.
Doubly linked lists, 58–61, 63, 72, 142, 225.
Downloadable programs and data, ii, iv–v, 163, 165, 181, 182, 206, 220, 230, 246.
Downpath, viii.
Draper, Stephen William, 171.
Dreweke, Alexander, 196.
Dual discriminator, 244.
Dual Horn clauses, 49, 215.
Dual Laplace limit, 165.
Dual of a Boolean function, 245.
Dual of a CSP, 139.
Dudeney, Henry Ernest, 64.
Dull, Brutus Cyclops, 160.
Dürer, Albrecht, 115, 203.
Duval, Jean-Pierre, 260.
Dynamic programming, 162, 164, 172.
Dynamic variable ordering, 65–69, 144.

e (base of natural logarithms), as source of "random" data, 141.
ea(f) (essential arity of f), 91–92, 151.
ECAI: *European Conference on Artificial Intelligence* (1982–), formerly called *Artificial Intelligence and Simulation of Behavior* (1974–1980).
Echelon form, 244.
Edges of a graph, 3, 9.
Efficiency, 54–56, 121, 147, 239.
Eigenvalues, 170.
EJOR: *European Journal of Operational Research* (1977–).
El-Zanati, Saad Ibrahim (سعد إبراهيم الزناتي), 183, 190, 191.
Elementary bipartite graphs, 198.
Elkies, Noam David (נעם דאד אלקס), 122.
Ell tetrominoes, 147, 167.
Elliott, Gordon Lewis, 109.
Embedding of graphs, iv, *see* SIP.
  isometric, 29–30, 128–129, 194.
  strict, *see* ISIP.
Empty queue, 145.
Empty set ($\emptyset$), 96, 99, 106, 240.
Empty (unsatisfiable) clause, 43.
Empty (unsatisfiable) relation, 152, 213, 245, 252.
empty_q() subroutine, 72, 145.
Encoding into SAT clauses, iv, 39–42, 46–48, 135, 137, 207–212.
Endomorphisms, 86, 106, 153.
Energy, 4–5, 113.
Equality relation, 107, 152, 245, 248, 256.
Equality selection relation, 96, 152.
Equivalence relations, 103, 129, 154, 157.
Equivalent relations, 138.
Ernst, Bruno (= Joannes (= Hans) Adrianus Friedrich de Rijk), 171.
Error-correcting codes, 175.
Essential arity, 91–92, 151.
Essential spectrum, 151.
Essentially different solutions, 17, 32, 125, 126, 131, 187.
Eulerian numbers, 219.
Eulerian trail, 113.
Even/odd coordinate system, 135, 163.
Exact cover problem, 31, 39, 114, 130, 136–138, 141, 146, 190.
  with colors, *see* XCC problem.
Exercises, notes on, iv.
Exponential generating function, 115.
  bivariate, 165, 241.
Exponentially large, 218.
Extended binary tree, 135.
External nodes, 94.
Extreme (p/q) string, 114.
"Extreme" XC problem, 64, 145.

Faces of a polyhedron, 9–10.
Factorial series, 121.
Factorization, 185.
"Fail early" strategy, 68.
Failed literals, 216.
Failure rate based heuristic (FRB), 68.
Family of sets, 245.
Fano, Robert Mario, hypergraph, 215.
Far vertices, 133.
FC: Forward consistency, 50.
Feder Bermann, Tomás, 4, 111, 239, 254.
Feedback vertex set, 135.
Fibonacci, Leonardo, of Pisa (= Leonardo filio Bonacii Pisano), numbers, 42, 118, 161, 169, 208.
Field (in algebra), 151.
  finite, 79, 127.
Fikes, Richard Earl, 107, 109, 158.
Fillomino puzzles, iv, 56, 74, 141.
Filter all domains, 83, 112.
Finite fields, 79, 127.
`FIRST` array, 17, 26, 120, 125.
First item of an option, 145, 232.
First item of every option, 72.
First moment principle, 196.
Fischer, Ingrid, 196.
Fischer, Julien Siegfried, 111.
Five-letter words of English, 29, 74, 138.
Fix a variable, *see* Assignment of value to variable.
Fixit stack, 71–72, 145.
`FLAG` variable, 63, 143.
Flippo, Maarten Lucas, 112.
Floating point calculations, 68, 229, 230.

Flower snark graphs, 131.
Flower snark line graphs, 44, 136–137.
Floyd, Robert W, 57, 107.
Focus, 69, 208.
Foggia, Pasquale, 200.
Football teams, 204.
Forced moves, 65, 67, 73, 78, 144, 227, 235.
Forcing representations, 48.
Forests, vi, 38, 104, 128, 130.
Fork graph, 23.
Forking, 103–105, 154–155, 257–258.
Forward consistency, 50–54, 58, 69, 71, 73, 109, 138–140, 216, 218, 229.
  versus Domain consistency, 51–56, 74–76, 138, 140.
Forward step, 142–143.
Fourfold symmetry, 203, 214, 220–221.
Fourier, Jean Baptiste Joseph, transform, 163.
Fractal, 118.
FRB heuristic, 68, 73, 144, 230, 234, 238.
FRB† heuristic, 75, 230, 234.
Free boundary, 14, 169, 171.
Free trees, 23, 38, 122, 134.
Freese, Ralph Stanley, v, 257.
Freuder, Eugene Charles, 55, 110, 239.
Friedman, Erich Jay, 223.
fsnark(q), 44, 136–137.
Full function, 91–92, 151.
Fun, 6, 16, 32, 38, 44, 76.
Fun fact, 121.

Gμ: One gigamem (one billion memory accesses), 73.
G_m^π (“random” graceful graph), 121.
GAD (globally all different) filtering, 34–35, 37, 112, 131–133.
Gadgets, 88–91, 96, 98, 150, 154.
  universal, 90.
Gallian, Joseph Anthony, 28.
Galois, Évariste, 151.
  connections, 93, 110.
games graphs, 134.
García-Molina, Héctor, 4.
Gardner, Martin, 108, 159–160, 168.
Garey, Michael Randolph, 194.
Gasarch, William Ian, xiii.
Gaschnig, John Gary, 109, 225.
Gault, Richard Leslie, 246.
Gauß (= Gauss), Johann Friderich Carl (= Carl Friedrich), elimination algorithm, 79, 107.
GCP (general combinatorial problem), 3, 113.
Geek art, xiv, 16, 119, 201–203, 253.
Geiger, David Scott, 110.
General combinatorial problem, 3, 113.
General position, 11, 117, 171, 172.
Generalized majority-minority operation, 153.
Generalized toruses, 184.
Generating all solutions, *see* Visiting all solutions.
Generating functions, 5, 113–115, 149, 165, 208, 241, 243, 244.
Gent, Ian Philip, 7, 198.
Geometric series, 217.
Gerdjikov, Stefan Vladomirov (Герджиков, Стефан Владимиров), 222.
Gessel, Ira Martin, 162, 165.
Giant component, 253.
Gigamems (billions of memory accesses), 73.
Gill, Mukhtiar Kaur (ਮੁਖਤਿਆਰ ਕੌਰ ਗਿਲ), = Acharya, Mukti (ਮੁਖਤ ਅਚਾਰੀਆ), 184.
GIP (graph isomorphism problem), 30.
Givry, Simon de, 56.
Glasgow solver, 201, 203.
Global constraints, 6, 107, 112, 156.
Global variables, 63.
Globally All Different filtering, 34–35, 37, 112, 131–133.
Gocht, Stephan, 112.
Goedgebeur, Jan, 179.
Goldberg, Reuben Garrett Lucius, 89.
Golden ratio (ϕ), 163, 170.
  as source of “random” data, 116.
Golomb, Solomon Wolf, 108, 109, 120, 121.
González-Morris, Germán Antonio, 223.
Googol (10^{100}), 141.
Gordon, Daniel Martin, 189, 191.
Goriely, Alain Isaïe, 30.
Grabarchuk, Serhiy Oleksiyovych (Грабарчук, Сергій Олексійович), 64.
Graceful digraphs, 24–28.
Graceful labeling, iv, 16–28, 38, 55, 108, 120–128, 178, 182–183.
Graceful permutations, 19, 120, 121.
Graceful tree conjecture, 24, 28.
Graham, Ronald Lewis (葛立恒), 194.
Graph coloring, 41, 64, 138, 159, 259.
Graph isomorphism problem, 30.
Graph labeling, *see* Graceful labeling.
Graph representation, 17, 26, 125, 187.
Graph theory, iii.
Graphs, small, 121, 126.
Gray, Frank, code, 122, 182.
  for 4-vertex graphs, 134.
Greatest common divisor, 249.
Greedy heuristics, 9.
Green, Chris, 220.
Grid graphs ($P_m \mathbin{\square} P_n$), 33, 123, 124, 131, 132, 138.
Ground states, 5.
Gruenberger, Fred Joseph, 159.
GTC: The graceful tree conjecture, 24, 28.
Gyssens, Marc Luc Henri Karel, 110, 111, 241.

H-coloring problem, 159.
Half edges, 10, 163.
Hall, Marshall, Jr., 198.
Hall, Philip, 197–198, 251.
  set, 131–132, 197–198, 200.
Halo of a polyomino, 222–223.
Hamiltonian cycles and paths, 37, 146, 160.
Hamming, Richard Wesley,
  distance, 191–192.
  embedding, 128.
  graph, 128.
Haralick, Robert Martin, 109.
Hardy, Godfrey Harold, 188.
Hartert, Renaud Fortunat, 110.
Hartke, Stephen Gold, 190.
Harvey, Warwick, 112.
Hash code for a graph, 179, 180.
Haystack problem, 137.
HC network, 12–13, 117–119.
HC picture, 10–16, 117–119.
Head of list, *see* List head.
Heap-ordered lists, 208.
Heawood, Percy John, 166.
  graph, 179, 215.
Height of restricted growth string, 251.
Hell, Pavol, 159.
Hello world, 6.
Hémery, Fred, 67, 68, 109.
Heuristic function $h(i)$, 67, 73, 144, 228.
Hexadecimal code, extended, 188, 205.
Hexadecimal constants, v.
Hexagon, 65, 197.
Hexagonal grid, 119.
Hexominoes, 221.
Hidden variables, 47, 53, 142, 148, 213, 215.
$\mathrm{hide}(i,c)$, 62–64, 143, 230.
Hints for d-cliques, 45, 145, 233.
Hinz, Andreas Michael, 209.
Historical notes, 8, 68, 86, 107–111, 158, 159, 165, 166, 171, 175, 176, 182, 183, 188, 189, 191, 197, 198, 201, 204, 207, 213, 215–219, 237, 238, 246, 251, 252, 254, 255, 257.
Histoscapes, 9–13, 52–53, 115, 139, 173.
Ho, Boon Suan (何文轩), 205.
Homogeneous equations, 190.
Homomorphisms, 3–4, 30, 111, 113, 147.
Honaker, Garland Lee, Jr., 75.
Hopcroft, John Edward, algorithm with Karp, ix–xi, 37–38, 131, 135.
Horizontal and vertical symmetry, 203.
Horn, Alfred, clauses, 49, 148.
Horn core algorithm, 49, 53, 139, 216.
Horn-SAT, 259.
Horsley, Daniel James, v.
House of Graphs, 179, 180.
Hsu, Derbiau Frank (許德標), 126, 188.
Huffman, David Albert, 10–16, 107, 171.
Hugo, Victor Marie, 204.
Huhn, András Péter, 252.
Hui, Roger Kwok-Wah (許國華), 238.
Human Connectome Project, 31.
Hunt, James Henry Leigh, 264.
Hwang, Joey Cho Yee (黃祖怡), 218.
Hyperbindings, 215.
Hypergraphs, 47, 215.
Hypersolid pentominoes, 79.

Icosahedron, 121.
Idempotent operators, 97, 98, 148, 246, 253–255.
Identity relation, 107, 152, 245, 248, 256.
IJCAI: *International Joint Conference on Artificial Intelligence* (1969–), 225.
ILOG Solver, 110.
Image of a graph or vertex, 28.
Implicational constraints, 80–82, 85–86, 110, 147, 149.
Impossible objects, 15–16, 119.
Inactive options, 232.
Inactive variables, 50, 54, 138.
Inclusive ancestor in a tree, 129.
Incompatible options, 54, 69, 77, 144.
Independent sets, 113, 204.
Indicator problems, 87–91, 96, 98, 110, 149, 150, 153, 245–247, 255, 256.
Induced subgraph isomorphism problem, *see* ISIP.
Inefficiency, 239.
Inequality relation ($x \neq y$), *see* Nonequality relation.
Initial domains, 1, 33, 36, 132, 142, 154, 200.
Inner degree, 134.
Inprocessing, 55.
Input states, 155.
Instantiation, *see* Assignment of value to variable.
Intel Xeon computer, 211.
Interior angles, 168.
Internal code number, 67.
Internal nodes, 94.
Internal numbers, 61.
Internet, ii, iv–vi, ix, 30, 166, 179, 205, 242.
Intersecting sets, 149.
Intersection of planes, 14–15.
Intersection of relations, 152, 241.
Intersection of sets, 147.
Intervals, 112, 143, 185, 219.
$\mathrm{Inv}(O)$, 95–96, 110.
Invariant properties of data structure, 62.
Invariant properties of relations, *see* Polymorphisms.
Inverse arrays, 59, 62.
Inverse of a permutation, 140.
Inverse variables, 8.
Inversions of a permutation, 164.
Irreflexive binary relation, 3.

Ising, Ernst, 4.
  configurations, 113.
  model, 4.
ISIP (induced subgraph isomorphism problem), 29–30, 108, 128, 130, 134–136, 200.
Isolated vertices, vii, 17, 23, 121, 174, 204.
Isometric embeddings, 29–30, 128–129, 194.
Isomorphism between graphs, 30, 113, 179, 196.
ITA Software, 238.
`ITEM` array, 61–64, 225.
Items and options, 1, 77.
Iteratively closed systems, 251.
`ITM` field, 61–64, 143.

J (all-ones matrix), 155, 241.
Jacobi, Carl Gustav Jacob, symbol, 119.
Jacquet, Philippe Pierre, 165.
Jakovac, Marko, 209.
Janson, Carl Svante, 218.
Janssen, Philippe, 216.
Järvisalo, Matti Juhani, 158.
Jeavons, Peter George, v, 3, 4, 100, 110, 111, 113, 159, 240, 241, 246, 247, 254.
Jégou, Philippe Rémy Paul, 216.
Jelliss, George Peter, 75.
Joachimsthal, Ferdinand, 166.
Johnson, David Stifler, 194.
Johnson, Samuel, 262.
Join graph, 216.
Join of relations ($R \bowtie S$), 4, 96, 241.
  lossless, 48.
Joke, 191.
Junctions, 10–14, 117–119, 163, 171–173.

k-ary constraints, 109, 142, 158, 225.
  regarded as binary constraints, 47, 137, 139; *see also* Hidden variables.
k-ary operations, 86, 94.
k-center problem, 146.
k-dimensional matching, 159.
k-tuples, 147.
k-wise projections, 99, 154.
$K_{2,1,1}$, 191.
$K_{3,3}$, 168.
$K_{m,n}$ (complete bipartite graph), 123–124, 180, 188, 242.
$\vec{K}_{m,n}$ (oriented complete bigraph), 126, 206.
K_n (complete graph), 3, 19, 108, 120–122, 124, 195; *see also* Cliques.
Kμ: One kilomem (one thousand memory accesses), 73.
Kaluzhnin, Lev Arkadievich (Калужнин, Лев Аркадьевич), 86, 110.
Kane, Daniel Mertz, 170.
Kaneko, Masanobu (金子昌信), 241.
Kaplansky, Irving, 241.
Karnaugh, Maurice, map, 188.
Karp, Richard Manning, algorithm with Hopcroft, ix–xi, 37–38, 131, 135.
Kazda, Alexandr, 252.
KC graphs ($K_n \mathbin{\square} C_r$), 22, 121, 176.
Kearnes, Keith Alan, 255.
Keith, Michael, 75.
Kenig, Michael Joseph, Jr., 183.
Kilomems (thousands of memory accesses), 73.
Kingwise adjacent, 222.
Kingwise connected, 85, 142, 241.
Kirkman, Thomas Penyngton, 191.
Kirousis, Lefteris Miltiades (Κυρούσης, Ελευθέριος Μιλτιάδης), 110, 119, 173.
Kiss, Emil W., 252.
Kissat solver, 43, 211.
Kite designs, 44.
Klavžar, Sandi, 207, 209.
Kløve, Torleiv, 175.
Knight graph (N_n), 133–134, 146, 234, 237.
  embedded in queen graph, 134.
Knight of Holinesse, 1.
Knight plus queen (superqueen), 157.
Knight's grids, 133.
Knight's tours, iv, 75–76, 146.
Knights and queens, nonattacking, 146–147.
Knuth, Donald Ervin (高德纳), i, v, 38, 43, 55, 61, 75, 111, 146, 163, 165, 166, 177, 179–182, 206, 208, 209, 220, 225, 227, 229, 232, 242.
Knuth, Nancy Jill Carter (高精兰), v.
`@kobouzu17`, 220.
Kőnig, Dénes, 198.
Koplowitz, Herbert Philip, 160.
Kotěšovec, Václav, 261.
Kotov, Viktor Nikolaevich (Котов, Виктор Николаевич), 86, 110.
Kotzig, Anton, 24, 175, 178, 179, 182, 183.
Kozik, Marcin, 251.
KP graphs ($K_n \mathbin{\square} P_r$), 20–22, 120, 121, 123, 176.
Krasner, Marc Isaacovich (Краснер, Марк Исаакобич), 186.
Krokhin, Andrei (Крохин, Андрей Анатольевич), 111, 254, 255.
Kumudakshi Suvarna (ಕುಮುದಾಕ್ಷಿ ಸುವರ್ಣ), 189.
Kurchan, Rodolfo Marcelo, 223.

$L(2,1)$ labelings, *see* Radio colorings.
$L_{m,n}$ (lollipop graphs), 180, 189, 204.
$L(r,s)$ lattice of partitions, 197.
Labeled graph embedding, 130.
LAD (locally all different) filtering, 33–37, 131, 132, 200.
Ladner, Richard Emil, 97.
Lagrange, Joseph Louis (= Giuseppe Lodovico Lagrangia = Giuseppe Ludovico (= Luigi) De la Grange Tournier), 166.
Langford, Charles Dudley, pairs, 65.

Laplace (= de la Place), Pierre Simon, Marquis de, limit constants, 165.
Larrosa Bondia, Francisco Javier, 134.
Latin squares, 154, 209.
Lattice, 197; *see also* Boolean lattice, Semilattice.
Lax Maróti pair, 98, 153.
Lax Siggers term, 153, 253.
Lax weak near unanimity operation, 97, 152–153.
Laxity, 153.
Lazy updates, 148.
le Clément de Saint-Marcq, Vianney Emmanuel Adrien Martin Marie Joseph, 110.
Leader, vii, viii.
Leaper, 204, 261.
Learning, 108.
Least common multiple, 249.
Least upper bound, 83, 148.
Lecoutre, Christophe, 67, 68, 109–111, 214, 216, 223.
Lee, Chester Chi Yuan (李始元) = Chi Lee (李濟), distance, 27.
Lee, Jimmy Ho-Man (李浩文), 112.
`LEN` field, 55, 63, 208.
Lenz, Wilhelm, 4.
Les Misérables, 204.
Level, 144, 227, 228.
Levels, 66.
Lexicographic order, 106, 115, 154, 157, 163, 259.
Li, Hongbo (李宏博), 68.
Li, Zhanshan (李占山), 68.
Lichtenstein, David Isaac Joseph, 173.
LIFE magazine, 263.
Line graphs, 44, 199, 209.
Line labeling problem, 9–16, 38, 52, 107–109, 115–119, 139.
Linear equations and linear inequalities, 119.
Linear equations mod d, 79, 86, 149.
Linear independence, 190, 244.
Linear programming, 173.
Linear time, 80–82, 173, 239.
Linked lists, 17, 26, 82, 142, 145, 147.
LIPIcs: *Leibniz International Proceedings in Informatics* (2008–).
List head, 59, 147, 225.
Literals, 1, 90, 158.
`LO` array, 17, 23, 26, 120, 122, 125, 189.
Lo Shu magic square, 203.
Lobjois, Lionel Jean, 56.
`LOC` field, 61–64, 143.
Local constraints versus global constraints, 107, 112.
Locally All Different filtering, 33–37, 131, 132, 200.
Log encoding, 2, 39–42, 44, 46, 136, 137, 207–211.
Lollipop graphs ($L_{m,n}$), 180, 189, 204.
Lonesum matrices, 242.
Longest cycle, 175.
Longest path, 120.
Lossless join dependencies, 48.
Lovász, László, 198.
Lower bounds for resolution, 136.
Loyd, Samuel, 237.
Lucas, François Édouard Anatole, numbers, 118, 171, 208.
Luks, Eugene Michael, 196.
LWNU: Lax weak near unanimity, 97, 152–153.
Lyndon, Roger Conant, words, *see* Prime strings.

m-fold composition, 248.
Mμ: One megamem (one million memory accesses), 73.
Mν: One meganode (one million nodes), 73.
$\binom{[m]}{k}$ (combinations as k-tuples), 99, 256.
Mackworth, Alan Keith, 48, 109, 158.
MacMahon, Percy Alexander, 64, 65.
Magic masks, 90.
Magic squares, 203.
Magical 4 × 4 torus, 188.
Magnetization, 113.
Mahéo, Maryvonne, 183.
Maintaining domain consistency, 52–56, 69–73, 110.
Majority operator, 100, 244.
Majorization, 196–197.
Maltsev (= Mal’cev), Anatoly Ivanovich (Мальцев, Анатолий Иванович), 254–255, 257.
  polymorphisms, 103–105, 149, 151, 153–155, 254–255, 257–259.
  stubs, 104, 154–155.
Marijnissen, Imko Cornelis Wouter Maria, 112.
Marković, Petar (Марковић, Петар), 255.
Maróti, Miklós, 111, 251–252.
  pairs, 98, 153, 155.
  pairs, double, 255.
Matching, 38.
  bipartite, ix, 33–34, 37, 131–132, 135.
  k-dimensional, 159.
`MATE` table, 174.
Mathematics versus computer science, 239.
Matoušek, Jiří, 196.
Matrices of 0s and 1s, 80, 84, 148–149, 239–240.
Matrix coordinates versus Cartesian coordinates, 162.
Matrix representation of binary relation, 80, 84, 148–149, 239–240.
Matula, David William, 38, 134, 135.
Max-closed constraints, 82–83, 89, 95, 99, 110, 147–149.

Maximum independent set in bipartite graph, xi.
MAXSAT: Satisfy as many clauses as possible, 5.
mc_revise(c, v), 83.
MCC problem: Multiple covering with colors, 38, 76–79, 114, 146, 147, 179, 234, 238.
  benchmarks for, v, 78–79.
MCC solvers, iv, 235–236.
McCreesh, Ciaran Arthur, v, 35, 112, 200, 201, 203.
McGill, John, Jr., 175.
McGinty, Abigail, 157.
McGregor, James Jardine, 109, 201.
M$^\text{c}$Ilree, Matthew John, 112.
McKenzie, Ralph Nelson Whitfield, 111, 251, 252, 255.
Mebane, Palmer Croasdale, 221.
Median ($\langle xyz\rangle$), v, 85, 95, 100, 138, 148, 152, 213, 241, 246, 255.
Median algebra, 252.
Median-closed constraints, 148.
Median labels, 193.
Megamems (millions of memory accesses), 73.
Meganodes (millions of nodes), 73.
Mélot, Hadrien, 179.
Mems (accesses to 64-bit words), 64, 163, 227.
  per node, 54, 74.
Mercer, John Herndon, ii, 258.
Meromorphic function, 166.
Mézard, Marc Jean Marcel, 5, 159.
Miguel, Ian James, 198.
Milton, John, 1.
Milutinović, Uroš, 209.
Min-and-max-closed constraints, 149.
Min-closed constraints, 147, 149, 240.
Min operation, 99.
Minimal non-Hamming graphs, 128, 193.
Minimal polynomial, 249.
Minimal stub, 100.
Minimum element of domain, 142.
Minimum remaining values heuristic, *see* MRV heuristic.
MINIZINC format, 111–112.
Miracle, 18, 122.
Mirror image, 7, 117, 160.
Mitchell, David Geoffrey, 218.
MIX, 251.
Mixed-radix representation, 177.
ML language, 86.
`MMIX` computer, ii, 58, 60, 133.
Mnemonic, 255.
MNH: Minimal non-Hamming, 128, 193.
Möbius, August Ferdinand, polynomial, 186.
Model theory, 2.
Modstep problem, 55–56, 74, 140.
Moiseev, Stanislav Vladimirovich (Моисеев, Станислав Владимирович), 252.
Molecules, 130.
Monoforkings, 154, 257.
Monominoes, 221.
Monotone functions, 83.
  Boolean, 87, 95, 240, 245, 246, 254.
  self-dual Boolean, 95, 254.
Montanari, Andrea, 5, 159.
Montanari, Ugo, 107, 155, 213.
Montgomery, Richard Harford, 28, 127.
Monus operation ($x \mathbin{\dot-} y = \max\{0, x-y\}$), 78.
Moon, John Wesley, 204.
Moshkovitz, Dana Aaronson (מושקוביץ דנה), 111.
Motif graph, 36.
Motley dissections, 79.
MRV heuristic (minimum remaining values), 54, 63, 65–67, 73, 78, 136, 139, 140, 143, 144, 204, 208, 227, 230, 234.
MRV† heuristic, 75.
Muchnik, Albert Abramovich (Мучник, Альберт Абрамович), 252.
Multiple covering with colors, *see* MCC problem.
Multiplicities, 235.
Multisets, 8, 130, 131.
Multivalued encoding, 40–42, 44, 207–212.
Multivalued logic, 110.
Muñiz, Alexandre Owen, 236.
Music, 74.
Mutually intersecting sets, 149.
Mycielski, Jan, graph M_4, 65, 135.

n-cube, 122, 124.
n queens problem, 48, 109, 138, 146, 157.
n superqueens problem, 157.
n-tone rows, 74.
(n, k)-universal sets, 257.
Name, choosing a technical, 86.
`NAME` array, 17, 26, 120, 125.
NAND relation, 149.
Narayana, Tadepalli Venkata (తాడేపల్లి వెంకట నారాయణ), numbers, 243.
Nārāyaṇa Paṇḍita, son of Nṛsiṁha (नारायण पण्डित, नृसिंहस्य पुत्रः), cows sequence, 161.
Natural join, 4.
Near α-labeling, *see* Ordered graceful labeling.
Near unanimity operations, 97–103, 152–154.
Near vertices, 133.
Nearest common ancestor, 193.
Negative table constraints, 135, 146.
  versus positive table constraints, 213.
Nešetřil, Jaroslav, 159.
Nested parentheses, 143, 156.
Networks, 108.
`NEXT` array, 26, 125.
`NEXTH` and `NEXTL` arrays, 17, 120, 187.

Nguyen, Alain, 9.
Niemelä, Ilkka Niilo Fredrik, 158.
Nightingale, Peter William, 198.
Niven, Todd Matthew, 251.
No-three-in-a-line, 79, 146.
No-touch puzzle, 160.
`NODE` array, 61–64.
Nodes, 54, 73–74.
Nogoods, 1, 46, 108, 112.
Nonattacking knights and queens, 146–147.
Nonattacking queens, *see* n queens problem.
Noncommuting variables, 169.
Nondeterministic automaton, 155.
Nonequality relation, 3, 40–41, 91, 150, 248.
Nonominoes, 74.
Nonsharp preference heuristic, 79.
Nordström, Karl Jakob, 112.
Not all equal, 150.
Not equality, *see* Nonequality relation.
Notational conventions, v.
  $X^{\nearrow}$ and $Y^{\swarrow}$ (polarities), 92.
  $G \subseteq H$ (G is subgraph of H), 30.
  $G \sqsubseteq H$ (G is induced subgraph of H), 30.
  $r \star s$ (star composition), 255.
  $t[x,y]$ (t-of-the-way from x to y), 254.
  $[x_1 \ldots x_k]$ (the value of $f(x_1,\ldots,x_k)$ in a polymorphism), 86–87.
  $\langle xyz\rangle$ (median), v.
  $\langle R\rangle_f$ (completion of R with respect to f), 99, 101.
  $o \parallel o'$ (options are compatible), 54.
  $x_1 \ldots x_k \parallel x'_1 \ldots x'_k$ (tuples are unlike), 113.
Nouguier, Bernard, 216.
NP-complete, 90–91, 96–98, 111, 119, 130, 159, 194, 253.
NP-hard, 38.
NP-intermediate, 97.
NU: Near unanimity, 97–103, 152–154.
Null matrix, 241.
Nullary (constant) operation, 2, 94, 113, 250.

O (all-zeros matrix), 155, 241.
O'Beirne, Thomas Hay, 160.
O'Donnell, Ryan William, 255.
`OACTIVE` pointer, 63, 231.
Octahedron, 9, 178.
Octal notation, 116, 191.
Octants, 116.
Octominoes, 142.
OEIS®: The On-Line Encyclopedia of Integer Sequences® (`oeis.org`), 177, 178, 181, 182, 184, 188, 191, 219, 237, 241, 243, 245, 252.
Ollis, Matthew Anthony, 175.
One-in-three satisfiability, 90.
One-to-one function, 261.
Online algorithms, 99, 102, 155.
Online resources, *see* Downloadable programs and data.
Operation: A function from D^k to D, 86, 94.
Operator: *see* Operation, 94.
Operator norm, 170.
Opršal, Jakub, 254.
$\mathrm{opt}(x,x')$, 227, 235.
$\mathrm{opt\_out}(o)$, 72, 144–145, 234.
Option list, 61, 62.
Options, 1, 2, 54, 77.
`OR` (bitwise OR), 201.
OR relation, 86–88, 149.
Orchard pattern, 237.
Order encoding, 2, 40–42, 44–46, 207.
Order-reversing correspondence, 93, 95.
Ordered graceful labeling, 123, 124.
Ordered graceful rainbow labeling, 191.
Ore, Øystein, 93.
Organ-pipe order, 65, 229.
  inverse, 170.
Orientations of edges, 27.
Oriented complete bipartite graph, 126.
Oriented cycle, vi, 26, 126, 150, 159.
Oriented forest, vi.
Oriented path, 26, 82, 126.
Origami, 10.
Orthographic projection, 11.
Orthohedral polyhedra, 173.
Östergård, Patric Ralf Johan, 190.
Otter, Richard Robert, 23.
Output states, 155.
Overflow, floating point, 229.
Overflow of stamps, 145.

$\wp$ (tautology), 211.
P $\neq$ NP (?), 97.
#P-complete, 259.
P_n (path of order n), 19–20, 120, 121, 124, 180.
$P_{\vec{n}}$ (oriented path of order n), 26, 126, 188.
(p,q)-leaper, 204.
(p/q)-string, 114.
Pack a given shape, 146.
Pahl, Felix Andreas (= `joriki`), 237.
Pairwise consistency, 216.
Pairwise ordering trick, 141, 160, 196.
Palindromes, 161, 260.
Palindromials, 125.
Papadimitriou, Christos Harilaos (Παπαδημητρίου, Χρίστος Χαριλάου), 119, 173, 196.
Parades, 242.
Parallomino graph, 123, 243.
Parameters, tuning, 43–44.
ParamILS, 43.
Parent vertex, vii, 128, 134.
Parity, 22, 104, 126, 183, 184, 213, 221, 230, 234, 258.
Partial assignment, 84, 260.

Partial ordering, 83, 148.
  of multisets, 197.
Partition function, 5, 113.
Partitioning into multisets, 130.
Partitions of integers, 197.
Partridge puzzle, 78, 79.
Pasting graphs together, 44.
Path consistency, 53, 84–85, 107, 139, 148, 241.
Path of order n (P_n), 19–20, 120, 121, 124, 180.
Patterns in permutations, 219.
Paw graph, 23, 180, 204.
Pearl, Judea (יהודה פרל), 215.
Pegg, Edward Taylor, Jr., 178.
Penrose, Lionel Sharples, 15.
Penrose, Roger, 15.
  n-gon, 119.
  triangle, 15–16.
Pentacubes, 65.
Pentominoes, 64–66, 78, 79, 124, 221.
Perez, Guillaume Didier, 110.
Perfect matching, 198.
Perfect Packing puzzle, 79.
Permutation matrix, 241.
Permutations, 17, 59, 115–116, 132, 153, 186, 211, 242.
  graceful, 19, 120, 121.
  inverse of, 140.
  inversions of, 164.
  of a multiset, 8.
  patterns in, 219.
  reversal of, 20, 219.
  slow growth, 56, 74–75, 140–141.
Perron, Laurent Frédéric, 110.
PERT: Program Evaluation and Review Technique, 83.
Petersen, Julius Peter Christian, graph, 129, 132, 135.
Petrie, Charles Joseph, Jr., 213.
Petrie, Karen Elizabeth Jefferson, 176.
Phi (ϕ), as source of "random" data, 116.
Philippsen, Michael, 196.
Photographs, 9.
Physics (statistical mechanics), 4–5.
Pi (π), as source of "random" data, 18, 59, 60, 74, 76, 90, 91, 100, 116, 121, 124, 141, 150, 164.
Pigeonhole principle, 198, 211.
Pigeonhole problem, 229.
Piles of polyominoes, 78, 236.
Pinched Sierpiński gasket graph, xiv, 42–44.
Pinched Sierpiński simplex graph, 137.
Pinsker, Michael, 254.
Pipeline of a computer, 227.
Pippenger, Nicholas John, 251.
Pixel, 107, 146.
Pixley, Alden French, 100, 251, 252.
Planar graphs, 11, 117, 172, 173, 177.
Plane in three dimensions, 167.
Plane partitions, 114.
Plummer, Michael David, 198.
Pointwise product matrix, 239.
Poirot, Hercule, 157.
Pokrovskiy, Alexey Alexeevich (Покровский, Алексей Алексеевич), 28, 127.
Pol(Γ), 95–97, 110.
Polarities, 92–95, 150–151.
Pollak, Henry Otto, 194.
Poly-Bernoulli numbers, 241.
Polyhedra, 114; *see also* 3VP.
Polymorphisms, 86–89, 110, 149–155, 241.
  exploitation of, 99–106.
Polynomial factorization, 124.
Polynomial time, 84, 85, 99, 103, 128, 149, 153, 196, 239, 258, 259.
  delay, 106, 155.
Polyominoes, 65, 141, 142, 146, 147, 243.
  piling, 236.
Population count (νx), sideways sum, v, 58, 161, 193, 247.
`POS` field, 61–64.
Pöschel, Reinhard, 110.
Positive table constraints, 135.
  versus negative table constraints, 213.
Post, Emil Leon, 95, 251, 256.
Potent constraints, 90–92, 95, 246.
Preclusion, 109.
Preclusion clauses, 46, 212.
Predecessors of a vertex, 187.
Predicate, *see* Relation, 2.
Prefix encoding, 40–41, 45, 135, 136, 207–211, 129.
Preorder of a tree, 129, 189.
Preprocessing, 55, 65, 74, 75, 211, 216.
Prestwich, Steven David, 41, 207.
Primary item, 2, 54, 62–64, 77–78, 145, 208–209, 253.
Prime factorization, 97, 249.
Prime queen attacking problem, 74–76, 146–147.
Prime strings, 156.
Primitive polynomial modulo p, 127.
Principal ideals, 249.
Product of relations ($R' \otimes R''$), 96, 152, 252.
Programming languages, 86.
Projection from 3D to 2D, *see* Line labeling problem.
Projection function (π_j), 88, 90, 95, 153, 213, 246–250, 258.
Projection of a relation ($\Pi_\sigma R$), 48, 96, 99–101, 152, 241.
Projective plane, 127.
Proof, computer-generated, 22.
Proof logging, 112, 157.
Propagation of constraints, 48–56, 108, 112.
Prosser, Patrick Thomas, v, 35, 200.

Prover–Delayer game, 218.
Puget, Jean-François, 176.
Punnim, Narong (ณรงค์ ปั้นนิ่ม), 191.
Pure literals, 212.
Pure vertices, 44–45, 137, 209.
Purging, 70–71.
Purified items, 231, 233.
Puzzles, 6, 17, 64–65, 74, 78–79, 116, 120, 141, 142, 147.
Pyramid, 9.

q-nomial coefficients $\left(\binom{n}{k}_q\right)$, 197.
Quasi-inverse function, 153.
Quasi-WNU: LWNU, 252.
Quasigroup, 154.
Quaternary (4-ary) constraint, 2, 162, 164.
Quaternary discriminator, 251.
Queen graph (Q_n), 64, 134, 137, 234.
Queens, attacking, *see* Prime queen attacking problem.
Queens, dominating, *see* 5-queens domination problem.
Queens, nonattacking, *see* n queens problem.
Queens and knights, nonattacking, 146–147.
Queens-and-knights problem, 67, 144, 237.
Queue (FIFO), 52–53, 71, 109, 145, 205–206.
  versus stack (LIFO), 144.
Quicksort, 224.
Quimper, Claude-Guy, 201.

$\langle R\rangle_f$ (completion of R with respect to f), 99, 101.
r-regular graphs, 121.
Radio colorings, 65, 135, 159.
Radio link frequency assignment problem, 56.
Radix list sort, 233.
Rainbow graceful labelings, 28, 108, 127–128.
Rainbow path problem, 39, 205.
Rampon, Jean-Xavier Fernand, 112.
Random circuit, 140.
Random CSP, 55, 140.
Random free tree, 135.
Random graceful graph, 121.
Random graph, 130, 188.
Random satisfiability, 5.
Randomization, 154, 179.
Randomized algorithms, 18, 122.
Range versus domain, 113.
Ranulac, Britt (‘*canular*’ pseudonym of Frédéric Marty), 186.
Rational numbers, 151.
Reachability, 3.
Real numbers, 125, 249.
Real-world data, 9.
Realizable HC networks, 15–16, 116, 117, 168, 173.
Recurrence, 140, 149, 165, 176, 177, 208, 219.
  binary, 122.
Recursive description, 240.
Recursive procedure, 50, 77, 205.
Reduced encoding, 40–42, 44, 135–136, 207–212.
Reduced matrix, 148.
Redundant constraints, 8.
Reentrant knight’s tour, *see* Closed knight’s tour.
Reflection of a solution, 7, 117, 156, 157.
Reflexive binary relation, 148.
Refutation length, 136.
Régin, Jean-Charles Pierre Henri, 109, 110, 198.
Relation: A property that holds for certain tuples of elements, 1–4, 86, 150.
  encoding as SAT clauses, iv, 39–42, 46–48, 135, 137, 207–212.
  encoding as XC, XCC, or MCC options, 38–39.
  table constraints, 135, 146, 213.
  tractable, 79–85.
Relational clones, 96, 152.
Relational databases, 4, 96, 216.
Relational structures, 2, 111.
Reluctant doubling, 182.
Removable value, 131.
Removed option, 71.
Representation of graphs and digraphs, 17, 26, 27.
Representation of sets, 56–60, 147.
Residual supports, 224.
Resolution, 137.
Restarts, 112, 182, 203.
Restricted growth strings, 128, 156, 251.
Restriction problem, 111.
Reutersvärd, Oscar Gustav Adolf, 14, 16, 119, 171.
Reversal of permutation, 20, 219.
Reverse model, 19.
Reversible memory, 57.
Reversible sparse bitsets, 60, 110, 142.
revise(c, v), 52–53, 83, 109.
Revolving-door Gray code for 4-vertex graphs, 134.
RGS: A restricted growth string, 128, 156, 251.
Richardson, Samuel, 262.
Rim representation, 162.
Riordan, John Francis, 219, 241.
Rivest, Ronald Linn, clauses R', 113.
RLFAP: Radio link frequency assignment problem, 56.
Rokicki, Tomas Gerhard, 18, 181, 182, 203, 204.
Roman numerals, 166.

Romov, Boris Aronovich (Ромов, Борис Аронович), 86, 110.
Rook graph, 134.
Rookwise connected, 141, 147, 220–221.
Root of a tree, 134–135.
Root of unity, 185.
Rooted graceful labeling, 121, 181, 182.
Rosa, Alexander, 24, 108, 183.
Rosenberg, Ivo Jiří (= George), 246.
Rossi, Francesca, 111, 213.
Rotating by 45°, 221.
Rounding error, 15.
Roussel, Olivier Michel Joseph, 111.
Row convex constraints, 84–85, 110, 148–149.
Row sums, 242.
Rows of musical tones, 74.
Ruler function ρx, v, 58, 183, 224.
Running time, 43–45, 54, 64, 65, 69, 78, 207–211.

S_c (scope of constraint c), 52.
(s,t)-caterpillar, 124.
Saarinen, Tapio, 220.
Sabin, Daniel, 55, 110, 239.
`SADD` (sideways addition), 58.
Safra, Shmuel (= Muli; שמואל ספרא), 111.
Saggese, Alessia, 200.
Saint-Marcq, Vianney Emmanuel Adrien Martin Marie Joseph, le Clément de, 110.
Saïs, Lakhdar (ⵙⴰⵢⵙ ⵍⴰⵅⴹⴰⵔ, لخضر سايس), 67, 68.
Sakinaga, Waku (崎永惑), 219.
Sansone, Carlo, 200.
Sapozhnikov, Artem Vasilievich (Сапожников, Артём Васильевич), 217.
SAT: The satisfiability problem, iii, 1–2, 137.
  represented as CSP, 1, 113.
SAT solvers, 1, 39–48, 137, 210.
Scene analysis, iv, 9.
Schaefer, Thomas Jerome, 97.
Schaus, Pierre Jean Philippe Daniel, 110.
Schiex, Thomas Daniel, 56.
Schulte, Christian, 112.
Scope of a constraint, 52–53, 239.
Search tree, 66–67, 144.
  size of, 136, 218.
Second moment principle, 196.
`SECOND` pointer, 62–64, 73, 143, 226–227.
Secondary item, 2, 54, 63, 77, 120, 160, 161, 186, 196, 205, 208–209, 213, 217, 219, 220, 225, 228–230, 234, 237, 238, 246.
Seed of a stub, 99, 104.
Selection relation, 96.
Self-converse digraphs, 25, 187–190.
Self-dual Boolean functions, 95, 150, 245, 254.
Self-dual ternary functions, 255.
Self-transpose latin square, 209.
Semilattice, 83–84, 148.
Senderov, Valery Anatol'evich (Сендеров, Валерий Анатольевич), 186.
Sentinel link, vii.
`SET` array, 61–64.
Set inclusion digraph, 24, 125–126.
Set partitions, 103, 251, 257.
Shadows in scenes, 107.
Sharp preference heuristic, 79, 143, 160, 217.
Sheppard, David Allen, 23, 184.
Shifting, 147.
Shortcut relation, 90, 150, 246, 253.
Shortest cycles, 108.
Shortest distance, 29, 128–129.
Shortest paths, 107.
Siblings, 62, 228, 236.
Sicherman, George Leprechaun, v.
Sideways addition, v, 58, 161, 193, 247.
Sidorov, Konstantin (Сидоров, Константин Сергеевич), 112.
Sierpiński, Wacław Franciszek, 41–42.
  gasket graph, 41–42, 135–136.
  gasket graph, pinched, xiv, 42–44, 135–136.
  graphs, proper, 136–137.
  simplex graphs, 44–45, 136–137.
  tetrahedron graph, 44–45, 136–137.
  tetrahedron graph, augmented, 45, 137.
  triangle graph, 41–42, 135–136.
Siggers, Mark Halvor, 255.
  term, 153, 253.
Signed skeleton of a polyhedron, 116.
Similar relational structures, 3.
Simonis, Helmut Matthias, 8.
simplex graphs, 131.
Simultaneous linear equations, 79.
Singer, James, 127.
Singleton domain consistency, 53, 139.
Singly linked lists, 145, 147.
Sink component, vi–viii.
Sink vertex, 82, 140, 147, 248, 252.
SIP: Subgraph isomorphism problem, 3, 30–38, 108–109, 128–135, 200.
  survey, 37.
`SIZE` field, 61–64, 77–78.
Skeleton of a polyhedron, 116, 123, 124.
`SLACK` field, 77–78.
Slim nonominoes, 74.
Slot in assembly line, 8, 160–161.
Slow growth permutations, 56, 74–75, 140–141.
SLUR algorithm, 216.
Small-and-slim nonominoes, 74.
Smile of order n, 118.

Smith, Barbara Mary, 176.
Smith, Henry John Stephen, normal form, 244.
Smits, Jeff, 112.
Snake path: An induced path, 29, 64.
Snevily, Hunter Saint Clair, 184.
Snyder, Thomas Marshall, 223.
Solid pentominoes, 65.
Solnon, Christine Roseline Michèle, v, 9, 33, 37, 61, 110, 134, 144, 195, 196, 203, 204.
Solution set, 102, 106, 149, 155, 156.
Sorting, 72, 131, 133, 233.
Source vertices, 140, 248, 253.
Spacer nodes, 61–62, 145, 230.
Spanning trees, 122, 221.
Sparse bitset, 60, 110, 142.
Sparse-set representation, iv, 59–64, 72–73, 110, 142, 181, 200, 227.
  deletion, 59, 227, 234.
Spectrum of a clone, 151, 152, 245.
Spenser, Edmund, 1.
Sphinx, 9–11, 14.
Spin, 4.
Spivak, Alexander Vasilievich (Спивак, Александр Васильевич), 186.
SPJ-algebra: select, project, join, 96.
Square dissection puzzle, 74.
Square torus, 164.
Squarefree polynomials, 249.
Squashed boxes, 119, 172, 194.
Squashing functions, 153, 254.
Stabilizer of a predicate, 110.
Stack, 57, 63, 71–73, 144, 214.
Stage stamp, 233.
Stages, 66–67, 144, 227, 228.
Staircase polygons, 243.
Stallman, Richard Matthew, 108.
Stamping, 52–53, 57–58, 72, 73, 109, 142, 144, 145, 233.
Standard line labeling, 13–14, 118–119.
Stanford GraphBase, ii, 29, 134, 191.
Stanford InfoLab, v.
Stanley, Richard Peter, 219.
Stappers, Filip Jan Jos, v, 43, 77, 146, 177, 186, 188, 189, 205, 208, 223, 244, 261.
Star composition, 255.
Star graph, 19.
Star of David pattern, 171.
State variable: A variable that helps to govern a process, 57.
States, 155–156.
Statistical mechanics, 4–5.
Stirling, James, partition numbers, 242, 251, 253.
Strachey, Christopher, 86.
Stravinsky, Igor Fëdorovich (Стравинский, Игорь Фёдорович), 1.
Strict embedding, 29–30, 128–135, 192, 200, 201.
Strict inequality, 173.
String polynomials, 169.
Strong components, vi–ix, 81–82, 131–132, 152, 198.
Strong prime queen attacking, 74–76, 146–147.
Strongly NP-complete, 194.
Strongly realizable, 15–16, 173.
Strongly rigid constraints, 246.
Structured constraint enforcement, 162.
Stubs, 100–105, 111, 154–155, 258–259.
Stuckey, Peter James, v, 111, 112.
Subcube labels, 129.
Subdag isomorphism, 38.
Subgraph isomorphism, *see* SIP, ISIP.
subsets graphs, 129.
Subtree isomorphism, 38, 134–135.
Successors of a vertex, 26, 125.
Sudakov, Benjamin (= Benny; בנימין סודקוב), 28, 127.
Sudoku, 17, 34, 65, 141.
Sugihara, Kokichi (杉原厚吉), v, 171, 173.
Superachievers, 253, 257.
Superexponential, 23, 122.
Superposition of functions, 94.
Superqueen, 157.
Supplemental labels, 35–36, 108, 132–133, 196.
  edge labels, 36, 132–133.
  graphs, 37, 132–133.
  pair labels, 36, 132–133.
Support encoding into SAT, 40–42, 44, 47, 49, 137, 138, 207–212, 215.
Support matrix, 70–72, 144.
Support vectors, 139–140.
Supported words, 192.
Supporting tuples, 51, 109, 140, 239.
Suresh Singh, Gnanadas (ஞானதாஸ் சுரேஷ் சிங்), 177.
Surprise, 29, 44, 45, 55, 56, 68, 74, 75, 209, 256.
  not a, 89, 90.
Sussenguth, Edward Henry, Jr., 108.
Sussman, Gerald Jay, 108.
Symmetric binary relation, 3, 249.
Symmetric Boolean ternary relations, 88–90, 150.
Symmetric digraphs, 27.
Symmetrical solutions only, 203.
Symmetry breaking (removal), 7, 17, 21, 23, 25, 54, 65, 74, 114, 120, 125, 137, 156, 160, 162, 176, 220, 223, 235, 236, 244, 246, 254, 255, 260, 261.
System of distinct representatives, 197.
Szeider, Stefan Hans, 205.
Szilassi, Lajos, polyhedron, 9–11, 14, 114, 116, 117.
Szymanek, Radosław Włodzimierz, 223.

T_0 (largest possible stage), 72, 227, 233.
T_0 (set of all solutions), 103, 106.
Table constraint, 46–48, 110, 135, 142.
Tableau shape, 162.
Tapestry, 79, 238.
Tardif, Claude Henri Joseph, 248.
Tarjan, Robert Endre, v, vi, viii.
  algorithm for strong components, vi–ix, 81–82.
Tautology, 158.
Taylor, Walter Fuller, term, 98, 153, 254.
Tee tetrominoes, 221.
Temperature, 5.
Tensor product of graphs, *see* Direct product of graphs.
Terms, 94.
Ternary (3-ary) constraint, 2, 53, 88, 138, 149, 253.
  irreducible, 137–138.
Ternary (3-ary) domain, 39, 49, 87, 90, 97, 138, 140, 150, 152, 154.
Ternary Boolean relations, *see* Boolean ternary relations.
Ternary Hamming graph, 129.
Ternary-to-binary encodings, 207.
Tetrahedron, 9–11, 166.
Tetrominoes, 64, 147, 221.
Thomas, Robin, 196.
Thought experiment, 99, 102.
Three-colorable, 38.
Three-faced, 9.
Three-valent polyhedral objects (3VP), 9–16, 115–119.
Tied-path graphs, 192.
Tilted squares, 238.
Time stamps, 52–53, 109.
Tolerance of weak digracefulness, 189.
Tone rows, 74.
Topological order, 81.
Toroidal patterns, 222.
Toroidal queens, 216.
Torto, 74.
Torus graph, 124.
Torus polyhedron, 164.
Totient function ($\varphi(n)$), 187, 188.
Tournament digraphs, 27, 126.
Trace of matrix, 118, 169–171.
Trace theory, 125.
Tractable families of CSPs, 79–85, 99–106, 110.
Trademarks, ii.
Trail, 57, 60, 142.
trail(x), 224.
Tramu, Jacques, 75.
Transcendental number, 249.
Transition rule, 155.
Transitive law, 103, 125, 130, 148.
Transitive tournament, 189.
Transpose of matrix, 65, 84, 171, 260.
Trees for algebraic formulas, 94.
Trees in subcube labels, 129.
Trees in subgraph isomorphism, 38, 130.
Treewidth, 79, 196.
Triangles, 42.
  colored, 65.
Triangular grids, 131, 135.
Tribonacci numbers, 161.
Trie structure, 104–105, 155, 257.
Trigger stacks, 71–73, 145.
Trihedral world, 9–16, 115–119.
Trimble, James Brian Rory, v, 35, 205.
Tripartite digraph, 131–132.
Trivalent (3-regular) graphs, 44, 121.
Trominoes, 221.
Truncated tetrahedron, 15.
Truth tables, 89, 212.
Tullis, Ian Dashiell Duckworth, 79, 147.
Tuple: An ordered sequence $(x_1, \ldots, x_k)$ of elements, often written simply $x_1 \ldots x_k$, 2, 13, 147, 223.
Turgeon, Jean Maurice, 179.
Twain, Mark (= Clemens, Samuel Langhorne), iii.
Tweaking, 77.
Twindragon fractal, 118.
Two-fan, 80, 240, 244.
Types of identifiers, 86.

Ullman, Jeffrey David, 4.
Ullmann, Julian Richard, 108–109, 201.
Unary (1-ary) constraint, 2, 53, 80, 91, 96, 98, 106, 147, 149, 225, 240, 254.
Unary encoding, 41, *see* Order encoding.
Unary operations, 94, 154.
Unary polymorphisms, 86, 106, 153.
Undeletion, 59.
Undoing, 72, 142.
Uniform hypergraph, 127.
Uniformly random solutions, 115.
Union-find algorithm, 177, 178.
Unique solution, problems that have a, 42, 116, 141, 160, 171, 186, 191, 197, 201, 204, 223, 237–239.
  graceful labeling, 19, 20, 22, 23, 121, 176–181, 189, 190.
Unit clauses, 207.
Unit propagations, 48, 137, 138.
Unit vector, 150, 259.
United States graphs, 16–18, 32–35, 131–132.
Universal algebra, 2, 94, 111, 153, 255.
Universal graphs, 134.
Universal relation, *see* Complete relation.
Universal set of tuples, 256.
'Unlike' relation, 113.
Unmatchable (blank) color, 237.
Unmatched girl in maximum matching, 205.
Up-up-or-down-down permutation, 115.
Updates when dancing, 64.

`USA` graph and subgraphs, 16–18, 32–35, 131, 132.

Valeriote, Matthew Anthony, 252.
Valid tuple, 142.
Validation code, 233.
Valiente Feruglio, Gabriel Alejandro, 134.
Valley folds, 10.
Valley in search tree, 224–225.
van Beek, Peter George, 110, 111, 168, 216.
van Dongen, Marc Remko Casper, 111, 211, 217.
Van Hentenryck, Pascal René Marc, 8, 110.
Van Kessel, Philippe Thibault, 201.
Vanden Eynden, Charles Lawrence, 183, 190, 191.
Vardi, Moshe Ya'akov (משה יעקב ורדי), 4, 111, 239, 254.
Variable ordering heuristics, dynamic, 65–69, 144.
Variables, 1, 8, 155.
Variance, 217.
Vector spaces, 258.
Vento, Mario, 200.
Verhaeghe, Hélène Marie-Jeanne Nelly, 224.
Vertex covers, 113.
Vertices, 3, 9.
Vesztergombi, Katalin, 242.
VF2 and VF3 solvers, 200.
Viable bindings, 50–52, 181.
Video, 205.
Vilarem, Marie-Catherine Maurer, 216.
Vision, computer, iv, 9–16, 107–108.
Visiting all solutions, 63, 72, 82, 106, 148, 155.
Volume of a tetrahedron, 166.
Vortex in a matrix, 115.

$\mathit{waerden}(j,k,n)$ problem, 43.
Wainwright, Robert Thomas, 79.
Walks, 161.
WalkSAT, 9, 41.
Walsh, Toby, 7, 111, 216.
Waltz, David Leigh, 107–109.
Warners, Johannes (= Joost) Pieter, 56.
Weak near unanimity operations, 97–98, 106, 151–153.
Weakened encoding, 40–42, 44, 136, 207–209.
Weakly digraceful labelings, 127.
Weakly realizable HC pictures, 15, 119.
Weakly viable bindings, 50, 55.
Webster, Daniel, 1.
Weigel, Peter Heinrich, 76, 146, 214, 229, 235, 237, 251.
Weight of primary item, *see* WTD heuristic.
Wermuth, Udo Wilhelm Emil, v.
Werth, Tobias, 196.
Whirlpool permutations, 115–116.
Widom, Jennifer, 4.
Wildcards, 129, 156, 224.
Willard, Ross David, v, 250, 252, 254, 255.
  pairing, 151.
Windmill dominoes, 64.
Windsor, Aaron Andrew, 183.
Winkler, Peter Mann, 128, 194.
Winston, Patrick Henry, 108, 169.
Wipe out, 51, 139, 214.
Witnesses, 101, 102.
WNU operations, 97–98, 106, 151–153.
Word rectangles, 74, 79.
Word search puzzles, 65.
Word squares, 56.
`WORDS`(n), the n most common five-letter words of English, 29, 74, 128, 138.
Wörlein, Marc, 196.
Wraparound edges, 22.
Wright, Edward Maitland, 188.
WTD heuristic, 67–68, 73, 144, 230, 234, 238.
WTD† heuristic, 75, 230.

X pentominoes, 221.
XC problem: Exact cover problem, 31, 39, 114, 130, 136–138, 141, 146, 190.
XCC problem: Exact covering with colors, iii, 1, 17, 38, 76, 117, 120, 125, 138, 140, 144, 146, 149, 152, 166, 182, 205, 247.
  as a CSP, 2, 54, 62.
  benchmarks, v, 64.
XCC representation of CSP, 2, 113, 135.
XCC solvers, iv, 63–64, 72–73, 227–228.
XCSP format, 111.

Y pentominoes, 65.
Yablonsky, Sergei Vsevolodovich (Яблонский, Сергей Всеволодович), 110.
Yanov, Yuri Ivanovich (Янов, Юрий Иванович), 252.
Yap, Roland Hock Chuan (葉福泉), 109.
Yin, Minghao (殷明浩), 68.

Zampelli, Stéphane, 196.
ZDD: A zero-suppressed decision diagram, 20, 120, 224.
Zemljič, Sara Sabrina, 209.
Zhang, Tianwei (张天玮), 205.
Zhang, Yuanlin (张元林), 109.
Zhu, Zichen (朱子晨), 112.
Zhuk, Dmitriy Nikolaevich (Жук, Дмитрий Николаевич), v, 97, 98, 111, 248, 252, 254.