# Complexity of Propositional Proofs Under a Promise*

Nachum Dershowitz[1,2] and Iddo Tzameret[1]

[1] School of Computer Science, Tel Aviv University, Ramat Aviv 69978, Israel
[2] Microsoft Research, Redmond, WA 98052, USA
{nachumd,tzameret}@post.tau.ac.il

*"Goods Satisfactory or Money Refunded"* – The Eaton Promise

**Abstract.** We study – within the framework of propositional proof complexity – the problem of certifying unsatisfiability of CNF formulas under the promise that any satisfiable formula has many satisfying assignments, where "many" stands for an explicitly specified function $\Lambda$ in the number of variables $n$. To this end, we develop propositional proof systems under different measures of promises (that is, different $\Lambda$) as extensions of resolution. This is done by augmenting resolution with axioms that, roughly, can eliminate sets of truth assignments defined by Boolean circuits. We then investigate the complexity of such systems, obtaining an exponential separation in the average-case between resolution under different size promises:

(i) Resolution has polynomial-size refutations for all unsatisfiable 3CNF formulas when the promise is $\varepsilon \cdot 2^n$, for any constant $0 < \varepsilon < 1$.
(ii) There are no sub-exponential size resolution refutations for random 3CNF formulas, when the promise is $2^{\delta n}$ (and the number of clauses is $o(n^{3/2})$), for any constant $0 < \delta < 1$.

**Keywords:** proof complexity, resolution, random 3CNF, promise problems.

## 1 Introduction

Demonstrating unsatisfiability of propositional formulas is one of the most fundamental problems in complexity theory, as well as in hardware and software validation. Any standard sound and complete propositional proof system has the ability to separate the set of unsatisfiable formulas in conjunctive normal form (CNF) from the set of CNF formulas having at least one satisfying assignment, in the sense that every unsatisfiable CNF has a refutation in the system, while no satisfiable CNF has one. Our goal is to develop and study, within the framework of propositional proof complexity, systems that are "sound and complete" in a relaxed sense: they can separate the set of unsatisfiable CNF formulas

* This work was carried out in partial fulfillment of the requirements for the Ph.D. degree of the second author and was supported in part by the Israel Science Foundation (grant no. 250/05).

L. Arge et al. (Eds.): ICALP 2007, LNCS 4596, pp. 291–302, 2007.

from the set of CNF formulas having *sufficiently many* satisfying assignments (where the term "sufficiently many" stands for an explicitly given function of the number of variables in the CNF). We call such proof systems *promise refutation systems*, as they are complete and sound for the set of CNF formulas promised to be either unsatisfiable or to have many satisfying assignments.

As the proof systems we develop here intend to prove the *unsatisfiability* of CNF formulas (in other words, to *refute* them, which is to validate their negation), we will work solely with *refutation* systems, and shall speak about refutations and proofs interchangeably, always intending refutations, unless otherwise stated. In particular, we work with refutation systems that extend the widely studied resolution refutation system.

Our first task is to introduce a natural model for promise propositional refutation systems. This is accomplished by augmenting the standard resolution refutation system (or any other propositional proof system extending resolution) with an additional collection of axioms, the *promise axioms*. Each refutation in a promise refutation system can make use of at most one promise axiom. The promise axioms are meant to capture the idea that we can "ignore" or "discard" a certain number of truth assignments from the space of all truth assignments, and still be able to certify (due to the promise) whether the given CNF is unsatisfiable or not. The number of assignments that a promise axiom is allowed to discard depends on the promise we are given, and specifically it needs to be less than the number of assignments promised to satisfy a given CNF (unless it is unsatisfiable).

Assuming we have a promise that a satisfiable CNF has more than $\Lambda$ satisfying assignments then we can discard up to $\Lambda$ assignments. We refer to $\Lambda$ as the *promise*. This way the refutation system is guaranteed not to contain refutations of CNF formulas having more than $\Lambda$ satisfying assignments, as even after discarding (at most $\Lambda$) assignments we still have at least one satisfying assignment left, and on the other hand, any unsatisfiable CNF formula has a refutation in the system, as resolution already has a refutation of it. We now explain (somewhat informally) what it means to *discard* assignments and how promise axioms formulate the notion of discarding the *correct number* of truth assignments.

Essentially, we say that a truth assignment $\boldsymbol{a}$ is discarded by some Boolean formula if $\boldsymbol{a}$ falsifies the formula. More formally, let $X := \{x_1, ..., x_n\}$ be the set of the underlying variables of a given CNF, called the *original variables*. Let $A$ be some CNF formula in the $X$ variables, and assume that $A$ also contains variables not from $X$ called *extension variables*. Let $\boldsymbol{a} \in \{0,1\}^n$ be a truth assignment to the $X$ variables, and assume that there is no extension of $\boldsymbol{a}$ (to the extension variables) that satisfies $A$. Thus, any assignment satisfying $A$ must satisfy also $X \not\equiv \boldsymbol{a}$ (that is, $A \models X \not\equiv \boldsymbol{a}$), and so any (implicationally) complete proof system can prove $X \not\equiv \boldsymbol{a}$ from $A$, or, in the case of a refutation system, can refute $X \equiv \boldsymbol{a}$, given $A$. In this case, we say that the assignment $\boldsymbol{a}$ is *discarded by* $A$.

The promise axioms we present have two main properties:

(I) They discard assignments from the space of possible assignments to the variables $X$.
(II) They express the fact that not too many assignments to the variables $X$ are being discarded (in a manner made precise).

The first property is achieved as follows: Let $C$ be any Boolean circuit with $n$ output bits. Then we can formulate a CNF formula denoted by $A$ (using extension variables) expressing the statement that the (vector of) variables $X$ is equal to the output of $C$. This enables $A$ to *discard every truth assignment to the $X$ variables outside the image of the Boolean map defined by $C$* (as if an assignment $\boldsymbol{a}$ to the $X$ variables is not in the image of $C$ then no extension of $\boldsymbol{a}$ can satisfy $A$, assuming the formulation of $A$ is correct). (The actual definition is a bit different than described here, due to technical reasons; see Sect. 3).

The second property is achieved as follows: Assume we can make the statement that the *domain* of the map defined by the Boolean circuit $C$ above is of size at least $2^n - \Lambda$ explicit (see Sect. 3 for more details on this). Then, in order for the second property to hold it is sufficient that the axiom formulates the statement that the circuit $C$ defines an *injective* map (and thus the image of the map contains enough truth assignments), which can be done quite naturally.

Given a certain promise and its associated promise axiom, we call a refutation of resolution augmented with the promise axiom a *resolution refutation under the (given) promise*.

Our second task, besides introducing the model of promise refutation systems, is to investigate the basic properties of this model and in particular to determine its average-case proof complexity with respect to different size of promises (see below for a summery of our findings in this respect).

### 1.1 Background and Motivation

In propositional proof complexity theory, it is standard to consider an *abstract* or *formal* propositional proof system (usually called *a Cook-Reckhow proof system*, following [3]) as a polynomial-time algorithm $A$ that receives a Boolean formula $F$ (usually in CNF) and a string $\pi$ over some finite alphabet ("the (proposed) refutation" of $F$), such that there exists a $\pi$ with $A(F, \pi) = 1$ if and only if $F$ is unsatisfiable. (A string $\pi$ for which $A(F, \pi) = 1$ is also called a *witness* for the unsatisfiability of $F$.) Equipped with this abstract definition of propositional proof systems, showing that *for every* abstract proof system there exists some family of formulas $F$ for which there is no polynomially-bounded family of proofs $\pi$ of $F$ is equivalent to showing $\mathbf{NP} \neq \mathbf{coNP}$.

For this reason (among others), it is customary in proof complexity theory to concentrate on specific (sometimes provably weaker) proof systems for which proofs have a simple structure. This makes the complexity analysis of such proof systems simpler. Prominent examples of such systems are Frege systems and weaker subsystems of Frege, the most notable of which is the resolution refutation system, which also plays an important rôle in many automated theorem provers.

In accordance with this, we shall be interested not with abstract promise proof systems (that is, not with finding general witnesses for unsatisfiability, possibly under a promise), but rather with specific and more structured proof systems, and specifically with refutation systems built-up as extensions of resolution.

A natural relaxation of the problem of unsatisfiability certification is to require that, if a CNF is satisfiable, then it actually have many satisfying assignments. As mentioned above, we call the specific number of assignments (as a function of the number of variables $n$) required to satisfy a satisfiable CNF formula, the promise. Accordingly, one can define an *abstract promise proof system* in an analogous manner to the definition of an abstract proof system. It is thus natural to ask whether giving such a promise can help in obtaining shorter proofs of unsatisfiability.

In the case of a *big* promise, that is, a constant fraction of the space of all truth assignments ($\Lambda = \varepsilon \cdot 2^n$, for a constant $0 < \varepsilon < 1$), there is already a *deterministic polynomial-time algorithm* for any fixed natural number $k$ that certifies the unsatisfiability of all unsatisfiable $k$CNF formulas under the promise: The algorithm receives a $k$CNF that is either unsatisfiable or has more than $\Lambda$ satisfying assignments and answers whether the formula is unsatisfiable (in case the formula is satisfiable, the algorithm provides a satisfying assignment); see the papers by Hirsch [6] and Trevisan [7] for such efficient algorithms. This trivially implies the existence of polynomial-size witnesses for any unsatisfiable $k$CNF under the promise $\varepsilon \cdot 2^n$. But does already resolution admit such short witnesses of unsatisfiability (that is, resolution refutations) under a big promise? We show that the answer is positive (for all unsatisfiable 3CNF formulas).

In the case of a *smaller* promise, by which we mean $\Lambda = 2^{\delta n}$ for a constant $0 < \delta < 1$, it is possible to efficiently transform any CNF over $n$ variables to a new CNF with $n' = \lceil n/(1-\delta) \rceil$ variables such that the original CNF is satisfiable if and only if the new CNF has at least $2^{\delta n'}$ satisfying assignments. This can be achieved by adding "dummy variables" (e.g. variables that do not occur at all in the formula or by adding any satisfiable CNF consisting of these dummy variables to the original CNF). Thus, the *worst-case* complexity of certifying CNF unsatisfiability under such a promise is polynomially equivalent to the worst-case complexity of certifying CNF unsatisfiability without a promise. However, it is still possible that a promise of $2^{\delta n}$ might give some advantage (that is, a super-polynomial speedup over refutations without a promise) in certifying the unsatisfiability of certain (but not all) CNF formulas; for instance, in the *average-case*.[1]

Feige, Kim and Ofek [5] showed that when the number of clauses is $\Omega(n^{7/5})$ there exist polynomial-size witnesses that witness the unsatisfiability of 3CNF

[1] Note that if we add dummy variables to a 3CNF then we obtain an "atypical instance" of a 3CNF. Thus, assuming we have polynomial-size witnesses of unsatisfiability of 3CNF formulas under a small promise in the average-case (the "typical case"), the reduction alone (that is, adding dummy variables) does *not* automatically yield polynomial-size witnesses for 3CNF formulas in the average-case without a promise as well.

formulas in the *average-case*. On the other hand, Beame et al. [1] and Ben-Sasson and Wigderson [2] showed that resolution does not provide sub-exponential refutations for 3CNF formulas in the average-case when the number of clauses is at most $n^{(3/2)-\epsilon}$, for any constant $0 < \epsilon < 1/2$. This shows that general witnessing of 3CNF unsatisfiability is strictly stronger than resolution refutations. But is it possible that under a promise of $2^{\delta n}$ resolution can do better in the average-case? We show that the answer is negative.

There are two main motivations for studying propositional proofs under a given promise and their complexity. The first is to answer the natural question whether CNF unsatisfiability certification enjoys any advantage given a certain promise. As already mentioned, the answer is positive when the promise is a constant fraction of all the truth assignments, and our results imply that this phenomenon already occurs for resolution. For a smaller promise of $2^{\delta n}$, we can show that at least in the case of resolution refutations of most 3CNF formulas (of certain clause-to-variable density) the answer is negative. In fact, we can show that the answer stays negative even when the promise is bigger than $2^{\delta n}$, and specifically when $\Lambda = 2^n/2^{n^{\xi}}$ for some constant $0 < \xi < 1$. Overall, our results establish the first unsatisfiability certification model in which a promise of a certain size is known to help (i.e., allows for more efficient certifications) in the average-case, while a promise of smaller size does not help.

The second motivation, is more intrinsic to proof complexity theory. It is a general goal to develop natural frameworks for propositional proofs that are not sound in the strict sense, but rather possess an approximate notion of soundness (like showing that certain "approximations" give speed-ups). For this purpose, the proof systems we propose formalize – in a natural way – the notion of separating unsatisfiable CNF formulas from those that have many satisfying assignments. The promise axioms we present also allow for a natural way of controlling the size of the promise, which in addition leads to an exponential separation between different size promises.

This paper introduces the concept of propositional proofs under a promise, analyzes the proof complexity of these proof systems with respect to different promise sizes, giving a separation between promises of different sizes, and also illustrates several new facts about the widely studied resolution proof system.

### 1.2 Results

We show that resolution refutations are already enough to efficiently separate unsatisfiable 3CNF formulas from those 3CNF formulas with an arbitrarily small constant fraction of satisfying assignments. In particular, in Section 4, we show the following:

**First Main Result:** Let $0 < \varepsilon < 1$ be some constant and let $\Lambda = \varepsilon \cdot 2^n$ be the given promise. Then *every* unsatisfiable 3CNF with $n$ variables has a polynomial-size (in $n$) resolution refutation under the promise $\Lambda$.

The proof resembles a deterministic algorithm of Trevisan [7] for approximating the number of satisfying assignments of $k$CNF formulas.

In contrast to the case of a big promise, we also show that, at least for resolution, a small promise of $\Lambda = 2^{\delta n}$ (for any constant $0 < \delta < 1$) does not give any advantage over standard resolution (resolution without the promise axioms) in most cases (that is, in the average-case). Specifically, in Section 5, we show the following:

**Second Main Result:** Let $0 < \delta < 1$ be any constant and let $\Lambda = 2^{\delta n}$ be the given promise. Then, there is an exponential lower bound on the size of resolution refutations of random 3CNF formulas under the promise $\Lambda$, when the number of clauses is $o(n^{3/2})$.

This lower bound actually applies to a more general model of promise proofs: It remains valid even if we allow (somehow) the promise proofs to discard *arbitrarily chosen* sets of truth assignments (of $\Lambda = 2^{\delta n}$ size), and not necessarily those sets that are definable by (small) Boolean circuits. In fact, the lower bound applies even to a bigger promise of $\Lambda = 2^{n-n^{\xi}}$, for some constant $0 < \xi < 1$.

The proof strategy of this lower bound follows that of Ben-Sasson and Wigderson [2] (the *size-width tradeoff* approach), and so the rate of the lower bound matches the one in that paper. The main novel observation is that under the appropriate modifications this strategy also works when one restricts the set of all truth assignments to a smaller set (that is, from $2^n$ down to $2^n - 2^{\delta n}$ for a constant $0 < \delta < 1$, and in fact down to $2^n - 2^{n-n^{\xi}}$, for some constant $0 < \xi < 1$).

It is important to note that the two main results above show that the decision to discard sets of truth assignments defined by *Boolean circuits* does not effect the results in any way, and thus should not be regarded as a restriction of the model of promise refutations (at least not for resolution). To see this, note that we could allow a promise refutation to discard *arbitrarily chosen* sets of truth assignments (of the appropriate size determined by the given promise); that is, sets of truth assignments that are not necessarily definable by (small) Boolean circuits. However, although this modification strengthens the model it is not really necessary for the *upper bound* in the First Main Result, as this upper bound is already valid when one discards sets of truth assignments by (small) Boolean circuits. On the other hand, as mentioned above, the *lower bound* in the Second Main Result is already valid when one allows a promise refutation to discard any *arbitrarily chosen* set of truth assignments (of the appropriate size).

The exact model of promise propositional proof systems is developed in Sect. 3. It is preceded by preliminaries and terminological conventions.

## 2 Preliminaries

**Resolution refutation system.** Resolution is a complete and sound proof system for unsatisfiable CNF formulas. Let $C$ and $D$ be two clauses containing neither $x_i$ nor $\neg x_i$, the *resolution rule* allows one to derive $C \vee D$ from $C \vee x_i$ and $D \vee \neg x_i$. The clause $C \vee D$ is the *resolvent* of the clauses $C \vee x_i$ and $D \vee \neg x_i$ on the variable $x_i$. The *weakening rule* allows to derive the clause $C \vee D$ from the clause $C$, for any two clauses $C, D$.

**Definition 1 (Resolution).** *A* resolution proof of the clause $D$ from a CNF formula $K$ *is a sequence of clauses* $D_1, D_2, \ldots, D_\ell$*, such that: (1) each clause* $D_j$ *is either a clause of* $K$ *or a resolvent of two previous clauses in the sequence or derived by the weakening rule from a previous clause in the sequence; (2) the last clause* $D_\ell = D$*. The size of a resolution proof is the total number of clauses in it. A* resolution refutation *of a CNF formula* $K$ *is a resolution proof of the empty clause* $\square$ *from* $K$ *(the empty clause stands for* FALSE*).*

Let $K$ be an unsatisfiable CNF formula. The *resolution refutation size of* $K$ is the minimal size of a resolution refutation of $K$. If $K$ has a polynomial-size resolution refutation we say that resolution can *efficiently certify* the unsatisfiability of $K$. Similarly, if the clause $D$ has a polynomial-size resolution proof from $K$ we say that $D$ *is efficiently provable from* $K$.

**Boolean circuit encoding.** The promise axioms we introduce use Boolean circuits to define the set of assignments to be discarded (see Sect. 3). Therefore, as resolution operates only with clauses, we need to encode Boolean circuits as collections of clauses (CNF formulas). For most purposes, we will not need an explicit description of how this encoding is done. Nevertheless, in Sect. 4 we need to ensure that resolution can efficiently prove several basic facts about the encoded circuits. For this reason, and for the sake of concreteness of the promise axioms (see Definition 4), we provide the precise definition of the encoding in the full version of this paper [4], in addition to proving some of the encoding's basic (proof theoretical) properties.

## 3 Promise Proof Systems

In this section, we define precisely the model of refutations under a promise. As discussed in the introduction, we work with the resolution refutation system as our underlying system and augment it with a new set of axioms that we call the *promise axioms*. We call this proof system *promise resolution*. The promise axioms are meant to express the fact that we can discard a certain number of truth assignments from the space of all truth assignments and still be able to certify (due to the promise) whether the input CNF is unsatisfiable or not. Each promise resolution refutation can use at most one promise axiom.

From now on, we assume that the underlying variables of the CNF formulas that are meant to be refuted are taken from the set $X := \{x_1, \ldots, x_n\}$. The $X$ variables are called the *original variables*. Any other variable that appears in a (promise resolution) refutation is an *extension variable*.

**Definition 2 (CNF formulas under a promise).** *Let* $\Lambda$ *be a fixed function in* $n$ *(the number of* $X$ *variables) such that* $0 \leq \Lambda(n) \leq 2^n$*. The function* $\Lambda$ *is called the* promise*. The set of* CNF formulas under the promise $\Lambda$ *consists of all CNF formulas in the* $X$ *variables that are either unsatisfiable or have more then* $\Lambda(n)$ *satisfying assignments (for* $n = |X|$*).*

The refutation systems we build are sound and complete for the set of CNF formulas under a (given) promise. That is, every unsatisfiable CNF formula has a refutation in the system (this corresponds to completeness), while no CNF having $n$ variables and more than $\Lambda(n)$ satisfying assignments has a refutation in it (this corresponds to soundness under the promise). The soundness (under the promise) is achieved by requiring that resolution should *prove the fact that we discard the right number of assignments* (see Sect. 3.1 for details).

**Definition 3 (Assignment discarding).** *Let $A$ be a CNF in the $X$ variables that can contain (but not necessarily does) extension variables (that is, variables not from $X$). We say that an assignment to the $X$ variables $\boldsymbol{a}$ is* discarded *by $A$ if there is no extension of $\boldsymbol{a}$ (to the extension variables in $A$) that satisfies $A$.*

### 3.1 Promise Axioms

**Big promise.** We first concentrate on a promise of a *constant fraction of assignments*. Let the promise (see Definition 2) be $\Lambda = \varepsilon \cdot 2^n$, for a constant $0 < \varepsilon < 1$ (we fix this $\Lambda$ throughout this subsection), and let $r = \lceil \log(1/\varepsilon) \rceil$ and $t = 2^r - 1$. Let $C$ be a sequence of Boolean circuits $C := (C^{(1)}, \ldots, C^{(t)})$. Assume that each $C^{(i)}$ has $n - r$ input bits and $n$ output bits and computes the Boolean map $f_i : \{0,1\}^{n-r} \to \{0,1\}^n$. Assume further that the $f_i$'s are all injective maps and that the images of all these maps are pairwise disjoint. Denote by $\mathrm{Im}(f_i)$ the image of the map $f_i$. For simplicity, we call the union $\bigcup_{i=1}^{t} \mathrm{Im}(f_i)$ *the image of* $C$ and denote it by $\mathrm{Im}(C)$. By the definition of $r$, we have $2^{n-r} \leq \varepsilon \cdot 2^n = \Lambda$, and by the injectivity and pairwise disjointness of the images of the $f_i$'s we have:

$$|\mathrm{Im}(C)| = t \cdot 2^{n-r} = (2^r - 1) \cdot 2^{n-r} = 2^n - 2^{n-r} \geq 2^n - \Lambda\,. \tag{1}$$

Therefore, *we can treat* $\mathrm{Im}(C)$ *as the set of all possible truth assignments for the original variables $X$, without loosing soundness*: If $K$ is unsatisfiable then there is no assignment in $\mathrm{Im}(C)$ that satisfies $K$; and if $K$ is satisfiable then according to the promise it has more than $\Lambda$ satisfying assignments, which means that there is at least one assignment in $\mathrm{Im}(C)$ that satisfies $K$. This idea is formulated as a propositional formula, as follows:

**Definition 4 (Promise Axiom for $\Lambda = \varepsilon \cdot 2^n$).** *Let the promise be $\Lambda = \varepsilon \cdot 2^n$, for a constant $0 < \varepsilon < 1$, and let $r = \lceil \log(1/\varepsilon) \rceil$ and $t = 2^r - 1$. Let $C$ be a sequence of Boolean circuits $C := (C^{(1)}, \ldots, C^{(t)})$. Assume that each $C^{(i)}$ has $n - r$ input bits and $n$ output bits and let $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ be two disjoint sets of $n-r$ extension variables each. The promise axiom* $\mathrm{PRM}_{C,\Lambda}$ *is the CNF encoding of the following Boolean formula (see the encoding in the full version [4]):*

$$\left( \bigwedge_{i=1}^{t} \left( C^{(i)}(\boldsymbol{W_1}) \equiv C^{(i)}(\boldsymbol{W_2}) \to \boldsymbol{W_1} \equiv \boldsymbol{W_2} \right) \wedge \bigwedge_{1 \leq i < j \leq t} C^{(i)}(\boldsymbol{W_1}) \not\equiv C^{(j)}(\boldsymbol{W_2}) \right) \longrightarrow \bigvee_{i=1}^{t} C^{(i)}(\boldsymbol{W_1}) \equiv X.$$

(The notation $\boldsymbol{W}_1 \equiv \boldsymbol{W}_2$ means that the $i$th variable in $\boldsymbol{W}_1$ is logically equivalent to the $i$th variable in $\boldsymbol{W}_2$, and similarly for $C^{(i)}(\boldsymbol{W_1}) \equiv C^{(i)}(\boldsymbol{W_2})$; see the full version of this paper for more details.) The promise axiom $\mathrm{PRM}_{C,\Lambda}$ expresses the fact that if each circuit in $C$ computes an injective map (this is formulated as $\bigwedge_{i=1}^{t}(C^{(i)}(\boldsymbol{W_1}) \equiv C^{(i)}(\boldsymbol{W_2}) \rightarrow \boldsymbol{W_1} \equiv \boldsymbol{W_2})$), and if the images of the maps computed by each pair of circuits in $C$ are disjoint (this is formulated as $\bigwedge_{1\leq i<j\leq t} C^{(i)}(\boldsymbol{W_1}) \not\equiv C^{(j)}(\boldsymbol{W_2})$), then we can assume that the assignments to the original variables $X$ are taken from the image of $C$ (this is formulated as $\bigvee_{i=1}^{t} C^{(i)}(\boldsymbol{W_1}) \equiv X$). The fact that the image of $C$ is of size at least $2^n - \Lambda$ is expressed (due to Eq. (1)) by the number of input bits (i.e., $n-r$) of each circuit in $C$ and the number of circuits in $C$ (i.e., $t$). Also note that the promise axiom is of polynomial-size as long as the circuits in $C$ are (since $1/\varepsilon$ is a constant).

The following claim states that the promise axioms are sound with respect to the promise $\Lambda$ in the sense that they do not discard too many truth assignments (see the full version [4] for the proof):

*Claim.* The promise axiom $PRM_{C,\Lambda}$ discards at most $\Lambda$ truth assignments. That is, there are at most $\Lambda$ distinct assignments $\boldsymbol{a}$ to the $X$ variables such that $PRM_{C,\Lambda} \models X \not\equiv \boldsymbol{a}$.

**Smaller promise.** We are also interested in formulating promise axioms for promises smaller than $\varepsilon \cdot 2^n$. Specifically, we are interested in the promise $\Lambda = 2^{\delta n}$ for a constant $0 < \delta < 1$. For such a promise, the promise axiom is essentially similar to Definition 4, except that the number of input bits of each circuit in $C$ needs to be modified accordingly. Due to space limitations, we do not describe the formulation of this kind of promise axiom (and refer the interested reader to the full version [4]).

### 3.2 Promise Resolution

**Definition 5 (Promise resolution).** *Let $\Lambda$ be the promise (see Definition 2) and let $K$ be a CNF in the $X$ variables. A* promise resolution (under the promise $\Lambda$) proof of the clause $D$ from a CNF formula $K$ *is a sequence of clauses $D_1, D_2, \ldots, D_\ell$ such that: (1) Each clause $D_j$ is either a clause of $K$ or a clause of a promise axiom* $\mathrm{PRM}_{C,\Lambda}$ *(where* $\mathrm{PRM}_{C,\Lambda}$ *is either a big or a smaller promise axiom as defined, for instance, in Definitions 4, and $C$ is an arbitrary sequence of circuits with the prescribed input and output number of bits) or a resolvent of two previous clauses in the sequence or a weakening of a previous clause; (2) The sequence contains (the clauses of) at most one promise axiom; (3) The last clause $D_\ell = D$. The* size *and* refutations *of promise resolution is defined the same as for resolution.*

Note that promise resolution is a Cook-Reckhow proof system (see the introduction for a definition), in the sense that it is possible to efficiently verify whether a given CNF is an instance of the promise axiom, and hence to verify whether a sequence of clauses constitute a legitimate promise refutation. This can be done by "decoding" the CNF that encodes the promise axiom $\mathrm{PRM}_{C,\Lambda}$ and then checking that each circuit in $C$ has the right number of input and output bits.

**Proposition 1.** *Let $\Lambda$ be the promise (where $\Lambda$ is either $\varepsilon \cdot 2^n$ or $2^{\delta n}$, for $0 < \varepsilon, \delta < 1$). Then, promise resolution under promise $\Lambda$ is a sound and complete proof system for the set of CNF formulas under the promise $\Lambda$. In other words, every unsatisfiable CNF has a promise resolution refutation, and every CNF that has more than $\Lambda$ satisfying assignments does not have promise resolution refutations.*

*Proof.* Completeness stems from completeness of resolution. Soundness under the promise $\Lambda$ stems from Claim 3.1. (This claim refers to the big promise, but a similar argument holds also for the smaller promise axiom.)

The full paper [4] contains a brief discussion of the definition of promise proofs and the choice made in formulating the above promise axioms.

## 4 Big Promise – The Upper Bound

We sketch a proof showing that under the promise $\Lambda = \varepsilon \cdot 2^n$ , for any constant $0 < \varepsilon < 1$, resolution can efficiently certify the unsatisfiability of all unsatisfiable 3CNF formulas. The method resembles the algorithm presented by Trevisan [7] for approximating the number of satisfying assignments of a $k$CNF formula.

The idea behind the refutations in this section is based on the following observation: Given an unsatisfiable 3CNF formula $K$ and a constant $c$, either there are $3(c-1)$ variables that hit[2] all the clauses in $K$ or there are at least $c$ clauses in $K$ over $3c$ *distinct* variables denoted by $K'$ (that is, each variable in $K'$ appears only once). In the first case, we can consider all the possible truth assignments to the $3c$ variables inside resolution: If $K$ is unsatisfiable then any such truth assignment yields an unsatisfiable 2CNF formula, which can be efficiently refuted in resolution. In the second case, we can make use of a promise axiom to efficiently refute $K'$ (this set of clauses has less then $\Lambda$ satisfying assignments, for sufficiently large $c$). Specifically, in the second case, we construct a sequence of small circuits $C$ for which any satisfying assignment for $K'$ is *provably in resolution* (with polynomial-size proofs) outside the image of $C$. The following is the main result of this section:

**Theorem 1.** *Let $0 < \varepsilon < 1$ be a constant and let $\Lambda = \varepsilon \cdot 2^n$ be the given promise. Then* every *unsatisfiable 3CNF with $n$ variables has a polynomial-size (in $n$) resolution refutation under the promise $\Lambda$.*

This theorem is a consequence of the three lemmas that follow (their proofs appear in the full version [4]):

**Lemma 1.** *Let $K$ be a 3CNF formula. For every integer $c$ one of the following holds: (i) there is a set of at most $3(c-1)$ variables that hit all the clauses in $K$; or (ii) there is a sub-collection of clauses from $K$, denoted $K'$, with at least $c$ clauses and where each variable appears only once in $K'$.*

[2] A set of variables $S$ that *hit all the clauses in a CNF formula $K$* is a set of variables for which every clause in $K$ contains some variable from $S$.

If case (i) of the prior lemma holds, then the following lemma suffices to efficiently refute the 3CNF:

**Lemma 2.** *Let $c$ be constant and $K$ be an unsatisfiable 3CNF formula in the $X$ variables (where $n = |X|$). Assume that there is a set $S \subseteq X$ of at most $3(c-1)$ variables that hit all the clauses in $K$. Then, there is a polynomial-size (in $n$) resolution refutation of $K$.*

If case (ii) in Lemma 1 holds, then it suffices to show that resolution under a big promise can efficiently refute any 3CNF formula $T$ with a constant number of clauses (for a sufficiently large constant), where *each variable in $T$ occurs only once* (such a $T$ is of course satisfiable, but it has less than an $\varepsilon$ fraction of satisfying assignments for a sufficiently large number of clauses):

**Lemma 3.** *Fix the constant $c = 3\lceil \log_{7/8}(\varepsilon/2) \rceil$. Let $\Lambda = \varepsilon \cdot 2^n$, where $0 < \varepsilon < 1$ is a constant and $n$ is sufficiently large. Assume that $T$ is a 3CNF with $c/3$ clauses (and $c$ variables) over the $X$ variables, where each variable in $T$ occurs only once inside $T$. Then, there is a polynomial-size resolution refutation of $T$ under the promise $\Lambda$.*

The proof of Lemma 3 consists of constructing a sequence of polynomial-size circuits $C$ (where the parameters of the circuits in $C$ are taken from Definition 4; that is, $r = \lceil \log(1/\varepsilon) \rceil$ and $t = 2^r - 1$), such that: (i) resolution can efficiently prove the injectivity and the pairwise disjointness of the images of the circuits in $C$; and (ii) there is a polynomial-size refutation of $T$ and $\mathrm{PRM}_{\Lambda,C}$. In other words, there is a polynomial-size derivation of the empty clause from the clauses of both $T$ and $\mathrm{PRM}_{\Lambda,C}$.

## 5 Smaller Promise – The Lower Bound

In this section, we state an exponential lower bound on the size of resolution refutations under the promise $2^{\delta n}$, for any constant $0 \leq \delta \leq 1$. The lower bound applies to random 3CNF formulas with $o(n^{3/2})$ number of clauses (where $n$ is the number of variables in the 3CNF). This lower bound matches the known lower bound on resolution refutation-size for random 3CNF formulas (without any promise). Basically, the proof strategy of our lower bound is similar to that of [2], except that we need to take care that every step in the proof works with the augmented (smaller) promise axiom.

The lower bound is somewhat stronger than described above in two respects. First, we show that restricting the set of all $2^n$ truth assignments to *any* smaller set (not just those sets defined by small circuits) that consists of $2^n - 2^{\delta n}$ assignments (for any constant $0 \leq \delta \leq 1$), does not give resolution any advantage in the average-case. One can think of such a restriction as modifying the semantic implication relation $\models$ to take into account only assignments from some prescribed set of assignments $S$, such that $|S| = 2^n - 2^{\delta n}$ (in other words, for two formulas $A, B$, we have that $A \models B$ under the restriction to $S$ iff any truth

assignment *from* $S$ that satisfies $A$ also satisfies $B$). Formally, this means that the lower bound does not use the fact that the restricted domain of size $2^n - 2^{\delta n}$ is defined by a sequence $C$ of polynomial-size circuits (nor the fact that the circuits in $C$ ought to have polynomial-size resolution proofs of their injectivity and pairwise disjointness). Second, we could allow for a promise that is bigger than $2^{\delta n}$, and in particular for a promise of $2^{n(1-1/n^{1-\xi})} = 2^n/2^{n^\xi}$, for some constant $0 < \xi < 1$.

The following defines the usual *average-case* setting of 3CNF formulas:

**Definition 6 (Random 3CNF formulas).** *For a* 3CNF *formula $K$ with $n$ variables $X$ and $\beta \cdot n$ clauses, we say that $\beta$ is the* density *of $K$. A* random 3CNF formula *on $n$ variables and density $\beta$ is defined by picking $\beta \cdot n$ clauses from the set of all $2^3 \cdot \binom{n}{3}$ clauses, independently and indistinguishably distributed, with repetitions.*

Finally, the next theorem gives our lower bound. A complete proof appears in the full version [4].

**Theorem 2.** *Let $0 < \delta < 1$ and $0 < \epsilon < 1/2$. With high probability a random 3CNF formula with $\beta = n^{1/2-\epsilon}$ requires a size $\exp(\Omega(\beta^{-4/(1-\epsilon)} \cdot n))$ resolution refutation under the promise $\Lambda = 2^{\delta n}$.*

## Acknowledgments

The second author is indebted to Ran Raz for very helpful conversations that led to the present paper. We also wish to thank Jan Krajíček for commenting on an earlier version of this paper and Eli Ben-Sasson and Amnon Ta-Shma for useful correspondence and conversations.

## References

1. Beame, P., Karp, R., Pitassi, T., Saks, M.: The efficiency of resolution and Davis-Putnam procedures. SIAM J. Comput. 31(4), 1048–1075 (2002)
2. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow—resolution made simple. J. ACM 48(2), 149–169 (2001)
3. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. The Journal of Symbolic Logic 44(1), 36–50 (1979)
4. Dershowitz, N., Tzameret, I.,: Complexity of propositional proofs under a promise (full version) (2007) `http://www.cs.tau.ac.il/~tzameret/PromiseProofs.pdf`
5. Feige, U., Kim, J., Ofek, E.: Witnesses for non-satisfiability of dense random 3CNF formulas. In: Proc. 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 497–508 (October 2006)
6. Hirsch, E.: A fast deterministic algorithm for formulas that have many satisfying assignments. Logic Journal of the IGPL 6(1), 59–71 (1998)
7. Trevisan, L.: A note on approximate counting for $k$-DNF. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 417–426. Springer, Heidelberg (2004)