On Sets and Graphs

Eugenio G. Omodeo • Alberto Policriti
Alexandru I. Tomescu

On Sets and Graphs

Perspectives on Logic and Combinatorics

Springer

Eugenio G. Omodeo
DMG/DMI (Department of Mathematics and Geosciences, Section of Studies in Mathematics and Information Technology)
University of Trieste
Trieste, Italy

Alberto Policriti
DMIF (Department of Mathematics Computer Science, and Physics)
University of Udine
Udine, Italy

Alexandru I. Tomescu
Department of Computer Science Helsinki Institute for Information Technology HIIT
University of Helsinki
Helsinki, Finland

ISBN 978-3-319-54980-4 ISBN 978-3-319-54981-1 (eBook)
DOI 10.1007/978-3-319-54981-1

Library of Congress Control Number: 2017935975

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Paola, Pietro, and Sara

To Giovanna

To Johanna and Aava

Foreword

Graphs are certainly the single most important data structure in computer science, mirroring somewhat the rôle sets play in mathematics. In fact, there is a very close and obvious relationship between graphs and sets, viz., each graph can be represented as a pair of sets, so very nearly we have something like a functor between graphs and sets. In programming, this symmetry is somewhat relaxed, since sets may as well be used for system construction, at least in the early design phases of software development (later sets vanish more and more into the processes and into the data structures actually implemented).

This book looks into the relationship between sets and graphs from different angles. One notes first that the set representation of graphs takes in fact for granted that the theory of sets is developed to such an extent that it may be usable for this representation. Looking into the development of set theory, however, one sees that there are many varieties of set theories around, so one probably wants to fix the ground before one can stand on it. In this monograph, this leads to the definition of sets that may be represented by graphs (in a curious reversal of rôles), demanding the question of characterizing those sets and those graphs for which such foundation issues arise. It gives rise to *set graphs*, roughly those graphs which underlie hereditary sets, which carry such an important burden in the foundation of set theory.

The authors examine the interplay of sets and graphs on a fairly fundamental level, formulating their set theory very carefully in such a way that some of the proofs become accessible to a proof assistant. They show that:

- one can find for every weakly extensional, acyclic digraph a Mostowski collapse by finite sets so that no two vertices are sent to the same set by the decoration,
- every graph admits an orientation which is weakly extensional and acyclic, so that one can regard its edges as membership arcs deprived of their natural orientation,

through the set-based proof-checker Referee (also known as the grown-up version of ÆtnaNova, it results from joint work of the two present Italian authors with Jacob T. Schwartz and D. Cantone). A brief introduction into Referee is given as well. Interestingly, Referee brings into the game the axiom of choice and not one of its—admittedly less well known—competitors like the axiom of determinacy. Technically, this happens through the `arb` operator familiar from languages like SETL.

One might ask whether this approach is a major route our discipline will be pursuing in the future: preparing the ground for machine-assisted proofs and then doing the proofs proper by a proof assistant. Well, probably. In some areas in logic and in complexity theory, proofs get so involved, so long, and so complicated that one scientist alone might no longer be able to conquer them. The analogy to programs comes to mind—programs tend to become so involved, so long, so complicated, and so many-faceted that a single programmer or a team of developers cannot really be its master any longer. But there is an important difference: programs are ultimately written for being executed on a computer, but a proof is intended to convey intellectual insight (and, of course, fun to its explorer), knowledge administering systems notwithstanding.

Since the universe of discourse is finite (but in the last chapter), combinatorial questions arise as well; the authors show that their approach renders counting certain graphs fairly straightforward and direct. They apply some Markov chain technique to the generation of random graphs, which at first sight looks like an unlikely match for the questions for which these chains are employed by the authors, but the technique soon turns out to an apt tool.

But one does not have to stop at finite objects, as the last chapter shows. Here the necessary set theory needs to be expanded, so that infinite sets can be handled. This excursion into infinity helps formulating and proving some properties which arise when well-foundedness of the basic relations is relaxed, and it gives surprising insights into Ramsey's theorem from finite combinatorics.

We should be glad that this book has been written. It helps us to understand the interplay of sets and graphs, in particular when wanting to look at graphs from an axiomatic set theory point of view. The book is also attractive because it formulates and solves some interesting combinatorial counting problems by directly manipulating the objects involved (rather than, say, setting up a generating function of some sort and manipulating it, which of course has also its merits). Finally, the use of the Referee system is not a mere add-on. It rather conveys some illuminating views into the inner working of some proofs requiring the power of set theory on both ends, viz., the tool, and the objects to be manipulated.

Professor Emeritus of Software Technology
and of Mathematics
Technische Universität Dortmund
Bochum, December 21, 2016

Ernst-Erich Doberkat

Preface

Will we say anything new on sets and graphs? Today's scholars are so closely acquainted with such entities—in theoretical computer science no less than in mathematics—that a book devoted to sets and graphs runs the risk of arousing, at best, an indulgent curiosity.

Why graphs? Without a good mastery of graphs, the design and analysis of computer algorithms would be unaffordable.

Why sets? Set Theory is commonly, though often implicitly, placed in the background of any other mathematical disciplines.

In particular, a graph, as customarily defined, is just a pair of finite sets: a set of vertices and a set of edges, of which the latter is included in the Cartesian square of the former. We see this reduction of graphs to sets neither as detracting from the autonomy of Graph Theory nor, in any deep or practical sense, revealing. One of the programmatic traits of this book is that graphs will endorse sets to the same extent to which sets will endorse graphs.

There is little controversy nowadays on the basilar role of sets and on the relevance of graphs for the development of algorithms. An obvious corollary then is: even algorithms are reducible, via Graph Theory, to Set Theory. As an indirect recognition of this, sets and associative maps (to wit, sets of ordered pairs) have become built-in types in many programming languages. But, in our daily experience, what abstractly speaking is just a set gets used in restrained ways; accordingly, in specific contexts, specialized concrete representations become preferable to others. As soon as concerns about representation arise, graphs pop up anew to the fore. This hints at why we will invest in cross-fertilization rather than giving priority to either graphs or sets in our investigation. Our undertaking, as we see it, is to exploit sets to shed light on certain issues that concern graphs while retaining our freedom to overturn perspective, whenever fit.

In the formal approach to Set Theory, a *set* is an object containing nothing but other sets as elements. This view brings uniformity into the foundations of mathematics and radically differs from the informal, naïve view according to which a set is a collection of elements whose nature we refrain from entering into. When we work in a universe where *everything is a set*, membership becomes the only relation we have to worry about and we find ourselves very close to Graph Theory.

This suggests a way in which graphs can represent sets: vertices will stand for sets and the edge relation will mimic the membership relation. This representation gives rise to interesting problems; in particular, it calls into play *hypersets*, whose notion somewhat generalizes the today prevailing conception of sets by permitting, e.g., membership to form cycles. By rooting sets into graphs, we are led to many combinatorial, structural, and computational questions: we will, in fact, study sets under the spotlights of combinatorial enumeration, canonical encodings by numbers, and random generation.

This book will privilege combinatorial questions over algorithmic issues in order to result in a more incisive manifesto. One of the exceptions will be our addressing the validity problem for set-theoretic sentences. With regard to solvable cases of that problem, the validity analysis is supported by graphs specially contrived to diagram collections of potential counterexamples to an alleged theorem. Such graphs also convey the causes of the combinatorial explosion that sometimes hinders logical analysis from going through.

Given a set, we can easily manage to build a graph that reflects the inner structure of that set. Doing this in the above-suggested manner will result in Directed edges. Can we conversely, but starting with a graph whose edges are undirected, find a set that conveniently represents it? Much of this book is devoted to an investigation on the graphs devoid of orientation that underlie sets, which we call *set graphs*. Through the study of their structure, we bring to light which graphs can be "implicitly" represented by sets. We elucidate the complexity status of the recognition problem for set graphs, characterize their class in terms of hereditary graph classes, and put forth polynomial algorithms for certain graph classes. Very little extant literature is concerned with the class of set graphs in its entirety, but many subclasses of it—suffice it to mention *claw-free* graphs here—have been studied since long. The set reading of graphs leads to simpler proofs of various classical results on claw-free graphs. We have taken advantage of the set-theoretic flavor of those proofs to formalize two of them with moderate effort in the set-based proof-checker Referee; the resulting new proofs are presented in this book in wealth of detail.

We will refrain, in general, from embarking on profound discussions about the infinite. Infinite sets do exist to us, mainly because we wish to adhere to the axioms of a standard set theory. Our hesitancy about the infinite is not grounded in philosophy; we simply decided to limit our focus to finite combinatorics.

In seeming conflict with the intention just stated, at a well-advanced phase of the exposition, we will not resist talking about a doubly stranded spiral formed by two infinite sets and describing it by means of slick set-theoretic formulae. An infinite graph will instantly show up that represents our spiral. Using it, we will shed light on a celebrated result, fundamental to *finite* combinatorics, namely, Ramsey's theorem.

A Word on the Audience for Whom This Book Is Intended

Much of the content of this book originates from papers recently published on scientific research journals, from which we selected topics which should be accessible with moderate effort also to nonspecialists, in particular to graduate students. The exercises put at the end of each chapter enable the reader to try her/his hand on the various topics so as to develop personally a deeper understanding of the subject matter.

This book assumes from the reader some familiarity with basic algorithm complexity and with standard programming techniques; anyhow, our algorithmic specifications will take the form of pseudo-code. To make our presentation as self-contained as possible on various topics we treat, we summarize some presupposed notions (e.g., NP-completeness) in panels which are spread all over the text.

The subject matter touches upon proof technology at some point; it is hence desirable that our reader has had some previous exposure to first-order predicate logic and formal methods. On the other hand, little knowledge of Set Theory and Graph Theory is assumed. For this reason, we try to present what is needed from those two areas of mathematics in a reasonably self-contained way, emphasizing concepts likely to be important in continuation of the work begun here, rather than technicalities. Foundational issues, for example, consideration of the strength or necessity of axioms or the precise relationship of our formal treatment to other weaker or stronger formalisms studied in the literature, are neglected.

Content of This Book

The introduction gives—mainly through examples—a rapid overview of the authors' way of combining the study of sets with the study of graphs.

Chapter 2 introduces two languages apt to support formal reasoning within Set Theory and Graph Theory, respectively. The primitive endowments of these languages rely on first-order predicate logic, and they are deliberately minimal; convenient syntactic extensions are then introduced conservatively. In the case of Set Theory, an inventory of sentences is produced from which the postulates of various specific axiom systems can be drawn: among those, the Zermelo-Fraenkel axioms; two axioms of opposite contents, namely, von Neumann's axiom of foundation and Aczel's anti-foundation axiom; and another incompatible pair of sentences, namely, Zermelo's infinity axiom (inessential for most of this book) and Tarski's axiom enforcing that only finite sets exist. In the case of graphs, the essential terminology is introduced, sometimes by resorting to the formal language in order to specify basic graph-theoretic notions such as acyclicity; we also hint at how significant graph theories often result from forbidding specific subgraphs.

Chapter 3 is also concerned with basics. It introduces two hierarchies of sets, one formed by the hereditarily finite sets, the other one—the celebrated von Neumann's *cumulative hierarchy*—also encompassing sets of infinite cardinality or rank. By virtue of the hierarchical construction of these set universes, membership is a well-founded relation over them, but this chapter also introduces two non-well-founded variants of the universe of hereditarily finite sets. The hierarchical hereditarily finite sets are intertwined with natural numbers, thanks to their anti-lexicographic order, first studied by Ackermann and also recalled in this chapter. Each hereditarily finite set is described in full by what we dub its *pointed membership graph*, whose acyclicity flags whether it is a hierarchical set (as opposed to a proper *hyperset*). A class of graphs named *membership graph* is also considered, which is broader than the class of pointed membership graphs; unlike in a *pointed* membership graph, there is no privileged vertex in a membership graph. Graphs closely akin to membership graphs, whose edges are also meant to mimic membership, though to a lesser degree of detail, are then exploited to solve some favorable cases of the *Entscheidungsproblem*, namely, to determine whether a set-theoretic formula subject to very stringent syntactical constraints is satisfiable or not.

The rest of the book is subdivided into two parts. Chapters 4 and 5 (along with their supplementary Appendix A) explain under what circumstances, and how, sets can conveniently model graphs; Chapters 6 through 8 investigate the converse issue: when is it convenient to represent sets by graphs? Notice that the use of graphs for handling the set-satisfaction problem mentioned at the end of the preceding paragraph pertains—at a relatively abstract level—to the latter circle of ideas.

Specifically, Chap. 4 undertakes the study of the class of *set graphs*, each of which results from a membership graph whose edge orientation has been forgotten. Set graphs hence have undirected edges; moreover, they are connected; their class includes graphs endowed with Hamiltonian paths and the so-called claw-free graphs. Given a graph with undirected edges, the problem of establishing whether or not it is a set graph is NP-complete; it amounts to finding an orientation of the edges such that the resulting directed graph neither forms cycles nor has distinct vertices endowed with the same children.

Chapter 5 presents recent proofs of two classical propositions concerning claw-free graphs. Since these proofs rely on the fact that claw-free graphs are set graphs, it is a straightforward task to develop them formally with the assistance of a proof-checker knowledgeable on sets, named Referee. Thus, by reporting on a concrete proof-checking experiment, this chapter offers a light introduction to proof technology. To convey the character of a proof script verifiable by means of our automated proof assistant, Appendix A shows excerpts of a formal proof of the fact that every connected graph has a vertex whose removal does not disrupt its connectedness.

Chapter 6 illustrates the usefulness of the set-to-graph correspondence by resorting to it for an explicit count of how many sets t of cardinality n enjoy this property: every element of t is also a subset of t. This result regards those hereditarily finite sets over which membership is well founded. The Ackermann encoding of their graphs is then revisited, to show that it can be obtained by means of a partition-refinement technique borrowed from algorithmics. A virtue of this technique is that it extends naturally to provide an ordering of non-well-founded hereditarily finite sets. This ordering can then be exploited for encoding non-well-founded hereditarily finite sets by dyadic rational numbers.

Chapter 7 studies how to generate a well-founded set of "size" n at random, so that each set of size n has equal probability to occur. Procedures of this kind can be of use for testing the correctness of algorithm implementations or for testing conjectures about data. Three general methods of generating combinatorial objects uniformly at random are described, two of which are based on the so-called combinatorial decomposition of the objects, while one is based on a Markov chain.

Chapter 8 broadens the scope of discussion to infinite sets and graphs. In its formulation dating back to Zermelo's original axiomatic system, the infinity axiom exhibits a single infinite set; as shown in this chapter, this axiom can be superseded by a sentence of greater syntactical simplicity, which brings into play two infinite sets twisted together. The simplest infinite graph underlying a pair of infinite sets that comply with this updated axiom can be used as a sort of abacus for speculating on finite combinatorics, in particular while addressing Ramsey's celebrated theorem.

Acknowledgments

We are grateful to Alberto Casagrande, Domenico Cantone, Giovanna D'Agostino, Agostino Dovier, Ernst-Erich Doberkat, Martin Golumbic, Alberto Marcone, Martin Milanič, Carla Piazza, and Romeo Rizzi for reading various parts of this book and to Salvatore Paxia, who enabled us to keep the Referee / ÆtnaNova proof-checking system alive. We also thank Alexandru-Alin Tudor for creating the three-dimensional images shown in Figs. 8.7 and 8.11.

This work was partially supported by the Academy of Finland under grants 250345 (CoECGR) and 274977, by the project *Specifica e verifica di algoritmi tramite strumenti basati sulla teoria degli insiemi* funded (2013) by INdAM/GNCS (Istituto Nazionale di Alta Matematica "F. Severi," Gruppo Nazionale per il Calcolo Scientifico) and by the project FRA-UniTS (2014) *Learning specifications and robustness in signal analysis*.

University of Trieste — Eugenio G. Omodeo
University of Udine — Alberto Policriti
University of Helsinki — Alexandru I. Tomescu
Helsinki/Trieste/Udine/Bucureşti/Berlin
November/December, 2016

Copyright Acknowledgments

The figures from Examples 4.3 and 4.4, Fig. 4.7 and some text excerpts from Chap. 4 have been reproduced with permission from Elsevier from [63]. Figures 4.3, 4.5, the figure from Exercise 4.5 and some text excerpts have been reproduced with permission from Elsevier from [64]. Some text excerpts have been reproduced with permission from Elsevier from [65].

The figure in Example 5.1, Figs. 5.2, 5.3, 5.4, 5.5 and some text excerpts from Chap. 5 have been reproduced with permission from Oxford University Press from [81]. The figures from Examples 5.3, 5.4, 5.5, Figs. 5.7, 5.8, 5.9, 5.11, 5.12, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, and some text excerpts from Chap. 5 have been reproduced with permission from Springer from [80].

Some text excerpts from Chap. 6 have been reproduced with permission from Elsevier from [94] and [99]. The figures in Examples 6.5, 6.6 and some text excerpts from Chap. 6 have been reproduced with permission from IOS Press from [25].

Figures 8.5 and 8.9 and some text excerpts from Chap. 8 have been reproduced with permission from Oxford University Press from [77]. Figure 8.19 and some text excerpts from Chap. 8 have been reproduced with permission from Cambridge University Press from [78].

Contents