

Preface

An Invitation to the Dance

It is an underappreciated fact that sets may have various types of complexity, and not all types are in harmony with each other. The primary goal of this book is to unify and make more widely accessible a vibrant stream of research—the theory of semi-feasible computation—that perfectly showcases the richness of, and contrasts between, the central types of complexity.

The semi-feasible sets, which are most commonly referred to as the P-selective sets, are those sets L for which there is a deterministic polynomial-time algorithm that, when given as input any two strings of which at least one belongs to L , will output one of them that is in L . The reason we say that the semi-feasible sets showcase the contrasts among types of complexity is that it is well-known that many semi-feasible sets have no recursive algorithms (thus their time complexity cannot be upper-bounded by standard time-complexity classes), yet all semi-feasible sets are simple in a wide range of other natural senses. In particular, the semi-feasible sets have small circuits, they are in the extended low hierarchy, and they cannot be NP-complete unless $P = NP$.

The semi-feasible sets are fascinating for many reasons. First, as mentioned above, they showcase the fact that mere deterministic time complexity is not the only potential type of complexity in the world of computation. Sets that are complex in terms of deterministic time—such as nonrecursive P-selective sets—may nonetheless be simple in many other computationally natural senses. A second reason that the semi-feasible sets are interesting is that they crisply capture the complexity of (standard left cuts of) real numbers, and a recent refinement of the semi-feasible sets has been shown to capture the complexity of complexity-bounded real numbers.

A third and more historical reason for interest in the semi-feasible sets is that they form the complexity-theoretic analog of a key class from recursive function theory; the semi-feasible sets are exactly what one gets if one alters the definition of the semi-recursive sets by changing the selector function from “recursive” to “polynomial-time computable.” In the late 1960s, the semi-recursive sets yielded great insights into distinguishing the power of reductions in the recursion-theoretic context. In 1979, Alan Selman launched a program that used—successfully, in the context of structural con-

nections to exponential time—semi-feasible sets to understand the structure of polynomial-time reductions.

A fourth and somewhat surprising reason to study semi-feasible sets is that the semi-feasible sets (in their nondeterministic version) conditionally resolve the important issue of whether NP machines can cull down to one the large number of potential solutions of satisfiable formulas. In particular, the study of the semi-feasible sets has established (see Section 2.4) that NP lacks such “unique solutions” unless the polynomial hierarchy collapses.

A fifth reason to study the semi-feasible sets is that the notion of semi-feasibility is both natural and attractive, and fits well into two related broad themes of computer science: making computers “smarter” even on problems that may be too complex to solve exactly, and allowing computers to make decisions even when they lack absolute “knowledge” of the goodness of the choices involved. For computers to be able to act and interact more intelligently with users, thus helping make computing more intuitive to those users, it would be nice for the computers themselves to show some “intuition” when making decisions, i.e., to act more boldly and intuitively—perhaps making membership claims that they might not “know” to be absolutely right or wrong, but that merely skate on intuition. Selectivity theory studies the sets for which a polynomial-time algorithm given two inputs—viewed as options, and potentially even as actions toward some goal—can intuit one to try, i.e., one for which to say “yes, if I had to take a flier and declare one of those options to be a good one, I’d go with *this* one.” Algorithms satisfying the rules of selectivity will have the property that if there is any good choice—one having whatever properties are possessed by the options in the set—offered to them, they will make a good choice. Curiously enough, due to the possibility of there being no good choice among the options being considered, or there being no bad choice among the options being considered, the algorithms will not necessarily “know” whether their choice is good or whether any option they pass up is bad. Nonetheless, we know that they are acting intelligently: If there was a good option among the inputs, a good option was chosen. Thus, by studying selectivity theory, we study the extent to which polynomial-time decision-making can be made “smart.” Speaking more broadly, one may say that selectivity theory formalizes a natural notion of intuition and intuitive computing.

We feel that this is the right time for such a book as this. Research into semi-feasible computation has already developed a rich set of tools, yet is young enough to have an abundance of fresh open issues. Though the primary goal of this book is to unify semi-feasibility research and make it accessible, another major goal is to lay out a path along which the reader can meet and engage the open problems in this research area. And wonderful open problems do remain. Though during the past fifteen years many long-standing issues were resolved, and the semi-feasible sets were shown to be deeply connected to issues of uniqueness, self-reducibility, and nondeterminism, these very ad-

vances themselves motivated new questions. The confluence of exciting open issues and a rich and expanding set of technical tools with which to study the semi-feasible sets make this perhaps the best of times to join the search for knowledge about semi-feasible computation. We hope this book will serve as both an invitation and a pathway.

Logistics

No previous knowledge of semi-feasible computation is required to read this book. We start with the definition of semi-feasibility and move on from there. However, though we include in the text or the appendix full definitions of each complexity-theoretic notion the book uses, we do assume that the reader has the basic comfort with computational complexity concepts—and the ability to grasp new definitions—that one would gain from a typical first course on computational complexity theory. (Among the textbooks, at various levels of difficulty, on computational complexity are those of Balcázar, Díaz, and Gabarró [BDG95,BDG90], Bovet and Crescenzi [BC93], Du and Ko [DK00], Hemaspaandra and Ogihara [HO02], Homer and Selman [HS01], Papadimitriou [Pap94], and Sipser [Sip97, Part Three]).

This text can be the focus of a second course on computational complexity theory. In particular, we feel that this material is very appropriate as a seminar course for first- or second-year graduate students who have already taken a first computational complexity course. We have found that both theory and non-theory students value and much enjoy the concreteness and “tour of the cutting edge” aspects of a course devoted to semi-feasible computation.

In virtually all of Chapters 1 through 6, the text contains no citations. The citations in these sections can be found in the Bibliographic Notes sections that end each chapter. The “we” used in this book (e.g., “we define,” “we prove”) refers to the reader and the authors as we together explore the theory of semi-feasible computation. Nonetheless, some of the research this book covers was done by the authors and their coauthors, and we sincerely thank those coauthors with whom we have explored semi-feasible computation: E. Allender, H. Buhrman, P. van Emde Boas, E. Hemaspaandra, H. Hempel, A. Hoene, Z. Jiang, A. Naik, C. Nasipak, A. Nickelsen, M. Ogihara, K. Parkins, J. Rothe, A. Selman, T. Thierauf, J. Wang, O. Watanabe, M. Zaki, and M. Zimand. Such research was generously funded by the following grants, whose support we gratefully acknowledge: HC&M-ERB4050PL93-0516, NSF-CCR-8957604, NSF-INT-9116781/JSPS-ENGR-207, NSF-CCR-9322513, NSF-INT-9513368/DAAD-315-PRO-fo-ab, NSF-INT-9815095/DAAD-315-PPP-gü-ab, and NWO-R-62-561.

We are extremely grateful to C. Homan, T. Tantau, and M. Thakur for proofreading the entire book, and to W. Gasarch, M. de Graaf, S. Homer, K. Regan, J. Rothe, D. Sivakumar, M. Stol, and J. Verbeek, each of whom did a detailed proofreading of one or more chapters of an earlier draft of this

book. This book benefited greatly from their suggestions and insights. We also thank the many other people who helped us with advice, discussions, suggestions, most-recent-version-of-paper information, or literature pointers, including E. Allender, H. Buhrman, J. Cai, L. Fortnow, E. Hemaspaandra, G. Magklis, A. Nickelsen, M. Ogihara, and F. Veltman. We are grateful to the Springer series editors—W. Brauer, G. Rozenberg, and A. Salomaa—and staff—A. Hofmann, F. Holzwarth, U. Stricker, T. Toomey, H. Wössner, and especially I. Mayer—for their advice and help.

Above all, we thank our families for their love and encouragement.

Rochester, New York, September 2002

Lane A. Hemaspaandra

Amsterdam, September 2002

Leen Torenvliet