DRAT and Propagation Redundancy Proofs Without New Variables*

Sam Buss[†]
Department of Mathematics
University of California, San Diego
La Jolla, CA 92093-0112, USA
sbuss@ucsd.edu

Neil Thapen[‡]
Institute of Mathematics
Czech Academy of Sciences
Prague, Czech Republic
thapen@math.cas.cz

February 16, 2021

Abstract

We study the complexity of a range of propositional proof systems which allow inference rules of the form: from a set of clauses Γ derive the set of clauses $\Gamma \cup \{C\}$ where, due to some syntactic condition, $\Gamma \cup \{C\}$ is satisfiable if Γ is, but where Γ does not necessarily imply C. These inference rules include BC, RAT, SPR and PR (respectively short for *blocked clauses*, *resolution asymmetric tautologies*, *subset propagation redundancy* and *propagation redundancy*), which arose from work in satisfiability (SAT) solving. We introduce a new, more general rule SR (*substitution redundancy*).

If the new clause C is allowed to include new variables then the systems based on these rules are all equivalent to extended resolution. We focus on restricted systems that do not allow new variables. The systems with deletion, where we can delete a clause from our set at any time, are denoted DBC$^-$, DRAT$^-$, DSPR$^-$, DPR$^-$ and DSR$^-$. The systems without deletion are BC$^-$, RAT$^-$, SPR$^-$, PR$^-$ and SR$^-$.

With deletion, we show that DRAT$^-$, DSPR$^-$ and DPR$^-$ are equivalent. By earlier work of Kiesl, Rebola-Pardo and Heule [24], they are also equivalent to DBC$^-$. Without deletion, we show that SPR$^-$ can simulate PR$^-$

*A preliminary version [8] of this paper appeared in the Proceedings of the 2019 Conference on Theory and Applications of Satisfiability Testing (SAT).

[†]This work was initiated on a visit of the first author to the Czech Academy of Sciences in July 2018, supported by ERC advanced grant 339691 (FEALORA). The first author was also supported by Simons Foundation grant 578919.

[‡]Partially supported by GA ČR project 19-05497S. The Institute of Mathematics of the Czech Academy of Sciences is supported by RVO:67985840.

provided only short clauses are inferred by SPR inferences. We also show that many of the well-known "hard" principles have small SPR^- refutations. These include the pigeonhole principle, bit pigeonhole principle, parity principle, Tseitin tautologies and clique-coloring tautologies. SPR^- can also handle or-fication and xor-ification, and lifting with an index gadget. Our final result is an exponential size lower bound for RAT^- refutations, giving exponential separations between RAT^- and both DRAT^- and SPR^-.

1 Introduction

SAT solvers are routinely used for a range of large-scale instances of satisfiability. It is widely realized that when a solver reports that a SAT instance Γ is unsatisfiable, it should also produce a *proof* that it is unsatisfiable. This is of particular importance as SAT solvers become increasingly complex, combining many techniques, and thus are more subject to software bugs or even design problems.

The first proof systems proposed for SAT solvers were based on reverse unit propagation (RUP, or $\vdash_1$ in the notation of this paper) inferences [14, 44] as this is sufficient to handle both resolution inferences and the usual CDCL clause learning schemes. However, RUP inferences only support logical implication, and in particular do not accommodate many "inprocessing" rules. Inprocessing rules support inferences which do not respect logical implication; instead they only guarantee *equisatisfiability* where the (un)satisfiability of the set of clauses is preserved [23]. Inprocessing inferences have been formalized in terms of sophisticated inference rules including DRAT (*deletion, reverse asymmetric tautology*), PR (*propagation redundancy*), SPR (*subset* PR) in a series of papers including [23, 18, 17, 45] — see Section 1.2 for definitions. These inference rules can be viewed as introducing clauses that hold "without loss of generality" [38], and thus preserve (un)satisfiability. An important feature of these systems is that they can be used both as proof systems to verify unsatisfiability, and as inference systems to facilitate searching for either a satisfying assignment or a proof of unsatisfiability.[1]

The DRAT system is very powerful as it can simulate extended resolution [29, 24]. This simulation is straightforward, but depends on DRAT's ability to introduce new variables; we simply show that the usual extension axioms are RAT. However, there are a number of results [16, 21, 19, 20] indicating that DRAT and PR are still powerful when restricted to use few new variables, or even no new variables. In particular, [21, 19, 20] showed that the pigeonhole principle clauses have short

[1]The deletion rule is very helpful to improve proof search and can extend the power of the inferences rules, see Corollary 5.5; however, it must be used carefully to preserve equisatisfiabity. The present paper only considers *refutation systems*, and thus the deletion rule can be used without restriction.

(polynomial size) refutations in the PR proof system. The paper [21] showed that Satisfaction Driven Clause Learning (SDCL) can discover PR proofs of the pigeonhole principle automatically; in the application studied by [21], the SDCL search appears to have exponential runtime, but is much more efficient than the usual CDCL search. There are at present no broadly applicable proof search heuristics for how to usefully introduce new variables with the extension rule. It is possible however that there are useful heuristics for searching for proofs that do not use new variables in DRAT and PR and related systems. For these reasons, DRAT and PR and related systems (even when new variables are not allowed) hold the potential for substantial improvements in the power of SAT solvers.

The present paper extends the theoretical knowledge of these proof systems viewed as refutation systems. We pay particular attention to proof systems that do not allow new variables. The remainder of Section 1 introduces the proof systems BC (*blocked clauses*), RAT, SPR, PR and SR (*substitution redundancy*). (Only SR is new to this paper.) These systems have variants which allow deletion, called DBC, DRAT, DSPR, DPR and DSR. There are also variants of all these systems restricted to not allow new variables: we denote these with a superscript "$-$" as BC^-, DBC^-, RAT^-, DRAT^-, etc.

Section 2 studies the relation between these systems and extended resolution. We show in particular that any proof system containing BC^- and closed under restrictions simulates extended resolution. Here a proof system $\mathcal{P}$ is said to *simulate* a proof system $\mathcal{Q}$ if any $\mathcal{Q}$-proof can be converted, in polynomial time, into a $\mathcal{P}$-proof of the same result. Two systems are *equivalent* if they simulate each other; otherwise they are *separated*. We also show that the systems discussed above all have equivalent canonical NP pairs (a coarser notion of equivalence).

Section 3 extends known results that DBC^- simulates DRAT^- [24] and that DRAT, limited to only one extra variable, simulates DPR^- [16]. Theorem 3.3 proves that DRAT^- simulates DPR^-. As a consequence, DBC^- can also simulate DPR^-. We then give a partial simulation of PR^- by SPR^- — our size bound is exponential in the size of the "discrepancy" of the PR inferences, but in many cases, the discrepancy will be logarithmic or even smaller.

Section 4 proves new polynomial upper bounds on the size of SPR^- proofs for many of the "hard" tautologies from proof complexity. (Recall that SPR^- allows neither deletion nor the use of new variables.) These include the pigeonhole principle, the bit pigeonhole principle, the parity principle, the clique-coloring principle, and the Tseitin tautologies. We also show that obfuscation by or-fication, xor-ification and lifting with a indexing gadget do not work against SPR^-. Prior results gave SPR^- proofs for the pigeonhole principle (PHP) [19, 20], and PR^- proofs for the Tseitin tautologies and the 2-1 PHP [16]. These results raise the question of whether SPR^- (with no new variables!) can simulate Frege systems, for instance. Some possible principles that might separate SPR$^-$ from Frege systems are the graph PHP principle, 3-XOR tautologies and the even coloring principle; these are discussed at the end of Section 4. However, the even coloring principle does have short DSPR$^-$ proofs, and it is plausible that the graph PHP principle has short SPR$^-$ proofs.

Section 5 shows that RAT$^-$ (with neither new variables nor deletion) cannot simulate either DRAT$^-$ (without new variables, but with deletion) or SPR$^-$ (with neither new variables nor deletion). This follows from a size lower bound for RAT$^-$ proofs of the bit pigeonhole principle (BPHP). We first prove a width lower bound, by showing that any RAT inference in a small-width refutation of BPHP can be replaced with a small-width resolution derivation, and then derive the size bound. We use that BPHP behaves well when the sign of a variable is flipped.

The known relationships between these systems, including our results, are summarized in the diagram below. Recall that e.g. BC is the full system, DBC$^-$ is the system with deletion but no new variables, and BC$^-$ is the system with neither deletion nor new variables. An arrow shows that the upper system simulates the lower one. Equivalence $\equiv$ indicates that the systems simulate each other.

$$\begin{tikzcd}
& \mathrm{ER} \equiv \mathrm{SR} \equiv \mathrm{PR} \equiv \mathrm{SPR} \equiv \mathrm{RAT} \equiv \mathrm{BC} \arrow[d] & \\
& \mathrm{DSR}^- \arrow[dl] \arrow[dr] & \\
\mathrm{DPR}^- \equiv \mathrm{DSPR}^- \equiv \mathrm{DRAT}^- \equiv \mathrm{DBC}^- \arrow[dr] & & \mathrm{SR}^- \arrow[dl] \\
& \mathrm{PR}^- \arrow[d, "*"] & \\
& \mathrm{SPR}^- \arrow[d, "\not\equiv"] & \\
& \mathrm{RAT}^- \arrow[d] & \\
& \mathrm{BC}^- \arrow[d, "\not\equiv"] & \\
& \mathrm{Res} &
\end{tikzcd}$$

The arrow from PR$^-$ and SPR$^-$ is marked $*$ to indicate that there is a simulation in the other direction under the additional assumption that the discrepancies (see Definition 3.9) of PR inferences are logarithmically bounded.

We summarize the rules underlying these systems in a table. The details and the necessary definitions are in Section 1.2 below – in particular see Theorem 1.10 for this definition of RAT.

BC	(a restriction of RAT)	*blocked clause*
RAT	τ is α with one variable flipped	*reverse asymmetric tautology*
SPR	τ is a partial assignment, $dom(\tau) = dom(\alpha)$	*subset propagation redundant*
PR	τ is a partial assignment	*propagation redundant*
SR	no extra conditions	*substitution redundant*

As presented here the rules (except for BC) have the form: derive C from Γ, if there is a substitution τ satisfying $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ plus the conditions shown, where α is $\overline{C}$. The implication $\vdash_1$ is defined below in terms of reverse unit propagation (RUP).

We remark that the question of whether new variables help reasoning with blocked clause inferences was already studied by Kullmann in the context of the system Generalized Extended Resolution (GER) [29]. As far as we know, GER does not correspond exactly to any of the systems we consider. [29] showed that allowing new variables does not reduce GER proof length when the blocked clause rule is restricted to introducing clauses of length at most two.

1.1 Preliminaries

We use the usual conventions for clauses, variables, literals, truth assignments, satisfaction, etc. *Var* and *Lit* denote the sets of all variables and all literals. A set of literals is called *tautological* if it contains a pair of complementary literals p and $\overline{p}$. A *clause* is a non-tautological[2] set of literals; we use $C, D, \ldots$ to denote clauses. The empty clause is denoted $\bot$, and is always false. 0 and 1 denote respectively *False* and *True*; and $\overline{0}$ and $\overline{1}$ are respectively 1 and 0. We use both $C \cup D$ or $C \vee D$ to denote unions of clauses, but usually write $C \vee D$ when the union is a clause. The notation $C = D \dot{\vee} E$ indicates that $C = D \vee E$ is a clause and D and E have no variables in common. If Γ is a set of clauses, $C \vee \Gamma$ is the set $\{C \vee D : D \in \Gamma \text{ and } C \vee D \text{ is a clause}\}$.

A *partial assignment* τ is a mapping with domain a set of variables and range contained in $\{0, 1\}$. It acts on literals by letting $\tau(\overline{p}) = \overline{\tau(p)}$. It is called a *total assignment* if it sets all variables. We sometimes identify a partial assignment τ with the set of unit clauses asserting that τ holds. For C a clause, $\overline{C}$ denotes the partial assignment whose domain is the variables of C and which asserts that C is false. For example, if $C = x \vee \overline{y} \vee z$ then, depending on context, $\overline{C}$ will denote either the set containing the three unit clauses $\overline{x}$ and y and $\overline{z}$, or the partial assignment α with domain $dom(\alpha) = \{x, y, z\}$ such that $\alpha(x) = 0$, $\alpha(y) = 1$ and $\alpha(z) = 0$.

A *substitution* generalizes the notion of a partial assignment by allowing variables to be mapped also to literals. Formally, a substitution σ is a map from $Var \cup$

[2]Disallowing tautological clauses makes the rest of the definitions more natural. In particular, we can identify clauses with the negations of partial assignments.

$\{0,1\}$ to $Lit \cup \{0,1\}$ which is the identity on $\{0,1\}$. Note that a substitution may cause different literals to become identified.[3] A partial assignment τ can be viewed as a substitution, by defining $\tau(x) = x$ for all variables x outside the domain of τ. The *domain* of a substitution σ is the set of variables x for which $\sigma(x) \neq x$.

Suppose C is a clause and σ is a substitution (or a partial assigment viewed as a substitution). Let $\sigma(C) = \{\sigma(p) : p \in C\}$. We say σ *satisfies* C, written $\sigma \vDash C$, if $1 \in \sigma(C)$ or $\sigma(C)$ is tautological. When $\sigma \nvDash C$, the *restriction* $C_{|\sigma}$ is defined by letting $C_{|\sigma}$ equal $\sigma(C) \setminus \{0\}$. Thus $C_{|\sigma}$ is a clause expressing the meaning of C under σ. For Γ a set of clauses, the restriction of Γ under σ is

$$\Gamma_{|\sigma} \;=\; \{\, C_{|\sigma} :\; C \in \Gamma \text{ and } \sigma \nvDash C \,\}.$$

The composition of two substitutions is denoted $\tau \circ \pi$, meaning that $(\tau \circ \pi)(x) = \tau(\pi(x))$, and in particular $(\tau \circ \pi)(x) = \pi(x)$ if $\pi(x) \in \{0,1\}$. For partial assignments τ and π, this means that $dom(\tau \circ \pi) = dom(\tau) \cup dom(\pi)$ and

$$(\tau \circ \pi)(x) \;=\; \begin{cases} \pi(x) & \text{if } x \in dom(\pi) \\ \tau(x) & \text{if } x \in dom(\tau) \setminus dom(\pi). \end{cases}$$

Lemma 1.1. *For a set of clauses Γ and substitutions τ and π, $\Gamma_{|\tau \circ \pi} = (\Gamma_{|\pi})_{|\tau}$. In particular, $\tau \vDash \Gamma_{|\pi}$ if and only if $\tau \circ \pi \vDash \Gamma$.*

Proof. Notice $\tau \circ \pi \vDash C$ if and only if $\pi \vDash C$ or $(\pi \nvDash C \wedge \tau \vDash C_{|\pi})$. Thus

$$\begin{aligned} (\Gamma_{|\pi})_{|\tau} &= \left\{ (C_{|\pi})_{|\tau} : C \in \Gamma,\ \pi \nvDash C,\ \tau \nvDash C_{|\pi} \right\} \\ &= \left\{ \tau \circ \pi(C) \setminus \{0\} : C \in \Gamma,\ \pi \nvDash C,\ \tau \nvDash C_{|\pi} \right\} \\ &= \left\{ C_{|\tau \circ \pi} : C \in \Gamma,\ \tau \circ \pi \nvDash C \right\} = \Gamma_{|\tau \circ \pi}. \end{aligned}$$ □

A set of clauses Γ *semantically implies* a clause C, written $\Gamma \vDash C$, if every total assignment satisfying Γ also satisfies C. As is well-known, $\Gamma \vDash C$ holds if and only if there is a *resolution derivation* of some $C' \subseteq C$; that is, C' is derived from Γ using *resolution inferences* of the form

$$\frac{p \,\dot{\vee}\, D \qquad \overline{p} \,\dot{\vee}\, E}{D \vee E}. \tag{1}$$

If the derived clause C' is the empty clause $\bot$, then the derivation is called a *resolution refutation* of Γ. By the soundness and completeness of resolution, $\Gamma \vDash \bot$, that is, Γ is unsatisfiable, if and only if there is a resolution refutation of Γ.

[3][40] defined a notion of "homomorphisms" that is similar to substitutions. Substitutions, however, allow variables to be mapped also to constants. Our SR inference, defined below, uses $\vdash_1$; this was not used with homomorphisms in [40].

If either D or E is empty, then the resolution inference (1) is an instance of *unit propagation*. A refutation using only such inferences is called a *unit propagation refutation*. Recall that we can write $\overline{C}$ for the set of unit clauses $\{\overline{p} : p \in C\}$.

Definition 1.2. We write $\Gamma \vdash_1 \bot$ to denote that there is a unit propagation refutation of Γ. We define $\Gamma \vdash_1 C$ to mean $\Gamma \cup \overline{C} \vdash_1 \bot$. For a set of clauses Δ, we write $\Gamma \vdash_1 \Delta$ to mean $\Gamma \vdash_1 C$ for every $C \in \Delta$.

Fact 1.3. *If $\Gamma \vdash_1 \bot$ and α is any partial assignment or substitution, then $\Gamma_{|\alpha} \vdash_1 \bot$.*

In the literature, when $\Gamma \vdash_1 C$ then C is said to be derivable from Γ by *reverse unit propagation* (RUP), or is called an *asymmetric tautology* (AT) with respect to Γ [44, 23, 18]. Of course, $\Gamma \vdash_1 C$ implies that $\Gamma \vDash C$. The advantage of working with $\vdash_1$ is that there is a simple polynomial time algorithm to determine whether $\Gamma \vdash_1 C$. We have the following basic property of $\vdash_1$ (going back to [10]):

Lemma 1.4. *If C is derivable from Γ by a single resolution inference, then $\Gamma \vdash_1 C$. Conversely, if $\Gamma \vdash_1 C$, then some $C' \subseteq C$ has a resolution derivation from Γ of length at most n, where n is the total number of literals occurring in clauses in Γ.*

Proof. First suppose that $C = D \vee E$ and clauses $p \dot\vee D$ and $\overline{p} \dot\vee E$ appear in Γ. Then by resolving these with the unit clauses in $\overline{C}$ we can derive the two unit clauses p and $\overline{p}$, then resolve these together to get the empty clause.

Now suppose that $\Gamma \vdash_1 C$. Then there is a unit propagation derivation of $\bot$ from $\Gamma \cup \overline{C}$, which is of length at most n. Removing all resolutions against unit clauses $\overline{p}$ for $p \in C$, this can be turned into a resolution derivation of C or of some $C' \subseteq C$ from Γ. □

Lemma 1.5. *Let $C \vee D$ be a clause (so $C \cup D$ is not tautological), and set $\alpha = \overline{C}$. Then*

$$\Gamma_{|\alpha} \vdash_1 D \setminus C \iff \Gamma_{|\alpha} \vdash_1 D \iff \Gamma \vdash_1 C \vee D.$$

Proof. The left-to-right directions are immediate from the definitions, since $\Gamma_{|\alpha}$ is derivable from $\Gamma \cup \alpha$ using unit propagation. To show that $\Gamma \vdash_1 C \vee D$ implies $\Gamma_{|\alpha} \vdash_1 D \setminus C$, suppose $\Gamma \cup \alpha \cup \overline{D} \vdash_1 \bot$ and apply Fact 1.3. □

1.2 Inference rules

We will describe a series if inference rules which can be used to add a clause C to a set of clauses Γ. In increasing order of strength the rules are

$$\mathrm{BC} \leftarrow \mathrm{RAT} \leftarrow \mathrm{SPR} \leftarrow \mathrm{PR} \leftarrow \mathrm{SR}.$$

We will show that in each case the sets Γ and $\Gamma \cup \{C\}$ are *equisatisfiable*, that is, either they are both satisfiable or both unsatisfiable. The definitions follow [23, 18, 20], except for the new notion SR of "substitution redundancy".[4] All of these rules can be viewed as allowing the introduction of clauses that hold "without loss of generality" [38]. The rules are summarized in a table earlier in this section.

Let Γ be a set of clauses and C a clause with a distinguished literal p, so that C has the form $p \dot{\vee} C'$.

Definition 1.6. ([27, 28]) The clause C is a *blocked clause* (BC) with respect to p and Γ if, for every clause D of the form $\overline{p} \dot{\vee} D'$ in Γ, the set $C' \cup D'$ is tautological.

Notice that the condition "$C' \cup D'$ is tautological" above would be equivalent to $\emptyset \vdash_1 C' \vee D'$, except that our notation does not allow us to write the expression $C' \vee D'$ if $C' \cup D'$ is tautological, since it is not a clause. Since $\overline{p}$ does not appear in C' or D', it would also be equivalent to $\emptyset \vdash_1 p \vee C' \vee D'$. Compare with the definition of RAT below.

Definition 1.7. ([23, 16, 45]) A clause C is a *resolution asymmetric tautology* (RAT) with respect to p and Γ if, for every clause D of the form $\overline{p} \dot{\vee} D'$ in Γ, either $C' \cup D'$ is tautological or

$$\Gamma \vdash_1 p \vee C' \vee D'.$$

Here we write $p \vee C'$ instead of C to emphasize that we include the literal p (some definitions of RAT omit it). Clearly, being BC implies being RAT.

Example 1.8 ([29])**.** Let Γ be a set of clauses in which the variable x does not occur, but the variables p and q may occur. Consider the three clauses

$$x \vee \overline{p} \vee \overline{q} \qquad \overline{x} \vee p \qquad \overline{x} \vee q$$

which together express that $x \leftrightarrow (p \wedge q)$. Let $\Gamma_1 \subset \Gamma_2 \subset \Gamma_3$ be Γ with the three clauses above successively added. Then $x \vee \overline{p} \vee \overline{q}$ is BC with respect to Γ and x, because no clause in Γ contains $\overline{x}$, so there is nothing to check. The second clause $\overline{x} \vee p$ is BC with respect to Γ_1 and $\overline{x}$ because the only clause in Γ_1 containing x is $x \vee \overline{p} \vee \overline{q}$, and resolving this with $\overline{x} \vee p$ gives a tautological conclusion. The third clause $\overline{x} \vee q$ is BC with respect to Γ_2 and $\overline{x}$ in a similar way.

[4] M. Heule [personal communication, 2018] has independently formulated an inference rule "permutation redundancy" (πPR) which allows only substitutions which set some variables to constants and acts as a permutation on the remaining literals. This is a special case of SR; but unlike SR, πPR does not allow identifying distinct literals. However, we do not know the strength of $\pi\mathrm{PR}^-$ relative to SR^- (even if deletion is allowed for both systems).

It follows from the example that we can use the BC rule to simulate extended resolution if we are allowed to introduce new variables; see Section 2.1.

We say the clause C is RAT with respect to Γ if it is RAT with respect to p and Γ for some literal p in C, and similarly for BC.

Theorem 1.9. ([29, 23]) *If C is* BC *or* RAT *with respect to* Γ*, then* Γ *and* $\Gamma \cup \{C\}$ *are equisatisfiable.*

Proof. It suffices to show that if Γ is satisfiable, then so is $\Gamma \cup \{C\}$. Let τ be any total assignment satisfying Γ. We may assume $\tau \vDash \overline{C}$, as otherwise we are done. Let τ' be τ with the value of $\tau(p)$ switched to satisfy p. Then τ' satisfies C, along with every clause in Γ which does not contain $\overline{p}$. Let $D = \overline{p} \dot{\vee} D'$ be any clause in Γ which contains $\overline{p}$. It follows from the RAT assumption that $\Gamma \vDash C \vee D'$, so $\tau \vDash D'$ since $\tau \vDash \overline{C}$. Hence $\tau' \vDash D'$ and thus $\tau' \vDash D$. This shows that $\tau' \vDash \Gamma \cup \{C\}$. $\square$

For the rest of this section, let α be the partial assignment $\overline{C}$. In a moment we will introduce the rules SPR, PR and SR. These are variants of a common form, and we begin by showing that RAT can also be expressed in a similar way (in the literature this form of RAT is called *literal propagation redundant* or LPR).

Theorem 1.10. ([20]) *A clause C is* RAT *with respect to p and* Γ *if and only if* $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ *where τ is the partial assignment identical to α except at p, with* $\tau(p) = 1$.

Proof. First suppose that C satifies the second condition. Consider any clause D of the form $\overline{p} \dot{\vee} D'$ in Γ. We need to show that either $C \cup D'$ is tautological or $\Gamma \vdash_1 C \vee D'$. Suppose $C \cup D'$ is not tautological. Then $\alpha \nvDash D'$, $\tau \nvDash D$, and by Lemma 1.5 it is enough to show $\Gamma_{|\alpha} \vdash_1 D'$. But this now follows from $\Gamma_{|\alpha} \vdash_1 D_{|\tau}$, since $D_{|\tau} = D'_{|\alpha} \subseteq D'$.

Now suppose C is RAT with respect to p and Γ. Consider any $D \in \Gamma$ such that $\tau \nvDash D$ and thus $D_{|\tau} \in \Gamma_{|\tau}$. We must show that $\Gamma_{|\alpha} \vdash_1 D_{|\tau}$. If $\overline{p} \notin D$ this is trivial, since then $D_{|\tau} = D_{|\alpha} \in \Gamma_{|\alpha}$. Otherwise $D = \overline{p} \dot{\vee} D'$, where $\alpha \nvDash D'$ since $\tau \nvDash D$, so $C \cup D'$ is not tautological. By the RAT property, $\Gamma \vdash_1 C \vee D'$. By Lemma 1.5 this implies $\Gamma_{|\alpha} \vdash_1 D' \setminus C$. But $D' \setminus C = D'_{|\alpha} = D_{|\tau}$. $\square$

Definition 1.11. ([20]) A clause C is *subset propagation redundant* (SPR) with respect to Γ if there is a partial assignment τ with $dom(\tau) = dom(\alpha)$ such that $\tau \vDash C$ and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$.

Definition 1.12. ([20]) A clause C is *propagation redundant* (PR) with respect to Γ if there is a partial assignment τ such that $\tau \vDash C$ and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$.

Definition 1.13. A clause C is *substitution redundant* (SR) with respect to Γ if there is a substitution τ such that $\tau \vDash C$ and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$.

Example 1.14 (based on [20]). Let Γ be the pigeonhole principle PHP_n (see Section 4.1) in variables $p_{i,j}$ expressing that pigeon i goes to hole j. Let C be the clause $\overline{p_{1,0}} \vee p_{0,0}$ so that α is the partial assignment $p_{1,0} \wedge \overline{p_{0,0}}$.

Let π be the substitution which swaps pigeons 0 and 1; that is, $\pi(p_{0,j}) = p_{1,j}$ and $\pi(p_{1,j}) = p_{0,j}$ for every hole j, and π is otherwise the identity. Notice that, by the symmetries of the pigeonhole principle, $\Gamma_{|\pi} = \Gamma$ and thus $\Gamma_{|\alpha} = (\Gamma_{|\pi})_{|\alpha} = \Gamma_{|\alpha\circ\pi}$. Let $\tau = \alpha\circ\pi$, so τ is the same as π except that $\tau(p_{0,0}) = 1$ and $\tau(p_{1,0}) = 0$.

Then $\tau \vDash C$ and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ (since they are the same set of clauses). Hence we have shown that C is SR with respect to Γ.

We go on to sketch a polynomial size DSR^- refutation of Γ, that is, one that uses SR inferences, resolution and deletion but introduces no new variables (see Section 1.4 below). Resolve C with the hole axiom $\overline{p_{1,0}} \vee \overline{p_{0,0}}$ to derive the unit clause $\overline{p_{1,0}}$. Delete C, so that we are now working with the set of clauses $\Gamma \cup \{\overline{p_{1,0}}\}$. Let C' be the clause $\overline{p_{2,0}} \vee p_{0,0}$ and let α' be its negation $p_{2,0} \wedge \overline{p_{0,0}}$. Let π' be the substitution which swaps pigeons 0 and 2 and let $\tau' = \alpha'\circ\pi'$. As before $\tau' \vDash C'$ and $(\Gamma \cup \{\overline{p_{1,0}}\})_{|\alpha'} \vdash_1 (\Gamma \cup \{\overline{p_{1,0}}\})_{|\tau'}$, since neither α' nor τ' affects $p_{1,0}$ so these are again the same set of clauses. Hence we may derive C' by a SR inference, then resolve with the hole axiom $\overline{p_{2,0}} \vee \overline{p_{0,0}}$ to get $\overline{p_{2,0}}$.

Carrying on in this way, we eventually derive $\Gamma \cup \{\overline{p_{1,0}}\} \cup \cdots \cup \{\overline{p_{n-1,0}}\}$. We now resolve each unit clause $\overline{p_{i,0}}$ with the pigeon axiom for pigeon i, for $i = 1, \dots, n-1$. After some deletions, we are left with clauses asserting that pigeons $1, \dots, n-1$ map injectively to holes $1, \dots, n-2$. This is essentially PHP_{n-1}. We carry on inductively to derive PHP_{n-2} etc. and can easily derive a contradiction when we get to PHP_2.

Section 4.1 contains a more careful version of this argument, refuting PHP_n using SPR inferences and no deletion.

Theorem 1.15. *If C is* SR *with respect to* Γ*, then* Γ *and* $\Gamma \cup \{C\}$ *are equisatisfiable. Hence the same is true for* SPR *and* PR.

Theorem 1.15 trivially implies the same statement for BC and RAT (this was Theorem 1.9 above).

Proof. Again it is sufficient to show that if Γ is satisfiable, then so is $\Gamma \cup \{C\}$. Suppose we have a substitution τ such that $\tau \vDash C$ and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$. Let π be any total assignment satisfying Γ. If $\pi \vDash C$ then we are done. Otherwise $\pi \vDash \overline{C}$, so $\pi \supseteq \alpha$ and π satisfies $\Gamma_{|\alpha}$ by Lemma 1.1. Thus, by the assumption, $\pi \vDash \Gamma_{|\tau}$. Therefore $\pi\circ\tau \vDash \Gamma$ by Lemma 1.1, and $\pi\circ\tau \vDash C$ since $\tau \vDash C$. $\square$

This proof of Theorem 1.15 still goes through if we replace $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ with the weaker assumption $\Gamma_{|\alpha} \vDash \Gamma_{|\tau}$. The advantage of using $\vdash_1$ is that it is efficiently checkable. Consequently, the conditions of being BC, RAT, SPR, PR or SR with respect to Γ are all polynomial-time checkable, as long as we include the partial assignment or substitution τ as part of the input.

1.3 Proof systems with new variables

This section introduces proof systems based on the BC, RAT, SPR, PR and SR inferences. Some of the systems also allow the use of the deletion rule: these systems are denoted DBC, DRAT, etc. All the proof systems are *refutation systems*. They start with a set of clauses Γ, and successively derive sets Γ_i of clauses, first $\Gamma_0 = \Gamma$, then $\Gamma_1, \Gamma_2, \dots, \Gamma_m$ until reaching a set Γ_m containing the empty clause. It will always be the case that if Γ_i is satisfiable, then Γ_{i+1} is satisfiable. Since the empty clause $\bot$ is in Γ_m, this last set is not satisfiable. This suffices to show that Γ is not satisfiable.

Definition 1.16. A BC, RAT, SPR, PR or SR proof (a refutation) of Γ is a sequence $\Gamma_0, \dots, \Gamma_m$ such that $\Gamma_0 = \Gamma$, $\bot \in \Gamma_m$ and each $\Gamma_{i+1} = \Gamma_i \cup \{C\}$, where either

- $\Gamma_i \vdash_1 C$ or
- C is BC, RAT, SPR, PR, or SR (respectively) with respect to Γ_i.

For BC or RAT steps, the proof must specify which p is used, and for SPR, PR or SR, it must specify which τ.

There is no constraint on the variables that appear in clauses C introduced in BC, RAT, etc. steps. They are free to include new variables that did not occur in $\Gamma_0, \dots, \Gamma_i$.

Definition 1.17. A DBC, DRAT, DSPR, DPR or DSR proof allows the same rules of inference (respectively) as Definition 1.16, plus the *deletion* inference rule:

- $\Gamma_{i+1} = \Gamma_i \setminus \{C\}$ for some $C \in \Gamma_i$.

Resolution can be simulated by RUP inferences (Lemma 1.4), so all the systems introduced in this and the next subsection simulate resolution. Furthermore, by Theorems 1.9 and 1.15, they are sound. Since the inferences are defined using $\vdash_1$, they are polynomial time verifiable, as the description of τ is included with every SPR, PR or SR inference. Hence they are all proof systems in the sense of Cook-Reckhow [12, 13].

The deletion rule deserves more explanation. First, we allow *any* clause to be deleted, even the initial clauses from Γ. So it is possible that Γ_i is unsatisfiable but Γ_{i+1} is satisfiable after a deletion. For us, this is okay since we focus on refuting sets of unsatisfiable clauses, not on finding satisfying assignments of satisfiable sets of clauses. SAT solvers generally wish to maintain the equisatisfiability property: they use deletion extensively to prune the search time, but are careful only to perform deletions that preserve both satisfiability and unsatisfiability, generally as justified by the BC or RAT rules. Since applying RAT, or more generally PR or SR, can change the satisfying assignment, a SAT solver may also need to keep a proof log with information about how to reverse the steps of the proof once a satisfying assignment is found (see [23]).

Second, deletion is important for us because the property of being BC, RAT etc. involves a universal quantification over the current set of clauses Γ_i. So deletion can make the systems more powerful, as removing clauses from Γ_i can make more inferences possible. For example, the unit clause x is BC with respect to the set $\{x \vee y\}$, since the literal $\overline{x}$ does not appear, but is not even SR with respect to the set $\{x \vee y, \overline{x}\}$, since $\{x \vee y, \overline{x}\}$ and $\{x \vee y, \overline{x}, x\}$ are not equisatisfiable. An early paper on this by Kullmann [29] exploited deletions to generalize the power of BC inferences.

As we will show in Section 2, all the systems defined so far are equivalent to extended resolution, because of the ability to freely introduce new variables. The main topic of the paper is the systems we introduce next, which lack this ability.

1.4 Proof systems without new variables

Definition 1.18. A BC refutation of Γ *without new variables*, or, for short, a BC^- refutation of Γ, is a BC refutation of Γ in which only variables from Γ appear. The systems RAT^-, PR^- etc. and DBC^-, DRAT^-, DPR^- etc. are defined similarly.

There is an alternative natural definition of "without new variables", which requires not just that a refutation of Γ uses only variables that are used in Γ, but also that once a variable has been eliminated from all clauses through the use of deletion, it may not be reused subsequently in the refutation. An equivalent way to state this is that a clause C inferred by a BC, RAT, SPR, PR or SR inference cannot involve any variable which does not occur in the *current* set of clauses.

This stronger definition is in fact essentially equivalent to Definition 1.18, for a somewhat trivial reason. More precisely, any refutation that satisfies Definition 1.18 can be converted into a refutation that satisfies the stronger condition with at worst a polynomial increase in the size of the refutation. We state the proof for DBC^-, but the same argument works verbatim for the other systems DRAT^-, DSPR^-, DPR^- and DSR^-.

Suppose Π is a DBC^- refutation of Γ in the sense of Definition 1.17, and consider a variable x. Suppose x is present in $\Gamma = \Gamma_0$ and in Γ_i, is not present in Γ_{i+1} through Γ_j, but is present again in Γ_{j+1}. The derivation of Γ_{i+1} from Γ_i deleted a single clause $x \vee C$; for definiteness we assume this clause contains x positively. The derivation of Γ_{j+1} introduced a clause $x \vee D$ with a BC inference; we may assume without loss of generality that x occurs with the same sign in $x \vee D$ as in $x \vee C$, since otherwise the sign of x could be changed throughout the refutation from Γ_{j+1} onwards.

The refutation Π is modified as follows. Before deleting the clause $x \vee C$, infer the unit clause x by a BC inference; this is valid trivially, since $\overline{x}$ does not occur in Γ_i. Then continue the derivation with the unit clause x added to $\Gamma_i, \ldots, \Gamma_j$. Since there are no other uses of x in $\Gamma_i, \ldots, \Gamma_j$, these steps in the refutation remain valid (by part (b) of Lemma 1.20 below). Upon reaching Γ_j, infer $x \vee D$ with a BC inference relative to the variable x. This is allowed since $\overline{x}$ does not appear in Γ_j. Then delete the unit clause x to obtain again Γ_{j+1}. Repeating this for every gap in Π where x disappears, and then doing the same construction for every variable, yields a DBC$^-$ refutation that satisfies the stronger condition.

1.5 Two useful lemmas

We conclude this subsection with two technical lemmas, which we will use in several places to simplify the construction of proofs.

All the inference rules BC, RAT, SPR, PR and SR are "non-monotone", in the sense that it is possible that $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ holds but $\Gamma'_{|\alpha} \vdash_1 \Gamma'_{|\tau}$ fails, for $\Gamma \subseteq \Gamma'$. In particular, adding more clauses to Γ may invalidate a BC, RAT, SPR, PR or SR inference. Conversely, removing clauses from Γ may allow new clauses to be inferred by one of these inferences. This is one reason for the importance of the deletion rule.

The next lemma is a useful technical tool that will sometimes let us avoid using deletion. It states conditions under which the extra clauses in Γ' do not invalidate a RAT, SPR, PR or SR inference.[5]

Definition 1.19. A clause C *subsumes* a clause D if $C \subseteq D$. A set Γ of clauses *subsumes* a set Γ' if each clause of Γ' is subsumed by some clause of Γ.

Lemma 1.20. *Suppose α and τ are substitutions and $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ holds. Also suppose $\Gamma \subseteq \Gamma'$.*

(a) *If Γ subsumes Γ', then $\Gamma'_{|\alpha} \vdash_1 \Gamma'_{|\tau}$.*

(b) *If Γ' is Γ plus one or more clauses involving only variables that are not in the domain of either α or τ, then $\Gamma'_{|\alpha} \vdash_1 \Gamma'_{|\tau}$.*

[5]The conclusion of Lemma 1.20 is true also for BC inferences.

Consequently, in either case, if C can be inferred from Γ *by a* RAT, SPR, PR *or* SR *rule, then C can also be inferred from* Γ' *by the same rule.*

Proof. We prove (a). Suppose $D \in \Gamma'$ and $\tau \not\models D$. We must show $\Gamma'_{|\alpha} \vdash_1 D_{|\tau}$. Let $E \in \Gamma$ with $E \subseteq D$. Then $\tau \not\models E$, so by assumption $\Gamma_{|\alpha} \vdash_1 E_{|\tau}$. Also $E_{|\tau} \subseteq D_{|\tau}$, so $\Gamma_{|\alpha} \vdash_1 D_{|\tau}$. It follows that $\Gamma'_{|\alpha} \vdash_1 D_{|\tau}$, since $\Gamma \subseteq \Gamma'$.

The proof of (b) is immediate from the definitions. □

Our last lemma gives a kind of normal form for propagation redundancy. Namely, it implies that when C is PR with respect to Γ, we may assume without loss of generality that $dom(\tau)$ includes $dom(\alpha)$. We will use this later to show a limited simulation of PR by SPR.

Lemma 1.21. *Suppose C is* PR *with respect to* Γ*, witnessed by a partial assignment* τ*. Then* $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\alpha\circ\tau}$*, where* α *is the partial assignment* $\overline{C}$*.*

Proof. Let $\pi = \alpha\circ\tau$. Suppose $E \in \Gamma$ is such that $\pi \not\models E$. We must show that $\Gamma_{|\alpha} \vdash_1 E_{|\pi}$. We can decompose E as $E_1 \vee E_2 \vee E_3$ where E_1 contains the literals in $dom(\tau)$, E_2 the literals in $dom(\alpha) \setminus dom(\tau)$ and E_3 the remaining literals. Then $E_{|\tau} = E_2 \vee E_3$ and by the PR assumption $\Gamma_{|\alpha} \vdash_1 E_{|\tau}$, so there is a derivation $\Gamma_{|\alpha} \cup \overline{E_2} \cup \overline{E_3} \vdash_1 \bot$. But neither $\Gamma_{|\alpha}$ nor $\overline{E_3}$ contain any variables from $dom(\alpha)$, so the literals in $\overline{E_2}$ are not used in this derivation. Hence $\Gamma_{|\alpha} \cup \overline{E_3} \vdash_1 \bot$, which completes the proof since $E_3 = E_{|\pi}$. □

2 Relations with extended resolution

2.1 With new variables

It is known that RAT, and even BC, can simulate extended resolution if new variables are allowed [29]. In extended resolution for any variables p, q we are allowed to introduce a new variable x together with three clauses expressing that $x \leftrightarrow (p \wedge q)$. As shown in Example 1.8, we can instead introduce these clauses using BC inferences. Thus all the systems described above which allow new variables simulate extended resolution. The converse holds as well:

Theorem 2.1. *The system* ER *simulates* DSR*, and hence every other system above.*

Proof. (Sketch) It is known that the theorem holds for DPR in place of DSR. Namely, [24] gives an explicit simulation of DRAT by extended resolution, and [16] gives an explicit simulation of DPR by DRAT. Thus extended resolution simulates DPR.

We sketch a direct proof of the simulation of DSR by extended resolution. Suppose $\Gamma_0, \ldots, \Gamma_m$ is a DSR proof and in particular $\Gamma_{i+1} = \Gamma_i \cup \{C\}$ is introduced by an SR inference from Γ_i with a substitution τ. Let $x_1, \ldots, x_s$ be all variables occurring in Γ_{i+1} including any new variables introduced in C. Using the extension rule, introduce new variables $x'_1, \ldots, x'_s$ along with extension variables and extension axioms expressing

$$x'_j \leftrightarrow (x_j \wedge C) \vee (\tau(x_j) \wedge \neg C).$$

Here $\tau(x_j)$ stands for a fixed symbol from $Lit \cup \{0, 1\}$. Let $\Gamma_{i+1}(\vec{x}/\vec{x}')$ be the set of clauses obtained from Γ_i by replacing each variable x_j with x'_j. It can be proved using only resolution, using the extension axioms, that if all clauses in Γ_i hold then all clauses in $\Gamma_{i+1}(\vec{x}/\vec{x}')$ hold. The extended resolution proof then proceeds inductively on i using the new variables x'_j in place of the old variables x_j.

Another way to prove the full theorem is via the theories of bounded arithmetic S^1_2 [9] and PV [11]. Namely, suppose we are given a DSR proof $\Gamma_0, \ldots, \Gamma_m$ and a satisfying assignment π_0 for Γ_0. By induction, there exists a satisfying assignment π_i for each Γ_i. The inductive step, for an SR rule deriving $\Gamma_{i+1} = \Gamma_i \cup \{C\}$, witnessed by a substitution τ, is to set $\pi_{i+1} = \pi_i$ if $\pi_i \vDash C$ and otherwise to set $\pi_{i+1} = \pi_i \circ \tau$, as in the proof of Theorem 1.15. The inductive hypothesis can be written as a Σ^b_1 formulas and the induction has m steps, so this is formally Σ^b_1 length-induction (or Σ^b_1-LIND) which is available in S^1_2. Thus S^1_2 can prove the soundness of DSR. By conservativity of S^1_2 over PV [9], the theory PV also proves the soundness of DSR. A fundamental property of PV is that PV proofs translate into uniform families of polynomial size extended resolution refutations [11]. Thus ER efficiently proves the soundness of DSR. It follows that ER simulates DSR. □

2.2 Without new variables

In the systems without the ability to freely add new variables, we can still imitate extended resolution by adding dummy variables to the formula we want to refute. This was observed already in [29].

For $m \geq 1$, define X^m to be the set consisting of only the two clauses

$$y \vee x_1 \vee \cdots \vee x_m \qquad \text{and} \qquad y.$$

Lemma 2.2. *Suppose Γ has an* ER *refutation Π of size m, and that Γ and X^m have no variables in common. Then $\Gamma \cup X^m$ has a* BC$^-$*-refutation Π^* of size $O(m)$, which can furthermore be constructed from Π in polynomial time.*

Proof. We describe how to change Π into Π^*. We first rename all extension variables to use names from $\{x_1, \ldots, x_m\}$ and replace all resolution steps with $\vdash_1$ inferences. Now consider an extension rule in Π which introduces the three extension clauses (1.8) expressing $x_i \leftrightarrow (p \wedge q)$, where we may assume that p and q are either variables of Γ or from $\{x_1, \dots, x_{i-1}\}$. We simulate this by introducing successively the three clauses

$$x_i \vee \overline{p} \vee \overline{q} \qquad \overline{x}_i \vee p \vee \overline{y} \qquad \overline{x}_i \vee q \vee \overline{y}$$

using the BC rule. The first clause, $x_i \vee \overline{p} \vee \overline{q}$, is BC with respect to x_i, because $\overline{x}_i$ has not appeared yet. The second clause is BC with respect to $\overline{x_i}$, because x_i appears only in two earlier clauses, namely $y \vee x_1 \vee \cdots \vee x_m$, which contains y, and $x_i \vee \overline{p} \vee \overline{q}$, which contains $\overline{p}$. In both cases the resolvent with $\overline{x}_i \vee p \vee \overline{y}$ is tautological. The third clause is similar. The unit clause y is in X^m, so we can then derive the remaining two needed extension clauses $\overline{x}_i \vee p$ and $\overline{x}_i \vee q$ by two $\vdash_1$ inferences. □

In the terminology of [32], Lemma 2.2 shows that BC$^-$ *effectively simulates* ER, in that we are allowed to transform the formula as well as the refutation when we move from ER to BC$^-$.

The next corollary is essentially from [29]. It shows how to use the lemma to construct examples of usually-hard formulas which have short proofs in BC$^-$. (We will give less artificial examples of short SPR$^-$ proofs in Section 4.) Let $m(n)$ be the polynomial size upper bound on ER refutations of the pigeonhole principle PHP$_n$ which follows from [13] — see Section 4.1 for the definition of the PHP$_n$ clauses.

Corollary 2.3. *The set of clauses* $\mathrm{PHP}_n \cup X^{m(n)}$ *has polynomial size proofs in* BC$^-$*, but requires exponential size proofs in constant depth Frege.*

Proof. The upper bound follows from Lemma 2.2. For the lower bound, let Π be a refutation in depth-d Frege. Then we can restrict Π by setting $y = 1$ to obtain a depth-d refutation of PHP$_n$ of the same size. By [26, 31], this must have exponential size. □

The same argument can give a more general result. A propositional proof system $\mathcal{P}$ is *closed under restrictions* if, given any $\mathcal{P}$-refutation of Γ and any partial assignment ρ, we can construct a $\mathcal{P}$-refutation of $\Gamma_{|\rho}$ in polynomial time. Most of the commonly-studied proof systems such as resolution, Frege, etc. are closed under restrictions. On the other hand, it follows from results in this paper that BC$^-$ and RAT$^-$ are not closed under restrictions. To see this, let Γ be $\mathrm{BPHP}_n \cup X^{m(n)}$ where BPHP$_n$ is the bit pigeonhole principle (see Section 4.2) and m is a suitable function. Then Γ has short BC$^-$ refutations, since BPHP$_n$ has short refutations in ER. But BPHP$_n$ is a restriction of Γ, as in Corollary 2.3, and has no short RAT$^-$ refutations by Theorem 5.4 below.

Theorem 2.4. *Let $\mathcal{P}$ be any propositional proof system which is closed under restrictions. If $\mathcal{P}$ simulates* BC$^-$*, then $\mathcal{P}$ simulates* ER.

Proof. Suppose Γ has a refutation Π in ER of length m. Take a copy of X^m in disjoint variables from Γ. By Lemma 2.2 we can construct a BC^--refutation of $\Gamma \cup X^m$. Since $\mathcal{P}$ simulates BC^-, we can then construct a $\mathcal{P}$-refutation of $\Gamma \cup X^m$. Let ρ be the restriction which just sets $y = 1$, so that $(\Gamma \cup X^m)_{|\rho} = \Gamma$. By the assumption that $\mathcal{P}$ is closed under restrictions, we can construct a $\mathcal{P}$ refutation of Γ. All constructions are polynomial time. □

Corollary 2.5. *If the Frege proof system simulates* BC^-*, then Frege and* ER *are equivalent.*

Hence it is unlikely that Frege simulates BC^-, since Frege is expected to be strictly weaker than ER.

2.3 Canonical NP pairs

The notion of *disjoint NP pairs* was first introduced by Grollmann and Selman [15]. Razborov [37] showed how a propositional proof system $\mathcal{P}$ gives rise to a canonical disjoint NP pair, which gives a measure of the strength of the system. It is known that if a propositional proof system $\mathcal{P}_1$ simulates a system $\mathcal{P}_2$, then there is a many-one reduction from the canonical NP pair for $\mathcal{P}_2$ to the canonical NP pair for $\mathcal{P}_1$ [37, 35]. We can use Lemma 2.2 to prove that the systems BC^- through DSR^- cannot be distinguished from each other or ER by their canonical NP pairs, even though they do not all simulate each other.

Definition 2.6. A *disjoint NP pair* is a pair (U, V) of NP sets such that $U \cap V = \emptyset$. A *many-one reduction* from a disjoint NP pair (U, V) to a disjoint NP pair (U', V') is a polynomial time function f mapping U to U' and mapping V to V'.

To motivate this definition a little, a disjoint NP pair (U, V) is said to be *polynomially separable* if there is a polynomial time function f which, given $x \in U \cup V$, correctly identifies whether $x \in U$ or $x \in V$. Clearly if (U, V) is many-one reducible to (U', V'), then if (U', V') is polynomially separable so is (U, V).

Definition 2.7. SAT is the set of pairs $(\Gamma, 1^m)$ such that Γ is a satisfiable set of clauses and $m \geq 1$ is an arbitrary integer. Let $\mathcal{P}$ be a propositional proof system for refuting sets of clauses. Then $\mathrm{REF}(\mathcal{P})$ is the set of pairs $(\Gamma, 1^m)$ such that Γ has a $\mathcal{P}$-refutation of length at most m. Notice that SAT and $\mathrm{REF}(\mathcal{P})$ are both NP. We define the *canonical disjoint NP pair*, or *canonical NP pair*, associated with $\mathcal{P}$ to be $(\mathrm{REF}(\mathcal{P}), \mathrm{SAT})$.

The canonical NP pair for a proof system $\mathcal{P}$ defines the following problem. Given a pair $(\Gamma, 1^m)$, the soundness of $\mathcal{P}$ implies that it is impossible that both

(a) Γ is satisfiable and (b) Γ has a proof in $\mathcal{P}$ of length $\leq m$. The promise problem is to identify one of (a) and (b) which does *not* hold. (If neither (a) nor (b) holds, then either answer may be given.)

Theorem 2.8. *There are many-one reductions in both directions between the canonical NP pair for* ER *and the canonical NP pairs for all the systems in Section 1.3.*

Proof. As a simulation implies a reduction between canonical NP pairs, all we need to show is a reduction of the canonical NP pair for ER to the canonical NP pair for the weakest system BC$^-$, that is, of (REF(ER), SAT) to (REF(BC$^-$), SAT). Suppose $(\Gamma, 1^m)$ is given as a query to (REF(ER), SAT). We must produce some Γ^* and m^* such that

1. Γ^* is satisfiable if Γ is,
2. Γ^* has a BC$^-$-refutation of size m^* if Γ has an ER-refutation of size m, and
3. m^* is bounded by a polynomial in m.

We use Lemma 2.2, letting Γ^* be $\Gamma \cup X^m$ for X^m in variables disjoint from Γ, and letting m^* be the bound on the size of the BC$^-$-refutation of Γ^*. $\square$

3 Simulations

3.1 DRAT$^-$ simulates DPR$^-$

The following relations were known between DBC$^-$, DRAT$^-$ and DPR$^-$.

Theorem 3.1. ([24]) DBC$^-$ *simulates* DRAT$^-$. *(Hence they are equivalent).*

Theorem 3.2. ([16]) *Suppose* Γ *has a* DPR *refutation* Π. *Then it has a* DRAT *refutation constructible in polynomial time from* Π*, using at most one variable not appearing in* Π.

We prove:

Theorem 3.3. DRAT$^-$ *simulates* DPR$^-$.

Hence the systems DBC$^-$, DRAT$^-$, DSPR$^-$ and DPR$^-$ are all equivalent. The theorem relies on the following main lemma used in the proof of Theorem 3.2. We include a proof for completeness.

Lemma 3.4. ([16]) *Suppose* C *is* PR *with respect to* Γ. *Then there is a polynomial size* DRAT *derivation of* $\Gamma \cup \{C\}$ *from* Γ*, using at most one variable not appearing in* Γ *or* C.

Proof. We have $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$, where $\alpha = \overline{C}$ and $\tau \vDash C$. Let x be a new variable. We describe the construction step-by-step.

Step 1. For each $D \in \Gamma$ which is not satisfied by τ, derive $D_{|\tau} \vee \overline{x}$ by RAT on $\overline{x}$. This is possible, as x does not appear anywhere yet.

Step 2. Derive $C \vee x$ by RAT on x. The only clauses in which $\overline{x}$ appears are those of the form $D_{|\tau} \vee \overline{x}$ introduced in step 1, and from Lemma 1.5 and the assumption that $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ we have that $\Gamma \vdash_1 D_{|\tau} \vee C$.

Step 3. For each $E \in \Gamma$ satisfied by τ, derive $E \vee x$ by a $\vdash_1$ step and delete E.

Step 4. For each literal p in τ, derive $\overline{x} \vee p$ by RAT on p. To see that this satisfies the RAT condition, consider any clause $G = G' \dot{\vee} \overline{p}$ with which $\overline{x} \vee p$ could be resolved. If $\tau \vDash G$, then by steps 2 and 3 above, G must also contain x, so the resolvent $G' \cup \overline{x}$ is a tautology. If $\tau \nvDash G$, then G must be one of the clauses $D \in \Gamma$ or $D_{|\tau} \vee \overline{x}$ from step 1, which means that we have already derived $G_{|\tau} \vee \overline{x}$, which subsumes the resolvent $G' \vee \overline{x}$.

Step 5. Consider each clause $E \vee x$ introduced in step 2 or 3. In either case $\tau \vDash E$, so E contains some literal p in τ. Therefore we can derive E by resolving $E \vee x$ with $\overline{x} \vee p$. Thus we derive C and all clauses from Γ deleted in step 3.

Finally delete all the new clauses except for C. $\square$

Definition 3.5. Let Γ be a set of clauses and x any variable. Then $\Gamma^{(x)}$ consists of every clause in Γ which does not mention x, together with every clause of the form $E \vee F$ where both $x \dot{\vee} E$ and $\overline{x} \dot{\vee} F$ are in Γ.

In other words, $\Gamma^{(x)}$ is formed from Γ by doing all possible resolutions with respect to x and then deleting all clauses containing either x or $\overline{x}$. (This is exactly like the first step of the Davis-Putnam procedure. In [29] the notation DP_x is used instead of $\Gamma^{(x)}$.)

Lemma 3.6. *There is a polynomial size* DRAT *derivation of* Γ *from* $\Gamma^{(x)}$*, using only variables from* Γ.

Proof. We first derive every clause of the form $E \dot{\vee} x$ in Γ, by RAT on x. As $\overline{x}$ has not appeared yet, the RAT condition is satisfied. Then we derive each clause of the form $F \dot{\vee} \overline{x}$ in Γ, by RAT on $\overline{x}$. The only possible resolutions are with clauses of the form $E \dot{\vee} x$ which we have just introduced, but in this case either $E \cup F$ is tautological or $E \vee F$ is in $\Gamma^{(x)}$ so $\Gamma^{(x)} \vdash_1 \overline{x} \vee F \vee E$. Finally we delete all clauses not in Γ. $\square$

The next two lemmas show that, under suitable conditions, if we can derive C from Γ in DPR^-, then we can derive it from $\Gamma^{(x)}$. We will use a kind of normal form for PR inferences. Say that a clause C is PR_0 with respect to Γ if there is a partial assignment τ such that $\tau \vDash C$, all variables in C are in $dom(\tau)$, and

$$C \vee \Gamma_{|\tau} \subseteq \Gamma. \tag{2}$$

(Recall that the notation $C \vee \Gamma_{|\tau}$ means the set of clauses $C \vee D$ for $D \in \Gamma_{|\tau}$.) The PR_0 inference rule lets us derive $\Gamma \cup \{C\}$ from Γ when (2) holds. Letting $\alpha = \overline{C}$ it is easy to see that (2) implies $\Gamma_{|\tau} \subseteq \Gamma_{|\alpha}$, so in particular $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$, and hence this is a special case of the PR rule.

Lemma 3.7. *Any* PR *inference can be replaced with a* PR_0 *inference together with polynomially many* $\vdash_1$ *and deletion steps, using no new variables.*

Proof. Suppose $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$, where $\alpha = \overline{C}$ and $\tau \vDash C$. By Lemma 1.21 we may assume $dom(\alpha) \subseteq dom(\tau)$ so $dom(\tau)$ contains all variables in C. Let $\Delta = C \vee \Gamma_{|\tau}$ and $\Gamma^* = \Gamma \cup \Delta$. Note that $\Delta_{|\tau}$ is empty, as τ satisfies C. This implies that $C \vee \Gamma^*_{|\tau} = C \vee \Gamma_{|\tau} \subseteq \Gamma^*$, so C is PR_0 with respect to Γ^*. Furthermore the condition $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ and Lemma 1.5 imply that every clause in Δ is derivable from Γ by a $\vdash_1$ step. Thus we can derive Γ^* from Γ by $\vdash_1$ steps, then introduce C by the PR_0 rule, and recover $\Gamma \cup \{C\}$ by deleting everything else. □

Lemma 3.8. *Suppose C is* PR_0 *with respect to* Γ*, witnessed by* τ *with* $x \notin dom(\tau)$*. Then* C *is* PR_0 *with respect to* $\Gamma^{(x)}$.

Proof. The PR_0 condition implies that the variable x does not occur in C. We are given that $C \vee \Gamma_{|\tau} \subseteq \Gamma$ and want to show that $C \vee \Gamma^{(x)}_{|\tau} \subseteq \Gamma^{(x)}$. So let $D \in \Gamma^{(x)}$ with $\tau \nvDash D$. First suppose D is in Γ and x does not occur in D. Then $C \vee D_{|\tau} \in \Gamma$ by assumption, so $C \vee D_{|\tau} \in \Gamma^{(x)}$. Otherwise, $D = E \vee F$ where both $E \dot{\vee} x$ and $F \dot{\vee} \overline{x}$ are in Γ. Then by assumption both $C \vee E_{|\tau} \vee x$ and $C \vee F_{|\tau} \vee \overline{x}$ are in Γ. Hence $C \vee D_{|\tau} = C \vee E_{|\tau} \vee F_{|\tau} \in \Gamma^{(x)}$. □

We can now prove Theorem 3.3, that DRAT^- simulates DPR^-.

Proof of Theorem 3.3. We are given a DPR^- refutation of some set Δ, using only the variables in Δ. By Lemma 3.7 we may assume without loss of generality that the refutation uses only $\vdash_1$, deletion and PR_0 steps. Consider a PR_0 inference in this refutation, which derives $\Gamma \cup \{C\}$ from a set of clauses Γ, witnessed by a partial assignment τ. We want to derive $\Gamma \cup \{C\}$ from Γ in DRAT using only variables in Δ.

Suppose τ is a total assignment to all variables in Γ. The set Γ is necessarily unsatisfiable, as otherwise it could not occur as a line in a refutation. Therefore $\Gamma_{|\tau}$ is simply $\bot$, so the PR_0 condition tells us that $C \in \Gamma$ and we do not need to do anything.

Otherwise, there is some variable x which occurs in Γ but is outside the domain of τ, and thus in particular does not occur in C. We first use $\vdash_1$ and deletion steps to replace Γ with $\Gamma^{(x)}$. By Lemma 3.8, C is PR_0, and thus PR, with respect to $\Gamma^{(x)}$. By Lemma 3.4 there is a short DRAT derivation of $\Gamma^{(x)} \cup \{C\}$ from $\Gamma^{(x)}$, using one new variable which does not occur in $\Gamma^{(x)}$ or C. We choose x for this variable. Finally, observing that here $\Gamma^{(x)} \cup \{C\} = (\Gamma \cup \{C\})^{(x)}$, we recover $\Gamma \cup \{C\}$ using Lemma 3.6. $\square$

3.2 Towards a simulation of PR^- by SPR^-

Our next result shows how to replace a PR inference with SPR inferences, without additional variables. It is not a polynomial simulation of PR^- by SPR^- however, as it depends exponentially on the "discrepancy" as defined next. Recall that C is PR with respect to Γ if $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$, where $\alpha = \overline{C}$ and τ is a partial assignment satisfying C. We will keep this notation throughout this section. C is SPR with respect to Γ if additionally $dom(\tau) = dom(\alpha)$.

Definition 3.9. The *discrepancy* of a PR inference is $|dom(\tau) \setminus dom(\alpha)|$. That is, it is the number of variables which are assigned by τ but not by α.

Theorem 3.10. *Suppose that Γ has a* PR *refutation Π of size S in which every* PR *inference has discrepancy bounded by δ. Then Γ has a* SPR *refutation of size $O(2^\delta S)$ which does not use any variables not present in Π.*

When the discrepancy is logarithmically bounded, Theorem 3.10 gives polynomial size SPR refutations automatically. We need a couple of lemmas before proving the theorem.

Lemma 3.11. *Suppose $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ and β is a partial assignment extending α, such that $dom(\beta) \subseteq dom(\tau)$. Then $\Gamma_{|\beta} \vdash_1 \Gamma_{|\tau}$*

Proof. Suppose $E \in \Gamma_{|\tau}$. Then E contains no variables from β, so $\overline{E}_{|\beta} = \overline{E}$, and by assumption there is a refutation $\Gamma_{|\alpha}, \overline{E} \vdash_1 \bot$. Thus $\Gamma_{|\beta}, \overline{E} \vdash_1 \bot$ by Fact 1.3. $\square$

Proof of Theorem 3.10. Our main task is to show that a PR inference with discrepancy bounded by δ can be simulated by multiple SPR inferences, while bounding the increase in proof size in terms of δ. Suppose C is derivable from Γ by a PR inference. That is, $\Gamma_{|\alpha} \vdash_1 \Gamma_{|\tau}$ where $\alpha = \overline{C}$ and $\tau \vDash C$, and by Lemma 1.21 we may assume that $dom(\tau) \supseteq dom(\alpha)$. List the variables in $dom(\tau) \setminus dom(\alpha)$ as $p_1, \dots, p_s$, where $s \leq \delta$.

Enumerate as $D_1, \dots, D_{2^s}$ all clauses containing exactly the variables $p_1, \dots, p_s$ with some pattern of negations. Let $\sigma_i = \overline{C \vee D_i}$, so that $\sigma_i \supseteq \alpha$ and $dom(\sigma_i) =$

$dom(\tau)$. By Lemma 3.11, $\Gamma_{|\sigma_i} \vdash_1 \Gamma_{|\tau}$. Since $\tau \vDash C \vee D_j$ for every j, in fact

$$\Gamma_{|\sigma_i} \vdash_1 (\Gamma \cup \{C \vee D_1, \dots, C \vee D_{i-1}\})_{|\tau}.$$

Thus we may introduce all clauses $C \vee D_1, \dots, C \vee D_{2^s}$ one after another by SPR inferences. We can then use $2^s - 1$ resolution steps to derive C.

The result is a set $\Gamma' \supseteq \Gamma$ which contains C plus many extra clauses subsumed by C, which must be carried through the rest of the refutation, as we do not have the deletion rule. But by Lemma 1.20(a) this is not a problem, as the presence of these additional subsumed clauses does not affect the validity of later PR inferences. □

4 Upper bounds for some hard tautologies

This section proves that SPR$^-$ — without new variables — can give polynomial size refutations for many of the usual "hard" propositional principles. Heule, Kiesl and Biere [20, 19] showed that the tautologies based on the pigeonhole principle (PHP) have polynomial size SPR$^-$ proofs, and Heule and Biere discuss polynomial size PR$^-$ proofs of the Tseitin tautologies and the 2-1 pigeonhole principle in [16]. The SPR$^-$ proof of the PHP tautologies can be viewed as a version of the original extended resolution proof of PHP given by Cook and Reckhow [13]; see also [29] for an adaptation of the original proof to use BC inferences.

Here we describe polynomial size SPR$^-$ proofs for several well-known principles, namely the pigeonhole principle, the bit pigeonhole principle, the parity principle, the clique-coloring principles, and the Tseitin tautologies. We also show that orification, xorification, and typical cases of lifting can be handled in SPR$^-$.

The existence of such small proofs is surprising, since they use only clauses in the original literals, and it is well-known that such clauses are limited in what they can express. However, SPR$^-$ proofs can exploit the underlying symmetries of the principles to introduce new clauses, in effect arguing that properties can be assumed to hold "without loss of generality" (see [38]).

It is open whether extended resolution, or the Frege proof system, can be simulated by PR$^-$ or DPR$^-$, or more generally by DSR$^-$. The examples below show that any separation of these systems must involve a new technique.

Our proofs use the same basic idea as the sketch in Example 1.14. One complication is that we are now working with SPR rather than SR inferences. This requires us to make the individual inferences more complicated – for example the assignments α in Example 1.14 set one pigeon, while those in Section 4.1 below set two pigeons. Another is that we want to avoid using any deletion steps. This means that, when showing that an SPR inference is valid, we have to consider every clause introduced so far. For this reason we will do all necessary SPR inferences at the start, in a careful order, before we do any resolution steps. This is the purpose of Lemma 4.2.

Definition 4.1. A Γ-*symmetry* is an invertible substitution π such that $\Gamma_{|\pi} = \Gamma$.

We will use the observation that, if π is a Γ-symmetry and $\alpha = \overline{C}$ is a partial assignment, then by Lemma 1.1 we have

$$\Gamma_{|\alpha} = (\Gamma_{|\pi})_{|\alpha} = \Gamma_{|\alpha\circ\pi}.$$

Hence, if $\alpha\circ\pi \vDash C$, we can infer C from Γ by an SR inference with $\tau = \alpha\circ\pi$. If furthermore all literals in the domain and image of π are in $dom(\alpha)$, then $\alpha\circ\pi$ behaves as a partial assignment and $dom(\alpha\circ\pi) = dom(\alpha)$, so this becomes an SPR inference.

We introduce one new piece of notation, writing $\overline{\alpha}$ for the clause expressing that the partial assignment α does not hold (so $C = \overline{\alpha}$ if and only if $\alpha = \overline{C}$). Two partial assignments are called *disjoint* if their domains are disjoint. The next lemma describes sufficient conditions for introducing, successively, the clauses $\overline{\alpha_i}$ for $i = 0, 1, 2, \dots$ using only SPR inferences.

Lemma 4.2. *Suppose $(\alpha_0, \tau_0), \dots, (\alpha_m, \tau_m)$ is a sequence of pairs of partial assignments such that for each i,*

1. $\Gamma_{|\alpha_i} = \Gamma_{|\tau_i}$
2. *α_i and τ_i are contradictory and have the same domain*
3. *for all $j < i$, the assignments α_j and τ_i are either disjoint or contradictory.*

Then we can derive $\Gamma \cup \{\overline{\alpha_i} : i = 0, \dots, m\}$ from Γ by a sequence of SPR *inferences.*

Proof. We write C_i for $\overline{\alpha_i}$. By item 2, $\tau_i \vDash C_i$. Thus it is enough to show that for each i,

$$\big(\Gamma \cup \{C_0, \dots, C_{i-1}\}\big)_{|\alpha_i} \supseteq \big(\Gamma \cup \{C_0, \dots, C_{i-1}\}\big)_{|\tau_i}.$$

We have $\Gamma_{|\alpha_i} = \Gamma_{|\tau_i}$. For each $j < i$, either α_j and τ_i are disjoint, and consequently $(C_j)_{|\alpha_i} = (C_j)_{|\tau_i} = C_j$, or they are contradictory and so $\tau_i \vDash C_j$ and C_j vanishes from the right hand side. $\square$

In the lemma, if we added to 2. the condition that α_i and τ_i disagree on only a single variable, then by Theorem 1.10 we could derive the clauses $\overline{\alpha_i}$ by RAT inferences rather than needing SPR inferences. However in the applications below they typically differ on more than one variable, so our proofs are in SPR$^-$, not in RAT$^-$.

4.1 Pigeonhole principle

Let $n \geq 1$ and $[n]$ denote $\{0, \ldots, n-1\}$. The pigeonhole principle PHP_n consists of the clauses

$$\bigvee_{j \in [n]} p_{i,j} \quad \text{for each fixed } i \in [n+1] \qquad \text{(pigeon axioms)}$$

$$\overline{p_{i,j}} \vee \overline{p_{i',j}} \quad \text{for all } i < i' \in [n+1] \text{ and } j \in [n] \qquad \text{(hole axioms).}$$

Theorem 4.3 ([20])**.** PHP_n *has polynomial size* SPR^- *refutations.*

Proof. Our strategy is to first derive all unit clauses $\overline{p_{j,0}}$ for $j > 0$, which effectively takes pigeon 0 and hole 0 out of the picture and reduces PHP_n to a renamed instance of PHP_{n-1}. We repeat this construction to reduce to a renamed instance of PHP_{n-2}, etc. At each step, we will need to use several clauses introduced by SPR inferences. We use Lemma 4.2 to introduce all necessary clauses at one go at the start of the construction.

Let $\alpha_{i,j,k}$ be the assignment setting $p_{i,k} = 1$, $p_{j,i} = 1$ and all other variables $p_{\ell,k}, p_{\ell,i}$ for holes k and i to 0. Let $\pi_{k,i}$ be the PHP_n-symmetry which switches holes k and i, that is, maps $p_{\ell,i} \mapsto p_{\ell,k}$ and $p_{\ell,k} \mapsto p_{\ell,i}$ for every pigeon ℓ. Let $\tau_{i,j,k}$ be $\alpha_{i,j,k} \circ \pi_{k,i}$, so in particular $\tau_{i,j,k}$ sets $p_{i,i} = 1$ and $p_{j,k} = 1$. By the properties of symmetries, we have $(\mathrm{PHP}_n)_{|\alpha_{i,j,k}} = (\mathrm{PHP}_n)_{|\tau_{i,j,k}}$.

For $i = 0, \ldots, n-2$ define

$$A_i := \{(\alpha_{i,j,k}, \tau_{i,j,k}) : i < j < n+1,\ i < k < n\}.$$

Any $\tau_{i,j,k}$ appearing in A_i contradicts every $\alpha_{i,j',k'}$ appearing in A_i, since they disagree about which pigeon maps to hole i. On the other hand, if $i' < i$ and $\alpha_{i',j',k'}$ appears in $A_{i'}$ and is not disjoint from $\tau_{i,j,k}$, then they must share some hole. So either $i = k'$ or $k = k'$, and in either case they disagree about hole k'.

Hence we can apply Lemma 4.2 to derive all clauses $\overline{\alpha_{i,j,k}}$ such that $i < j < n+1$ and $i < k < n$. Note $\overline{\alpha_{i,j,k}}$ is the clause $\overline{p_{i,k}} \vee \bigvee_{\ell \neq i} p_{\ell,k} \vee \overline{p_{j,i}} \vee \bigvee_{\ell \neq j} p_{\ell,i}$, which we resolve with hole axioms to get $\overline{p_{i,k}} \vee \overline{p_{j,i}}$.

Now we use induction on $i = 0, \ldots, n-1$ to derive all unit clauses $\overline{p_{j,i}}$ for all j with $i < j < n+1$. Fix $j > i$. For each hole $k > i$ we have $\overline{p_{i,k}} \vee \overline{p_{j,i}}$ (or if $i = n-1$ there is no such k). We have $\overline{p_{i,i}} \vee \overline{p_{j,i}}$ since it is a hole axiom, and for each $k < i$, we have $\overline{p_{i,k}}$ from the inductive hypothesis. Resolving all these with the axiom $\bigvee_k p_{i,k}$ gives $\overline{p_{j,i}}$.

Finally the unit clauses $\overline{p_{n,i}}$ for $i < n$ together contradict the axiom $\bigvee_i p_{n,i}$. $\square$

4.2 Bit pigeonhole principle

Let $n = 2^k$. The *bit pigeonhole principle* contradiction, BPHP_n, asserts that each of $n + 1$ pigeons can be assigned a distinct k-bit binary string. For each pigeon x, with $0 \le x < n+1$, it has variables $p_1^x, \dots, p_k^x$ for the bits of the string assigned to x. We think of strings $y \in \{0,1\}^k$ as holes. When convenient we will identify holes with numbers $y < n$. We write $(x{\to}y)$ for the conjunction $\bigwedge_i (p_i^x = y_i)$ asserting that pigeon x goes to hole y, where $p_i^x = 1$ is the literal p_i^x and $p_i^x = 0$ is the literal $\overline{p_i^x}$, and where y_i is the i-th bit of y. We write $(x{\not\to}y)$ for its negation: $\bigvee_i (p_i^x \neq y_i)$. The axioms of BPHP_n are then

$$(x{\not\to}y) \vee (x'{\not\to}y) \quad \text{for all holes } y \text{ and all distinct pigeons } x, x'.$$

Notice that the set $\{(x{\not\to}y) : y < n\}$ consists of the 2^k clauses containing the variables $p_1^x, \dots, p_k^x$ with all patterns of negations. We can derive $\bot$ from this set in $2^k - 1$ resolution steps.

Theorem 4.4. *The* BPHP_n *clauses have polynomial size* SPR^- *refutations.*

The theorem is proved below. It is essentially the same as the proof of PHP in [20] (or Theorem 4.3 above). For each $m < n - 1$ and each pair $x, y > m$, we define a clause

$$C_{m,x,y} := (m{\not\to}y) \vee (x{\not\to}m).$$

Note we allow $x = y$. Let Γ be the set of all such clauses $C_{m,x,y}$. We will show these clauses can be introduced by SPR inferences, but first we show they suffice to derive BPHP_n.

Lemma 4.5. $\mathrm{BPHP}_n \cup \Gamma$ *has a polynomial size resolution refutation.*

Proof. Using induction on $m = 0, 1, 2, \dots, n-1$ we derive all clauses $(x{\not\to}m)$ such that $x > m$. So suppose $m < n$ and $x > m$. For each $y > m$, we have the clause $(m{\not\to}y) \vee (x{\not\to}m)$, as this is $C_{m,x,y}$. We also have the clause $(m{\not\to}m) \vee (x{\not\to}m)$, as this is an axiom of BPHP_n. Finally, for each $m' < m$, we have $(m{\not\to}m')$ by the inductive hypothesis (or, in the base case $m = 0$, there are no such clauses). Resolving all these together gives $(x{\not\to}m)$.

At the end we have in particular derived all the clauses $(n{\not\to}m)$ such that $m < n$. Resolving all these clauses together yields $\bot$. □

Thus it is enough to show that we can introduce all clauses in Γ using SPR inferences. We use Lemma 4.2. For $m < n - 1$ and each pair $x, y > m$, define partial assignments

$$\alpha_{m,x,y} := (m{\to}y) \wedge (x{\to}m)$$
$$\tau_{m,x,y} := (m{\to}m) \wedge (x{\to}y)$$

so that $C_{m,x,y} = \overline{\alpha_{m,x,y}}$ and $\tau_{m,x,y} = \alpha_{m,x,y} \circ \pi$ where π swaps all variables for pigeons m and x. Hence $(\mathrm{BPHP}_n)_{|\alpha_{m,x,y}} = (\mathrm{BPHP}_n)_{|\tau_{m,x,y}}$ as required.

For the other conditions for Lemma 4.2, first observe that assignments $\alpha_{m,x,y}$ and $\tau_{m,x',y'}$ are always inconsistent, since they map m to different places. Now suppose that $m < m'$ and $\alpha_{m,x,y}$ and $\tau_{m',x',y'}$ are not disjoint. Then they must have some pigeon in common, so either $m' = x$ or $x' = x$. In both cases $\tau_{m',x',y'}$ contradicts $(x \to m)$, in the first case because it maps x to m', and in the second because it maps x to y' with $y' > m'$.

4.3 Parity principle

The *parity principle* states that there is no (undirected) graph on an odd number of vertices in which each vertex has degree exactly one (see [1, 4]). For n odd, let PAR_n be a set of clauses expressing (a violation of) the parity principle on n vertices, with variables $x_{i,j}$ for the $\binom{n}{2}$ many values $0 \le i < j < n$, where we identify the variable $x_{i,j}$ with $x_{j,i}$. We write $[n]$ for $\{0, \dots, n-1\}$. PAR_n consists of the clauses

$$\bigvee_{j \neq i} x_{i,j} \quad \text{for each fixed } i \in [n] \qquad \text{(“pigeon” axioms)}$$

$$\overline{x_{i,j}} \vee \overline{x_{i,j'}} \quad \text{for all distinct } i, j, j' \in [n] \qquad \text{(“hole” axioms).}$$

Theorem 4.6. *The* PAR_n *clauses have polynomial size* SPR^- *refutations.*

Proof. Let $n = 2m + 1$. For $i < m$ and distinct j, k with $2i + 1 < j, k < n$ define $\alpha_{i,j,k}$ to be the partial assignment which matches $2i$ to j and $2i + 1$ to k, and sets all other adjacent variables to 0. That is, $x_{2i,j} = 1$ and $x_{2i,j'} = 0$ for all $j' \neq j$, and $x_{2i+1,k} = 1$ and $x_{2i+1,k'} = 0$ for all $k' \neq k$. Similarly define $\tau_{i,j,k}$ to be the partial assignment which matches $2i$ to $2i + 1$ and j to k, and sets all other adjacent variables to 0, so that $\tau_{i,j,k} = \alpha_{i,j,k} \circ \pi$ where π swaps vertices $2i + 1$ and j. It is easy to see that the conditions of Lemma 4.2 are satisfied. Therefore, we can introduce all clauses $\overline{\alpha_{i,j,k}}$ by SPR inferences.

We now inductively derive the unit clauses $x_{2i,2i+1}$ for $i = 0, 1, \dots, m-1$. Once we have these, refuting PAR_n becomes trivial. So suppose we have $x_{2i',2i'+1}$ for all $i' < i$ and want to derive $x_{2i,2i+1}$. Consider any $r < 2i$. First suppose r is even, so $r = 2m$ for some $m < i$. We resolve the “hole” axiom $\overline{x_{2m,2i}} \vee \overline{x_{2m,2m+1}}$ with $x_{2m,2m+1}$ to get $\overline{x_{2m,2i}}$, which is the same clause as $\overline{x_{2i,r}}$. A similar argument works for r odd, and we can also obtain $\overline{x_{2i+1,r}}$ in a similar way.

Resolving the clauses $\overline{x_{2i,r}}$ and $\overline{x_{2i+1,r}}$ for $r < 2i$ with the “pigeon” axioms for vertices $2i$ and $2i + 1$ gives clauses

$$x_{2i,2i+1} \vee \bigvee_{r > 2i+1} x_{2i,r} \quad \text{and} \quad x_{2i,2i+1} \vee \bigvee_{r > 2i+1} x_{2i+1,r}.$$

Now by resolving clauses $\overline{\alpha_{i,j,k}}$ with suitable "hole" axioms we can get $\overline{x_{2i,j}} \vee \overline{x_{2i+1,k}}$ for all distinct $j, k > 2i + 1$. Resolving these with the clauses above gives $x_{2i,2i+1}$, as required. □

4.4 Clique-coloring principle

The *clique-coloring principle* $\mathrm{CC}_{n,m}$ states, informally, that a graph with n vertices cannot have both a clique of size m and a coloring of size $m - 1$ (see [25, 33]). For $m \leq n$ integers, $\mathrm{CC}_{n,m}$ uses variables $p_{a,i}$, $q_{i,c}$ and $x_{i,j}$ where $a \in [m]$ and $c \in [m-1]$ and $i, j \in [n]$ with $i \neq j$. Again, $x_{i,j}$ is identified with $x_{j,i}$. The intuition is that $x_{i,j}$ indicates that vertices i and j are joined by an edge, $p_{a,i}$ asserts that i is the a-th vertex of a clique, and $q_{i,c}$ indicates that vertex i is assigned color c. The clauses of $\mathrm{CC}_{n,m}$ are

(i) $\bigvee_i p_{a,i}$ for each $a \in [m]$

(ii) $\overline{p_{a,i}} \vee \overline{p_{a',i}}$ for distinct $a, a' \in [m]$ and each $i \in [n]$

(iii) $\bigvee_c q_{i,c}$ for each $i \in [n]$

(iv) $\overline{q_{i,c}} \vee \overline{q_{i,c'}}$ for each $i \in [n]$ and distinct $c, c' \in [m-1]$

(v) $\overline{p_{a,i}} \vee \overline{p_{a',j}} \vee x_{i,j}$ for each distinct $a, a' \in [m]$ and distinct $i, j \in [n]$

(vi) $\overline{q_{i,c}} \vee \overline{q_{j,c}} \vee \overline{x_{i,j}}$ for each $c \in [m-1]$ and distinct $i, j \in [n]$.

Theorem 4.7. *The* $\mathrm{CC}_{n,m}$ *clauses have polynomial size* SPR^- *refutations.*

Proof. The intuition for the SPR^- proof is that we introduce clauses stating that the first r clique members are assigned vertices that are colored by the first r colors; iteratively for $r = 1, 2, \ldots$.

Write $(a{\to}i{\to}c)$ for the assignment which sets

$$p_{a,i} = 1 \quad \text{and} \quad p_{a',i} = 0 \text{ for all } a' \neq a$$
$$q_{i,c} = 1 \quad \text{and} \quad q_{i,c'} = 0 \text{ for all } c' \neq c.$$

For all $r < m-2$, all indices $a > r$, all colors $c > r$ and all distinct vertices $i, j \in [n]$, define

$$\alpha^r_{a,i,j,c} := (a{\to}j{\to}r) \wedge (r{\to}i{\to}c)$$
$$\tau^r_{a,i,j,c} := (a{\to}j{\to}c) \wedge (r{\to}i{\to}r).$$

Let Γ consist of axioms (i), (ii) and (v), containing p and x variables but no q variables, and let Δ consist of the remaining axioms (iii), (iv) and (vi), containing

q and x variables but no p variables. Let us write α for $\alpha^r_{a,i,j,c}$ and τ for $\tau^r_{a,i,j,c}$. Then $\Gamma_{|\alpha} = \Gamma_{|\tau}$ since α and τ are the same on p variables. Let α' and τ' be respectively α and τ restricted to q variables. Then $\Delta_{|\alpha'} = \Delta_{|\tau'}$ since $\tau' = \alpha' \circ \pi$ where π is the Δ-symmetry which swaps vertices i and j. Hence also $\Delta_{|\alpha} = \Delta_{|\tau}$.

We will show that the conditions of Lemma 4.2 are satisfied, so we can introduce all clauses $\overline{\alpha^r_{a,i,j,c}}$ by SPR inferences. The first condition was just discussed. For the second condition, first notice that $\alpha^r_{a,i,j,c}$ and $\tau^r_{a,i,j,c}$ set the same variables.

Now suppose r, a, c, i, j are such that $r < a < m$, that $r < c < m-1$, and that $i, j \in [n]$ are distinct. Suppose r', a', c', i', j' satisfy the same conditions, with $r' \leq r$. We want to show that if $\tau := \tau^r_{a,i,j,c}$ and $\alpha := \alpha^{r'}_{a',i',j',c'}$ are not disjoint, then they are contradictory. Notice that showing this will necessarily use the literals $\overline{p_{a',i}}$ and $\overline{q_{i,c'}}$ in the definition of our assignments, and that it will be enough to show that α and τ disagree about either which index or which color is assigned to a vertex i. First suppose $r' = r$. Assuming α and τ are not disjoint, we must be in one of the following four cases.

1. $i' = i$. Then α maps vertex i to color $c' > r$ while τ maps i to color r.
2. $i' = j$. Then α maps index $r < a$ to vertex j while τ maps index a to j.
3. $j' = i$. Then α maps index $a' > r$ to vertex i while τ maps index r to i.
4. $j' = j$. Then α maps vertex j to color $r < c$ while τ maps j to color c.

Now suppose $r' < r$. Assuming α and τ are not disjoint, we have the same cases.

1. $i' = i$. Then α maps index $r' < r$ to vertex i while τ maps index r to i.
2. $i' = j$. Then α maps index $r' < a$ to vertex j while τ maps index a to j.
3. $j' = i$. Then α maps vertex i to color $r' < r$ while τ maps i to color r.
4. $j' = j$. Then α maps vertex j to color $r' < c$ while τ maps j to color c.

Thus the conditions are met and we can introduce the clauses $\overline{\alpha^r_{a,i,j,c}}$, that is,

$$\overline{p_{a,j}} \vee \bigvee_{a' \neq a} p_{a',j} \vee \overline{q_{j,r}} \vee \bigvee_{r' \neq r} q_{j,r'} \vee \overline{p_{r,i}} \vee \bigvee_{r' \neq r} p_{r',i} \vee \overline{q_{i,c}} \vee \bigvee_{c' \neq c} q_{i,c'},$$

for all $r < a < m$, all $r < c < m-1$ and all distinct $i, j \in [n]$. Now let $C^r_{a,i,j,c}$ be the clause

$$\overline{p_{a,j}} \vee \overline{q_{j,r}} \vee \overline{p_{r,i}} \vee \overline{q_{i,c}}.$$

We derive this by resolving $\overline{\alpha^r_{a,i,j,c}}$ with instances of axiom (ii) to remove the literals $p_{a',j}$ and $p_{r',i}$ and then with instances of axiom (iv) to remove the literals $q_{j,r'}$

and $q_{i,c'}$. We now want to derive, for each r, each a with $r < a < m$ and each $j \in [n]$, the clause

$$\overline{p_{a,j}} \vee \bigvee_{c>r} q_{j,c} \tag{3}$$

which can be read as "if $a > r$ goes to j, then j goes to some $c > r$". Intuitively, this removes indices and colors $0, \ldots, r$ from $\mathrm{CC}_{n,m}$, thus reducing it to a CNF isomorphic to $\mathrm{CC}_{n,m-r-1}$.

Suppose inductively that we have already derived (3) for all $r' < r$. In particular we have $\overline{p_{a,j}} \vee \bigvee_{c>r-1} q_{j,c}$, or for $r = 0$ we use the axiom $\bigvee_c q_{j,c}$. We resolve this with the clauses $C^r_{a,i,j,c}$ for all $c > r$ to get

$$\overline{p_{a,j}} \vee \overline{q_{j,r}} \vee \overline{p_{r,i}} \vee q_{i,r}. \tag{4}$$

By resolving together suitable instances of axioms (v) and (vi) we obtain

$$\overline{p_{a,j}} \vee \overline{p_{r,i}} \vee \overline{q_{j,r}} \vee \overline{q_{i,r}}$$

and resolving this with (4) removes the variable $q_{i,r}$ to give $\overline{p_{a,j}} \vee \overline{q_{j,r}} \vee \overline{p_{r,i}}$. We derive this for every i, and then resolve with the axiom $\bigvee_i p_{r,i}$ to get $\overline{p_{a,j}} \vee \overline{q_{j,r}}$, and finally again with our inductively given clause $\overline{p_{a,j}} \vee \bigvee_{c>r-1} q_{j,c}$ to get $\overline{p_{a,j}} \vee \bigvee_{c>r} q_{j,c}$ as required. □

4.5 Tseitin tautologies

The *Tseitin tautologies* $\mathrm{TS}_{G,\gamma}$ are well-studied hard examples for many proof systems (see [41, 42]). Let G be an undirected graph with n vertices, with each vertex i labelled with a charge $\gamma(i) \in \{0, 1\}$ such that the total charge on G is odd. For each edge e of G there is a variable x_e. Then $\mathrm{TS}_{G,\gamma}$ consists of clauses expressing that, for each vertex i, the parity of the values x_e over the edges e touching i is equal to the charge $\gamma(i)$. For a vertex i of degree d, this requires 2^{d-1} clauses, using one clause to rule out each assignment to the edges touching i with the wrong parity. If G has constant degree then this has size polynomial in n, but in general the size may be exponential in n. It is well-known to be unsatisfiable.

The next lemma is a basic property of Tseitin contradictions. Note that it does not depend on γ. By cycle we mean a simple cycle, with no repeated vertices.

Lemma 4.8. *Let K be any cycle in G. Then the substitution π_K which flips the sign of every literal on K is a $\mathrm{TS}_{G,\gamma}$-symmetry.*

Lemma 4.9. *If every node in G has degree at least 3, then G contains a cycle of length at most $2 \log n$.*

Proof. Pick any vertex i and let H be the subgraph consisting of all vertices reachable from i in at most $\log n$ steps. Then H cannot be a tree, as otherwise by the assumption on degree it would contain more than n vertices. Hence it must contain some vertex reachable from i in two different ways. $\square$

Theorem 4.10. *The* $\mathrm{TS}_{G,\gamma}$ *clauses have polynomial size* SPR^- *refutations.*

Proof. We will construct a sequence of triples $(G_0, \gamma_0, \ell_0), \dots, (G_m, \gamma_m, \ell_m)$ where (G_0, γ_0) is (G, γ), each G_{i+1} is a subgraph of G_i formed by deleting one edge and removing any isolated vertices, γ_i is an odd assignment of charges to G_i, and ℓ_i is a literal corresponding to an edge in $G_i \setminus G_{i+1}$. Let

$$\Gamma_i = \mathrm{TS}_{G_0,\gamma_0} \cup \{\ell_0\} \cup \cdots \cup \mathrm{TS}_{G_i,\gamma_i} \cup \{\ell_i\}.$$

As we go we will construct an SPR^- derivation containing sets of clauses Γ_i' extending and subsumed by Γ_i, and we will eventually reach a stage m where Γ_m is trivially refutable. The values of ℓ_i, G_{i+1} and γ_{i+1} are defined from G_i and γ_i according to the next three cases.

Case 1: G_i contains a vertex j of degree 1. Let $\{j, k\}$ be the edge touching j. If k has degree 2 or more, we define (G_{i+1}, γ_{i+1}) by letting G_{i+1} be G_i with edge $\{j, k\}$ and vertex j removed, and letting γ_{i+1} be γ_i restricted to G_{i+1} and with $\gamma_{i+1}(k) = \gamma_i(k) + \gamma_i(j)$. If k has degree 1 and the same charge as j, then we let G_{i+1} be G_i with both j and k removed (with unchanged charges). In both cases, every clause in $\mathrm{TS}_{G_{i+1},\gamma_{i+1}}$ is derivable from $\mathrm{TS}_{G_i,\gamma_i}$ by a $\vdash_1$ step, as the Tseitin condition on j in $\mathrm{TS}_{G_i,\gamma_i}$ is a unit clause; we set ℓ_i to be the literal contained in this clause. If k has degree 1 and opposite charge from j, then we can already derive a contradiction from $\mathrm{TS}_{G_i,\gamma_i}$ by one $\vdash_1$ step.

Case 2: G_i contains no vertices of degree 1 or 2. Apply Lemma 4.9 to find a cycle K in G_i of length at most $2 \log n$ and let e be the first edge in K. Our goal is to derive the unit clause $\overline{x}_e$ and remove e from G_i.

Let α be any assignment to the variables on K which sets x_e to 1, and let τ be the opposite assignment. Using Lemma 4.8 applied simultaneously to all graphs $G_0, \dots, G_i$ we have $(\Gamma_i)_{|\alpha} = (\Gamma_i)_{|\tau}$, as the unit clauses ℓ_i are unaffected by these restrictions. Hence by Lemma 4.2, SPR^- inferences can be used to introduce all clauses $\overline{\alpha}$, of which there are at most $2^{2 \log n - 1}$. We resolve them all together to get the unit clause $\overline{x}_e$. This subsumes all other clauses introduced so far in this step; we set ℓ_i to be $\overline{x_e}$, and by Lemma 1.20(a), we may ignore these subsumed clauses in future inferences. (Therefore we avoid needing the deletion rule.) We define (G_{i+1}, γ_{i+1}) by deleting edge e from G_i and leaving γ_i unchanged. All clauses in $\mathrm{TS}_{G_{i+1},\gamma_{i+1}}$ can now be derived from $\mathrm{TS}_{G_i,\gamma_i}$ and $\overline{x}_e$ by single $\vdash_1$ steps.

Case 3: G_i contains no vertices of degree 1, but may contain vertices of degree 2. We will adapt the argument of case 2. Redefine a *path* to be a sequence of edges connected by degree-2 vertices. By temporarily replacing paths in G_i with edges, we can apply Lemma 4.9 to find a cycle K in G_i consisting of edge-disjoint paths $p_1, \dots, p_m$ where $m \leq 2 \log n$. Let x_j be the variable associated with the first edge in p_j. For each j, there are precisely two assignments to the variables in p_j which do not immediately falsify some axiom of $\mathrm{TS}_{G_i,\gamma_i}$. Let α be a partial assignment which picks one of these two assignments for each p_j, and such that $\alpha(x_1) = 1$. As in case 2, SPR^- inferences can be used to introduce $\overline{\alpha}$ for each α of this form.

Let us look at the part of $\overline{\alpha}$ consisting of literals from path p_j. This has the form $z^j_1 \vee \cdots \vee z^j_r$, where z^j_1 is x_j with positive or negative sign and for each k, by the choice of α, there are Tseitin axioms expressing that z^j_k and z^j_{k+1} have the same value. Hence if we set $z^j_1 = 0$ we can set all literals in this clause to 0 by unit propagation. Applying the same argument to all parts of α shows that we can derive $z^1_1 \vee \cdots \vee z^m_1$ from $\overline{\alpha}$ and $\mathrm{TS}_{G_i,\gamma_i}$ with a single $\vdash_1$ step. We introduce all 2^{m-1} such clauses, one for each α, all with $z^1_1 = \overline{x_1}$. We resolve them together to get the unit clause $\overline{x_1}$, then proceed as in case 2.

For the size bound, each case above requires us to derive at most $n \cdot |\mathrm{TS}_{G,\gamma}|$ clauses, and the refutation can take at most n steps. □

4.6 Or-ification and xor-ification

Orification and xorification have been widely used to make hard instances of propositional tautologies, see [6, 5, 43]. This and the next section discuss how SPR^- inferences can be used to "undo" the effects of orification, xorification, and lifting without using any new variables. As a consequence, these techniques are not likely to be helpful in establishing lower bounds for the size of PR^- refutations.

Typically, one "orifies" many variables at once; however, for the purposes of this paper, we describe orification of a single variable. Let Γ be a set of clauses, and x a variable. For the m-fold *orification* of x, we introduce new variables $x_1, \dots, x_m$, with the intent of replacing x with $x_1 \vee x_2 \vee \cdots \vee x_m$. Specifically, each clause $x \vee C$ in Γ is replaced with $x_1 \vee \cdots \vee x_m \vee C$, and each clause $\overline{x} \vee C$ is replaced with the m-many clauses $\overline{x_j} \vee C$. Let $\Gamma^{\vee}$ denote the results of this orification of x. We claim that SPR^- inferences may be used to derive Γ (with x renamed to x_1) from $\Gamma^{\vee}$, undoing the orification, as follows. We first use SPR^- inferences to derive each clause $x_1 \vee \overline{x_j}$ for $j > 1$. This is done using Lemma 4.2, with α_j setting x_1 to 0 and x_j to 1, and τ_j setting x_1 to 1 and x_j to 0, so that τ_j is α_j with x_1 and x_j swapped. Thus any clause $x_1 \vee \cdots \vee x_m \vee C$ in $\Gamma^{\vee}$ can be resolved with these to yield $x_1 \vee C$, and for clauses $\overline{x_1} \vee C$ in $\Gamma^{\vee}$ we do not need to change anything.

Xorification of x is a similar construction, but now we introduce m new variables with the intent of letting x be expressed by $x_1 \oplus x_2 \oplus \cdots \oplus x_m$. Each clause $x \vee C$ in Γ (respectively, $\overline{x} \vee C$ in Γ) is replaced by 2^{m-1} many clauses $x_1^\sigma \vee x_2^\sigma \vee \cdots \vee x_m^\sigma \vee C$ where σ is a partial assignment setting an odd number (respectively, an even number) of the variables x_j to 1. To undo the xorification it is enough to derive the unit clauses $\overline{x_j}$ for $j > 1$. So for each $j > 1$, we first use Lemma 4.2 to introduce the clause $x_1 \vee \overline{x}_j$, using the same partial assignments as in the previous paragraph, and the clause $\overline{x_1} \vee \overline{x_j}$, using assignments α_j setting x_1 and x_j both to 1, and τ_j setting x_1 and x_j both to 0, so that τ_j is α_j with the signs of both x_1 and x_j flipped. Resolving these gives $\overline{x}_j$. This subsumes $x_1 \vee \overline{x}_j$ and $\overline{x_1} \vee \overline{x_j}$, so by Lemma 1.20(a), we may ignore these two clauses in later SPR$^-$ steps, and can thus use the same argument to derive the clauses $\overline{x}_i$ for $i \neq j$, since α_i and τ_i do not affect the clause $\overline{x}_j$.

4.7 Lifting

Lifting is a technique for leveraging lower bounds on decision trees to obtain lower bounds in stronger computational models, see [36, 3, 22].

The most common form of lifting is the "indexing gadget" where a single variable x is replaced by $\ell + 2^\ell$ new variables $y_1, \dots, y_\ell$ and $z_0, \dots, z_{2^\ell - 1}$. The intent is that the variables $y_1, \dots, y_\ell$ specify an integer $i \in [2^\ell]$, and z_i gives the value of x. As in Section 4.2, we write $(\vec{y} \to i)$ for the conjunction $\bigwedge_j (y_j = i_j)$ where i_j is the j-th bit of i, and write $(\vec{y} \not\to i)$ for its negation $\bigvee_j (y_j \neq i_j)$. Thus, x is equivalent to the CNF formula $\bigwedge_{i \in [2^\ell]} \big((\vec{y} \not\to i) \vee z_i\big)$ and $\overline{x}$ is equivalent to the CNF formula $\bigwedge_{i \in [2^\ell]} \big((\vec{y} \not\to i) \vee \overline{z_i}\big)$.

Let Γ is a set of clauses with an SPR$^-$ refutation. The indexing gadget applied to Γ on the variable x does the following to modify Γ to produce set of lifted clauses Γ': Each clause $x \dot\vee C$ containing x is replaced by the 2^ℓ clauses $(\vec{y} \not\to i) \vee z_i \vee C$ for $i \in [2^\ell]$, and each clause $\overline{x} \dot\vee C$ containing $\overline{x}$ is replaced by the 2^ℓ clauses $(\vec{y} \not\to i) \vee \overline{z_i} \vee C$.

For all $i \neq 0$ and all $a, b \in \{0, 1\}$, let $\alpha_{i,a,b}$ and $\tau_{i,a,b}$ be the partial assignments

$$\begin{aligned} \alpha_{i,a,b} &:= (\vec{y} \to i) \wedge z_0 = a \wedge z_i = b \\ \tau_{i,a,b} &:= (\vec{y} \to 0) \wedge z_0 = b \wedge z_i = a. \end{aligned}$$

Since $i \neq 0$ always holds, it is immediate that conditions 2. and 3. of Lemma 4.2 hold. For condition 1., observe that the set of clauses $\{(\vec{y} \not\to j) \vee z_j \vee C : j \in [2^\ell]\}$, restricted by $(\vec{y} \to i)$, becomes the single clause $z_i \vee C$, and restricted by $(\vec{y} \to 0)$ becomes $z_0 \vee C$. In this way $\Gamma'_{|\alpha_{i,a,b}} = \Gamma'_{|\tau_{i,a,b}}$ and condition 1. also holds. Therefore by Lemma 4.2, SPR$^-$ inferences can be used to derive all clauses $\overline{\alpha_{i,a,b}}$, namely all the clauses $(\vec{y} \not\to i) \vee z_0 \neq a \vee z_i \neq b$. For each fixed $i \neq 0$ this is four clauses, which can be resolved together to give the clause $(\vec{y}\not\leftrightarrow i)$. Then from these $2^\ell - 1$ clauses we can obtain by resolution each unit clause y_j for $j = 1, \ldots, \ell$. Finally, using unit propagation with these, we derive the clauses $z_0 \dot\vee C$ and $\overline{z_0} \dot\vee C$ for all original clauses $x \dot\vee C$ and $\overline{x} \dot\vee C$ in Γ. We have thus derived from Γ', using SPR$^-$ and resolution inferences, a copy Γ'' of all the clauses in Γ, except with x replaced with z_0. The other clauses in in Γ' or that were inferred during the process of deriving Γ'' are subsumed by either the unit clauses y_j or the clauses in Γ''. Thus applying part (b) and then part (a) of Lemma 1.20, they do not interfere with any future SPR$^-$ inferences refuting Γ''.

5 Lower bounds

This section gives an exponential separation between DRAT$^-$ and RAT$^-$, by showing that the bit pigeonhole principle BPHP$_n$ requires exponential size refutations in RAT$^-$. This lower bound still holds if we allow some deletions, as long as no initial clause of BPHP$_n$ is deleted. On the other hand, with unrestricted deletions, it follows from Theorems 3.1, 3.3 and 4.4 in this paper that it has polynomial size refutations in DRAT$^-$ and even in DBC$^-$, as well as in SPR$^-$.

Kullmann [29] has already proved related separations for generalized extended resolution (GER), which lies somewhere between DBC and BC in strength. That work shows separations between various subsystems of GER, and in particular gives an exponential lower bound on proofs of PHP$_n$ in the system GER with no new variables, by analyzing which clauses are blocked with respect to PHP$_n$.

We define the *pigeon-width* of a clause or assignment to equal the number of distinct pigeons that it mentions. Our size lower bound for BPHP$_n$ uses a conventional strategy: we first show a width lower bound (on pigeon-width), and then use a random restriction to show that a proof of subexponential size can be made into one of small pigeon-width. We do not aim for optimal constants.

We have to be careful about one technical point in the second step, which is that RAT$^-$ refutation size does not in general behave well under restrictions, as discussed in Section 2.2. So, rather than using restrictions as such to reduce width, we will define a partial random matching ρ of pigeons to holes and show that if BPHP$_n$ has a RAT$^-$ refutation of small size, then BPHP$_n$ $\cup$ ρ has one of small pigeon-width.

A useful tool in analyzing resolution derivations from a set of clauses Γ is the *Prover-Adversary game* on Γ (see e.g. [34, 2]). In the game, the Adversary claims to know a satisfying assignment for Γ, and the Prover tries to force her into a contradiction by querying the values of variables; the Prover can also forget variable assignments to save memory and simplify his strategy. A *position* in the game is a partial assignment α recording the contents of the Prover's memory. To fully specify the game we also need to specify the starting position.

As the next lemma shows, a strategy for the Prover in this game is essentially the same thing as a resolution derivation. But it is more intuitive to describe a strategy than a derivation, and the game also gives a natural way to show width lower bounds.

Lemma 5.1. *Consider a restriction of the Prover-Adversary game on a set of clauses* Γ *starting from position* $\overline{C}$ *in which the Prover's memory can hold information about at most* m *pigeons simultaneously. If the Prover has a winning strategy in this game, then* C *is derivable from* Γ *in pigeon-width* m*. If the Adversary has a winning strategy, then* C *is not derivable from* Γ *in pigeon-width* $m-1$.

Proof. We can think of a winning strategy for the Prover as a tree in which the nodes are labelled with a partial assignment (the position α) and with the Prover's action in that situation, that is: query a variable, forget a variable, or declare victory because α falsifies a clause from Γ. We can make this into a resolution refutation by replacing each label α with the clause negating it and interpreting the three actions as respectively resolution, weakening, and deriving a clause from an axiom by weakening.

For the other direction, it is enough to construct a winning strategy for the Prover from a derivation of pigeon-width $m-1$. This is the reverse of the process described above, except that we need to be careful with the resolution rule. Suppose an instance of the rule is: from $p \vee D$ and $\overline{p} \vee E$ derive $D \vee E$, where each clause mentions at most $m-1$ pigeons. In the Prover-strategy, this becomes: from position $\overline{D} \cup \overline{E}$, query p. If it is false, forget some variables from $\overline{E}$ to reach position $\overline{p} \cup \overline{D}$. If it is true, forget some variables from $\overline{D}$ to reach position $p \cup \overline{E}$. Thus the Prover may be in a position mentioning m pigeons immediately after p is queried. □

Lemma 5.2. *Let* β *be a partial assignment corresponding to a partial matching of* m *pigeons to holes. Then* $\mathrm{BPHP}_n \cup \beta$ *requires pigeon-width* $n-m$ *to refute in resolution.*

Proof. $\mathrm{BPHP}_n \cup \beta$ is essentially an unusual encoding of the pigeonhole principle with $n+1-m$ pigeons and $n-m$ holes. Thus, if the Prover is limited to remembering variables from at most $n-m$ pigeons, there is an easy strategy for the Adversary in the game starting from the empty position. Namely, she can always maintain a matching between the pigeons mentioned in the Prover's memory and the available holes. The result follows by Lemma 5.1. □

Theorem 5.3. *Let ρ be a partial matching of size at most $n/4$. Let Π be a* DRAT^- *refutation of* $\mathrm{BPHP}_n \cup \rho$ *in which no clause of* BPHP_n *is ever deleted. Then some clause in Π has pigeon-width more than $n/3$.*

Proof. Suppose for a contradiction there is a such a refutation Π in pigeon-width $n/3$. We consider each RAT inference in Π in turn, and show it can be eliminated and replaced with standard resolution reasoning, without increasing the pigeon-width.

Inductively suppose Γ is a set of clauses derivable from $\mathrm{BPHP}_n \cup \rho$ in pigeon-width $n/3$, using only resolution and weakening. Suppose a clause C in Π of the form $p \dot\vee C'$ is RAT with respect to Γ and p. Let $\alpha = \overline{C}$, so $\alpha(p) = 0$ and α mentions at most $n/3$ pigeons. We consider three cases.

Case 1: the assignment α is inconsistent with ρ. This means that ρ satisfies a literal which appears in C, so C can be derived from ρ by a single weakening step.

Case 2: the assignment $\alpha \cup \rho$ can be extended to a partial matching β of the pigeons it mentions. We will show that this cannot happen. Let x be the pigeon associated with the literal p. Let $y = \beta(x)$ and let y' be the hole β would map x to if the bit p were flipped to 1. If $y' = \beta(x')$ for some pigeon x' in the domain of β, let $\beta' = \beta$. Otherwise let $\beta' = \beta \cup \{(x', y')\}$ for some pigeon x' outside the domain of β.

Let H be the hole axiom $(x \not\to y') \vee (x' \not\to y')$ in Γ. The clause $(x \not\to y')$ contains the literal $\overline{p}$, since $(x \to y')$ contains p. So $H = \overline{p} \dot\vee H'$ for some clause H'. By the RAT condition, either $C' \cup H'$ is a tautology or $\Gamma \vdash_1 C \vee H'$. Either way, $\Gamma \cup \overline{C} \cup \overline{H'} \vdash_1 \bot$. Since $\beta' \supseteq \alpha$, β' falsifies C. It also falsifies H', since it satisfies $(x \to y') \wedge (x' \to y')$ except at p. It follows that $\Gamma \cup \beta' \vdash_1 \bot$. By assumption, Γ is derivable from $\mathrm{BPHP}_n \cup \rho$ in pigeon-width $n/3$, and β' extends ρ. Since unit propagation does not increase pigeon-width, this implies that $\mathrm{BPHP}_n \cup \beta'$ is refutable in resolution in pigeon-width $n/3$, by first deriving Γ and then using unit propagation. This contradicts Lemma 5.2 as β' is a matching of at most $n/3 + n/4 + 1$ pigeons.

Case 3: the assignment $\alpha \cup \rho$ cannot be extended to a partial matching of the pigeons it mentions. Consider the Prover-Adversary game on $\mathrm{BPHP}_n \cup \rho$ with starting position α. The Prover can ask all remaining bits of the pigeons mentioned in α, and since there is no suitable partial matching this forces the Adversary to reveal a collision and lose the game. This strategy has pigeon-width $n/3$; it follows that C is derivable from $\mathrm{BPHP}_n \cup \rho$ in resolution in this pigeon-width, as required. □

Theorem 5.4. *Let Π be a* DRAT^- *refutation of* BPHP_n *in which no clause of* BPHP_n *is ever deleted. Then Π has size at least $2^{n/60}$.*

Proof. Construct a random restriction ρ by selecting each pigeon independently with probability $1/5$ and then randomly matching the selected pigeons with distinct holes (there is an $(1/5)^{n+1}$ chance that there is no matching, because we selected all the pigeons — in this case we set all variables at random).

Let $m = n/4$. Let C be a clause mentioning at least m distinct pigeons $x_1, \ldots, x_m$ and choose literals $p_1, \ldots, p_m$ in C such that p_i belongs to pigeon x_i. The probability that p_i is satisfied by ρ is $1/10$. However, these events are not quite independent for different i, as the holes used by other pigeons are blocked for pigeon x_i. To deal with this, we may assume that pigeons $x_1, \ldots, x_m$, in that order, were the first pigeons considered in the construction of ρ. When we come to x_i, if we set it, then there are $n/2$ holes which would satisfy p_i, at least $n/2 - m \geq n/4$ of which are free; so of the free holes, the fraction which satisfy p_i is at least $1/3$. So the probability that ρ satisfies p_i, conditioned on it not satisfying any of $p_1, \ldots, p_{i-1}$, is at least $1/15$. Therefore the probability that C is not satisfied by ρ is at most $(1 - 1/15)^m < e^{-m/15} = e^{-n/60}$.

Now suppose Π contains no more than $2^{n/60}$ clauses of pigeon width at least $n/4$. By the union bound, for a random ρ, the probability that at least one of these clauses is not satisfied by ρ is at most $2^{n/60}e^{-n/60} = (2/e)^{n/60}$. Therefore most restrictions ρ satisfy all clauses in Π of pigeon-width at least $n/4$, and by the Chernoff bound we may choose such a ρ which also sets no more than $n/4$ pigeons.

We now observe inductively that for each clause C in Π, some subclause of C is derivable from $\mathrm{BPHP}_n \cup \rho$ in resolution in pigeon-width $n/3$, ultimately contradicting Lemma 5.2. If C has pigeon-width more than $n/3$, this follows because C is subsumed by ρ. Otherwise, if C is derived by a RAT inference, we repeat the proof of Theorem 5.3; in case 2 we additionally use the observation that if $\Gamma \vdash_1 C \vee H'$ and Γ' subsumes Γ, then $\Gamma' \vdash_1 C \vee H'$. $\square$

Corollary 5.5. RAT^- *does not simulate* DRAT^-. RAT^- *does not simulate* SPR^-.

Proof. By Theorem 4.4, BPHP_n has short proofs in SPR^-. Thus, by Theorem 3.3, this also holds for DRAT^- (and for DBC^- by Theorem 3.1). On the other hand, Theorem 5.4 just showed BPHP_n requires exponential size RAT^- proofs. $\square$

6 Open problems

There are a number of open questions about the systems with no new variables. Of particular importance is the question of the relative strengths of DPR^-, DSR^- and related systems. The results of [16, 19, 20] and the present paper show that DPR^-, and even the possibly weaker system SPR^-, are strong. DPR^- is a promising system for effective proof search algorithms, but it is open whether practical proof search algorithms can effectively exploit its strength. It is also open whether DPR^- or DSR^- simulates ER.

Another important question is to understand the strength of deletion for these systems. Of course, deletion is well-known to help the performance of SAT solvers in practice, if for no other reason, because unit propagation is faster when fewer clauses are present. In addition, for systems such as RAT, it is known that deletion can allow new inferences. The results in Sections 4 and 5 improve upon this by showing that RAT$^-$ does not simulate DRAT$^-$. This strengthens the case, at least in theory, for the importance of deletion.

In Section 4 we described small SPR$^-$ proofs of many of the known "hard" tautologies that have been shown to require exponential size proofs in constant depth Frege. It is open whether SPR$^-$ simulates Frege; and by these results, any separation of SPR$^-$ and Frege systems will likely require developing new techniques. Even more tantalizing, we can ask whether SR$^-$ simulates Frege.

There are several hard tautologies for which we do not whether there are as polynomial size SPR$^-$ proofs. Jakob Nordström [personal communication, 2019] suggested (random) 3-XOR SAT and the even coloring principle as examples. 3-XOR SAT has short cutting planes proofs via Gaussian elimination; it is open whether SPR$^-$ or DSPR$^-$ or even DSR$^-$ has polynomial size refutations for all unsatisfiable 3-XOR SAT principles. The even coloring principle is a special case of the Tseitin principle [30]: the graph has an odd number of edges, each vertex has even degree, and the initial clauses assert that, for each vertex, exactly one-half the incident edges are labeled 1. It is not hard to see that the even coloring principle can be weakened to the Tseitin principle by removing some clauses with the deletion rule. Hence there are polynomial size DSPR$^-$ refutations (with deletion) of the even coloring principle. It is open whether SPR$^-$ (without deletion) has polynomial size refutations for the even coloring principle.

Paul Beame [personal communication, 2018] suggested that the graph PHP principles (see [7]) may separate systems such as SPR$^-$ or even SR$^-$ from Frege systems. However, there are reasons to suspect that in fact the graph PHP principles also have short SPR$^-$ proofs. Namely, SPR inferences can infer a lot of clauses from the graph PHP clauses. If an instance of graph PHP has every pigeon with outdegree ≥ 2, then there must be an alternating cycle of pigeons $i_1, \dots i_{\ell+1}$ and holes $j_1, \dots j_\ell$ such that $i_\ell = i_1$, the edges (i_s, j_s) and (i_{s+1}, j_s) are all in the graph, and $\ell = O(\log n)$. Then an SPR inference can be used to learn the clause $\overline{x_{i_1,j_1}} \vee \overline{x_{i_2,j_2}} \vee \cdots \vee \overline{x_{i_\ell,j_\ell}}$, by using the fact that a satisfying assignment that falsifies this clause can be replaced by the assignment that maps instead each pigeon i_{s+1} to hole j_s.

This construction clearly means that SPR inferences can infer many clauses from the graph PHP clauses. However, we do not know how to use these to form a short SPR$^-$ proof of the graph PHP principles. It remains open whether a polynomial size SPR$^-$ proof exists.

Acknowledgements We thank the reviewers of the conference and journal versions of this paper for suggestions and comments that improved the paper. We also thank Jakob Nordström, Paul Beame, Marijn Heule, Thomas Kochmann and Oliver Kullmann for useful comments, questions and suggestions.

References

[1] M. Ajtai, *Parity and the pigeonhole principle*, in Feasible Mathematics: A Mathematical Sciences Institute Workshop held in Ithaca, New York, June 1989, Birkhäuser, 1990, pp. 1–24.

[2] A. Atserias and V. Dalmau, *A combinatorial characterization of resolution width*, Journal of Computer and System Sciences, 74 (2008), pp. 323–334.

[3] P. Beame, T. Huynh, and T. Pitassi, *Hardness amplification in proof complexity*, in Proc. 42nd Annual ACM Symposium on Theory of Computing (STOC'10), 2010, pp. 87–96.

[4] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák, *Lower bounds on Hilbert's Nullstellensatz and propositional proofs*, Proceedings of the London Mathematical Society, 73 (1996), pp. 1–26.

[5] E. Ben-Sasson, *Size space tradeoffs for resolution*, SIAM Journal on Computing, 38 (2009), pp. 2511–2525.

[6] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson, *Near optimal separation of tree-like and general resolution*, Combinatorica, 24 (2004), pp. 585–603.

[7] E. Ben-Sasson and A. Wigderson, *Short proofs are narrow — resolution made simple*, Journal of the ACM, 48 (2001), pp. 149–169.

[8] S. Buss and N. Thapen, *DRAT proofs, propagation redundancy, and extended resolution*, in Proc. 22nf Intl. Conference on Theory and Applications of Satisfiability Testing (SAT 2019), Springer-Verlag Lecture Notes in Computer Science 11628, 2019, pp. 71–89.

[9] S. R. Buss, *Bounded Arithmetic*, Bibliopolis, Naples, Italy, 1986. Revision of 1985 Princeton University Ph.D. thesis.

[10] C. L. Chang, *The unit proof and the input proof in theorem proving*, J. ACM, 17 (1970), pp. 698–707.

[11] S. A. Cook, *Feasibly constructive proofs and the propositional calculus*, in Proceedings of the Seventh Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, 1975, pp. 83–97.

[12] S. A. Cook and R. A. Reckhow, *On the lengths of proofs in the propositional calculus, preliminary version*, in Proceedings of the Sixth Annual ACM Symposium on the Theory of Computing, 1974, pp. 135–148.

[13] ——, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic, 44 (1979), pp. 36–50.

[14] E. I. Goldberg and Y. Novikov, *Verification of proofs of unsatisfiability for CNF formulas*, in Design, Automation and Test in Europe Conference (DATE), IEEE Computer Society, 2003, pp. 10886–10891.

[15] J. Grollmann and A. Selman, *Complexity measures for public-key cryptosystems*, SIAM Journal on Computing, 17 (1988), pp. 309–335.

[16] M. J. H. Heule and A. Biere, *What a difference a variable makes*, in Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference (TACAS 2018), Lecture Notes in Computer Science 10806, Springer Verlag, 2018, pp. 75–92.

[17] M. J. H. Heule, W. A. Hunt Jr., and N. Wetzler, *Trimming while checking clausal proofs*, in Formal Methods in Computer-Aided Design (FMCAD), IEEE, 2013, pp. 181–188.

[18] ——, *Verifying refutations with extended resolution*, in Automated Deduction - 24th International Conference (CADE), Lecture Notes in Computer Science 7898, Springer Verlag, 2013, pp. 345–359.

[19] M. J. H. Heule, B. Kiesl, and A. Biere, *Short proofs without new variables*, in Automated Deduction - 26th International Conference (CADE), Lecture Notes in Computer Science 10395, Springer Verlag, 2017, pp. 130–147.

[20] ——, *Strong extension-free proof systems*, Journal of Automated Reasoning, (2019), pp. 1–22. Extended version of [19].

[21] M. J. H. Heule, B. Kiesl, M. Seidl, and A. Biere, *PRuning through satisfaction*, in Hardware and Software: Verification and Testing - 13th International Haifa Verification Conference (HVC), Lecture Notes in Computer Science 10629, Springer Verlag, 2017, pp. 179–194.

[22] T. Huynh and J. Nordström, *On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract)*, in Proc. 44th Annual ACM Symposium on Theory of Computing (STOC'12), 2012, pp. 233–248.

[23] M. Järvisalo, M. J. H. Heule, and A. Biere, *Inprocessing rules*, in Automated Reasoning - 6th International Joint Conference (IJCAR), Lecture Notes in Computer Science 7364, Springer Verlag, 2012, pp. 355–270.

[24] B. Kiesl, A. Rebola-Pardo, and M. J. H. Heule, *Extended resolution simulates DRAT*, in Automated Reasoning - 6th International Joint Conference (IJCAR), Lecture Notes in Computer Science 10900, Springer Verlag, 2018, pp. 516–531.

[25] J. Krajíček, *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*, Journal of Symbolic Logic, 62 (1997), pp. 457–486.

[26] J. Krajíček, P. Pudlák, and A. Woods, *Exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle*, Random Structures and Algorithms, 7 (1995), pp. 15–39.

[27] O. Kullmann, *Worst-case analysis, 3-SAT decision and lower bounds: Approaches for improved SAT algorithms*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 35 (1997), pp. 261–313.

[28] ——, *New methods for 3-SAT decision and worst-case analysis*, Theoretical Computer Science, 223 (1999), pp. 1–72.

[29] ——, *On a generalizaton of extended resolution*, Discrete Applied Mathematics, 96-97 (1999), pp. 149–176.

[30] K. Markström, *Locality and hard SAT-instances*, Journal on Satisfiability, Boolean Modeling and Computation (JSAT), 2 (2006), pp. 221–227.

[31] T. Pitassi, P. Beame, and R. Impagliazzo, *Exponential lower bounds for the pigeonhole principle*, Computational Complexity, 3 (1993), pp. 97–140.

[32] T. Pitassi and R. Santhanam, *Effectively polynomial simulations*, in Innovations in Computer Science (ICS), 2010, pp. 370–381.

[33] P. Pudlák, *Lower bounds for resolution and cutting planes proofs and monotone computations*, Journal of Symbolic Logic, 62 (1997), pp. 981–998.

[34] ——, *Proofs as games*, American Mathematical Monthly, 107 (2000), pp. 541–550.

[35] ——, *On reducibility and symmetry of disjoint NP pairs*, Theoretical Computer Science, 205 (2003), pp. 323–339.

[36] R. RAZ AND P. MCKENZIE, *Separation of the monotone NC hierarchy*, Combinatorica, 19 (1999), pp. 403–435.

[37] A. A. RAZBOROV, *On provably disjoint NP-pairs*, Tech. Rep. TR94-006, Electronic Colloquium in Computational Complexity (ECCC), 1994. Also available as BRIC technical report RS-94-36.

[38] A. REBOLA-PARDO AND M. SUDA, *A theory of satisfiability-preserving proofs in SAT solving*, in Proc., 22nd Intl. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'22), EPiC Series in Computing 57, EasyChair, 2018, pp. 583–603.

[39] J. SIEKMANN AND G. WRIGHTSON, *Automation of Reasoning*, vol. 1&2, Springer-Verlag, Berlin, 1983.

[40] S. SZEIDER, *Homomorphisms of conjunctive normal forms*, Discrete Applied Mathematics, 130 (2003), pp. 351–365.

[41] G. S. TSEJTIN, *On the complexity of derivation in propositional logic*, Studies in Constructive Mathematics and Mathematical Logic, 2 (1968), pp. 115–125. Reprinted in: [39, vol 2], pp. 466-483.

[42] A. URQUHART, *Hard examples for resolution*, J. Assoc. Comput. Mach., 34 (1987), pp. 209–219.

[43] ——, *A near-optimal separation of regular and general resolution*, SIAM Journal on Computing, 40 (2011), pp. 107–121.

[44] A. VAN GELDER, *Verifying RUP proofs of propositional unsatisfiability*, in 10th International Symposium on Artificial Intelligence and Mathematics (ISAIM), 2008. http://isaim2008.unl.edu/index.php?page=proceedings.

[45] N. WETZLER, M. J. H. HEULE, AND W. A. HUNT JR., *DRAT-trim: Efficient checking and trimming using expressive clausal proofs*, in Theory and Applications of Satisfiability Testing - 17th International Conference (SAT), Lecture Notes in Computer Science 8561, Springer Verlag, 2014, pp. 422–429.