Input Distance and Lower Bounds for Propositional Resolution Proof Length

Allen Van Gelder

University of California, Santa Cruz CA 95060, USA,
WWW home page: `http://www.cse.ucsc.edu/~avg`

Abstract. Input Distance (Δ) is introduced as a metric for propositional resolution derivations. If $\mathcal{F} = C_i$ is a formula and D is a clause, then $\Delta(D, \mathcal{F})$ is defined as $\min_i |D - C_i|$. The Δ for a derivation is the maximum Δ of any clause in the derivation. Input Distance provides a refinement of the clause-width metric analyzed by Ben-Sasson and Wigderson (JACM 2001) in that it applies to families whose clause width grows, such as pigeon-hole formulas. They showed two upper bounds on $(W - \mathit{width}(\mathcal{F}))$, where W is the maximum clause width of a narrowest refutation of $\mathcal{F}$. It is shown here that (1) both bounds apply with $(W - \mathit{width}(\mathcal{F}))$ replaced by Δ; (2) for pigeon-hole formulas $\mathrm{PHP}(m, n)$, the minimum Δ for any refutation is $\Omega(n)$. A similar result is conjectured for the $GT(n)$ family analyzed by Bonet and Galesi (FOCS 1999).

1 Introduction

The reader is assumed to be generally familiar with the propositional satisfiability problem, CNF formulas, and resolution derivations. Some definitions are briefly reviewed in Section 2, but are not comprehensive.

Ben-Sasson and Wigderson [3] showed that, if the minimum-length general resolution refutation for a CNF formula $\mathcal{F}$ has S steps, and if the minimum-length tree-like refutation of $\mathcal{F}$ has S_T steps, then there is a (possibly different) refutation of $\mathcal{F}$ using clauses of width at most:

$$w(\mathcal{F} \vdash \bot) \leq w(\mathcal{F}) + c\sqrt{n \ln S}; \tag{1}$$

$$w(\mathcal{F} \vdash \bot) \leq w(\mathcal{F}) + \lg S_T. \tag{2}$$

Note that the $w(\mathcal{F})$ terms were omitted from their statement in the introduction, but appear in their statements of the theorems. The notation for this expression is:

- n is the number of propositional variables in $\mathcal{F}$;
- $w(\mathcal{F})$ is the width of the widest clause in $\mathcal{F}$;
- $w(\mathcal{F} \vdash \bot)$ denotes the minimum *resolution width* of π ranging over all resolution derivations that refute $\mathcal{F}$, where the *resolution width* of π, denoted $w_{\mathcal{F}}(\pi)$, is the width of the widest clause in π;
- c is a constant, independent of $\mathcal{F}$;

– ln and lg denote natural and binary logs, respectively.

All formulas and clauses are propositional, clauses are disjunctions of literals, formulas are in CNF, unless specified otherwise.

Our main results essentially eliminate the $w(\mathcal{F})$ terms in the Ben-Sasson and Wigderson theorems [3], and replace resolution width by $\Delta_{\mathcal{F}}(\pi)$, the *input distance*, as defined next, in Section 1.1. For families of formulas whose widest clause is bounded by a constant, input distance and resolution width are essentially equivalent measures.

1.1 Input Distance

We define *input distance* for nontautologous clauses (primarily derived clauses in a resolution proof) for input CNF formula $\mathcal{F}$.

Definition 1.1. (input distance) All clauses mentioned are non-tautologous. Let D be a clause; let C be an input clause, i.e., a clause of formula $\mathcal{F}$. The *input distance* of D from C is $|D - C|$, treating D and C as sets of literals, and using "−" for set difference. The *input distance* of D from $\mathcal{F}$, denoted $\Delta_{\mathcal{F}}(D)$, is the minimum over $C \in \mathcal{F}$ of the *input distances* of D from C.

For a resolution proof π the *input distance* of π from $\mathcal{F}$, denoted $\Delta_{\mathcal{F}}(\pi)$, is the maximum over $D \in \pi$ of the *input distances* of D from $\mathcal{F}$.

When $\mathcal{F}$ is understood from the context, $\Delta(D)$ and $\Delta(\pi)$ are written. Following Ben-Sasson and Wigderson [3], $\Delta(\mathcal{F} \vdash D)$ denotes the minimum of $\Delta_{\mathcal{F}}(\pi)$ over all π that are derivations of D from $\mathcal{F}$.

1.2 Summary of Results

The theorems shown here are that, if π is a resolution refutation of $\mathcal{F}$ and π uses all clauses of $\mathcal{F}$ and the length of π is S, then there is a refutation of $\mathcal{F}$ using clauses that have *input distance* from $\mathcal{F}$ that is at most:

$$\Delta(\mathcal{F} \vdash \bot) \leq c\sqrt{n \ln S}; \tag{3}$$

$$\Delta(\mathcal{F} \vdash \bot) \leq \lg S_T. \tag{4}$$

Also, we show that the pigeon-hole family of formulas $\mathrm{PHP}(m,n)$ require refutations with input distance $\Omega(n)$, although they contain clauses of width n. This result suggests that input distance provides a refinement of the clause-width metric as a measure of resolution difficulty. That is, when a family of formulas with increasing clause-width, such as $\mathrm{PHP}(m,n)$, is transformed into a bounded-width family, such as $\mathrm{EPHP}(m,n)$, and the bounded-width family has large resolution width, this is not simply because they rederive the wide clauses of the original family, then proceed to refute the original family. Rather, it is the case that wide clauses substantially different from those in the original family must be derived. However, note that input distance $\Omega(n)$ does not imply any useful lower bound on general resolution refutation length for $\mathrm{PHP}(m,n)$, at least not through any known theorem.

Table 1. Summary of notations.

$a, \ldots, z$	Literal; i.e., propositional variable or negated propositional variable.
$\neg x$	Complement of literal x; $\neg\neg x$ is not distinguished from x.
$\lvert x \rvert$	The propositional variable in literal x; i.e., $\lvert a \rvert = \lvert \neg a \rvert = a$.
$A, \ldots, Z$	Disjunctive clause, or set of literals, depending on context.
$\mathcal{A}, \ldots, \mathcal{H}$	CNF formula, or set of literals, depending on context.
π	Resolution derivation DAG.
σ	Total assignment, represented as the set of true literals.
$[p_1, \ldots, p_k]$	Clause consisting of literals $p_1, \ldots, p_k$.
$\bot$	The *empty clause*, which represents *false*.
$\top$	The *tautologous clause*, which represents *true*; (see Definition 2.2).
$\alpha, \ldots, \delta$	Subclause, in the notation $[p, q, \alpha]$, denoting a clause with literals p, q, and possibly other literals, α.
C^-	Read as "C, or some clause that subsumes C".
p	In a context where a unit clause is expected, $[p]$ may be abbreviated to p.
C, p	In a context where a formula is expected, $\{C\}$ may be abbreviated to C and $\{[p]\}$ may be abbreviated to p.
$+$, $-$	Set union and difference, as infix operators, where operands are formulas, possibly using the abbreviations above.
$\mathbf{res}(q, C, D)$	Resolvent of C and D, where q and $\neg q$ are the clashing literals (see Definition 2.2).
$C\lvert\mathcal{A}$, $\mathcal{F}\lvert\mathcal{A}$, $\pi\lvert\mathcal{A}$	C (respectively $\mathcal{F}$, π) *strengthened* by $\mathcal{A}$ (see Definition 2.4).

2 Preliminaries

2.1 Notation

This section collects notations and definitions used throughout the paper. Standard terminology for conjunctive normal form (CNF) formulas is used. Notations are summarized in Table 1. Although the general ideas of resolution and derivations are well known, there is no standard notation for many of the technical aspects, so it is necessary to specify our notation in detail.

Definition 2.1. (assignment, satisfaction, model) A partial assignment is a partial function from the set of variables into $\{false, true\}$. This partial function is extended to literals, clauses, and formulas in the standard way. If the partial assignment is a total function, it is called a *total assignment*, or simply an *assignment*.

A clause or formula is *satisfied* by a partial assignment if it is mapped to *true*; A partial assignment that satisfies a formula is called a *model* of that formula. □

A partial assignment is conventionally represented by the (necessarily consistent) set of *unit clauses* that are mapped into *true* by the partial assignment. Note that this representation is a very simple formula.

2.2 Resolution as a Total Function

Definition 2.2. (resolution, subsumption, tautologous) A clause is *tautologous* if it contains complementary literals. All tautologous clauses are considered to be indistinguishable and are denoted by $\top$.

If $C = [q, \alpha]$ and $D = [\neg q, \beta]$ are two non-tautologous clauses (α and β are subclauses), then

$$\mathbf{res}(q, C, D) = \mathbf{res}(q, D, C) = \mathbf{res}(\neg q, C, D) = \mathbf{res}(\neg q, D, C) = [\alpha, \beta]$$

defines the *resolution* operation, and $[\alpha, \beta]$ is called the *resolvent*, which may be tautologous. Resolution is extended to include $\top$ as an identity element:

$$\mathbf{res}(q, C, \top) = C$$

provided C contains q or $\neg q$.

Resolution is further extended to apply any two non-tautologous clauses and any literals, as follows. Fix a total order on the clauses definable with the n propositional variables such that $\bot$ is smallest, $\top$ is largest, and wider clauses are "bigger" than narrower clauses. Other details of the total order are not important.

If $C = [\alpha]$ does not contain q and $D = [\neg q, \beta]$ is non-tautologous, then

$$\mathbf{res}(q, C, D) = \mathbf{res}(q, D, C) = \mathbf{res}(\neg q, C, D) = \mathbf{res}(\neg q, D, C) = [\alpha]$$

If $C = [\alpha]$ and $D = [\beta]$ and neither contains q or $\neg q$, and both are non-tautologous, then

$$\begin{aligned}\mathbf{res}(q, C, D) &= \mathbf{res}(q, D, C) = \mathbf{res}(\neg q, C, D) = \mathbf{res}(\neg q, D, C) \\ &= \text{the smaller of } C \text{ and } D.\end{aligned}$$

With this generalized definition of resolution, we have an algebra, and the set of clauses (including $\top$) is a lattice, based on $\subseteq$, with the convention that every clause is a subset of $\top$. We shall see later that the benefit of this structure is that resolution "commutes up to subsumption" with *strengthening* (see Definition 2.4), so strengthening can be applied to any resolution derivation to produce another derivation.

If clause $C \subset D$, we say C *properly subsumes* D; if $C \subseteq D$, we say C *subsumes* D. Also, any non-tautologous clause properly subsumes $\top$. Notation D^- is read as "D, or some clause that subsumes D". □

Definition 2.3. (derivation, refutation) A *derivation* (short for *propositional resolution derivation*) from formula $\mathcal{F}$ is a *rooted*, directed acyclic graph (DAG) in which each vertex is labeled with a clause and possibly with a clashing literal. Let D be the clause label of vertex v. If $D = C \in \mathcal{F}$, then v has no out-edges and no clashing literal, and is called a *leaf*. Otherwise v is called a *resolution vertex*, has two out-edges, say to vertices with clause labels D_1 and D_2, and is also labeled with the clashing literal q such that

$$D = \mathbf{res}(q, D_1, D_2),$$

where **res** is the total function defined in Definition 2.2. In much of the discussion, vertices are referred to by their clause labels.

A derivation derives its root clause. When the root clause is $\bot$, the derivation is called a *refutation*. □

2.3 The Strengthening Operation

Definition 2.4. (strengthened formula, strengthened derivation) Let $\mathcal{A}$ be a partial assignment for formula $\mathcal{F}$. Let π be a derivation from $\mathcal{F}$. The clause $C|\mathcal{A}$, read "C strengthened by $\mathcal{A}$", and the formula $\mathcal{F}|\mathcal{A}$, read "$\mathcal{F}$ strengthened by $\mathcal{A}$", are defined as follows.

1. $C|\mathcal{A} = \top$, if C contains any literal that occurs in $\mathcal{A}$.
2. $C|\mathcal{A} = C - \{q \mid q \in C \text{ and } \neg q \in \mathcal{A}\}$, if C does not contain any literal that occurs in $\mathcal{A}$. This may be the empty clause.
3. $\mathcal{F}|\mathcal{A} = \{C|\mathcal{A} \mid C \in \mathcal{F}\}$; i.e., apply strengthening to each clause in $\mathcal{F}$. Usually, occurrences of $\top$ (produced by part (1)) are deleted in $\mathcal{F}|\mathcal{A}$.
4. $\pi|\mathcal{A}$ is the same DAG as π structurally, but the clauses labeling the vertices are changed as follows. If a leaf (input clause) of π contains C, then the corresponding leaf of $\pi|\mathcal{A}$ contains $C|\mathcal{A}$. Each derived clause of $\pi|\mathcal{A}$ uses resolution on the same clashing literal as the corresponding vertex of π.

The operation $\mathcal{F}|p$ (i.e., $\mathcal{F}|\{[p]\}$) is sometimes called "unit simplification". □

The term "strengthen" comes from the theorem-proving community [7]. Ben-Sasson and Wigderson [3] and others in the proof-complexity community use the term "restriction" for "unit simplification" or "strengthening by a single literal"; several different terms for this operation may be found in the literature.

Example 2.5. Let $\mathcal{F}$ consist of clauses $C_1 = [a, b]$, $C_2 = [\neg a, c]$, $C_3 = [\neg b, e]$, and $C_4 = [\neg c, \neg d]$. Let π consist of leaves C_1, C_2 and C_4 and the derived clauses

$$\begin{aligned} D_1 &= \mathbf{res}(a, C_1, C_2) = [b, c]\,, \\ D_2 &= \mathbf{res}(c, D_1, C_4) = [b, \neg d]\,, \\ D_3 &= \mathbf{res}(b, D_2, C_3) = [e, \neg d]\,. \end{aligned}$$

Then $\mathcal{F}|a = \{[c]\,, [\neg b, e]\,, [\neg c, \neg d]\}$, Also, $\mathcal{F}|\{a, c\} = \{[\neg b, e]\,, [\neg d]\}$.

Now consider $\pi|a$. The leaves are $C_1|a = \top$, $C_2|a = [c]$, $C_3|a = C_3$, and $C_4|a = C_4$. The derived clauses are E_1, E_2 and E_3, where:

$$\begin{aligned} E_1 &= \mathbf{res}(a, \top, [c]) = [c]\,; \\ E_2 &= \mathbf{res}(c, [c]\,, [\neg c, \neg d]) = [\neg d]\,; \\ E_3 &= \mathbf{res}(b, [\neg d]\,, [\neg b, e]) = [\neg d]\,. \end{aligned}$$

Notice that $E_i \neq D_i|a$ in any case, but $E_i = (D_i|a)^-$ in all cases. Also notice that the clashing literal is absent from one operand in the resolution for E_3, so the resolvent is just the other operand. □

Lemma 2.6. Given formula $\mathcal{F}$, and a strengthening literal p,

$$\mathbf{res}(q, D_1|p, D_2|p) \subseteq \mathbf{res}(q, D_1, D_2)|p.$$

Proof. The principal case that requires checking is when $q = p$ and $q \in D_1$ and $\neg q \in D_2$ (or *vice versa*). In this case,

$$\mathbf{res}(q, D_1, D_2)|p = \mathbf{res}(q, D_1, D_2) = (D_1 - p) \cup (D_2 - \neg p).$$

Then $\mathbf{res}(q, D_1|p, D_2|p) = D_2|p = (D_2 - \neg p)$. Therefore, $D_2|p \subseteq \mathbf{res}(q, D_1, D_2)|p$. $\square$

Lemma 2.7. Given formula $\mathcal{F}$, and a strengthening literal p, if π is a derivation of C from $\mathcal{F}$, then $\pi|p$ is a derivation of $(C|p)^-$ (a clause that subsumes $C|p$) from $\mathcal{F}|p$.

Proof. The proof is by induction on the structure of π with edge $v \to w$ interpreted to mean that v is greater than w. Thus, the base cases are the vertices that are clauses in $\mathcal{F}$, called the leaves. By Lemma 2.6, if a vertex of π contains the derived clause C, and the two adjacent operand vertices satisfy the lemma, then the corresponding vertex of $\pi|p$ contains $(C|p)^-$. $\square$

If C is the root of π and $C|p \neq \top$, then a $\top$-free derivation of $(C|p)^-$ can be constructed from $\pi|p$ by changing all resolution vertices that have exactly one $\top$ operand to "copy" vertices that use the non-$\top$ operand, then deleting all the $\top$ vertices, then compressing out all the copy vertices. Finally, the resulting DAG might have multiple sources, so delete all vertices that cannot be reached from the original root, which now contains $(C|p)^-$. This procedure does not change the *clause* in any vertex of $\pi|p$.

Notice that Ben-Sasson and Wigderson [3] define $\pi|p$ differently, as clause-by-clause strengthening (restriction, in their terminology) of the originally derived clauses. As Example 2.5 showed, this definition does not necessarily produce a derivation; they do not discuss this issue. The definitions used herein *do* ensure that the strengthening of a derivation is a derivation, without using weakening. The point of Lemma 2.7 is that the clauses derived from the strengthened formula are at least as strong as the clause-by-clause strengthenings of the originally derived clauses.

2.4 Input Distance and Strengthening

A few properties of input distance on clauses that result from strengthening are stated.

Lemma 2.8. Let C be a clause of $\mathcal{F}$ and let $\mathcal{A}$ be a partial assignment. If $C|\mathcal{A} \neq \top$ (i.e., $\mathcal{A}$ does not satisfy C), then $\Delta_{\mathcal{F}}(C|\mathcal{A}) = 0$.

Proof. $|(C|\mathcal{A}) - C| = 0$. $\square$

Lemma 2.9. Let D be a clause of $\mathcal{F}$, let $\mathcal{A}$ be a partial assignment, and let $\mathcal{G} = \mathcal{F}|\mathcal{A}$. If $D|\mathcal{A} \neq \top$ (i.e., $\mathcal{A}$ does not satisfy D), then $\Delta_{\mathcal{F}}(D) \leq \Delta_{\mathcal{G}}(D|\mathcal{A}) + |\mathcal{A}|$.

Proof. Suppose $C|\mathcal{A} \in \mathcal{G}$ is a clause for which $\Delta_{\mathcal{G}}(D|\mathcal{A}) = |(D|\mathcal{A}) - (C|\mathcal{A})|$. Then $|D - C| \leq |D - (C|\mathcal{A})| \leq |(D|\mathcal{A}) - (C|\mathcal{A})| + |\mathcal{A}|$. $\square$

3 Size vs. Input Distance Relationships

Ben-Sasson and Wigderson [3] derived size-width relationships that they describe as a "direct translation of [CEI96] to resolution derivations." Their informal statement, "if $\mathcal{F}$ has a *short* resolution refutation then it has a refutation with a small *width*," applies only when $\mathcal{F}$ has no wide clauses.

This section shows that by using input distance rather than clause width, the restriction on the width of $\mathcal{F}$ can be removed. That is, the relationships are strengthened by removing the additive term, $\mathit{width}(\mathcal{F})$.

The use of strengthening for recursive construction of refutations with special properties originates with Anderson and Bledsoe [2], who used it as a uniform framework for showing completeness of various restrictions on resolution, including linear resolution, set-of-support strategy, positive resolution, and others. Clegg *et al.* [5] used it in connection with Groebner-basis refutations. Ben-Sasson and Wigderson [3] used it to construct resolution refutations of small width. We use it here to construct resolution refutations of small input distance, closely following Ben-Sasson and Wigderson.

Lemma 3.1. Given formula $\mathcal{F}$, and a strengthening literal p, let $\mathcal{G} = \mathcal{F}|p$. If derivation π_1 derives clause D from $\mathcal{G}$ with input distance $\Delta_{\mathcal{G}}(\pi_1) = (d-1)$, then there is a derivation π_2 that derives $(D + \neg p)^-$ from $\mathcal{F}$ with input distance $\Delta_{\mathcal{F}}(\pi_2) \leq d$.

Proof. Since $\mathcal{G}$ contains neither p nor $\neg p$, we can assume w.l.o.g. that no vertices of π_1 have p or $\neg p$ as the clashing literal. Define π_2 to have the same DAG structure as π_1, and the same clashing literal at each vertex, but wherever a leaf of π_1 is labeled with $C|p$, label the corresponding leaf of π_2 with C. Each clause of $\mathcal{F}$ has at most one additional literal, $\neg p$, compared to the corresponding clause of $\mathcal{G}$, or else contains p. But no clauses of $\mathcal{F}$ containing p are leaves of π_2. Complete the clause labeling of π_2 according to the definition of resolution. Clearly π_2 derives $(D + \neg p)^-$. For each clause E in π_2, the corresponding clause in π_1 is $E|p$. By Lemma 2.9, $\Delta_{\mathcal{F}}(E) \leq \Delta_{\mathcal{G}}(E|p) + 1$. So $\Delta_{\mathcal{F}}(\pi_2) \leq d$. □

Lemma 3.2. Given formula $\mathcal{F}$, and a strengthening literal p, let $\mathcal{G} = \mathcal{F}|p$ and $\mathcal{H} = \mathcal{F}|\neg p$. If derivation π_1 derives $\perp$ from $\mathcal{G}$ with input distance $\Delta_{\mathcal{G}}(\pi_1) = d-1$, and derivation π_2 derives $\perp$ from $\mathcal{H}$ with input distance $\Delta_{\mathcal{H}}(\pi_2) = d$, then there is a derivation π_3 that derives $\perp$ from $\mathcal{F}$ with input distance $\Delta_{\mathcal{F}}(\pi_3) \leq d$.

Proof. Using Lemma 3.1, there is a derivation π_4 that derives $[\neg p]^-$ from $\mathcal{G}$ with input distance $\Delta_{\mathcal{F}}(\pi_4) \leq d$. If the root of π_4 is $\perp$, let $\pi_3 = \pi_4$ and we are done. Otherwise, construct π_3 as follows:

1. Use π_4 as the initial part of π_3. This part of π_3 has input distance at most d from $\mathcal{F}$.
2. Resolve every clause of $\mathcal{F}$ that contains p with the root of π_4, which contains $[\neg p]$. Call this set of resolvents $\mathcal{F}_1$. All of these resolvents have input distance 0 from $\mathcal{F}$ (Lemma 2.8), so they do not contribute to $\Delta_{\mathcal{F}}(\pi_3)$; also, they and are in $\mathcal{H}$.

3. Let $\mathcal{F}_2$ consist of those clauses in $\mathcal{F}$ that contain neither $\neg p$ nor p. Note that $\mathcal{F}_1 + \mathcal{F}_2 = \mathcal{H}$.
4. Complete the derivation π_3 according to the derivation π_2, using clauses from $\mathcal{F}_1$ and $\mathcal{F}_2$ in place of $\mathcal{H}$ at the leaves of π_2. Since $|D - C| \leq |(D - C|\neg p)|$ for any clauses, C, D, this part of π_3 has input distance at most d from $\mathcal{F}$.

Thus $\Delta_{\mathcal{F}}(\pi_3) \leq d$. □

Theorem 3.3. Let $\mathcal{F}$ be an unsatisfiable formula on $n \geq 1$ variables and let $d \geq 0$ be an integer. Let S_T be the size of the shortest tree-like refutation of $\mathcal{F}$. If $S_T \leq 2^d$, then $\mathcal{F}$ has a refutation π with input distance $\Delta_{\mathcal{F}}(\pi) \leq d$.

Proof. The proof is by induction on the pair (n, d) with the component-wise partial order, and follows Ben-Sasson and Wigderson [3], except that it uses input distance and Lemma 3.2 above. The bases cases are $d = 0$ or $n = 1$, and are immediate. For $d > 0$ and $n > 1$ assume the theorem holds for smaller pairs. Let x be the clashing literal at the root of π, a shortest tree-like refutation of $\mathcal{F}$. The children of the root are themselves the roots of tree-like derivations of x and $\neg x$; call them π_1 and π_0. Assume the size of π_1 is at most 2^{d-1}. But $\pi_1|\neg x$ is a tree-like refutation of $\mathcal{G} = \mathcal{F}|\neg x$. By the inductive hypothesis, $\mathcal{G}$ has a refutation π_2 with input distance $\Delta_{\mathcal{G}}(\pi_2) \leq d - 1$. Also, $\mathcal{H} = \mathcal{F}|x$ has at most $n - 1$ variables, so by the inductive hypothesis, $\mathcal{H}$ has a refutation with input distance $\Delta_{\mathcal{H}}(\pi_1) \leq d$. By Lemma 3.2, $\mathcal{F}$ has a refutation π with input distance $\Delta_{\mathcal{F}}(\pi) \leq d$. □

Corollary 3.4. $S_T(\mathcal{F}) \geq 2^{\Delta(\mathcal{F} \vdash \bot)}$.

Theorem 3.5. Let $\mathcal{F}$ be an unsatisfiable formula on $n \geq 1$ variables and let $d \geq 0$ be an integer. Let $S(\mathcal{F})$ be the size of the shortest refutation of $\mathcal{F}$. If $S(\mathcal{F}) \leq e^{\frac{d^2}{8n}}$, then $\mathcal{F}$ has a refutation π_1 with $\Delta_{\mathcal{F}}(\pi_1) \leq d$.

Proof. The proof is by induction on the pair (n, d) with the component-wise partial order, and follows Ben-Sasson and Wigderson [3], except that it uses input distance and Lemma 3.2 above. Their local variable d is renamed to f here and denotes the input distance that causes a clause to be classified as *fat*; $f = \lceil\sqrt{2n \ln S(\mathcal{F})}\rceil$. For any derivation π, let π^* be the set of clauses $D \in \pi$ with $\Delta_{\mathcal{F}}(D) > f$. Define $a = 2n/(2n - f)$. The theorem follows from this claim:

Claim: For all $b \geq 0$ and $1 \leq m \leq n$, if formula $\mathcal{G}$ has m variables and π is a refutation of $\mathcal{G}$ and $|\pi^*| < a^b$, then $\Delta(\mathcal{G} \vdash \bot) \leq f + b$.

Setting $b = f$ and $\mathcal{G} = \mathcal{F}$, and using the identity $-\ln(1 - f/2n) > f/2n$, ensures that $a^b \geq S(\mathcal{F})$, so ensures the hypothesis, $|\pi^*| < a^b$, is true. Setting $d = 2f$ proves the theorem.

The claim is proved by induction on on the pair (m, b) with the component-wise partial order. The base cases are $b = 0$ or $m = 1$, for which the claim is immediate, as $|\pi^*| = 0$. For $b > 0$ and $m > 1$, there is some literal x that appears in at least $|\pi^*| f/2n$ clauses of π^*. Let $\pi|x$ be as defined in Definition 2.4. By

Lemma 2.7 and the discussion following it, there is a $\top$-free derivation π_1 with the same nontautologous clauses as $\pi|x$. Then $|\pi_1^*| \leq (1 - f/2n)|\pi^*| \leq a^{b-1}$. But π_1 refutes $\mathcal{G}|x$, so by the inductive hypothesis, $\Delta(\mathcal{G}|x \vdash \bot) \leq f + b - 1$. Let π_0 be the $\top$-free version of $\pi|\neg x$, which refutes $\mathcal{G}|\neg x$. Since $\mathcal{G}|\neg x$ has fewer than m variables and $|\pi_0^*| \leq a^b$, by the inductive hypothesis, $\Delta(\mathcal{G}|\neg x \vdash \bot) \leq f + b$. Applying Lemma 3.2 proves the claim. $\square$

Corollary 3.6. $S(\mathcal{F}) \geq e^{\frac{\Delta(\mathcal{F} \vdash \bot)^2}{8n}}$.

4 Pigeon-Hole Formulas

The well-known family of Pigeon-Hole formulas for m pigeons and n holes ($\mathrm{PHP}(m, n)$) is defined by these clauses:

$$\begin{aligned} C_i &= [x_{i,1}, \ldots, x_{i,n}] && \text{for } 1 \leq i \leq m \\ B_{ijk} &= [\neg x_{i,k}, \neg x_{j,k}] && \text{for } 1 \leq i \leq m,\ 1 \leq j \leq m,\ 1 \leq k \leq n. \end{aligned}$$

For the standard version, $m = n + 1$. We shall show that any refutation of $\mathrm{PHP}(m, n)$ with $m > n$ has input distance $\Omega(n)$. An (already known) exponential lower bound for tree-like refutations follows by Corollary 3.4, but no useful lower bound for general refutations follows by Corollary 3.6, since the "n" in that corollary is the number of variables, which is nm in the notation of this section. The method follows Ben-Sasson and Wigderson [3], except that it uses input distance and the original PHP clauses of width n.

Theorem 4.1. Any refutation of $\mathrm{PHP}(m, n)$ with $m > n$ has input distance at least $n/3 - 2$.

Proof. For $1 \leq i \leq m$, define

$$\mathcal{A}_i = \{C_i, B_{ijk},\ 1 \leq j \leq m, 1 \leq k \leq n\}$$

which consists of all the constraints on pigeon i. Define $\mu(D)$, the *complexity* of a clause D, as the minimum number of $\mathcal{A}_i$'s needed to logically imply D. Then $\mu(\bot) = n + 1$ and $\mu(C) = 1$ where C is any input clause. Suppose I is the index set for a minimum-cardinality set of $\mathcal{A}_i$'s that imply D and $n/3 \leq |I| < 2n/3$. That is,

$$\Big(\bigwedge_{i \in I} \mathcal{A}_i\Big) \rightarrow D \tag{5}$$

is a tautology. Such an I must exist, because $\mu(\mathbf{res}(q, D_1, D_2)) \leq \mu(D_1) + \mu(D_2)$.

Equation (5) holds if and only if the following is unsatisfiable (note that $\neg(D)$ constitutes a set of unit clauses):

$$\Big(\bigwedge_{i \in I} \mathcal{A}_i\Big) \wedge \neg(D) \tag{6}$$

Let P_0 be the set of pigeons (first index of variables) that have negative literals in D; let P_1 be the set of pigeons that have positive literals in D. If D has at least $n/3$ negative literals, then its input distance is at least $n/3-2$; assume this is not the case. Therefore, $I - P_0$ is nonempty.

The plan of the proof is to show that, if $I - P_0$ is nonempty and D has fewer than $n/3$ negative literals, either there is an assignment that satisfies (6) or P_1 has at least $n/3 - 1$ pigeons. Table 2 illustrates some of the notation.

Since $I - P_0$ is nonempty, let $p \in I - P_0$ and let $I^* = I - \{p\}$. Thus p is some pigeon whose hole is not forced by $\neg(D)$. By the minimality of I there is an assignment σ that makes $\neg(D)$ true, makes $\mathcal{A}_p$ false and satisfies $\mathcal{A}_i$ for $i \in I^*$. W.l.o.g. let σ be chosen to have as few positive literals as possible. Then σ sets all $x_{ik} = 0$ for i not in $I \cup P_0 \cup P_1$ and $1 \leq k \leq n$. Further, σ sets all $x_{pk} = 0$, $1 \leq k \leq n$, since none of these positive literals occur in $\neg(D)$. Choose a function $k(i)$ for $i \in I^*$ such that $x_{i,k(i)} = 1$ in σ. Necessarily, $k(i_1) \neq k(i_2)$ for distinct $i_1, i_2 \in I^*$. Let K be the set of indexes in the range 1 through n that are *not* in the range of $k(i)$; $|K| \geq n/3 + 2$. These are the holes that are available for pigeon p.

Recall that σ sets $x_{pk} = 0$ for all $k \in K$. Also, σ sets $x_{ik} = 1$ for $i \in I^*$ and $k \neq k(i)$ only if $x_{ik} \in \neg(D)$. If, for any $k \in K$, x_{pk} can be flipped to 1 and x_{ik} can be set to 0 for all $i \neq p$ without falsifying $\neg(D)$, that would create a satisfying assignment for (6). Therefore, for each $k \in K$, $\neg(D)$ contains $\neg x_{pk}$ or x_{ik} for some $i \neq p$. Since $|K| > n/3$ and we assumed D has fewer than $n/3$ negative literals, it must be the case that $\neg(D)$ contains $\neg x_{pk}$ for some $k \in K$.

Finally, we argue that since $\neg(D)$ contains $\neg x_{pk}$, for some $k \in K$, it must contain $\neg x_{ik}$ for all $i \in I^*$. Suppose this fails for some i. Then modify σ by setting $x_{p,k(i)} = 1$, $x_{p,k} = 0$, $x_{i,k(i)} = 0$, and $x_{i,k} = 1$ (see Table 2). This produces a satisfying assignment for (6).

To summarize, if $\neg(D)$ contains $\neg x_{pk}$ for some $k \in K$, then D contains positive literals for at least $n/3 - 1$ different pigeons, i.e.,$|P_1| \geq n/3 - 1$, giving D an input distance of at least $n/3 - 2$. $\square$

Table 2. Changing σ to expose a faulty index set I, in proof of theorem.

$D = [\neg x_{11}, x_{32}, x_{52}]$, $\quad \neg(D) = [x_{11}] \wedge [\neg x_{32}] \wedge [\neg x_{52}]$, $\quad I = \{2, 3, 5\}$, $\quad I^* = \{2, 3\}$.

Original σ

pige–ons	*holes* 1	2	3	4
1	1	0	0	0
2	0	0	1	0
3	0	0	0	1
4	0	0	0	0
$p = 5$	0	0	0	0
6	0	0	0	0

i	$k(i)$
2	3
3	4

$K = \{1, 2\}$

Modified σ

pige–ons	*holes* 1	2	3	4
1	1	0	0	0
2	0	***1***	***0***	0
3	0	0	0	1
4	0	0	0	0
$p = 5$	0	***0***	***1***	0
6	0	0	0	0

5 Conclusion

We proposed the *input distance* metric as a refinement of clause width for studying the complexity of resolution. For families with wide clauses, the trade-off between resolution refutation size and input distance is sharper than the trade-off between resolution refutation size and clause width.

We showed that any refutation of $\mathrm{PHP}(m, n)$ requires input distance at least $n/3 - 2$. Moreover, the proof showed that this input distance can arise in two possible ways: by having $n/3$ negative literals in a derived clause, or by having $n/3 - 1$ positive literals *that refer to distinct pigeons.*

We conjecture that a similar exercise can be carried out for the family called $\mathrm{GT}(n)$ [6], which has general refutations of polynomial size [8], but for which tree-like refutations are exponential [4]. This family can be modified so that regular refutations are also exponential [1]. Bonet and Galesi introduced a bounded-width variant called $\mathrm{MGT}(n)$, and showed that refutations of $\mathrm{MGT}(n)$ have width $\Omega(n)$ [4]. However, the complexity function they used does not transfer straightforwardly to a lower bound on input distance, as there are clauses with input distance zero and complexity between $n/3$ and $2n/3$.

Some open problems remain. Can input distance improve the lower bound for *weak* $\mathrm{PHP}(m, n)$, where $m >> n$? Ben-Sasson and Wigderson [3] transformed this problem into a family with clause width proportional to $\log m$. Are there other natural families to which input distance can be applied? Is there a trade-off between regular refutation size and input distance?

References

1. Alekhnovich, M., Johannsen, J., Pitassi, T., Urquhart, A.: An exponential separation between regular and unrestricted resolution. In: Proc. 34th ACM Symposium on Theory of Computing. (2002) 448–456
2. Anderson, R., Bledsoe, W.W.: A linear format for resolution with merging and a new technique for establishing completeness. Journal of the ACM **17** (1970) 525–534
3. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow — resolution made simple. JACM **48** (2001) 149–168
4. Bonet, M., Galesi, N.: A study of proof search algorithms for resolution and polynomial calculus. In: Proc. 40th Symposium on Foundations of Computer Science. (1999) 422–432
5. Clegg, M., Edmonds, J., Impagliazzo, R.: Using the Groebner basis algorithm to find proofs of unsatisfiability. In: Proc. 28th ACM Symposium on Theory of Computing. (1996) 174–183
6. Krishnamurthy, B.: Short proofs for tricky formulas. Acta Informatica **22** (1985) 253–274
7. Letz, R., Mayr, K., Goller, C.: Controlled integration of the cut rule into connection tableau calculi. Journal of Automated Reasoning **13** (1994) 297–337
8. Stälmarck, G.: Short resolution proofs for a sequence of tricky formulas. Acta Informatica **33** (1996) 277–280