Electronic Colloquium on Computational Complexity, Report No. 33 (2024)

Regular resolution effectively simulates resolution

Sam Buss* Emre Yolcu†

February 24, 2024

Abstract

Regular resolution is a refinement of the resolution proof system requiring that no variable be resolved on more than once along any path in the proof. It is known that there exist sequences of formulas that require exponential-size proofs in regular resolution while admitting polynomial-size proofs in resolution. Thus, with respect to the usual notion of simulation, regular resolution is separated from resolution. An alternative, and weaker, notion for comparing proof systems is that of an "effective simulation," which allows the translation of the formula along with the proof when moving between proof systems. We prove that regular resolution is equivalent to resolution under effective simulations. As a corollary, we recover in a black-box fashion a recent result on the hardness of automating regular resolution.

Keywords: *resolution, regular resolution, effective simulation, automatability, proof complexity*

1 Introduction

Proof complexity studies the sizes of proofs[1] in propositional proof systems. A common question in proof complexity is that of the relative strengths of different systems. The comparison is performed typically with respect to the notion of a "simulation" [CR79, Definition 1.5]. A system P *simulates* another system Q if every Q-proof can be converted, with at most a polynomial increase in size, into a P-proof of the same formula. An alternative, and weaker, notion of simulation is the following, which is arguably more natural from an algorithmic point of view. (See Pitassi and Santhanam [PS10, Section 1] for a discussion.)

Definition 1.1 ([PS10, Definition 2.5])**.** Let P and Q be two proof systems for a class $\mathcal{C}$ of propositional formulas. The *size*, $|\Gamma|$, of a formula Γ is defined to equal the number of symbols in the formula. We say P *effectively simulates* Q if there exists a function $f\colon \mathcal{C} \times \mathbb{N} \to \mathcal{C}$ such that the following hold.

*Department of Mathematics, University of California, San Diego. Email: sbuss@ucsd.edu.
†Computer Science Department, Carnegie Mellon University. Email: eyolcu@cs.cmu.edu.

[1]Throughout this work, by "proof" we mean a refutation of satisfiability.

ISSN 1433-8092

- The formula $f(\Gamma, s)$ can be computed in time polynomial in $|\Gamma| + s$, and it is satisfiable if and only if Γ is.
- When s is at least the size of the smallest Q-proof of Γ, the formula $f(\Gamma, s)$ has a P-proof of size polynomial in $|\Gamma| + s$.

Remark 1.2. As remarked by Pitassi and Santhanam [PS10], the role of the size parameter s in the above definition might not be clear at first glance. It would be simpler to define a notion of *strict effective simulation* by omitting s and requiring that $f(\Gamma)$ be computable in time polynomial in $|\Gamma|$ and that $f(\Gamma)$ have a P-proof of size polynomial in the size of the smallest Q-proof of Γ. A major motivation behind Definition 1.1 is its relationship to "weak automatability" [AB04, Definition 4], which we define in Section 4.2, and the relaxed definition suffices for the relationship to hold.

Effective simulations exist in several instances where either no simulation is known or a separation exists. Examples include the following,[2] where $P \geq Q$ denotes that P effectively simulates Q:

- linear resolution $\geq$ resolution [BP07, BJ16]
- clause learning $\geq$ resolution [HBPV08, BHJ08]
- resolution $\geq$ k-DNF resolution [AB04]
- blocked clauses without new variables $\geq$ extended resolution [BT21]
- G_0 ("quantified Frege") $\geq$ any quantified propositional proof system [PS10]
- CP (cutting planes) $\geq$ CP with quadratic terms [Pud03]
- constant-depth polynomial calculus $\geq$ CP, Positivstellensatz calculus [IMP20]

In this work we prove that although regular resolution and resolution are separated with respect to the usual notion of simulation [Goe93, AJPU07, Urq11], the two systems are equivalent under effective simulations. As a technical side note, although in most of the known effective simulations the size parameter s is not needed (i.e., they are strict effective simulations), in our simulation it is necessary that f have access to the parameter. It is an open question whether regular resolution strictly effectively simulates resolution.

2 Preliminaries

We assume that the reader is familiar with propositional logic, proof complexity, and resolution. We review some concepts to describe our notation. For all $n \in \mathbb{N}$, we let $[n] := \{1, \ldots, n\}$. For X a nonempty list of variables, we write $\mathrm{poly}(X)$ to denote a quantity bounded by some polynomial in X.

[2] Some results use a version of Definition 1.1 that allows comparing proof systems over different languages [HHU07, Definition 4].

A *literal* is a propositional variable x or its negation $\overline{x}$. Overline denotes negation, and if p is the literal $\overline{x}$, then $\overline{p}$ is x. A *clause* is a disjunction of literals. We use $\perp$ to denote the empty clause. A formula in *conjunctive normal form* (CNF) is a conjunction of clauses. Throughout this work, by "formula" we mean a formula in CNF. We identify clauses with sets of literals and formulas with sets of clauses. For a formula Γ, we denote by $\mathrm{var}(\Gamma)$ the set of all the variables occurring in Γ. In particular, for a literal p of a variable x, we have $\mathrm{var}(p) = x$.

Definition 2.1. The *resolution rule* is

$$\frac{A \vee x \qquad B \vee \overline{x}}{A \vee B},$$

where A, B are clauses and x is a variable not occurring in A or B. We call $A \vee B$ the *resolvent* of $A \vee x$ and $B \vee \overline{x}$ on x.

Definition 2.2. A *resolution derivation* from a formula Γ is a sequence $\Pi = C_1, \ldots, C_s$ of distinct clauses such that for all $i \in [s]$, the clause C_i either occurs in Γ or is a resolvent of two earlier clauses in the sequence. If $C_s = \perp$, then Π is a *resolution refutation* of Γ. The *size* of Π is s.

Remark 2.3. A resolution derivation Π from Γ can be viewed as a directed acyclic graph: The nodes of the graph are the clauses in Π; every *initial clause* (i.e., a clause in Γ) has in-degree zero, and every other clause D has two incoming edges from the premises of the resolution inference that derives D. Thus, every node has in-degree zero or two. The *height* of a resolution derivation is the number of edges in the longest directed path in the graph.

Definition 2.4. A resolution derivation Π is *regular* if no variable is resolved on more than once along any directed path in Π.

In the rest of this paper, by "path" we mean a directed path.

3 Main result

Theorem 3.1. *Regular resolution effectively simulates resolution.*

Proof. Let Γ be a formula with n variables, and let h be a size parameter. We will define a new formula $f(\Gamma, h)$ such that if Γ has a resolution refutation of size s and height h, then $f(\Gamma, h)$ has a regular resolution refutation of size $\mathrm{poly}(s, n)$ and height $\mathrm{poly}(h, n)$.

The formula $f(\Gamma, h)$ is defined by introducing new variables and adding several 2-clauses to Γ. For each variable x of Γ and each $j \in [h-1]$, there is a new variable $W[x, j]$. We refer to j as the *level* of $W[x, j]$. We identify $W[x, h]$ with x. Thus, $f(\Gamma, h)$ has a total of hn variables, with $(h-1)n$ of them new. We extend the notation to define $W[p, j]$ for p a literal by letting $W[\overline{x}, j]$ denote the literal $\overline{W[x, j]}$.

We form $f(\Gamma, h)$ by adding to Γ the 2-clauses expressing for all $j \in [h-1]$ the equivalence $W[x,j] \leftrightarrow W[x,j+1]$. That is,

$$f(\Gamma, h) \coloneqq \Gamma \wedge \bigwedge_{x \in \mathrm{var}(\Gamma)} \bigwedge_{j \in [h-1]} \left[\left(\overline{W[x,j]} \vee W[x,j+1] \right) \wedge \left(W[x,j] \vee \overline{W[x,j+1]} \right) \right]. \quad (1)$$

The formula $f(\Gamma, h)$ is satisfiable if and only if Γ is satisfiable since the added clauses simply define new names for each variable of Γ.

Let $\Pi = C_1, \ldots, C_s$ be a size-s, height-h resolution refutation of Γ, viewed as a directed graph as described in Remark 2.3. To prove the theorem, we will describe how to turn Π into a regular resolution refutation Π' of $f(\Gamma, h)$ of size at most $6hns$ and height at most hn. The intuition for forming the regular refutation is that the new variables $W[x,j]$ are equivalent to x, and the general refutation Π can be turned into a regular refutation by replacing literals p with $W[p,j]$, letting j decrease as the refutation progresses. In this way, multiple resolutions on a variable x are replaced by resolutions on variables $W[x,j]$, with j decreasing along paths in the refutation so that no $W[x,j]$ is resolved on twice on any path.

Let D be a clause in Π. Without loss of generality, there is at least one path in Π from D to $\bot$. For a variable x, the *irregularity height* of x at D in Π, denoted $L_D(x)$, is defined to be the maximum number of inferences that use x as the resolution variable along any path in Π from D to $\bot$. For p a literal, we allow L_D to act on p by letting $L_D(p) \coloneqq L_D(\mathrm{var}(p))$. The value of $L_D(x)$ depends on Π of course, but this is suppressed in the notation. Note that $L_D(x) \leq h$.

For each clause D in Π, let

$$g(D) \coloneqq \bigvee_{p \in D} W[p, L_D(p)].$$

The regular resolution refutation Π' will have the form $P_0, P_1, \ldots, P_s$, where each P_i is a finite sequence of clauses. The sequence P_0 contains the initial clauses from Γ that appear in Π and the newly added 2-clauses in $f(\Gamma, h)$. For all $i \in [s]$, the sequence P_i will end with the clause $g(C_i)$. Since $g(\bot) = \bot$, the final clause of Π' will be $\bot$, so Π' will be a regular resolution refutation of $f(\Gamma, h)$.

The principal tool in forming each P_i will be "lowering" the levels of literals. Specifically, suppose that E and F are clauses of the forms

$$E = W[p_1, j_1] \vee \cdots \vee W[p_t, j_t] \quad \text{and} \quad F = W[p_1, k_1] \vee \cdots \vee W[p_t, k_t]$$

with $j_\ell \geq k_\ell$ for all $\ell \in [t]$. When this holds, we say E *dominates* F. Then F can be derived from E by resolving it with the new $f(\Gamma, h)$ clauses

$$W[p_\ell, m] \vee \overline{W[p_\ell, m+1]} \qquad \text{for } \ell \in [t] \text{ and } m = j_\ell - 1, j_\ell - 2, \ldots, k_\ell.$$

This is called *lowering* E to F.

We now describe how to inductively form the subderivations P_i of Π'. Consider the clause C_i, which we will write simply as C from this point on. The subderivation P_i needs to end with $g(C)$. There are two cases to consider.

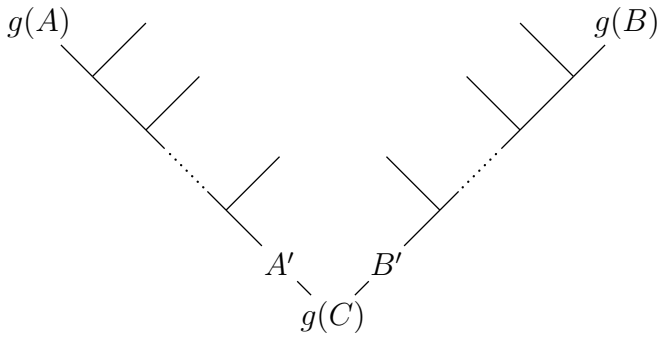

Figure 1: Lowering the premises for a resolution inference in Case 2. The unlabeled leaves correspond to initial clauses from $f(\Gamma, h) \setminus \Gamma$, included earlier in P_0.

Case 1: *C is an initial clause in Π.* Since $C \in \Gamma$, it already appears in P_0. Furthermore, C dominates $g(C)$, so we can lower C to obtain $g(C)$. The subderivation consists of (zero or more) resolutions with 2-clauses to perform the lowering.

Case 2: *C is the resolvent of A and B on y in Π.* The subderivation P_i will lower the earlier-derived clauses $g(A)$ and $g(B)$ to form clauses A' and B' that can be resolved to give the clause $g(C)$. For each literal p, define

$$\lambda(p) \coloneqq \begin{cases} L_C(p) + 1 & \text{if } \mathrm{var}(p) = y \\ L_C(p) & \text{otherwise.} \end{cases}$$

Then A' and B' are defined as

$$A' = \bigvee_{p \in A} W[p, \lambda(p)] \quad \text{and} \quad B' = \bigvee_{p \in B} W[p, \lambda(p)].$$

Since Π contains paths from A and from B that resolve on y and then pass through C, we have $\lambda(p) \leq j$ for all $W[p, j]$ in $g(A)$ or $g(B)$. Therefore, A' and B' are indeed dominated by $g(A)$ and $g(B)$, and thus can be derived in P_i. After that, P_i resolves A' and B' on $W[y, \lambda(y)]$ to derive $g(C)$. Figure 1 summarizes the derivation of $g(C)$ from $g(A)$ and $g(B)$.

The refutation Π' is completed once P_s is formed, as it derives $g(C_s) = \bot$. By construction, for every x, the levels j of the resolution variables $W[x, j]$ are decreasing along all paths in the refutation Π'. Therefore, Π' is a regular resolution refutation.

It is straightforward to see that Π' has size at most $s + 2hn + (2hn + 1)s \leq 6hns$. This bound is calculated as follows: Π' has at most s clauses from Γ. It also has the $\leq 2hn$ many 2-clauses added in the definition (1) of $f(\Gamma, h)$. The term $(2hn + 1)s$ is justified by noting that for each of the $\leq s$ resolution inferences in the original refutation Π, lowering is performed in Case 2 as needed on the literals in the clauses (at most h times on each of the $\leq 2n$ literals in the clauses being resolved) and then one resolution inference is performed by resolving on the lowered version of y.

The bound hn on the height of Π' follows from the fact that it is a regular resolution refutation of a formula with hn variables, since each variable can be resolved on at most once along any path in the refutation Π'. □

The proof of Theorem 3.1 establishes a statement stronger than necessary for the effective simulation to hold. Definition 1.1 allows f to depend on proof size; in our case it suffices for f to depend on height. This dependence is needed in our simulation to ensure that there are sufficiently many variables $W[x, j]$. We leave open whether the dependence can be eliminated:

Question 3.2. Does regular resolution strictly effectively simulate resolution?

4 Corollaries

Theorem 3.1 has some interesting consequences, given in Corollaries 4.2, 4.5, and 4.6. We need a few definitions before we can state the corollaries.

4.1 Closure under substitutions

Let $\mathbf{V}$ and $\mathbf{L}$ denote the sets of all variables and all literals, respectively. A *substitution* is a partial function $\sigma\colon \mathbf{V} \to \mathbf{L} \cup \{\mathit{False}, \mathit{True}\}$. We allow σ to act on literals by letting $\sigma(\overline{x}) \coloneqq \overline{\sigma(x)}$. We call a set *tautological* if it contains *True* or a pair of complementary literals. For a clause C, we define $\sigma(C) \coloneqq \{\sigma(p)\colon p \in C\}$ and

$$C|_\sigma \coloneqq \begin{cases} \mathit{True} & \text{if } \sigma(C) \text{ is tautological} \\ \sigma(C) \setminus \{\mathit{False}\} & \text{otherwise.} \end{cases}$$

For a formula Γ, we define $\Gamma|_\sigma \coloneqq \{C|_\sigma\colon C \in \Gamma \text{ and } C|_\sigma \neq \mathit{True}\}$.

Definition 4.1. A proof system P is *closed under substitutions* if for every formula Γ and every substitution σ, the formula $\Gamma|_\sigma$ has a P-proof of size polynomial in the size of the smallest P-proof of Γ. We say P is *p-closed under substitutions* if there exists an algorithm that, given a size-s P-proof of Γ and a substitution σ, outputs a P-proof of $\Gamma|_\sigma$ in time polynomial in s.

Most of the natural proof systems are closed under substitutions. Theorem 3.1 can be used to prove that this is not the case for regular resolution; in fact, due to the exponential separation between regular resolution and resolution [AJPU07], applying substitutions to formulas can increase their regular resolution refutation complexity exponentially.

Corollary 4.2. *Regular resolution is not closed under substitutions.*

Proof. Let $\{\Gamma_n\}_{n=1}^{\infty}$ be a family of formulas admitting polynomial-size refutations in resolution while requiring exponential-size refutations in regular resolution. For all n, let h_n denote the height of the smallest resolution refutation of Γ_n.

Let f be the formula transformation defined in the proof of Theorem 3.1, and consider a substitution σ_n that maps $W[x, j]$ to x for all $x \in \mathrm{var}(\Gamma_n)$ and $j \in [h_n - 1]$. The formula $f(\Gamma_n, h_n)$ can be refuted in polynomial size in regular resolution, whereas $f(\Gamma_n, h_n)|_{\sigma_n} = \Gamma_n$ requires exponential size. □

4.2 Automatability

Definition 4.3. A proof system P is *automatable* if there exists an algorithm A that, given an unsatisfiable formula Γ, outputs a P-proof of Γ in time polynomial in $|\Gamma| + s$, where s is the size of the smallest P-proof of Γ. We say P is *weakly automatable* if the algorithm A is allowed to output a proof in some other system.

Effective simulations give reductions between weak automatability of proof systems:

Proposition 4.4 ([PS10, Proposition 2.7])**.** *Let P and Q be proof systems. If P effectively simulates Q and P is weakly automatable, then Q is weakly automatable.*

Thus, the following is an immediate corollary of Theorem 3.1.

Corollary 4.5. *If regular resolution is weakly automatable, then so is resolution.*

Although effective simulations do not necessarily give reductions between (strong) automatability, in our case Corollary 4.5 can be extended:

Corollary 4.6. *If regular resolution is automatable, then so is resolution.*

Proof. Let A be an algorithm that automates regular resolution in time bounded by a polynomial t. Let f be the formula transformation defined in the proof of Theorem 3.1, and let u be a polynomial such that when s is at least the size of the smallest resolution refutation of Γ, the formula $f(\Gamma, s)$ has a regular resolution refutation of size $u(|\Gamma| + s)$.

Resolution can be automated as follows: Given an unsatisfiable formula Γ, for each $r = 0, 1, \ldots$, simulate A on $f(\Gamma, r)$ for $t(u(|\Gamma| + r))$ steps until, for some r, it outputs a regular resolution refutation Π of $f(\Gamma, r)$. Let σ be a substitution that maps $W[x, j]$ to x for all $x \in \mathrm{var}(\Gamma)$ and $j \in [r - 1]$. Resolution is p-closed under substitutions and we have $f(\Gamma, r)|_\sigma = \Gamma$, so convert Π into a resolution refutation Π' of Γ and output Π'. □

An analogous result is already known: Atserias and Müller [AM20] proved that resolution is not automatable unless $\mathsf{P} = \mathsf{NP}$, and it was observed afterwards that their result can be extended to regular resolution. However, this extension is a nontrivial step and requires at least an inspection of their proof. (See for instance the preprint by Bell [Bel20] for a detailed writeup.) In contrast, Corollary 4.6 recovers the same extension in a black-box fashion.

Acknowledgments

This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing. Sam Buss's research is supported in part by Simons Foundation grant 578919.

References

[AB04] Albert Atserias and María Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, 2004.

[AJPU07] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory of Computing*, 3:81–102, 2007.

[AM20] Albert Atserias and Moritz Müller. Automating resolution is NP-hard. *Journal of the ACM*, 67(5):31:1–31:17, 2020.

[Bel20] Zoë Bell. Automating regular or ordered resolution is NP-hard. Technical Report 105, Electronic Colloquium on Computational Complexity (ECCC), 2020.

[BHJ08] Samuel R. Buss, Jan Hoffmann, and Jan Johannsen. Resolution trees with lemmas: Resolution refinements that characterize DLL algorithms with clause learning. *Logical Methods in Computer Science*, 4(4:13):1–28, 2008.

[BJ16] Sam Buss and Jan Johannsen. On linear resolution. *Journal on Satisfiability, Boolean Modeling and Computation*, 10(1):23–35, 2016.

[BP07] Joshua Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. *The Journal of Symbolic Logic*, 72(4):1336–1352, 2007.

[BT21] Sam Buss and Neil Thapen. DRAT and propagation redundancy proofs without new variables. *Logical Methods in Computer Science*, 17(2):12:1–12:31, 2021.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[Goe93] Andreas Goerdt. Regular resolution versus unrestricted resolution. *SIAM Journal on Computing*, 22(4):661–683, 1993.

[HBPV08] Philipp Hertel, Fahiem Bacchus, Toniann Pitassi, and Allen Van Gelder. Clause learning can effectively p-simulate general propositional resolution. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 283–290. AAAI Press, 2008.

[HHU07] Alexander Hertel, Philipp Hertel, and Alasdair Urquhart. Formalizing dangerous SAT encodings. In *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, number 4501 in Lecture Notes in Computer Science, pages 159–172. Springer, 2007.

[IMP20] Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi. The surprising power of constant depth algebraic proofs. In *Proceedings of the 35th Symposium on Logic in Computer Science (LICS)*, pages 591–603. Association for Computing Machinery, 2020.

[PS10] Toniann Pitassi and Rahul Santhanam. Effectively polynomial simulations. In *Proceedings of the 1st Innovations in Computer Science (ICS)*, pages 370–382. Tsinghua University Press, 2010.

[Pud03] Pavel Pudlák. On reducibility and symmetry of disjoint NP pairs. *Theoretical Computer Science*, 295(1–3):323–339, 2003.

[Urq11] Alasdair Urquhart. A near-optimal separation of regular and general resolution. *SIAM Journal on Computing*, 40(1):107–121, 2011.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il