LNCS 3210

Jerzy Marcinkowski
Andrzej Tarlecki (Eds.)

Computer Science Logic

18th International Workshop, CSL 2004
13th Annual Conference of the EACSL
Karpacz, Poland, September 2004, Proceedings

Springer

Jerzy Marcinkowski Andrzej Tarlecki (Eds.)

Computer Science Logic

18th International Workshop, CSL 2004
13th Annual Conference of the EACSL
Karpacz, Poland, September 20-24, 2004
Proceedings

Springer

Volume Editors

Jerzy Marcinkowski
Wrocław University
Institute of Computer Science
Przesmyckiego 20, 51-151 Wrocław, Poland
E-mail: jma@ii.uni.wroc.pl

Andrzej Tarlecki
Warsaw University
Institute of Informatics
Banacha 2, 02-097 Warsaw, Poland
E-mail: tarlecki@mimuw.edu.pl

Library of Congress Control Number: 2004111518

CR Subject Classification (1998): F.4.1, F.4, I.2.3-4, F.3

ISSN 0302-9743
ISBN 3-540-23024-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11315872 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3210

Lecture Notes in Computer Science

For information about Vols. 1–3112

please contact your bookseller or Springer

Vol. 3241: D. Kranzlmüller (Ed.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XIII, 452 pages. 2004.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004. (Subseries LNBI).

Vol. 3232: R. Heery, L. Lyon (Eds.), Research and Advanced Technology for Digital Libraries. XV, 528 pages. 2004.

Vol. 3224: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection. XII, 315 pages. 2004.

Vol. 3223: K. Slind, A. Bunker, G. Gopalakrishnan (Eds.), Theorem Proving in Higher Order Logic. VIII, 337 pages. 2004.

Vol. 3221: S. Albers, T. Radzik (Eds.), Algorithms – ESA 2004. XVIII, 836 pages. 2004.

Vol. 3220: J.C. Lester, R.M. Vicari, F. Paraguaçu (Eds.), Intelligent Tutoring Systems. XXI, 920 pages. 2004.

Vol. 3210: J. Marcinkowski, A. Tarlecki (Eds.), Computer Science Logic. XI, 520 pages. 2004.

Vol. 3208: H.J. Ohlbach, S. Schaffert (Eds.), Principles and Practice of Semantic Web Reasoning. VII, 165 pages. 2004.

Vol. 3207: L.T. Jang, M. Guo, G.R. Gao, N.K. Jha, Embedded and Ubiquitous Computing. XX, 1116 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004. (Subseries LNAI).

Vol. 3205: N. Davies, E. Mynatt, I. Siio (Eds.), UbiComp 2004: Ubiquitous Computing. XVI, 452 pages. 2004.

Vol. 3203: J. Becker, M. Platzner, S. Vernalde (Eds.), Field Programmable Logic and Application. XXX, 1198 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004. (Subseries LNAI).

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004. (Subseries LNAI).

Vol. 3199: H. Schepers (Ed.), Software and Compilers for Embedded Systems. X, 259 pages. 2004.

Vol. 3198: G.-J. de Vreede, L.A. Guerrero, G. Marín Raventós (Eds.), Groupware: Design, Implementation and Use. XI, 378 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004. (Subseries LNAI).

Vol. 3193: P. Samarati, P. Ryan, D. Gollmann, R. Molva (Eds.), Computer Security – ESORICS 2004. X, 457 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004. (Subseries LNAI).

Vol. 3190: Y. Luo (Ed.), Cooperative Design, Visualization, and Engineering. IX, 248 pages. 2004.

Vol. 3189: P.-C. Yew, J. Xue (Eds.), Advances in Computer Systems Architecture. XVII, 598 pages. 2004.

Vol. 3186: Z. Bellahsène, T. Milo, M. Rys, D. Suciu, R. Unland (Eds.), Database and XML Technologies. X, 235 pages. 2004.

Vol. 3185: M. Bernardo, F. Corradini (Eds.), Formal Methods for the Design of Real-Time Systems. VII, 295 pages. 2004.

Vol. 3184: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust and Privacy in Digital Business. XI, 299 pages. 2004.

Vol. 3183: R. Traunmüller (Ed.), Electronic Government. XIX, 583 pages. 2004.

Vol. 3182: K. Bauknecht, M. Bichler, B. Pröll (Eds.), E-Commerce and Web Technologies. XI, 370 pages. 2004.

Vol. 3181: Y. Kambayashi, M. Mohania, W. Wöß (Eds.), Data Warehousing and Knowledge Discovery. XIV, 412 pages. 2004.

Vol. 3180: F. Galindo, M. Takizawa, R. Traunmüller (Eds.), Database and Expert Systems Applications. XXI, 972 pages. 2004.

Vol. 3179: F.J. Perales, B.A. Draper (Eds.), Articulated Motion and Deformable Objects. XI, 270 pages. 2004.

Vol. 3178: W. Jonker, M. Petkovic (Eds.), Secure Data Management. VIII, 219 pages. 2004.

Vol. 3177: Z.R. Yang, H. Yin, R. Everson (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2004. XVIII, 852 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. VIII, 241 pages. 2004. (Subseries LNAI).

Vol. 3175: C.E. Rasmussen, H.H. Bülthoff, B. Schölkopf, M.A. Giese (Eds.), Pattern Recognition. XVIII, 581 pages. 2004.

Vol. 3174: F. Yin, J. Wang, C. Guo (Eds.), Advances in Neural Networks - ISNN 2004. XXXV, 1021 pages. 2004.

Vol. 3173: F. Yin, J. Wang, C. Guo (Eds.), Advances in Neural Networks – ISNN 2004. XXXV, 1041 pages. 2004.

Vol. 3172: M. Dorigo, M. Birattari, C. Blum, L. M.Gambardella, F. Mondada, T. Stützle (Eds.), Ant Colony, Optimization and Swarm Intelligence. XII, 434 pages. 2004.

Vol. 3170: P. Gardner, N. Yoshida (Eds.), CONCUR 2004 - Concurrency Theory. XIII, 529 pages. 2004.

Vol. 3166: M. Rauterberg (Ed.), Entertainment Computing – ICEC 2004. XXIII, 617 pages. 2004.

Vol. 3163: S. Marinai, A. Dengel (Eds.), Document Analysis Systems VI. XII, 564 pages. 2004.

Vol. 3162: R. Downey, M. Fellows, F. Dehne (Eds.), Parameterized and Exact Computation. X, 293 pages. 2004.

Vol. 3160: S. Brewster, M. Dunlop (Eds.), Mobile Human-Computer Interaction – MobileHCI 2004. XVIII, 541 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004. (Subseries LNAI).

Vol. 3158: I. Nikolaidis, M. Barbeau, E. Kranakis (Eds.), Ad-Hoc, Mobile, and Wireless Networks. IX, 344 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004. (Subseries LNAI).

Vol. 3156: M. Joye, J.-J. Quisquater (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2004. XIII, 455 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004. (Subseries LNAI).

Vol. 3154: R.L. Nord (Ed.), Software Product Lines. XIV, 334 pages. 2004.

Vol. 3153: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), Mathematical Foundations of Computer Science 2004. XIV, 902 pages. 2004.

Vol. 3152: M. Franklin (Ed.), Advances in Cryptology – CRYPTO 2004. XI, 579 pages. 2004.

Vol. 3150: G.-Z. Yang, T. Jiang (Eds.), Medical Imaging and Augmented Reality. XII, 378 pages. 2004.

Vol. 3149: M. Danelutto, M. Vanneschi, D. Laforenza (Eds.), Euro-Par 2004 Parallel Processing. XXXIV, 1081 pages. 2004.

Vol. 3148: R. Giacobazzi (Ed.), Static Analysis. XI, 393 pages. 2004.

Vol. 3146: P. Érdi, A. Esposito, M. Marinaro, S. Scarpetta (Eds.), Computational Neuroscience: Cortical Dynamics. XI, 161 pages. 2004.

Vol. 3144: M. Papatriantafilou, P. Hunel (Eds.), Principles of Distributed Systems. XI, 246 pages. 2004.

Vol. 3143: W. Liu, Y. Shi, Q. Li (Eds.), Advances in Web-Based Learning – ICWL 2004. XIV, 459 pages. 2004.

Vol. 3142: J. Diaz, J. Karhumäki, A. Lepistö, D. Sannella (Eds.), Automata, Languages and Programming. XIX, 1253 pages. 2004.

Vol. 3140: N. Koch, P. Fraternali, M. Wirsing (Eds.), Web Engineering. XXI, 623 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004. (Subseries LNAI).

Vol. 3138: A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, D.d. Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXII, 1168 pages. 2004.

Vol. 3137: P. De Bra, W. Nejdl (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. XIV, 442 pages. 2004.

Vol. 3136: F. Meziane, E. Métais (Eds.), Natural Language Processing and Information Systems. XII, 436 pages. 2004.

Vol. 3134: C. Zannier, H. Erdogmus, L. Lindstrom (Eds.), Extreme Programming and Agile Methods - XP/Agile Universe 2004. XIV, 233 pages. 2004.

Vol. 3133: A.D. Pimentel, S. Vassiliadis (Eds.), Computer Systems: Architectures, Modeling, and Simulation. XIII, 562 pages. 2004.

Vol. 3132: B. Demoen, V. Lifschitz (Eds.), Logic Programming. XII, 480 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004. (Subseries LNAI).

Vol. 3130: A. Syropoulos, K. Berry, Y. Haralambous, B. Hughes, S. Peter, J. Plaice (Eds.), TeX, XML, and Digital Typography. VIII, 265 pages. 2004.

Vol. 3129: Q. Li, G. Wang, L. Feng (Eds.), Advances in Web-Age Information Management. XVII, 753 pages. 2004.

Vol. 3128: D. Asonov (Ed.), Querying Databases Privately. IX, 115 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004. (Subseries LNAI).

Vol. 3126: P. Dini, P. Lorenz, J.N.d. Souza (Eds.), Service Assurance with Partial and Intermittent Resources. XI, 312 pages. 2004.

Vol. 3125: D. Kozen (Ed.), Mathematics of Program Construction. X, 401 pages. 2004.

Vol. 3124: J.N. de Souza, P. Dini, P. Lorenz (Eds.), Telecommunications and Networking - ICT 2004. XXVI, 1390 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004. (Subseries LNAI).

Vol. 3122: K. Jansen, S. Khanna, J.D.P. Rolim, D. Ron (Eds.), Approximation, Randomization, and Combinatorial Optimization. IX, 428 pages. 2004.

Vol. 3121: S. Nikoletseas, J.D.P. Rolim (Eds.), Algorithmic Aspects of Wireless Sensor Networks. X, 201 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004. (Subseries LNAI).

Vol. 3119: A. Asperti, G. Bancerek, A. Trybulec (Eds.), Mathematical Knowledge Management. X, 393 pages. 2004.

Vol. 3118: K. Miesenberger, J. Klaus, W. Zagler, D. Burger (Eds.), Computer Helping People with Special Needs. XXIII, 1191 pages. 2004.

Vol. 3116: C. Rattray, S. Maharaj, C. Shankland (Eds.), Algebraic Methodology and Software Technology. XI, 569 pages. 2004.

Vol. 3115: P. Enser, Y. Kompatsiaris, N.E. O'Connor, A.F. Smeaton, A.W.M. Smeulders (Eds.), Image and Video Retrieval. XVII, 679 pages. 2004.

Vol. 3114: R. Alur, D.A. Peled (Eds.), Computer Aided Verification. XII, 536 pages. 2004.

Vol. 3113: J. Karhumäki, H. Maurer, G. Paun, G. Rozenberg (Eds.), Theory Is Forever. X, 283 pages. 2004.

Preface

This volume contains papers selected for presentation at the 2004 Annual Conference of the European Association for Computer Science Logic, held on September 20–24, 2004 in Karpacz, Poland.

The CSL conference series started as the International Workshops on Computer Science Logic, and then, after five meetings, became the Annual Conference of the European Association for Computer Science Logic. This conference was the 18th meeting, and the 13th EACSL conference.

Altogether 99 abstracts were submitted, followed by 88 papers. Each of these papers was refereed by at least three reviewers. Then, after a two-week electronic discussion, the Programme Committee selected 33 papers for presentation at the conference. Apart from the contributed papers, the Committee invited lectures from Albert Atserias, Martin Hyland, Dale Miller, Ken McMillan and Paweł Urzyczyn.

We would like to thank all PC members and the subreferees for their excellent work.

The electronic PC meeting would not be possible without good software support. We decided to use the GNU CyberChair system, created by Richard van de Stadt, and we are happy with this decision. We also would like to thank Michał Moskal who installed and ran CyberChair for us. Finally, we would like to thank ToMasz Wierzbicki, who helped with the preparation of this volume.

We gratefully acknowledge financial support for the conference received from the Polish Committee for Scientific Research, and Wrocław University.

July 2004 Jerzy Marcinkowski and Andrzej Tarlecki

Organization

CSL 2004 was organized by the Institute of Computer Science, Wrocław University.

Programme Committee

Krzysztof Apt (*Amsterdam*)
Hubert Comon (*Cachan*)
Anuj Dawar (*Cambridge*)
Thomas Ehrhard (*Marseille*)
Javier Esparza (*Stuttgart*)
Kousha Etessami (*Edinburgh*)
Erich Graedel (*Aachen*)
Robert Harper (*CMU*)
Johann Makowsky (*Technion*)
Jan Małuszyński (*Linkoping*)
Jerzy Marcinkowski (*Wrocław*, Chair)
Eugenio Moggi (*Geneva*)
Till Mossakowski (*Bremen*)
Hans de Nivelle (*Saarbrücken*)
Luke Ong (*Oxford*)
Andrew Pitts (*Cambridge*)
Michael Rusinowitch (*Nancy*)
David Schmidt (*Kansas*)
Jean-Marc Talbot (*Lille*)
Andrzej Tarlecki (*Warsaw*, Co-chair)

Referees

P. Abdullah
J.-M. Andreoli
J. Avigad
A. Avron
V. Balat
M. Bednarczyk
M. Benedikt
N. Benton
M. Bezem
P. Blain Levy
A.-J. Bouquet
O. Bournez
A. Bove
J. Bradfield
P. Bruscoli
A. Bucciarelli
W. Charatonik
B. Coecke
S. Conchon
V. Cortier
S. Dantchev
P. de Groote
D. Deharbe
G. Delzanno
N. Dershowitz
R. Di Cosmo
C. Dixon
W. Drabent
A. Edalat
A. Filinski
M. Fiore
C. Fuhrman
D. Galmiche
R. Gennari
G. Ghelli
D. Ghica
O. Grumberg
S. Guerrini
H. Herbelin
D. Hutter
N. Immerman
R. Impagliazzo
D. Janin
S. Kahrs
Y. Kazakov
E. Kieroński
C. Kirchner
B. König
B. Konikowska
J. Krajíček
S. Kreutzer
J.-L. Krivine
A. Kučera
G. Lafferriere
J. Laird
D. Larchey-Wendling
C. Lüth
M. Luttenberger
P. Madhusudan
C. Marché
J.-Y. Marion
R. Mayr
C. McBride

D. Miller
O. Miś
M. Müller-Olm
J. Niehren
R. Nieuwenhuis
U. Nilsson
D. Niwiński
A. Nowack
D. Nowak
N.N. Vorobjov
J. Obdrzalek
M. Otto
L. Pacholski
P. Panangaden
G. Pappas
W. Pawłowski
N. Peltier
F. Pfenning
L. Regnier
D. Richerby
C. Ringeissen
E. Ritter
M. Roggenbach
G. Rosolini
H. Ruess
C. Russo
A. Sabry
H. Schlingloff
R. Schmidt
L. Schröder
A. Schubert
S. Schwoon
N. Segerlind
V. Simonet
K. Sohr
L. Straßburger
T. Streicher
J. Stuber
A. Szałas
H. Thielecke
S. Tison
L. Tortora de Falco
T. Touili
T. Truderung
J. Tyszkiewicz
C. Urban
P. Urzyczyn
T. Uustalu
L. Van Begin
J. van de Pol
J. van Oosten
V. van Oostrom
R. Vestergaard
M. Virgile
T. Walsh
J. Wells
B. Werner
T. Wierzbicki
S. Wöhrle
M. Zawadowski

Table of Contents

Invited Lectures

Regular Papers

Notions of Average-Case Complexity for Random 3-SAT

Albert Atserias*

Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract. By viewing random 3-SAT as a distributional problem, we go over some of the notions of average-case complexity that were considered in the literature. We note that for dense formulas the problem is polynomial-time on average in the sense of Levin. For sparse formulas the question remains widely open despite several recent attempts.

1 Introduction

The satisfiability problem for propositional logic is central to computational complexity. The work of Cook [2] showed that the problem is NP-complete, even when restricted to 3-CNF formulas, and is thus hard in the worst-case unless P = NP. Later on, the optimization versions of 3-SAT were also considered and showed hard. Namely, Hastad [4] culminated the monumental work of the 1990s on PCPs by showing that the number of clauses that can be satisfied simultaneously in a 3-CNF formula cannot be approximated within a ratio better than $7/8$ in polynomial-time, unless P = NP. The current decade is perhaps time for studying the *average-case complexity* of 3-SAT. Is it hard on average as well, unless P = NP, or is it easier?

The program comes also motivated from the fact that a fairly natural probability distribution on 3-CNF formulas has attracted the attention of many different communities, from AI to statistical physics, through combinatorics and mathematical logic. Our aim here is to review the background for a complexity-theoretic approach to the average-case complexity of random 3-SAT. In this short note we focus on the different definitions of average-case complexity that were introduced in the literature and their relationship. In the talk we will overview some of the partial results towards settling the main open questions.

2 Notions of Average Case Complexity

For every $n \geq 0$, let $f_n : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function. In order to simplify notation, we will use f instead of f_n, and we will write $f = f_n$ to emphasize the fact that f is actually a sequence of functions parameterized by n. It will be understood from context that f denotes the sequence of functions $\{f_n\}$ in some cases, and the particular function f_n in others. We adopt the framework of *ensemble of distributions* suggested by Impagliazzo [5], where a different probability distribution is considered

* Supported in part by CICYT TIC2001-1577-C03-02 and the Future and Emerging Technologies programme of the EU under contract number IST-99-14186 (ALCOM-FT).

J. Marcinkowski and A. Tarlecki (Eds.): CSL 2004, LNCS 3210, pp. 1–5, 2004.

for each input size. So let $\mu = \mu_n$ be a probability distribution on $\{0,1\}^n$. We should think of μ as a sequence of distributions, one for each $n \geq 0$. The pair (f, μ) is called a *distributional problem*. Informally, the problem reads as follows: given a random input $x \in \{0,1\}^n$ drawn according to distribution μ_n, compute $f_n(x)$.

Levin's Average Case. Let (f, μ) be a distributional problem. Consider an algorithm A computing f, and let $T = T_n : \{0,1\}^n \rightarrow \mathbb{N}$ be its running time on inputs of length n. What should it mean that the running time T of A be polynomial on average with respect to μ? The obvious immediate candidate definition would be this: there exists a $k \geq 1$ such that $\mathrm{E}_\mu[T] = O(n^k)$ where E_μ denotes expectation with respect to μ. Unfortunately, this simple definition suffers from a serious problem: the class of functions that are polynomial on average under this definition is not closed under polynomial images. Indeed, if we let μ be the uniform distribution on $\{0,1\}^n$, and let T be such that $T(x) = n$ for all but one string x_0 in $\{0,1\}^n$ for which $T(x_0) = 2^n$, then $\mathrm{E}_\mu[T] = O(n)$ while $\mathrm{E}_\mu[T^2] = \Omega(2^n)$. This lack of robustness would spoil any attempt to build a theory of polynomial reducibilities among distributional problems. A satisfactory remedy to this was discovered by Levin and reformulated by Impagliazzo for ensembles of distributions: we say that T *is polynomial on average with respect to* μ if there exists a $k \geq 1$ such that $\mathrm{E}_\mu[T^{1/k}] = O(n)$. It is now immediate from this definition that the class of functions that is polynomial on average is closed under polynomial functions. We say that a distributional problem (f, μ) has a *polynomial-time on average algorithm* if there exists an algorithm A for f whose running time is polynomial on average with respect to μ.

Impagliazzo's Benign Algorithms. Let $f = f_n : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function. A *prudent algorithm* for f is one that, for every input $x \in \{0,1\}^n$, outputs either $f(x)$ or ?. We should think of ? as a "don't know answer". Clearly, a prudent algorithm is useful only if it rarely outputs ?. We say that a distributional problem (f, μ) has a *polynomial-time benign algorithm* if there exists a prudent algorithm $A(x, \delta)$ for f that is polynomial-time in $|x|$ and $1/\delta$, and such that $\mathrm{Pr}_\mu[A(x, \delta) = ?] \leq \delta$ where Pr_μ denotes probability with respect to μ. The last clause of this definition formalizes the idea that the algorithm "rarely outputs ?".

Impagliazzo [5] showed that the two notions introduced so far coincide, from which we conclude that the concept is fairly robust. We reproduce the proof since it is informative.

Theorem 1 (Impagliazzo). *Let (f, μ) be a distributional problem. Then, the following are equivalent:*

1. *(f, μ) has a polynomial-time on average algorithm.*
2. *(f, μ) has a polynomial-time benign algorithm.*

Proof: We start by 1. implies 2.: Let $\mathrm{E}_\mu[T^{1/k}] \leq cn$. By Markov's inequality, we have $\mathrm{Pr}_\mu[T(x) > (tcn)^k] \leq 1/t$. Thus, for building a benign algorithm, it suffices to run the polynomial-time on average algorithm for $(cn/\delta)^k$ steps, and if it does not terminate, output ?. Next we show that 2. implies 1.: Suppose the benign algorithm runs in $(n/\delta)^k$ steps. Run the benign algorithm with parameter $\delta = 1/2,\ 1/4,\ 1/8,\ \ldots$ until an output different from ? is returned. Then, the expectation of the $2k$-th root of the running time of this algorithm is bounded by $(2n)^{1/2} + 1/2(4n)^{1/2} + 1/4(8n)^{1/2} + \cdots = O(n^{1/2})$ since at most $1/2$ of the inputs return ? in the first round, at most $1/4$ in the second round, and so on. □

Certification Algorithms. Let (f, μ) be a distributional problem. A *sound algorithm* for f is one that, for every input $x \in \{0,1\}^n$, if it outputs 1, then indeed $f(x) = 1$. Clearly, a sound algorithm A is useful only if it outputs 1 on a large fraction of the "yes" instances, in other words, only if $|\Pr_\mu[f(x) = 1] - \Pr_\mu[A(x) = 1]|$ is small. In such cases we say that it is *almost complete*. We say that a distributional problem (f, μ) has a *polynomial-time certification algorithm* if there exists a sound algorithm $A(x, \delta)$ for f that is polynomial in $|x|$ and $1/\delta$, and such that $|\Pr_\mu[f(x) = 1] - \Pr_\mu[A(x) = 1]| \leq \delta$. The last clause of this definition formalizes the idea of almost completeness. The relationship is now very easy to see:

Lemma 1. *Let (f, μ) be a distributional problem. Then, if (f, μ) has a polynomial-time benign algorithm, then (f, μ) has a polynomial-time certification algorithm.*

Proof: Let $A(x, \delta)$ be the benign algorithm. Consider the following algorithm $B(x, \delta)$: run the benign algorithm $A(x, \delta)$, and if it outputs ?, output 0. Clearly, $B(x, \delta)$ is sound. Moreover, by soundness, we have $|\Pr_\mu[f(x) = 1] - \Pr_\mu[B(x, \delta) = 1]| = \Pr_\mu[f(x) \neq B(x, \delta)]$ which in turn is bounded by $\Pr_\mu[A(x, \delta) = ?] \leq \delta$. □

If we put Theorem 1 and Lemma 1 together we see that if (f, μ) has a polynomial-time on average algorithm, then (f, μ) has a polynomial-time certification algorithm. In the contrapositive form, if (f, μ) is hard to certify, then (f, μ) is hard on average. Although we do not know whether the converse relationship holds in general, we note in the next section that it holds for the particular case of random 3-SAT.

3 Random 3-SAT

Let $x_1, \ldots, x_n$ be n propositional variables. A literal is a variable or its negation. A 3-clause is a tuple of three literals. A 3-CNF formula is a set of 3-clauses. Note that the number of 3-clauses is exactly $(2n)^3$. Thus, a 3-CNF formula can be encoded by a binary string of length $(2n)^3$ denoting which clauses are present and which are not.

There are several probability distributions that have been considered in the literature. The one we adopt here is inspired from the theory of random graphs. The distribution $\mu = \mu_n$ is parameterized by a real number $p = p_n$ in $(0, 1)$ and consists in choosing each clause with independent probability p. This probability model is sometimes referred to as the model A. The model B considers the number of clauses m as fixed and chooses the formula uniformly within that set. Both these models have several variants according to whether clauses are ordered tuples or sets, and may, or may not, have repeated and complementary literals. As in the random graph model, which model to use is often a matter of convenience, and rarely an important issue as far as the results are concerned.

We are interested in the distributional problem $(UNSAT, \mu)$, where $UNSAT = UNSAT_n$ is simply the unsatisfiability problem on 3-CNF formulas with n variables, and $\mu = \mu_n$ is the probability distribution that we just described. Notice that here n is not exactly the length of the input, but it is polynomially related. Notice also that μ is parameterized by $p = p_n$, and the complexity of the distributional problem may very well depend on p. As a matter of fact, when p is large, it can be seen that $(UNSAT, \mu)$ has a benign polynomial-time algorithm. Before proving that, we first show that for all values of p that guarantee unsatisfiability of a random formula with overwhelming probability, the three notions of average-case complexity considered so far coincide.

Theorem 2. *Let $p \geq (\ln 2 + \epsilon)/n^2$, with $\epsilon > 0$. Then, the following are equivalent:*

1. *$(UNSAT, \mu)$ has a polynomial-time on average algorithm.*
2. *$(UNSAT, \mu)$ has a polynomial-time benign algorithm.*
3. *$(UNSAT, \mu)$ has a polynomial-time certification algorithm.*

Proof: By Theorem 1 and Lemma 1, it suffices to show that 3. implies 1. Let $A(x, \delta)$ be the certification algorithm. Assume its running time is $(n/\delta)^k$. Consider the following algorithm. Run the certification algorithm with parameter $\delta = 1/2,\ 1/4,\ 1/8, \ldots$ until either "unsatisfiable" is returned, in which case we return "unsatisfiable" as well, or the parameter becomes smaller than 2^{-n}, in which case we run through all 2^n truth assignments to check whether F is satisfiable or not. By soundness of the certification algorithm, it is clear that this algorithm is correct. Let us estimate the expectation of the r-th root of its running time for a constant r to be determined later.

When $p \geq (\ln 2 + \epsilon)/n^2$, the probability that a random formula is satisfiable is $2^{-\gamma n}$ for some constant $\gamma > 0$, as it is easy to see. Let us consider the set of "satisfiable" instances. For those instances, the running time of the algorithm can only be bounded by

$$\sum_{i=1}^{n} (2^i n)^k + 2^{cn}$$

for some constant $c \geq 1$, which is time 2^{dkn} for some other constant $d \geq c$. Hence, the "satisfiable" instances contribute to the expectation of the r-th root of the running time by at most $2^{-\gamma n} 2^{dkn/r}$. Let us now turn to the contribution to the expectation of the "unsatisfiable" instances. The expectation of the r-th root of the running time for those instances is bounded by

$$(2n)^{k/r} + 2^{-1}(4n)^{k/r} + 2^{-2}(8n)^{k/r} + \cdots + 2^{-n+1}(2^n n)^{k/r} + 2^{-n}(2^{cn})^{1/r}$$

since at most $1/2$ of the "unsatisfiable" instances miss the first round, at most $1/4$ of those miss the second round, and so on, until at most $1/2^n$ of the instances miss all rounds in which case the algorithm goes on to cycle through all 2^n truth assignments in time 2^{cn}. It is now straightforward to see that if we take r large enough, say $r > dk/\gamma$, then the total expectation of the r-root of the running time is $O(n)$. □

In general, the proof technique of this result applies to any distributional problem for which the "no" instances represent a fraction that is inversely polynomial with respect to the worst-case running time that it is required to solve the problem. Let us conclude this paper with the promised benign algorithm when p is large. The reader will notice that the proof below resembles the arguments in [1].

Theorem 3. *Let $p = \omega(1/n)$. Then $(UNSAT, \mu)$ has a polynomial-time benign algorithm.*

Proof Sketch: Consider the following algorithm. Let F be the input 3-CNF formula and let δ be the error parameter. Let $\gamma > 0$ be a small constant to be determined later. If $\delta < 2^{-\gamma n}$, simply run through all 2^n truth assignments to check whether F is satisfiable or not. If $\delta \geq 2^{-\gamma n}$, find the most popular variable x in F. Consider the set of 2-clauses in $F|_{x=0}$ and $F|_{x=1}$, and run a polynomial-time 2-SAT algorithm on the resulting 2-CNF formulas. If both are unsatisfiable, report "unsatisfiable". Otherwise, output ?.

It should be clear from its definition that the algorithm is prudent. It is also clear that the running time of the algorithm is polynomial in n and $1/\delta$. Indeed, when $\delta < 2^{-\gamma n}$, the running time is $2^{O(n)}$ which is polynomial in $1/\delta$, and in the other case the running time is polynomial in n. Let us argue that the probability that it outputs ? is smaller than δ. When $\delta < 2^{-\gamma n}$, the algorithm never outputs ?. So let us assume that $\delta \geq 2^{-\gamma n}$. Each variable appears in $\Theta(n^2)$ clauses. Hence, the expected number of occurrences of each variable is $\Theta(n^2 p)$, which is $\omega(n)$ since $p = \omega(1/n)$. It follows from concentration bounds that the probability that a particular variable appears less than half this number of times is $e^{-\omega(n)}$. Thus, by Markov's inequality, the probability that some variable appears $\omega(n)$ times is at least $1 - ne^{-\omega(n)}$. The number of 2-clauses in $F|_{x=0}$ and $F|_{x=1}$ is thus $\omega(n)$ with at least that much probability. Moreover, the resulting 2-CNF formulas are random, so the probability that one of them is satisfiable is bounded by $2^{-\Omega(n)}$, as is well-known. All in all, the probability that the algorithm does not report "unsatisfiable" is bounded by $2^{-\Omega(n)}$. Thus, the probability that the algorithm outputs ? is bounded by δ since $\delta \geq 2^{-\gamma n}$. Here γ is chosen to be the hidden constant in the $2^{-\Omega(n)}$ bound. □

It follows from this result and Theorem 1 that when $p = \omega(1/n)$, the distributional problem $(UNSAT, \mu)$ is solvable in polynomial-time on average in the sense of Levin. For $p = O(1/n^{3/2-\epsilon})$, recent work has focused on certification algorithms [3]. For $p = O(1/n^2)$, the problem is widely open.

References

1. P. Beame, R. Karp, T. Pitassi, and M. Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal of Computing*, pages 1048–1075, 2002.
2. S. Cook. The complexity of theorem proving procedures. In *3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.
3. J. Friedman and A. Goerdt. Recognizing more unsatisfiable random 3-SAT instances efficiently. In *28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 310–321. Springer-Verlag, 2001.
4. J. Hastad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–859, 2001.
5. R. Impagliazzo. A personal view of average-case complexity. In *10th IEEE Structure in Complexity Theory*, pages 134–147, 1995.