The limits of tractability in Resolution-based propositional proof systems

Stefan Dantchev and Barnaby Martin [1]

School of Engineering and Computing Sciences, Durham University, Science Labs, South Road, Durham DH1 3LE, U.K.

Abstract

We study classes of propositional contradictions based on the Least Number Principle (LNP) in the refutation system of Resolution and its generalisations with bounded conjunction, Res(k). We prove that any first-order sentence with no finite models that admits a Σ_1 interpretation of the LNP, relativised to a set that is quantifier-free definable, generates a sequence of propositional contradictions that have polynomially-sized refutations in the system Res(k), for some k. When one considers the LNP with total order we demonstrate that a Π_1 interpretation of this is sufficient to generate such a propositional sequence with polynomially-sized refutations in the system Res(k). On the other hand, we prove that a very simple first-order sentence that admits a Π_1 interpretation of the LNP (with partial and not total order) requires exponentially-sized refutations in Resolution.

Key words: Propositional proof complexity, Resolution with bounded conjunction, Lower bounds

1 Introduction

Many of the outstanding examples of propositional tautologies (contradictions) used in proving lower bounds for propositional proof (refutation) systems are derived in a uniform fashion from first-order (fo) principles. For refutation systems, one takes an fo sentence ϕ without finite models and derives a sequence of propositional contradictions the nth of which asserts that ϕ has a model of size n. The *Pigeonhole principle* (PHP) and *Least number principle* are perhaps the most popular fo principles in this area. The (negation of the) PHP asserts that there is an injection from a set of size n to a set

[1] The second author was supported by EPSRC grant EP/G020604/1.

of size $n-1$, and the (negation of the) LNP asserts that a strict partial order has no minimal element. Thus neither of these fo sentences has a finite model (though each has an infinite model).

A fairly recent sub-branch of Proof Complexity involves the studying of gap phenomena within propositional proof (refutation) systems. The first such result of this kind appeared in [1], where it was noted that the sequence of propositional contradictions derived from an fo sentence ϕ without finite models – as described above – either has 1.) a polynomially-sized refutation in tree-like Resolution, or 2.) requires fully exponential tree-like Resolution refutations. Moreover, the hard Case 2 prevails exactly when ϕ has some (infinite) model. Since the publication of [1], another gap has been discovered based on rank, and not size, for the integer linear programming-based refutation systems of Lovász-Schrijver and Sherali-Adams [2]. In these cases, the separating criterion is again whether or not an infinite model exists for ϕ, with the hard case – of polynomial instead of constant rank – prevailing if it does.

A gap for Resolution, if it exists at all, can not form along the the same lines. It is known that the LNP – which clearly has infinite models – has polynomially-sized refutations in Resolution (while the PHP requires exponential refutations). Further, it has been argued that Resolution is not so suitable a system in which to search for a gap, because of its instability with respect to relativisation. When one considers certain relativisations of the LNP, the ensuing principles become hard for Resolution – requiring exponential-sized refutations [3]. Model-theoretically, the LNP and its relativisations are very similar, whence the argument that looking for a model-theoretic separation in the case of Resolution might be difficult. Perhaps a more robust system in which to search for a separation might be Resolution-with-bounded-conjunction, Res(k), introduced by Krajíĉek in [4] - for any relativisation of the LNP there exists a k such that it admits polynomially-sized Res(k) refutations [3].

In this paper we explore the boundary of tractability – polynomially-sized refutations – in the systems Res(k). We prove that any fo sentence with no finite models that admits a Σ_1 interpretation of the LNP, relativised to a set that is quantifier-free (qf) definable, generates a sequence of propositional contradictions that have polynomially-sized refutations in the system Res(k), for some k. When one considers the LNP with total order we demonstrate that a Π_1 interpretation of this is sufficient to allow polynomially-sized refutations in Res(k), for some k. On the other hand, we prove that a very simple fo sentence that admits a Π_1 interpretation of the LNP (with partial and not total order) – and without relativisation – requires exponentially-sized refutations in Resolution. This fo sentence is exactly a Π_1-variant of the LNP. We conjecture that this same fo sentence requires exponentially-sized refutation in Res(k), for all k. We briefly explore a sequence of Π_{d+1}-variants of the LNP, and conjecture that they (in fact their negations) may be used to separate depth

d-Frege from depth $d+1$-Frege.

The paper is organised as follows. After the preliminaries, we give in Section 3 our upper bounds for Res(k). In Section 4 we give our lower bound for Resolution. Finally, in Section 5 we give some final remarks as well as our conjectures.

An extended abstract of this paper appeared as [5].

2 Preliminaries

Resolution and Res(k).

We denote by $\top$ and $\bot$ the Boolean values "true" and "false", respectively. A *literal* is either a propositional variable or a negated variable. We shall denote literals by small letters, usually ls. A *k-conjunction* (*k-disjunction*) is a conjunction (disjunction) of at most k literals. A *term* (*k-term*) is either a conjunction (k-conjunction) or a constant, $\top$ or $\bot$. We shall use capital letters to denote terms or k-terms, usually Cs for conjunctions and Ds for disjunctions. A *k-DNF* or *k-clause* (*k-CNF*) is a disjunction (conjunction) of an unbounded number of k-conjunctions (k-disjunctions). We shall use calligraphic capital letters to denote k-CNFs or k-DNFs, usually $\mathcal{C}$s for CNFs and $\mathcal{D}$s for DNFs. Sometimes, when clear from the context, we will say "clause" instead of "k-clause", even though, formally speaking, a clause is a 1-clause.

We can now describe the propositional refutation system Res(k), first introduced by Krajíček [6]. It is used *to refute* (i.e. to prove inconsistency) of a given set of k-clauses by deriving the empty clause from the initial clauses. There are four derivation rules:

(1) The $\wedge$-*introduction rule* is

$$\frac{\mathcal{D}_1 \vee \bigwedge_{j\in J_1} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j\in J_2} l_j}{\mathcal{D}_1 \vee \mathcal{D}_2 \vee \bigwedge_{j\in J_1\cup J_2} l_j},$$

provided that $|J_1 \cup J_2| \leq k$.

(2) The *cut (or resolution) rule* is

$$\frac{\mathcal{D}_1 \vee \bigvee_{j\in J} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j\in J} \neg l_j}{\mathcal{D}_1 \vee \mathcal{D}_2},$$

(3) The two *weakening rules* are

$$\frac{\mathcal{D}}{\mathcal{D} \vee \bigwedge_{j \in J} l_j} \quad \text{and} \quad \frac{\mathcal{D} \vee \bigwedge_{j \in J_1 \cup J_2} l_j}{\mathcal{D} \vee \bigwedge_{j \in J_1} l_j},$$

provided that $|J| \leq k$.

A Res(k)-proof can be considered as a directed acyclic graph (DAG), whose sources are the initial clauses, called also axioms, and whose only sink is the empty clause. We shall define *the size of a proof* to be the number of the internal nodes of the graph, i.e. the number of applications of a derivation rule, thus ignoring the size of the individual k-clauses in the refutation.

In principle the k from "Res(k)" could depend on n – an important special case is Res $(\log n)$. In the present paper, however, we shall be concerned only with Res (k) for some constant k.

Clearly, Res(1) is *(ordinary) Resolution*, working on 1-clauses, and using only the cut rule, which becomes the usual resolution rule, and the first weakening rule.

Equivalence between Res (k) and a special class of branching programs.

If we turn a Res (k) refutation of a given set of k-clauses $\mathcal{D}$ upside-down, i.e. reverse the edges of the underlying graph and negate the k-clauses on the vertices, we get a special kind of restricted branching k-program. The restrictions are as follows.

Each vertex is labelled by a k-CNF which partially represents the information that can be obtained along any path from the source to the vertex (this is a *record* in the parlance of [7]). Obviously, the (only) source is labelled with the constant $\top$. There are two kinds of queries, which can be made by a vertex:

(1) Querying a new k-disjunction, and branching on the answer, which can be depicted as follows.

$$\begin{array}{ccc} & \mathcal{C} & \\ & ?\bigvee_{j \in J} l_j & \\ \top \swarrow & & \searrow \bot \\ \mathcal{C} \wedge \bigvee_{j \in J} l_j & & \mathcal{C} \wedge \bigwedge_{j \in J} \neg l_j \end{array} \tag{1}$$

(2) Querying a known k-disjunction, and splitting it according to the answer:

$$\begin{array}{ccc} & \mathcal{C} \wedge \bigvee_{j \in J_1 \cup J_2} l_j & \\ & ? \bigvee_{j \in J_1} l_j & \\ \top \swarrow & & \searrow \bot \\ \mathcal{C} \wedge \bigvee_{j \in J_1} l_j & & \mathcal{C} \wedge \bigvee_{j \in J_2} l_j \end{array} \tag{2}$$

There are two ways of forgetting information,

$$\begin{array}{ccc} \mathcal{C}_1 \wedge \mathcal{C}_2 & & \mathcal{C} \wedge \bigvee_{j \in J_1} l_j \\ \downarrow & \text{and} & \downarrow \quad , \\ \mathcal{C}_1 & & \mathcal{C} \wedge \bigvee_{j \in J_1 \cup J_2} l_j \end{array} \tag{3}$$

the point being that forgetting allows us to equate the information obtained along two different branches and thus to merge them into a single new vertex. A sink of the branching k-program must be labelled with the negation of a k-clause from $\mathcal{D}$. Thus the branching k-program is supposed by default to solve the *Search problem for* $\mathcal{D}$: given an assignment of the variables, find a clause which is falsified under this assignment.

The equivalence between a Res (k) refutation of $\mathcal{D}$ and a branching k-program of the kind above is obvious. Naturally, if we allow querying single variables only, we get branching 1-programs – decision DAGs – that correspond to Resolution. If we do not allow the forgetting of information, we will not be able to merge distinct branches, so what we get is a class of decision trees that correspond precisely to the tree-like version of these refutation systems.

Finally, we mention that the queries of the form (1) and (2) as well as forget-rules of the form (3) give rise to a Prover-Adversary game (see [7] where this game was introduced for Resolution). In short, Adversary claims that $\mathcal{D}$ is satisfiable, and Prover tries to expose him. Prover always wins if her strategy is kept as a branching program of the form we have just explained, whilst a good (randomised) Adversary's strategy would show a lower bound on the branching program, and thus on any Res (k) refutation of $\mathcal{D}$.

Translation of fo sentences into propositional CNF formulae.

We shall use the relational language of first-order logic with equality, but without function symbols (this is for convenience only – note that one may simulate constants with added outermost existentially quantified variables).

For the sake of explaining the translation, we assume that such an fo sentence ϕ is given in prenex normal form with quantifier-free part in r-CNF for some r. We start with the easy case of Π_1 sentences:

$$\forall x_1, x_2, \ldots x_l \, \mathcal{F}\left(x_1, x_2, \ldots x_l\right),$$

where $\mathcal{F}$ is quantifier-free, and thus can be considered as a propositional formula over propositional variables of two different kinds: $R\left(x_{i_1}, x_{i_2}, \ldots x_{i_p}\right)$, where R is a p-ary relation symbol, and $(x_i = x_j)$. We now take the union of the clauses of $\mathcal{F}$ as $x_1, x_2, \ldots x_l$ range over $[n]^l$ (we shall always use $[n] = \{1, 2, \ldots n\}$ as a finite universe). The variables of the form $(x_i = x_j)$ evaluate to either true or false, and we are left with variables of the form $R\left(x_{i_1}, x_{i_2}, \ldots x_{i_p}\right)$ only. The general case, a Π_l sentence ϕ,

$$\forall x_1 \exists y_1 \ldots \forall x_l \exists y_l \, \mathcal{F}\left(\overline{x}, \overline{y}\right),$$

can be reduced to the previous case by Skolemisation. We introduce Skolem relations $S_i(x_1, x_2, \ldots x_i, y_i)$ for $1 \leq i \leq l$. $S_i\left(x_1, x_2, \ldots x_i, y_i\right)$ witnesses y_i for any given $x_1, x_2, \ldots x_i$, so we need to add clauses stating that such a witness always exists, i.e.

$$\bigvee_{y_i=1}^{n} S_i\left(x_1, x_2, \ldots x_i, y_i\right) \tag{4}$$

for all $(x_1, x_2, \ldots x_i) \in [n]^i$. The original sentence can then be transformed into the following purely universal sentence

$$\forall \overline{x}, \overline{y} \, \left(\bigwedge_{i=1}^{l} S_i\left(x_1, \ldots x_i, y_i\right) \right) \rightarrow \mathcal{F}\left(\overline{x}, \overline{y}\right). \tag{5}$$

We shall call the clauses (4) "big" (or Skolem) clauses, and the clauses that result in the translation (5) "small" clauses in order to emphasise the fact that the former contain n literals while the latter contain only a constant number of literals, independent from n. Indeed, since $\mathcal{F}$ is assumed to be an r-CNF, we can see the small clauses have width $l + r$ – note the equivalence of $(\bigwedge_{i=1}^{l} S_i) \rightarrow \mathcal{F}$ and $\bigvee_{i=1}^{l} \neg S_i \vee \mathcal{F}$.

For the given fo sentence ϕ, we denote its CNF propositional translation obtained as explained above by $\mathcal{C}_{\phi,n}$ where n is the size of the (finite) model.

Given a (propositional) variable of the form $R_i(x_1, x_2, \ldots, x_p)$ or $S_j(x_1, x_2, \ldots, x_p, y)$, we call $x_1, x_2, \ldots, x_p$ *arguments* of R_i or S_j. We call y the *witness* of S_i. We also call $x_1, x_2, \ldots x_p$ and y the *elements* of R_i or S_j.

Finally, we point out that the Skolemisation also gives us a transformation of the original Π_k sentence ϕ into a Π_2 sentence ϕ':

$$\forall \overline{x}, \overline{y} \exists \overline{z} \bigwedge_{i=1}^{k} S_i(x_1, \ldots x_i, z_i) \wedge \left(\bigwedge_{i=1}^{k} S_i(x_1, \ldots x_i, y_i) \rightarrow \mathcal{F}(\overline{x}, \overline{y}) \right). \tag{6}$$

Clearly ϕ' is equivalent to ϕ, i.e. ϕ' has the same set of countable models as ϕ except for the Skolem relations $S_i(x_1, x_2, \ldots x_i, y_i)$ that are explicit in ϕ' but not in ϕ.

Whenever we say that we refute an fo sentence ϕ in $\mathrm{Res}(k)$, we really mean that we first translate the sentence into a set of (1-)clauses, assuming a finite universe of size n, $\mathcal{C}_{\phi,n}$, and then refute $\mathcal{C}_{\phi,n}$ with $\mathrm{Res}(k)$. Naturally, the size of the refutation is then a function in n.

Least Number Principles

The *least number principle* is the assertion that every finite partial order has a minimal element. We will consider two versions of it (more accurately, its negation):

$$\begin{aligned}
\mathsf{LNP}: &\quad \forall x, y, z \exists w \; R(x,w) \wedge \neg R(x,x) \wedge (\neg R(x,y) \vee \neg R(y,z) \vee R(x,z)) \\
\mathsf{TLNP}: &\quad \forall x, y, z \exists w \; R(x,w) \wedge \neg R(x,x) \wedge (\neg R(x,y) \vee \neg R(y,z) \vee R(x,z)) \\
&\qquad \wedge (x = y \vee R(x,y) \vee R(y,x))
\end{aligned}$$

the latter of which enforces that the order is total. The translation of these to propositional contradictions is a little verbose, involving as it does the introduction of an essentially unnecessary Skolem relation. We will therefore prefer the slightly more natural versions as follows (note that our results go through for the more verbose versions). Recall the variables are $R(i,j)$, $i, j \in$

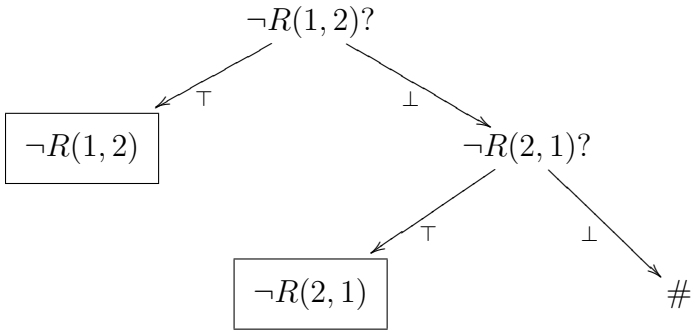
$[n]$; for LNP_n we have the clauses:

$$\begin{aligned}
&\neg R(i,i) && \text{for } i \in [n]\\
&\neg R(i,j) \vee \neg R(j,k) \vee R(i,k) && \text{for } i,j,k \in [n],\\
&R(1,j) \vee \ldots \vee R(n,j), && \text{for } j \in [n]
\end{aligned}$$

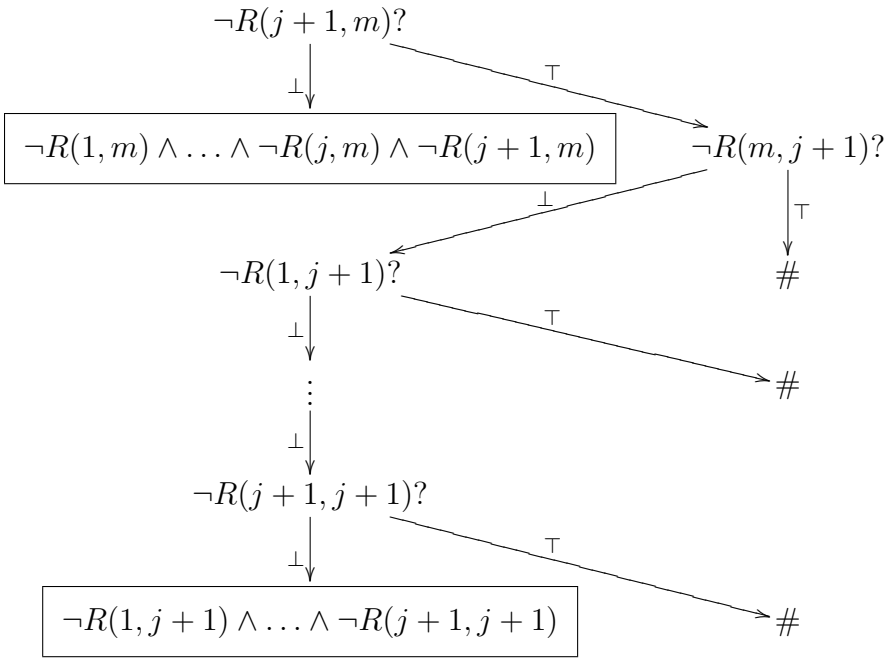
and for TLNP_n we add the clauses $R(i,j) \vee R(j,i)$ for $i \neq j$.

3 Short refutations in Res(k)

In this section we explore upper bounds in the systems Res(k). It is well-known that the Least number principle has polynomially-sized Resolution refutations. We will now consider these as branching 1-programs. Essentially, one maintains at each point a current minimal element m among the investigated $\{1,\ldots,j\}$, for $m \in [j]$. For the LNP_n, we consider major nodes, boxed below, of the program to be of the form $\boxed{\neg R(1,m) \wedge \ldots \wedge \neg R(j,m)}$, for $m \in [j]$. For pedagogical reasons (relating to the forthcoming proof of Proposition 1) we will query negated variables $\neg R(x,y)$. The branching program begins:



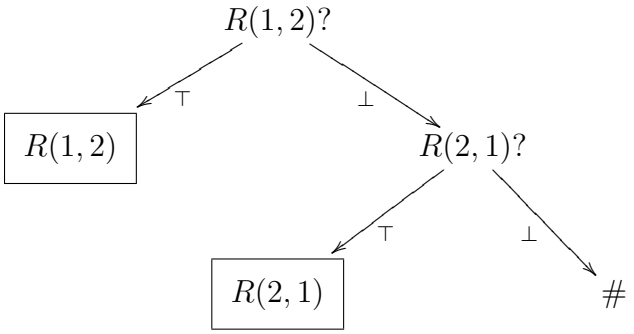
now, from a point $\boxed{\neg R(1,m) \wedge \ldots \wedge \neg R(j,m)}$ it continues



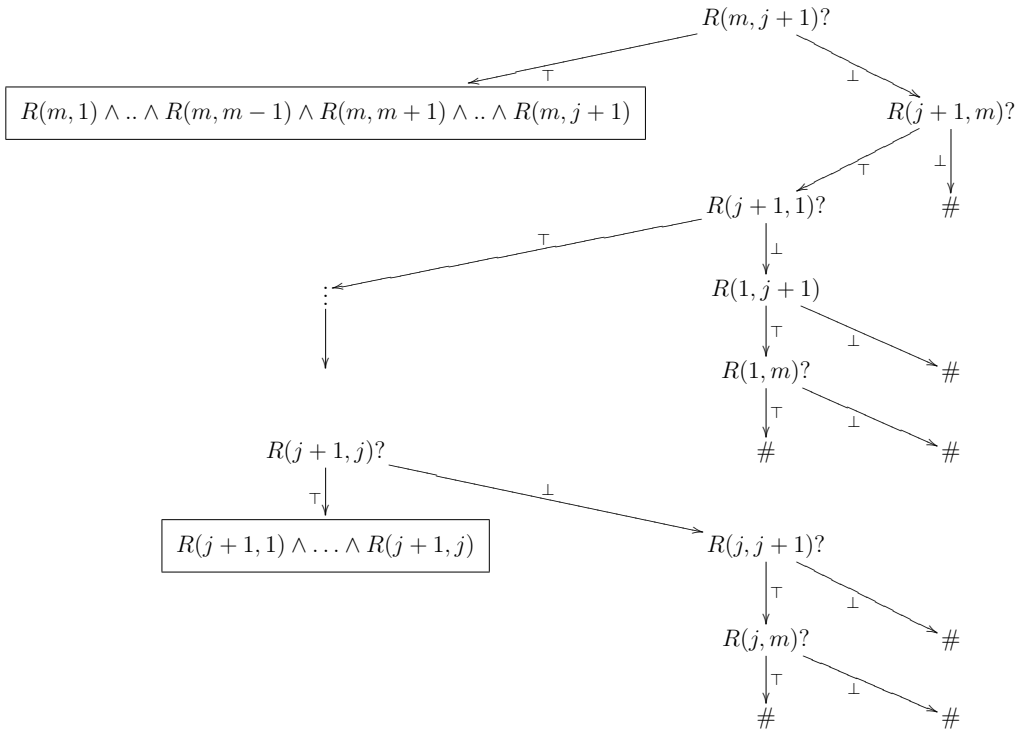
The branching 1-program ends with $\boxed{\neg R(1,m) \wedge \ldots \wedge \neg R(n,m)}$ which contradict the Skolem clauses. The total number of boxed nodes is bound by n^2 and the internal nodes in navigating between boxed nodes are fewer than, say, $2n$. It follows that the program is of size bound by $2n^3$. In the case of the TLNP_n we may go about our business in the dual fashion, with boxed nodes of the form

$$\boxed{R(m,1) \wedge \ldots \wedge R(m,m-1) \wedge R(m,m+1) \wedge \ldots \wedge R(m,j)},$$

for $m \in [j]$, beginning with:



now, from a point $\boxed{R(m,1) \wedge \ldots \wedge R(m,m-1) \wedge R(m,m+1) \wedge \ldots \wedge R(m,j)}$ it continues



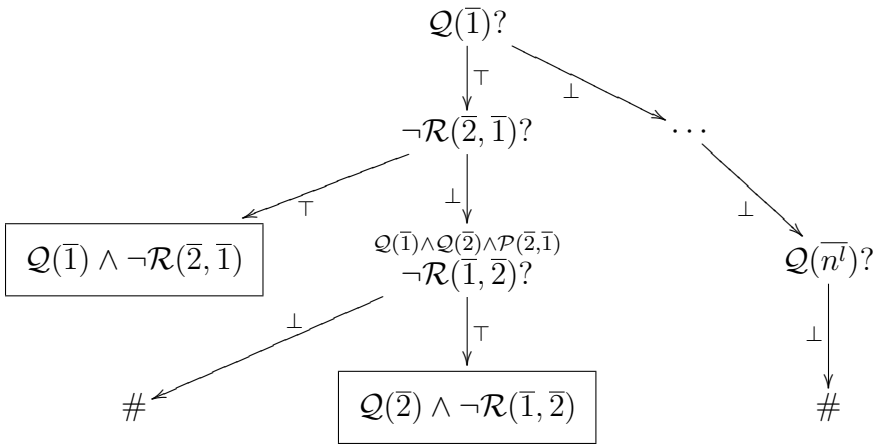
We say that an fo sentence ϕ admits a quantifier-free *interpretation of the relativised LNP* if there exist quantifier-free formulae $\mathcal{P}(\overline{x},\overline{y})$ and $\mathcal{Q}(\overline{x})$ such that, in all models $\mathfrak{A}$ of ϕ, $\mathcal{P}(\overline{x},\overline{y})$ defines a partial order without minimum on the set of tuples given by $\mathcal{Q}(\overline{x})$ (which is non-empty). We are now ready to state our first result.

Proposition 1 *Let ϕ be an fo sentence with no finite models, but some infinite model, s.t. ϕ admits a quantifier-free interpretation of the relativised LNP. Then there exists k s.t. the sequence $\mathcal{C}_{\phi,n}$ has polynomially-sized Res(k) refutations.*

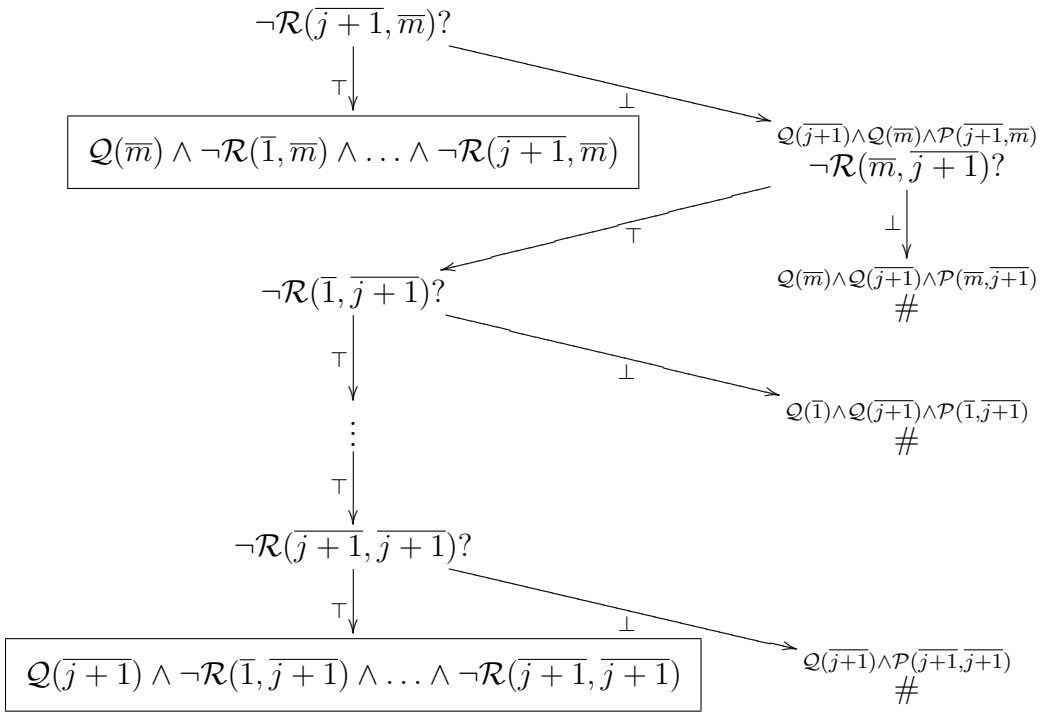
PROOF. In the following, suppose $\overline{x}$ and $\overline{y}$ are l-tuples. Let $\mathcal{Q}(\overline{x})$ be realised as some k_1-CNF $D_1^{\mathcal{Q}}(\overline{x}) \wedge \ldots \wedge D_s^{\mathcal{Q}}(\overline{x})$ and let $\neg\mathcal{R}(\overline{x},\overline{y}) := \neg\mathcal{P}(\overline{x},\overline{y}) \vee \neg\mathcal{Q}(\overline{x}) \vee \neg\mathcal{Q}(\overline{y})$ be realised as some k_2-CNF $D_1^{\mathcal{R}}(\overline{x},\overline{y}) \wedge \ldots \wedge D_t^{\mathcal{R}}(\overline{x},\overline{y})$. Set $k := \max\{k_1,k_2\}$. In a branching k-program, when we talk of questioning $\mathcal{Q}(\overline{x})$ and $\neg\mathcal{R}(\overline{x},\overline{y})$ we in fact mean that we ask the sequence of questions $D_1^{\mathcal{Q}}(\overline{x}) \wedge \ldots \wedge D_s^{\mathcal{Q}}(\overline{x})$ and $D_1^{\mathcal{R}}(\overline{x},\overline{y}) \wedge \ldots \wedge D_t^{\mathcal{R}}(\overline{x},\overline{y})$, obtaining either that they are all true or that one of them is false. This means the branching factor at

such a node will in fact be $s+1$ and $t+1$, instead of 2, corresponding to which of the k-disjuncts might be false (for the sake of simplicity, in future digrams, we will not draw these additional $s-1$ and $t-1$ branches). When $\neg\mathcal{R}(\overline{x},\overline{y})$ is answered false, we assume that $\mathcal{Q}(\overline{x}),\mathcal{Q}(\overline{y})$ and $\mathcal{P}(\overline{x},\overline{y})$ are immediately asked and answered true.

For $i \in [n^l]$, let $\overline{i}$ be the representation in n-ary of i. We consider major nodes, boxed below, of the program to be of the form $\boxed{\mathcal{Q}(\overline{m}) \wedge \neg\mathcal{R}(\overline{1},\overline{m}) \wedge \ldots \wedge \neg\mathcal{R}(\overline{j},\overline{m})}$, for $\overline{m} \in [\overline{j}]$. Our branching k-program begins:



Now from the point $\boxed{\mathcal{Q}(\overline{m}) \wedge \neg\mathcal{R}(\overline{1},\overline{m}) \wedge \ldots \wedge \neg\mathcal{R}(\overline{j},\overline{m})}$ we continue.



The k-program ends with $\boxed{\mathcal{Q}(\overline{m}) \wedge \neg\mathcal{R}(\overline{1},\overline{m}) \wedge \ldots \wedge \neg\mathcal{R}(\overline{n^l},\overline{m})}$, which yield

$$\mathcal{Q}(\overline{m}) \wedge (\neg\mathcal{P}(\overline{1},\overline{m}) \vee \neg\mathcal{Q}(\overline{1})) \wedge \ldots \wedge (\neg\mathcal{P}(\overline{n^l},\overline{m}) \vee \neg\mathcal{Q}(\overline{n^l})),$$

which contradict $\mathcal{C}_{\phi,n}$. The total number of boxed nodes is bound by n^{2l}. The internal nodes in navigating between boxed nodes are fewer than, say, $2(t+1)n^l$. And right at the beginning there is a branching of $(s+1)n^l$. It follows that the program is of size bound by $2(s+1)(t+1)n^{3l}$.

We say that an fo sentence ϕ admits a Σ_1 *interpretation of the relativised LNP* if there exist quantifier-free formulae $\mathcal{P}(\overline{z},\overline{x},\overline{y})$ and $\mathcal{Q}(\overline{x})$, where $\overline{x},\overline{y}$ are l-tuples and $\overline{z}$ is an l'-tuple, such that, in all models $\mathfrak{A}$ of ϕ, $\exists\overline{z}\ \mathcal{P}(\overline{z},\overline{x},\overline{y})$ defines a partial order without minimum on the set of tuples given by $\mathcal{Q}(\overline{x})$ (which is non-empty).

Proposition 2 *Let ϕ be an fo sentence with no finite models, but some infinite model, s.t. ϕ admits a Σ_1 interpretation of the relativised LNP. Then there exists k s.t. the sequence $\mathcal{C}_{\phi,n}$ has polynomially-sized Res(k) refutations.*

PROOF. The proof is similar to the one just given, except $\neg\mathcal{R}(\overline{x},\overline{y}) := \bigwedge_{\overline{z}\in[n^{l'}]} \neg\mathcal{P}(\overline{z},\overline{x},\overline{y}) \vee \neg\mathcal{Q}(\overline{x}) \vee \neg\mathcal{Q}(\overline{y})$ (where we assume that $\neg\mathcal{P}(\overline{z},\overline{x},\overline{y}) \vee \neg\mathcal{Q}(\overline{x}) \vee \neg\mathcal{Q}(\overline{y})$ is realised as some k-CNF $D_1^{\mathcal{R}}(\overline{z},\overline{x},\overline{y}) \wedge \ldots \wedge D_t^{\mathcal{R}}(\overline{z},\overline{x},\overline{y})$). Note that negative answers to questions $\neg\mathcal{R}(\overline{x},\overline{y})$ now have branching factor $tn^{l'}$.

We say that an fo sentence ϕ admits a Π_1 *interpretation of the relativised TLNP* if there exist quantifier-free formulae $\mathcal{P}(\overline{z},\overline{x},\overline{y})$ and $\mathcal{Q}(\overline{x})$, where $\overline{x},\overline{y}$ are l-tuples and $\overline{z}$ is an l'-tuple, such that, in all models $\mathfrak{A}$ of ϕ, $\forall\overline{z}\ \mathcal{P}(\overline{z},\overline{x},\overline{y})$ defines a total order without minimum on the set of tuples given by $\mathcal{Q}(\overline{x})$ (which is non-empty).

Proposition 3 *Let ϕ be an fo sentence with no finite models, but some infinite model, s.t. ϕ admits a Π_1 interpretation of the relativised TLNP. Then there exists k s.t. the sequence $\mathcal{C}_{\phi,n}$ has polynomially-sized Res(k) refutations.*

PROOF. The proof is similar to the one just given, except we use the dual method for refuting the TLNP.

4 Exponential refutations in Resolution

In this section we will prove an exponential lower bound on a variant of the LNP in Resolution; a result that is somehow a counterpoint to those of the previous section. A similar result for a relativised version of the LNP has appeared in [3]. Our variant, which is not relativised, will be designated the Π_1-LNP. It will be specified by the conjunction of the following

$$\forall x, y, z\ \neg(\forall u R(u, x, y)) \vee \neg(\forall u R(u, y, z)) \vee (\forall u R(u, x, z))$$
$$\forall x\ \neg(\forall u R(u, x, x))$$
$$\forall x \exists y\ (\forall u R(u, y, x)),$$

and which more naturally appear as

$$\forall x, y, z\ (\exists u \neg R(u, x, y)) \vee (\exists u \neg R(u, y, z)) \vee (\forall u R(u, x, z))$$
$$\forall x\ (\exists u \neg R(u, x, x))$$
$$\forall x \exists y\ (\forall u R(u, y, x)).$$

It is transparent that this admits a Π_1 interpretation of the LNP (with partial and not total order) and no relativisation. We will translate this slightly differently from our stated procedure, with the addition of only a single Skolem relation $S(x, y)$ (this is in order to maintain some simplicity, our argument would work equally well in the normal circumstance).

$$\forall x, y, z\ (\exists u \neg R(u, x, y)) \vee (\exists u \neg R(u, y, z)) \vee (\forall u R(u, x, z))$$
$$\forall x\ (\exists u \neg R(u, x, x))$$
$$\forall x \forall y \forall u\ \neg S(x, y) \vee R(u, y, x).$$
$$\forall x \exists y\ S(x, y).$$

We can now give these naturally as the clauses

$$\text{for each } w, x, y, z \in [n]\ \bigvee_{u\in[n]} \neg R(u, x, y) \vee \bigvee_{u\in[n]} \neg R(u, y, z) \vee R(w, x, z)$$
$$\text{for each } x \in [n]\qquad \bigvee_{u\in[n]} \neg R(u, x, x)$$
$$\text{for each } w, x, y, z \in [n]\ \neg S(x, y) \vee R(w, y, x).$$
$$\text{for each } x \in [n]\qquad \bigvee_{u\in[n]}\ S(x, y).$$

The main result of this section is the following.

Theorem 4 *Any Resolution refutation of* Π_1*-LNP*$_n$ *must be of size* $\geq 2^{\frac{n}{64}}$.

This will follow immediately from Lemma 10, below. We will derive our result through the probabilistic method, as appears, e.g., in [7]. Considering a decision DAG for the Π_1-LNP, we first prove that certain large records (conjunctions of facts) must appear. We then prove that a large - exponential - number of distinct large records ("bottlenecks") must appear, because otherwise there is a random restriction that generates a decision DAG for a (smaller) Π_1-LNP, that itself has no large records (a contradiction).

Let us imagine that n is divisible by four (we may handle the other cases in a similar manner). We will describe the following *random restrictions* to the Π_1-LNP. Consider a random partition of our universe into two equal parts A and B, with A again randomly divided into the two equal A_1 and A_2. Our random restriction will constrain the variables as follows.

- $S(x,y)$ and $R(w,y,x)$ for all w and $y \in A_1$ and $x \in A_2$.
- $\neg S(x,y)$ and $\neg R(w,y,x)$ for all w and $x \in A_1$ and $y \in A_2$.
- $\neg S(x,y)$ and $\neg R(w,y,x)$ for all w and $x \in A, y \in B$ or $x \in B, y \in A$.

Finally, we set all of the remaining variables of the form $R(w,y,x)$ [i.e. x,y both in A_1, A_2 or B], to $\top$ with probability $\frac{1}{4}$. Considering a decision DAG for the Π_1-LNP, we describe an element $x \in [n]$ as *busy* if either

$S(x,y)$ holds (for some y) or

$\bigwedge_{y \in Y} \neg S(x,y)$ holds (for a set of elements Y of size $> \frac{n}{2}$).

Further, we describe the pair (x,y) as *busy* if either

$R(w,y,x)$ holds (for some w) or

$\neg R(w,y,x)$ holds (for some w).

We will consider an $\frac{n}{2}$*-modified* variant of the Π_1-LNP$_n$ in which there are, for each x,y, at most $\frac{n}{2}$ distinct w s.t. $R(w,y,x)$ is set to $\top$.

Lemma 5 *Any decision DAG for an* $\frac{n}{2}$*-modified* Π_1*-LNP*$_n$ *contains a record with* $\geq \frac{n}{4}$ *busy entities.*

PROOF. We consider an Adversary strategy for the decision DAG. Adversary keeps in mind a set of comparisons $\mathcal{P}$, each of the form $x \prec y$, whose

transitive closure TC($\mathcal{P}$) is a partial order involving $\leq \frac{n}{2}$ elements. Initially, $\mathcal{P}$ is empty. While Prover asks questions of the form $S(x, y)$, Adversary answers $\perp$ if $x = y$, $x \prec y$ is in TC($\mathcal{P}$) or $\neg R(w, y, x)$ is on record; and $\top$ otherwise, adding $y \prec x$ to $\mathcal{P}$. If Prover asks questions of the form $R(w, y, x)$, Adversary should answer $\top$ if $y \prec x$ is in TC($\mathcal{P}$), and $\perp$ otherwise.

If $S(x, y)$ is forgotten then $y \prec x$ should be removed from $\mathcal{P}$, unless some $R(w, y, x)$ or $\neg S(x, y')$ for more than $\frac{n}{2}$ distinct y', or $\neg R(w_{y'}, y', x)$ for more than $\frac{n}{2}$ distinct y', is on record. If $\neg S(x, y)$ is forgotten and there are now $< \frac{n}{2}$ distinct y' s.t. $\neg S(x, y')$ is on record, then, if there is z with $z \prec x$ in $\mathcal{P}$, this should be removed unless $S(x, z)$, some $R(w, z, x)$, or $\neg R(w_{y'}, y', x)$ for more than $\frac{n}{2}$ distinct y', is on record.

If $R(w, y, x)$ is forgotten then $y \prec x$ should be removed from $\mathcal{P}$, unless $S(x, y)$, some other $R(w', y, x)$ or $\neg S(x, y')$ for more than $\frac{n}{2}$ distinct y', or $\neg R(w_{y'}, y', x)$ for more than $\frac{n}{2}$ distinct y', is on record. If $\neg R(w, y, x)$ is forgotten and there are now $< \frac{n}{2}$ elements y' s.t. $\neg R(w_{y'}, y', x)$ is on record, then, if there is z with $z \prec x$ in $\mathcal{P}$, this should be removed unless some other $R(w', z, x)$, $S(x, y)$ or $\neg S(x, y')$, for more than $\frac{n}{2}$ distinct y', is on record.

The Adversary strategy clearly does not fail until TC($\mathcal{P}$) has more than $\frac{n}{2}$ elements, which can not happen unless $\mathcal{P}$ contains at least $\frac{n}{4}$ pairs; and this can not happen without reference to $\frac{n}{4}$ busy entities.

Consider a set of clauses Γ obtained from Π_1-LNP$_n$ by imposing the random restrictions. We describe Γ as *good* if, for all x, y, **both** of which are in either of A_1, A_2 or B, there are $\leq \frac{n}{2}$ distinct w s.t. $R(w, y, x)$ is set (to $\top$). If Γ is not good it is bad.

Lemma 6 *The probability that* Γ *is bad is* $\leq \frac{3}{8} \cdot e^{\frac{-n}{12}}$.

PROOF. We use the following version of the Chernoff bound as appears in [8]. Let $X_1, X_2, \ldots X_n$ be independent $0-1$ random variables with $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then, for every δ, $0 < \delta \leq 1$, the following bound holds

$$\Pr[X \geq (1 + \delta)\mu] \leq e^{\frac{-\mu\delta^2}{3}}$$

In our case we have $p_i = \frac{1}{4}$ (and thus $\mu = \frac{n}{4}$) and $\delta = 1$, so the probability for a specific pair (x, y) to be "bad" is at most $e^{\frac{-n}{12}}$. The probability that a bad pair exists is then (by the union-bound on $(\frac{n}{2})^2 + 2 \cdot (\frac{n}{4})^2$ pairs) at most $\frac{3}{8} n^2 e^{\frac{-n}{12}}$.

Lemma 7 *Consider a good* Γ *obtained from* Π_1*-LNP*$_n$ *by imposing the random restrictions. Any decision DAG for* Γ *contains a record with* $\geq \frac{n}{16}$ *busy entities.*

PROOF. Any decision DAG for Γ contains a refutation of the $\frac{n}{2}$-modified Π_1-LNP on either A_1 or B. (It will never contain a refutation of an LNP on A_2 because these elements *have other elements below them in the partial order* – specifically, those in A_1. However, we asked for the criterion of goodness on A_2 to ensure against the possibility that, for all $x, y \in A_2$ and for all w, we get both $R(w, x, y)$ and $R(w, y, x)$, which would be an immediate contradiction.) The former may be treated as an instance of an $\frac{n}{8}$-modified Π_1-LNP$_{\frac{n}{4}}$; the latter as an instance of an $\frac{n}{4}$-modified Π_1-LNP$_{\frac{n}{2}}$. The argument for these follows as in Lemma 5.

Consider a decision DAG for Π_1-LNP$_n$. We describe a record involving $\geq \frac{n}{16}$ busy entities as a *bottleneck*.

Lemma 8 *With probability* $> 1 - (\frac{1}{2})^{\frac{n}{32}}$*, any bottleneck is falsified by the random restrictions.*

PROOF. Consider each busy entity within the given bottleneck. If x is busy by virtue of $S(x, y)$, then this is falsified with probability $\geq \frac{1}{2}$ (if $x \in A$, then $y \in B$ with probability $\geq \frac{1}{2}$; if $x \in B$, then $y \in A$ with probability $\geq \frac{1}{2}$). If x is busy by virtue of $\bigwedge_{y \in Y} \neg S(x, y)$, then this is falsified with probability $\geq \frac{1}{16}$ (in fact each conjunct $\neg S(x, y)$ is falsified with probability $\geq \frac{1}{16}$ as this is the likelihood of x being in A_2 while y is in A_1). If (x, y) is busy by virtue $R(w, y, x)$ then this is falsified with probability $\geq \frac{1}{2}$ (if $x \in A$, then $y \in B$ with probability $\geq \frac{1}{2}$; if $x \in B$, then $y \in A$ with probability $\geq \frac{1}{2}$). Finally, if (x, y) is busy by virtue of $\neg R(w, y, x)$ then this is falsified with probability $\geq \frac{1}{16}$ (this is the likelihood of x being in A_2 while y is in A_1).

Now, we do not quite have independence of the busy entities. For, example, if x is busy by virtue of $S(x, y)$ then it is more likely that some (x, y) is busy by virtue of some $R(w, y, x)$; however, these dependencies can only come in pairs. Therefore, we may consider that there are $\frac{n}{16}/2$ busy entities whose business is independent. It follows that a bottleneck is falsified by probability exceeding $1 - (1 - \max\{\frac{1}{2}, \frac{1}{16}\})^{\frac{n}{32}}$.

Lemma 9 *If there are* $< 2^{\frac{n}{64}}$ *bottlenecks then there is a random restriction that falsifies all bottlenecks.*

PROOF. The probability that a bottleneck survives the random restrictions is $\leq (\frac{1}{2})^{\frac{n}{32}}$. Thus, by the union bound, the probability that any of the $2^{\frac{n}{64}}$

bottlenecks survives is $\leq (\frac{1}{2})^{\frac{n}{32}}$. Now, again using the union bound, we deduce that the probability any bottleneck survives or the Γ induced by the random restrictions is bad is at most $(\frac{1}{2})^{\frac{n}{32}} + \frac{3}{8}n^2 e^{\frac{-n}{12}} < 1$ (at least for $n \geq 101$). It follows that there is some good random restriction that kills all bottlenecks.

Lemma 10 *Any decision DAG for Π_1-LNP$_n$ must contain $\geq 2^{\frac{n}{64}}$ bottlenecks.*

PROOF. If not it follows from the Lemma 9 that there is a random restriction that kills all bottlenecks and induces a good Γ. But then what remains incorporates a refutation of this good Γ which has a record involving $\geq \frac{n}{16}$ busy entities by Lemma 7. But, such a record would have been a bottleneck in the decision DAG for the original Π_1-LNP$_n$ – a contradiction.

5 Final remarks

We believe that our proof of Theorem 4 can be canonically extended to prove that the lower bound holds not only in Resolution, but also in Res(k).

Conjecture 11 *There exists $\epsilon_k > 0$ s.t. any Res(k) refutation of Π_1-LNP$_n$ must be of size $\geq 2^{\epsilon_k \cdot n}$.*

We may define the following further variants of the least number principle which we will designate the Π_d-LNP. They may be specified by the conjunction of the following

$$\forall x, y, z\ \neg(\forall u_1 \exists u_2 \ldots, Q u_d\, R(u_1, .., u_d, x, y)) \vee\ \neg(\forall u_1 \exists u_2 \ldots, Q u_d\, R(u_1, .., u_d, y, z)) \vee (\forall u_1 \exists u_2 \ldots, Q u_d\, R(u_1, .., u_d, x, z))$$

$$\forall x\ \neg(\forall u_1 \exists u_2 \ldots, Q u_d\, R(u_1, .., u_d, x, x))$$

$$\forall x \exists y\ (\forall u_1 \exists u_2 \ldots, Q u_d\, R(u_1, .., u_d, y, x)),$$

where Q is $\forall$ if d is odd and is $\exists$ if d is even. We conjecture that the (negations of the) principles Π_d-LNP may be used to separate depth d-Frege from depth $d+1$-Frege (for the definitions of these proof systems see [6]).

Conjecture 12 *The negation of the Π_{d+1}-LNP gives rise to a sequence of propositional tautologies $\mathcal{C}_n$ that admits polynomially-sized proofs in depth $d+1$-Frege but requires exponentially-sized proofs in depth d-Frege.*

References

[1] S. Riis, A complexity gap for tree-resolution, Computational Complexity 10 (2001) 179–209.

[2] S. Dantchev, Rank complexity gap for Lovász-Schrijver and Sherali-Adams proof systems, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007, ACM, 2007, pp. 311–317.

[3] S. Dantchev, S. Riis, On relativisation and complexity gap for Resolution-based proof systems, in: The 17th Annual Conference of the EACSL, Computer Science Logic, Vol. 2803 of LNCS, Springer, 2003, pp. 142–154.

[4] J. Krajíĉek, On the weak pigeonhole principle, Fundamenta Mathematica 170 (2001) 123–140.

[5] S. Dantchev, B. Martin, The limits of tractability in resolution-based propositional proof systems, in: The 6th Conference Computability in Europe, Vol. 6158 of LNCS, Springer, 2010, pp. 98–107.

[6] J. Krajíĉek, Bounded Arithmetic, Propositional Logic, and Complexity Theory, Cambridge University Press, 1995.

[7] P. Pudlák, Proofs as games, American Mathematical Monthly (2000) 541–550.

[8] M. Buot, Probability and computing: Randomized algorithms and probabilistic analysis. Michael Mitzenmacher and Eli Upfal, Journal of the American Statistical Association 101 (2006) 395–396.
URL `http://ideas.repec.org/a/bes/jnlasa/v101y2006p395-396.html`