Preface

This book contains a revised version of the dissertation the author wrote at the Department of Computer Science of the University of Chicago. The thesis was submitted to the Faculty of Physical Sciences in conformity with the requirements for the PhD degree in June 1999. It was honored with the 1999 ACM Doctoral Dissertation Award in May 2000.

Summary

Computational complexity is the study of the inherent difficulty of computational problems and the power of the tools we may use to solve them. It aims to describe how many resources we need to compute the solution as a function of the problem size. Typical resources include time on sequential and parallel architectures and memory space. As we want to abstract away from details of input representation and specifics of the computer model, we end up with classes of problems that we can solve within certain robust resource bounds such as polynomial time, parallel logarithmic time, and logarithmic space. Research in complexity theory boils down to determining the relationships between these classes – inclusions and separations.

In this dissertation, we focus on the role of randomness and look at various properties of hard problems in order to obtain separations. We also investigate the power of nondeterminism and alternation, as well as space versus time issues.

Randomness provides a resource that seems to help in various situations. We study its use in the area of proof checking. We show that every property that has a bounded-round interactive proof system has subexponential size classical proofs (for infinitely many input sizes) unless the polynomial-time hierarchy collapses. This provides the first strong evidence that graph nonisomorphism has subexponential size proofs. Under a stronger hypothesis we can scale the proof size down to polynomial size. We obtain our result by derandomizing Arthur-Merlin games. The same technique applies to various other randomized processes. We show how it works for the Valiant-Vazirani random hashing procedure which prunes the number of satisfying assignments of a propositional formula to one, the exact learning of Boolean circuits using equivalence queries and access to a satisfiability oracle, the construction of

matrices with high rigidity, and generating polynomial-size universal traversal sequences.

Completeness arguably constitutes the single most pervasive concept of computational complexity. A problem π is hard for a complexity class if we can efficiently reduce every problem in the class to π, i.e., we can efficiently solve any problem in the class when given access to an oracle providing solutions to instances of π. If π itself also belongs to the class, we call π complete for the class. Several complexity classes have complete problems under various reducibility notions. One often focuses on decision problems, or equivalently, on the corresponding language of yes instances. In this thesis, we develop techniques for separating complexity classes by isolating a structural difference between their complete languages. We look at various properties from this perspective:

Sparseness	–	the *density* of complete languages.
Autoreducibility	–	the *redundancy* of complete languages.
Resource-bounded measure	–	the *frequency* of complete languages.

Sparseness forms a candidate differentiating property that interests complexity theorists because of its connections to nonuniform complexity and to isomorphism questions. A language is sparse if it contains no more than polynomially many instances of each size. Showing that polynomial time has no sparse hard language under logarithmic space reductions would separate polynomial time from logarithmic space. We establish the logical completeness of this approach for reductions that can ask a bounded number of queries: If polynomial time differs from logarithmic space then there exists no sparse hard language for polynomial time under logarithmic space reductions with a bounded number of queries. The proof works for various other classes as well, e.g., for nondeterministic logarithmic space versus logarithmic space, and for logarithmic space versus parallel logarithmic time. Another instantiation states that no sparse hard language for nondeterministic polynomial time exists under polynomial-time randomized reductions with a bounded number of queries unless we can solve nondeterministic polynomial time in randomized polynomial time.

Autoreducibility defines the most general type of efficient reduction of a problem to itself. A problem is autoreducible if we can solve a given instance in polynomial time when allowed to ask an oracle the solution to any other instance. We establish that large complexity classes like doubly exponential space have complete languages that are not autoreducible, whereas the complete languages of smaller classes like exponential time all share the property of autoreducibility. The specific results we get yield alternate proofs of known separations. We also show that settling the question for doubly exponential time either way, would imply major new separations: It would either separate polynomial time from polynomial space, and nondeterministic logarithmic space from nondeterministic polynomial time, or else the polynomial-time hierarchy from exponential time.

Resource-bounded measure formalizes the notions of scarceness and abundance within complexity classes such as exponential time. From the separation point of view, the theory seems particularly suited for separating randomized polynomial time from exponential time. Within the formalism of resource-bounded measure, most languages turn out to be hard for randomized polynomial time under relatively powerful reductions. On the other hand, by establishing a small span theorem and using other approaches, we prove that exponential time and several subclasses only have a few complete or hard languages under weaker reducibilities. A very narrow gap between the power of the reductions remains, and bridging it would separate randomized polynomial time from exponential time.

Another approach for settling this problem, similar in spirit to the probabilistic method of combinatorics, tries to show that randomized polynomial time is a small subclass of exponential time. We prove the logical completeness of this strategy, i.e., if randomized polynomial time differs from exponential time then it is a small subclass of exponential time. As a byproduct, we obtain the first nontrivial example of a class for which the equivalent of Kolmogorov's 0-1 Law in resource-bounded measure holds.

One can view resource-bounded measure as a restriction of classical Lebesgue measure preserving properties like additivity and monotonicity. Whether invariance under permutations also carries over, remains open. If it does, then the class of autoreducible languages is small. We show that a resource-bounded version of permutation invariance holds if efficient pseudo-random generators of exponential security exist, and that if it holds, then randomized polynomial time differs from exponential time. We develop betting games as the basis for an alternate to resource-bounded measure for quantifying the frequency of properties within complexity classes, with permutation invariance built in.

Acknowledgments

To many people I owe a lot for their support during my graduate studies. It is my pleasure to thank them here.

First there are the people directly related to this dissertation. Above all is my advisor, Lance Fortnow. While still in Belgium, I had a hard time tracking down some of the references of the paper "MIP = NEXP" [17] which I had started reading. I finally emailed Lance, one of the authors of the paper. I gave him a list of three references which I couldn't get a hold of, and asked him whether he could send me copies. I wasn't expecting much of a reply. To my happy surprise, one week later, I found an envelope with the three papers in my mailbox! I contacted Lance again when I was considering a visit to the Department of Computer Science at the University of Chicago. Lance was very enthusiastic about it. He became my host during the visit and later my

advisor. I am thankful to Lance for bringing me to Chicago, for his inviting attitude towards research, and for his advice and support over the years.

I am also grateful to the other members of my committee, Laci Babai and Janos Simon. In particular, I would like to thank Laci for his inspiring lectures as well as for his support. I'll be happy if I'll be able to carry over even a fraction of Laci's inspiration to others. I thank Janos for his broad enlightening perspective on almost anything.

I am indebted to all members of the theory faculty, including Stuart Kurtz, Ketan Mulmuley, and Robert Soare, for the things I learned from them, and to the Department of Computer Science in general for the research and teaching opportunities I received. Thanks also go to Barbara Castro, Margaret Jaffey, Karin Lustre, Rory Millard, and Shirley Proby for the administrative help.

Much of the work reported in this thesis was a joint effort. For their collaboration and for their permission to include our results in my dissertation, I am grateful to my coauthors: Harry Buhrman, Lance Fortnow, Adam Klivans, Ken Regan, D. Sivakumar, Martin Strauss, and Leen Torenvliet. I would also like to thank Jack Lutz, Mitsu Ogihara, and Shiyu Zhou for the work we did together.

I would like to express my sincere gratitude towards Ashish Naik for his encouragement when I made my first steps in the world of computational complexity, as well as to Jin-Yi Cai, Mitsu Ogihara, and D. Sivakumar. I would like to thank Jin-Yi especially for having served as a reference on several occasions.

I have been very fortunate to get the chance to visit various institutes during my graduate studies. I am grateful to Eric Allender and the NSF for the summer of '96 which I spent at DIMACS. Eric was an excellent host and I have been very lucky to receive his advice and support ever since.

Harry Buhrman played a major role in my visit to CWI during the academic year '96–'97. I am thankful to him and to the European Union for the opportunities I got. Leen Torenvliet provided me with office space at the University of Amsterdam during that period.

I owe thanks to Allan Borodin, Steve Cook, Faith Fich, Charlie Rackoff, and to the Fields Institute in Toronto for inviting me to the Special Year on Computational Complexity from January till June '98. I am particularly grateful to Steve for the class he taught and for his interest and support.

I acknowledge Dirk Janssens and Jan Van den Bussche for the Belgian connection they offered.

On the more personal level, I would like to thank the following people for having been around: Amber S., Anna G., Barb M., Bernd B., Brian S., Dave K., Duke W., Dustin M., Fran G.-H., Gerry B., John R., John T., Juliana E., Kass S., Kati F., Kousha E., Louis S., Mark F., Micah A., Patrick P., Peter G., Peter K., Pradyut S., Robert S., Ronald d.W., Ruth M., Sandy K., Satya L., Shiyu Z., Silvia G., Sophie L., Steve M., Thomas W., Tom H., Tom M., and Vassilis A. Special thanks go to my office mate at the University of

Amsterdam, Bas Terwijn, to Valentine Kabanets in Toronto, and to my office and room mate in Chicago, Marcus Schäfer. Most of all, I would like to thank my room mate and best friend Behfar Bastani for all the discussions we had and the things we did together.

I also want to thank my long-time friends in Belgium: Bart, Jeroen, Patrick, and Peter. For their hospitality and interest, I am very grateful to Marijke, Bruno, Sigrid, and Shanti, my relatives on this side of the ocean. Similarly but on the other side of the ocean, I want to thank my grandma, sister, brother-in-law, and my nephews Jonas, Saul, and Tibo. Just seeing their smiles made every trip over the ocean more than worthwhile.

Finally, there are my parents. Words to describe my deep gratitude towards them done. Their continuous support – even when I made decisions which they would have preferred to see fail me differently – was invaluable.

Aan jullie, moeke en vake, draag ik dit werk met veel genoegen op!

Chicago, May 1999 *Dieter van Melkebeek*