Nondeterministic Instance Complexity and Hard-to-Prove Tautologies

V. Arvind[1], J. Köbler[2], M. Mundhenk[3], and J. Torán[4]

[1] Institute of Mathematical Sciences, C.I.T Campus, Madras 600113, India
[2] Humboldt-Universität zu Berlin, Institut für Informatik, D-10099 Berlin, Germany
[3] Universität Trier, Fachbereich IV – Informatik, D-54286 Trier, Germany
[4] Universität Ulm, Theoretische Informatik, D-89069 Ulm, Germany

Abstract. In this note we first formalize the notion of hard tautologies using a nondeterministic generalization of instance complexity. We then show, under reasonable complexity-theoretic assumptions, that there are infinitely many propositional tautologies that are hard to prove in any sound propositional proof system.

1 Introduction

In their seminal paper [4] Cook and Reckhow first formalized the study of lengths of proofs in propositional proof systems. They showed that NP = coNP if and only if there is a sound and complete propositional proof system S and polynomial p such that all tautologies F (i.e. all $F \in TAUT$) have proofs in S of size $p(|F|)$. The Cook-Reckhow paper led to an extensive and fruitful study of lengths of proofs for standard proof systems of increasing complexity and strength. A recent survey of the area by Beame and Pitassi is contained in [2]. A restatement of the Cook-Reckhow result is that if NP $\neq$ coNP then for every sound propositional proof system P and polynomial bound p there is an infinite collection of "hard" tautologies: i.e. the shortest proofs in P of each such tautology F is of length more than $p(|F|)$. A stronger notion of hard tautologies for a proof system is defined by Krajícek [8, Definition 14.2.1]. He defines a sequence of tautologies $\{\varphi_n\}$ to be hard for a proof system P if for every n the formula φ_n can be computed from 1^n in polynomial time, has size at least n, and there is no polynomial p so that $p(|\varphi_n|)$ bounds the size of the proof of φ_n in P for all n. Under a complexity assumption (slightly stronger than NP $\neq$ coNP) it is shown by Krajícek [8, Theorem 14.2.3] that for any non-optimal proof system P there is a sequence of hard tautologies for P. A more difficult question is whether there is a family of tautologies that is hard in the above sense for every sound proof system. Krajícek's construction does not yield this as the degree of the polynomial bounding the construction time of hard tautologies depends on the proof system and can become arbitrarily large. In a recent result Riis and Sitharam in

[12] show under the assumption NEXP $\neq$ coNEXP how to construct a sequence of tautologies $\{\phi_n\}_{n>0}$ that do not have polynomial-size proofs in any propositional proof system. The property of $\{\phi_n\}_{n>0}$ here is that the sequence is, in a formal logic sense, uniformly generated. Notice that this result is really about the hardness of the entire sequence and not about the individual tautologies ϕ_n. Indeed, the sequence $\{\phi_n\}_{n>0}$ might have an infinite subsequence that have polynomial-size proofs in some proof system.

Schöning [13] approaches the question of hard tautologies using the notion of complexity cores first defined in [10]. He shows in [13] that if NP $\neq$ coNP then there exist a constant ϵ and a collection $\mathcal{F}$ of tautologies of density at least $2^{\epsilon n}$ i.o. such that for *every* sound proof system for $TAUT$ and for every polynomial p, the shortest proof of F has length more than $p(|F|)$ for all but finitely many $F \in \mathcal{F}$. (The density part of this result is a consequence of the fact that $TAUT$ has a linear-time padding function.) Again, notice here that $\mathcal{F}$ could have individual tautologies that are easy for specific proof systems.

A natural question concerning hard tautologies that arises, which we address in this paper, is whether there exist individual tautologies which are "hard" for each sound proof system. Notice that none of the results mentioned above assert the existence of an infinite collection of tautologies in which every individual tautology is hard for every sound proof system. First, what do we mean by a tautology that is hard for all sound proof systems? We have to handle the following difficulty in the definition. For any $F \in TAUT$, we can always define a sound proof system S that includes F as an axiom. Clearly, F will have a trivial proof in the proof system S. The key point here is that F has to be somehow encoded into the proof system S, implying that the size of the description of S needs to be taken into account. We present a formal definition of hard tautology for a proof system based on the notion of instance complexity [7]. Intuitively, given a polynomial bound p for proof length, a tautology F is hard if for any given proof system P, either the shortest proof of F in P has size greater than $p(|F|)$ or else the size of P (considered as a string) is big enough to contain an encoding of F. We formalize the definition of hard tautology using the concepts of nondeterministic instance complexity and Kolmogorov complexity. We then prove, under reasonable complexity-theoretic assumptions, the existence of an infinite set of tautologies with the property that every single tautology in the set is hard (with respect to the discussed hardness notions) for every sound proof system.

2 Definitions

We fix the alphabet as $\Sigma = \{0, 1\}$. Let SAT denote the language of satisfiable propositional formulas suitably encoded as strings in Σ^*. Similarly, let $TAUT$ denote the language of propositional tautologies. For the definitions of standard complexity classes like P, NP, coNP, NEXP, the polynomial hierarchy PH etc. we refer the reader to a standard textbook, e.g. [1]. The class Θ_2^p (cf. [6]) – also described as $\mathrm{P}^{\mathrm{NP}}[O(\log n)]$ – consists of all sets decidable by a polynomial-time oracle machine that can make $O(\log n)$ queries to an NP oracle.

We start with a definition of proof systems that is equivalent to the original Cook-Reckhow formulation [4].

Definition 1. [2] *A (sound and complete) propositional proof system S is defined to be a polynomial-time predicate S such that for all F,*

$$F \in TAUT \Leftrightarrow \exists p \ : \ S(F,p).$$

In other words, a proof system can be identified with an efficient procedure for checking correctness of proofs. Thus, in complexity-theoretic terms we can also identify a propositional proof system S with a nondeterministic Turing machine M. The soundness and completeness properties of S can be expressed in terms of M as follows:

$$F \in L(M) \Rightarrow F \in TAUT \qquad \langle soundness \rangle$$

$$F \in TAUT \Rightarrow F \in L(M) \qquad \langle completeness \rangle$$

For the rest of the paper we use the characterization of propositional proof systems as nondeterministic Turing machines.

Nondeterministic instance complexity The notion of instance complexity was introduced by Orponen et al. in [7] as a measure of the complexity of individual instances. The motivation in that paper is to study whether the hardness of NP-complete problems, particularly SAT, can be pinpointed to individual hard instances in some formal sense.

We adapt the instance complexity idea to a nondeterministic setting that is suitable for defining hard individual tautologies. For a set $A \subseteq \Sigma^*$ we say that a nondeterministic machine M is A-consistent if $L(M) \subseteq A$.

Definition 2. *For a set A and a time bound t, the t-time-bounded* nondeterministic instance complexity *(nic for short) of x w.r.t. A is defined as:*

$$\begin{aligned} nic^t(x:A) \ = \ \min\{\ |M| \ \mid & \ M \textit{ is an } A\textit{-consistent } t\textit{-time-bounded} \\ & \ \textit{nondeterministic machine, and} \\ & \ M \textit{ decides correctly on } x \ \ \}. \end{aligned}$$

As required for instance complexity [7] and Kolmogorov complexity [9], we formalize *nic* in terms of a fixed universal machine U that executes nondeterministic programs. For a time bound function t, and two strings $q, x \in \Sigma^*$, we denote by $U^t(q,x)$ the result of running in the universal machine U program q on input x for $t(|x|)$ steps. Notice that the *nic* measure is (up to an additive constant) essentially the same w.r.t. efficient universal machines.

We now turn to the formal definition of a set A having hard instances w.r.t. the *nic* measure. Intuitively, we consider x to be a hard instance if there is no easier way for an A-consistent nondeterministic program to decide x than to explicitly encode x into the program. To define this formally, we consider a nondeterministic version of Kolmogorov complexity first defined in [3]. More precisely, given a time bound t and a string $x \in \Sigma^*$,

$$\begin{aligned} CND^t(x) \;=\; \min\{\, |M| \;\mid\; & M \text{ is a } t\text{-time-bounded nondeterministic} \\ & \text{Turing machine with } L(M) = \{x\} \,\} \end{aligned}$$

is the nondeterministic t-time-bounded decision Kolmogorov complexity of x w.r.t. M. In the standard way (see [9]) we can consider the *CND* measure to be defined w.r.t. a fixed universal machine. Notice that the *CND* measure is a nondeterministic generalization of Sipser's *CD* measure [14]. We note in passing that there is no difference between the nondeterministic Kolmogorov complexity of checking and generating [3] (unlike in the deterministic case where the C and *CD* time-bounded measures appear to be different [14]). The *CND* measure gives an immediate upper bound to nondeterministic instance complexity.

The next proposition will provide an upper bound on the size of a sound proof system that for a given tautology has proofs of a certain length.

Proposition 1. *For any language A, and time bound t, there is a constant $c > 0$ such that $nic^t(x : A) \leq CND^t(x) + c$, for all $x \in \Sigma^*$.*

We now define hard instances for a language A w.r.t. the *nic* measure.

Definition 3. *A language A is said to have hard instances w.r.t. the nic measure if for every polynomial t there are a polynomial t' and a constant c such that for infinitely many x we have, $nic^t(x : A) > CND^{t'}(x) - c$.*

We observe next that hard instances of A, if any, must necessarily be in A. This is in contrast to the deterministic instance complexity definition [7] for which hard instances could be in A or $\overline{A}$.

Proposition 2. *For any set A there is a constant c such that for all time bounds t and for all $x \notin A$, $nic^t(x : A) \leq c$.*

Proof. It follows from the fact that any nondeterministic machine that accepts $\emptyset$ is consistent with A and decides correctly on all instances $x \notin A$. $\square$

Observe that by Proposition 1, for every tautology F and time bound t, there is always a sound proof system P of size bounded by $CND^t(F) + c$ for some constant c, and in which F has proofs of size at most $t(|F|)$. Ideally we would like to obtain a matching lower bound for the *nic* measure, as stated in the following open question.

Problem 1. If NP $\neq$ coNP, then for every polynomial time bound t there are infinitely many tautologies F and a constant c such that $nic^t(F : TAUT) \geq CND^t(F) - c$ holds?

Because of the above observation, this result would be best possible. Although we cannot match the upper bound we prove somewhat weaker lower bounds using less restricted notions of Kolmogorov complexity instead of the *CND* measure.

In particular, we consider the standard unbounded Kolmogorov complexity measure C [9] and a relativized time-bounded Kolmogorov complexity measure C^B: $C^{B,t}(x)$ is defined to be the t time-bounded Kolmogorov complexity of string x, where the universal machine has oracle access to oracle B. We summarize below the main results of this paper proved in Section 3.

- If PH does not collapse to Θ_2^p then for every polynomial t and constant $c > 0$ there are infinitely many tautologies F such that $nic^t(F : TAUT) \geq c \log |F|$.
- NP $\neq$ coNP if and only if for every polynomial t there exist a constant $c > 0$ and infinitely many tautologies F such that $nic^t(F : TAUT) > C(F) - c$.
- NP $\neq$ coNP if and only if for every polynomial t there exist a constant $c > 0$, a polynomial t', and infinitely many tautologies F such that $nic^t(F : TAUT) > C^{SAT_2,t'}(F) - c$, where SAT_2 denotes all true quantified boolean formulas with $\exists\forall$ as quantifier prefix.

Each of the above results can be interpreted as stating that, under a reasonable complexity-theoretic assumption, there exist infinitely many tautologies F such that any sound proof system S in which F has short proofs must have large description (the description size is some function of $|F|$).

Notice that the latter two results are weaker than the ideal possible result mentioned above because for all F and all polynomials t we have: $C(F) \leq C^{SAT_2,t}(F) \leq CND^t(F)$. Also, observe that the first result proving an $O(\log |F|)$ lower bound on the nondeterministic instance complexity of infinitely many tautologies F is incomparable to the other results (and also to the statement in Problem 1) because it does not refer to the Kolmogorov complexity of F.

Hard instances and NP cores We review NP cores in TAUT defined by Schöning [13] and briefly compare it with nondeterministic instance complexity. An NP core in TAUT is an infinite subset $C \subseteq TAUT$ such that for all polynomials p and for all proof systems S, at most finitely many $F \in C$ have proofs in system S of length bounded by $p(|F|)$. In [13] it is shown that NP $\neq$ coNP if and only if TAUT has an NP core (in fact, of density $2^{\epsilon n}$, for a constant fraction $\epsilon > 0$).

Theorem 1. [13] *Assuming* NP $\neq$ coNP, *there exist a constant ϵ and a collection $\mathcal{F}$ of tautologies of density at least $2^{\epsilon n}$ i.o. such that for every sound proof system for $TAUT$ and for every polynomial p, the shortest proof of F has length more than $p(|F|)$ for all but finitely many $F \in \mathcal{F}$.*

The above theorem does not really talk about tautologies that are hard for each proof system. We can only make the following easy connection to the nondeterministic instance complexity of tautologies in an NP core of $TAUT$.

Proposition 3. *A set C is an* NP *core of a recursive set A (and hence also $TAUT$) if and only if for every polynomial t and constant c, $nic^t(x : A) > c$ for all but finitely many $x \in C$.*

3 The results

We prove two theorems which assert, in different ways, that there exist infinitely many hard tautologies assuming that the polynomial hierarchy does not collapse.

Theorem 2. *If* PH $\neq \Theta_2^p$ *then given any polynomial* t *and constant* c*, there exist infinitely many tautologies* F *such that* $nic^t(F : TAUT) > c \log |F|$.

Proof. Assume the contrary: Suppose there are a polynomial t and constant c such that for all $F \in TAUT : nic^t(F : TAUT) \leq c \log |F|$. In particular, for any n we have

$$\forall\, F \in TAUT^{\leq n} \;:\; nic^t(F : TAUT) \leq c \log n.$$

It suffices to show under this assumption that $\Sigma_2^p = \Theta_2^p$. Let M be an oracle-NP machine with TAUT as oracle. With no loss of generality, we can assume that on each computation path M makes precisely one positive oracle query to $TAUT$ and that too at the end of the path before it decides on acceptance. Let p denote a polynomial such that the running time of M on input x is bounded by $p(|x|)$, for all x. We will design a Θ_2^p machine N that accepts $L(M)$. Let x be a length m input to M. Let n denote $p(m)$ which bounds the size of $TAUT$ queries that $M(x)$ can make. Given the $O(\log n)$ bound on the nondeterministic instance complexity of tautologies in $TAUT^{\leq n}$, we only have to use the $TAUT$-consistent nondeterministic programs in the set $\Sigma^{\leq c \log n}$, whose size is bounded by some polynomial in n. But first we must get rid of the $TAUT$-inconsistent nondeterministic programs in $\Sigma^{\leq c \log n}$. We show that the inconsistent programs can be bundled out in an NP set, and by a census argument that can be carried out with a Θ_2^p computation we can exactly count the $TAUT$-inconsistent nondeterministic programs in $\Sigma^{\leq c \log n}$. After that a single NP computation can guess the $TAUT$-consistent subset of $\Sigma^{\leq c \log n}$ and use them to replace the $TAUT$ oracle.

$$BAD := \{\langle p, 0^n \rangle \mid p \in \Sigma^{\leq c \log n} \;:\; \exists\, F \in \Sigma^n - TAUT^{=n} \;:\; U^t(p, F) \text{ accepts } \}$$

Notice that BAD is in NP. In fact, BAD is a *sparse* NP set: $\|\{p \in \Sigma^{\leq c \log n} \mid \langle p, 0^n \rangle \in BAD\}\| = O(n^c)$. Let BAD_n denote the set $\{p \in \Sigma^{\leq c \log n} \mid \langle p, 0^n \rangle \in BAD\}$. We can use a standard census argument like the one in [6] to determine $\|BAD_n\|$ with a Θ_2^p computation. The Θ_2^p machine N will first compute $\|BAD_n\|$. Now it is easy to simulate $M^{TAUT}(x)$ with an NP computation: First guess the $\|BAD_n\|$ strings in BAD_n and verify with an NP computation that $\langle p, 0^n \rangle \in BAD$ for each guessed p. Having guessed and verified BAD_n, the remaining strings in $\Sigma^{\leq c \log n}$ are all $TAUT$-consistent programs. The NP computation now proceeds to simulate M^{TAUT}. When the simulation encounters a positive $TAUT$ query F, it accepts if and only if

$$\exists\, p \in (\Sigma^{\leq c \log n} - BAD_n) \;:\; U^t(p, F) \text{ accepts.}$$

To summarize, the Θ_2^p machine N on input x first computes $\|BAD_n\|$ with a Θ_2^p computation. What remains to simulate $M^{TAUT}(x)$ is just an NP computation. Hence PH collapses to Θ_2^p. $\square$

We now turn to the next result which is closer in spirit to the definition of hard-to-prove tautologies.

Theorem 3. NP $\neq$ coNP *if and only if for every polynomial t there exist a constant $c > 0$ and infinitely many tautologies F such that $nic^t(F : TAUT) > C(F) - c$.*

Proof. We first prove the forward implication. Our proof is an easy modification of the Fortnow-Kummer proof technique in [5] for the instance complexity conjecture [7] for NP-hard sets under honest reductions (this technique is also used by Mundhenk [11] to prove a weaker form of the general instance complexity conjecture of [7]). The idea is to build a deterministic Turing machine M that on input 0^n searches for and outputs a tautology F such that $nic^t(F : TAUT) > cn$ for a constant c. The fact that M outputs F on input 0^n implies that $C(F) \leq \log n$, proving the theorem. The proof of correctness argues that if M never halts on some input 0^n then, in fact, $TAUT \in$ NP contradicting the assumption. We give the formal details.

Fix $n \in \mathcal{N}$. Let $I_n := \{q \in \Sigma^* \mid |q| \leq 2n\}$ be the candidate nondeterministic programs. We describe the machine M:

1. **input** 0^n; $m := 0$;
2. $I_n := \{q \in \Sigma^* \mid |q| \leq 2n\}$; $m := 0$;
3. *Stage* m:
   (a) Spend m steps in computing an initial segment σ of $TAUT$ using brute-force search;
   (b) Spend m steps in simulating $U^t(q, x)$ for $q \in I_n$ and where $x \in \sigma$ is picked
   in lexicographic order;
   (c) If q is found incompatible with σ then eliminate q from I_n;
   (d) Do a prefix search for $F \in \Sigma^m$ with the following property

$$F \in TAUT^{=m} \ : \ \text{all programs in } I_n \text{ reject } F$$

   (e) If the prefix search finds such an F then **output** F and stop. Else goto *Stage* $m + 1$.

Claim. $TAUT \in$ NP if $M(0^n)$ does not terminate for some n.

To see this, suppose $M(0^n)$ does not terminate. Steps (a), (b), and (c) ensure that after some stage $m = m_0$, all $TAUT$-inconsistent programs in I_n are eliminated. Let I denote the remaining TAUT-consistent programs. Now, if the prefix search in Step (d) fails for each $m > m_0$ it follows that for all $m > m_0$

$$F \in TAUT \Leftrightarrow U^t(q, F) \text{ accepts for some } q \in I$$

which implies that $TAUT \in$ NP.

By assumption, therefore, $M(0^n)$ halts for each n. Let F_n be the formula output by the computation of $M(0^n)$. The prefix search ensures that $nic^t(F_n :$

$TAUT) > 2n$. The fact that $M(0^n)$ has output F_n implies that $C(F_n) \leq \log n$. Thus $nic^t(F_n : TAUT) > C(F_n) - c$, for a suitable constant c holds for all n, which completes the proof.

To see the reverse implication, notice that if NP = coNP then for some polynomial time bound t and some constant c we have $nic^t(F : TAUT) \leq c$. Now, since $C(F) > c$ for almost all F, it follows that $nic^t(F : TAUT) \leq C(F)$ for almost all F. □

Notice in the above proof that instead of $C(F)$ we can use polynomial-time bounded Kolmogorov complexity with a Σ_2^p oracle. This is due to the fact that with a Σ_2^p oracle the machine M can do the prefix search for F in polynomial time. Thus, we also have the following (stronger) result. Let SAT_2 denote the Σ_2^p-complete language consisting of all $\exists\forall$ quantified boolean formulas that are true.

Theorem 4. NP $\neq$ coNP *if and only if for every polynomial t there exist a constant $c > 0$, a polynomial t', and infinitely many tautologies F such that $nic^t(F : TAUT) > C^{SAT_2,t'}(F) - c$.*

4 Discussion

The interesting open question is to improve Theorems 3 and 4 by proving that if NP $\neq$ coNP then for every polynomial t there are infinitely many tautologies F, a polynomial t' and a constant c such that $nic^t(F : TAUT) \geq CND^{t'}(F) - c$ holds. The difficulty in adapting the Fortnow-Kummer proof technique to prove this is in carrying out the prefix search in Step 3(d) of the algorithm in the proof of Theorem 3. A sufficiently powerful oracle is required to carry out the prefix search for $F \in TAUT^{=n}$ in polynomial time. More accurately, the prefix search requires a leading existential quantifier followed by a universal quantifier (both for ensuring that $F \in TAUT^{=n}$ and checking that all nondeterministic programs in I_n reject F). Thus, we require a Σ_2^p oracle for the task; it is not clear if an NP oracle suffices.

This brings us to a related issue concerning the deterministic instance complexity result of [5]: In that paper a similar prefix search as mentioned above is carried out. Here the prefix search is for a hard instance $F \in \Sigma^n$ (not necessarily in SAT because, unlike for the *nic* measure, hard *ic* instances can be in SAT or $\overline{SAT}$). At first sight it appears that an NP oracle is required for the prefix search, but Fortnow and Kummer [5] apply the clever trick of using the programs in the set I to simulate this NP oracle and interpreting the program answers in some suitable way. However, the result of the prefix search could be a formula in SAT or $\overline{SAT}$, and there is no way of guaranteeing it to be in SAT, for instance. Thus, the proof in [5] cannot guarantee that there are infinitely many hard instances in SAT (although it certainly follows from their proof that either SAT or $\overline{SAT}$ has infinitely many hard instances). Thus, the following question concerning deterministic instance complexity of SAT remains open.

Problem 2. If $\mathrm{P} \neq \mathrm{NP}$, then for each polynomial t there are a polynomial t', a constant c, and infinitely many $F \in SAT$, such that $ic^t(F : SAT) > C^{t'}(F) - c$?

However, guided by an NP oracle, a prefix search for a hard instance $F \in SAT^{=n}$ can be carried out in polynomial time (the additional Fortnow-Kummer trick [5] of replacing the NP oracle with the set of remaining programs does not appear possible if we need to search for $F \in SAT$). This gives us the following result that can be proved on the same lines as Theorems 3 and 4.

Theorem 5. $\mathrm{P} \neq \mathrm{NP}$ *if and only if for each polynomial t there are polynomial t', a constant c, and infinitely many $F \in SAT$, such that $ic^t(F : SAT) > C^{SAT,t'}(F) - c$.*

The above result is very similar to [11, Theorem 5] which proves a weaker form of the instance complexity conjecture of [7].

In order to put the above result in better perspective we define a one-sided version of deterministic instance complexity. For $A \subseteq \Sigma^*$ a deterministic machine M is *one-sided A-consistent* if $L(M) \subseteq A$.

Definition 4. *For $A \subseteq \Sigma^*$ and a time bound t, the t-time-bounded* one-sided deterministic *instance complexity (ic_+ for short) of x w.r.t. A is defined as:*

$$ic^t_+(x : A) = \min\{\, |M| \;\mid M \textit{ is a one-sided } A\textit{-consistent } t\textit{-time-bounded deterministic Turing machine and } M \textit{ decides correctly on } x \,\}.$$

As usual, ic_+ is defined w.r.t. a fixed universal machine. The one-sidedness of the definition implies, as in the case of *nic*, that hard instances w.r.t. the ic_+ measure can only be in A. It follows from the fact that any deterministic machine that accepts $\emptyset$ is consistent with A and decides correctly on all instances $x \notin A$.

Proposition 4. *For any set A there is a constant c such that for all time bounds t and and for all $x \notin A$, $ic^t_+(x : A) \leq c$.*

We say A has hard instances w.r.t. the ic_+ measure if for each polynomial t there are a constant c and polynomial t' such that for infinitely many x, $ic^t_+(x : A) > C^{t'}(x) - c$. The following result can also be proved exactly as Theorem 3.

Theorem 6. $\mathrm{P} \neq \mathrm{NP}$ *if and only if for each polynomial t there are a polynomial t', constant c, and infinitely many $F \in SAT$, such that $ic^t_+(F : SAT) > C^{SAT,t'}(F) - c$.*

Acknowledgments. The authors thank Uwe Schöning for interesting discussions. The first author is grateful to Uwe Schöning for hosting his visit to Ulm university during which he was partially supported by an Alexander von Humboldt fellowship.

References

1. J. L. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science. Springer-Verlag, second edition, 1995.
2. P. Beame and T. Pitassi, *Propositional proof complexity: past, present, and future*, ECCC Report TR98-067, 1998.
3. H. Buhrman and L. Fortnow, *Resource-bounded Kolmogorov complexity revisited*, In Proceedings of the 14th Symposium on Theoretical Aspects of Computer Science, LNCS, 1200: 105–116, Springer, 1997.
4. S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic, 44(1):36–50, 1979.
5. L. Fortnow and M. Kummer, *On resource-bounded instance complexity*, Theoretical Computer Science A, 161:123–140, 1996.
6. J. Kadin, $\mathrm{P}^{\mathrm{NP}[\log n]}$ *and sparse Turing-complete sets for* NP, Journal of Computer and System Sciences, 39(3):282–298, 1989.
7. P. Orponen, K. Ko, U. Schöning, and O. Watanabe, *Instance complexity*, Journal of the ACM, 41:96–121, 1994.
8. J. Krajícek, *Bounded arithmetic, propositional logic, and complexity theory*, Cambridge University Press, 1995.
9. M. Li and P. Vitanyi, *Kolmogorov complexity and applications*, Springer Verlag, 1993.
10. N. Lynch, *On reducibility to complex or sparse sets*, Journal of the ACM, 22:341–345, 1975.
11. M. Mundhenk, *NP-hard sets have many hard instances*, In *Proc. MFCS*, LNCS 1295, 428–437, Springer Verlag, 1997.
12. S. Riis and M. Sitharam, *Generating hard tautologies using predicate logic and the symmetric group*, BRICS Report RS-98-19.
13. U. Schöning, *Complexity cores and hard to prove formulas*, In *Proc. CSL*, LNCS, Springer Verlag, 273–280, 1987.
14. M. Sipser, *A complexity theoretic approach to randomness*, In *Proc. ACM STOC*, 330–335, 1983.