BRICS

Basic Research in Computer Science

BRICS RS-99-29 S. Riis: A Complexity Gap for Tree-Resolution

A Complexity Gap for Tree-Resolution

Søren Riis

BRICS Report Series RS-99-29

ISSN 0909-0878 September 1999

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone:	+45 8942 3360
Telefax:	+45 8942 3255
Internet:	BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/29/`

A Complexity gap for tree-resolution

Søren Riis *

September 1999

Abstract

It is shown that any sequence ψ_n of tautologies which expresses the validity of a fixed combinatorial principle either is "easy" i.e. has polynomial size tree-resolution proofs or is "difficult" i.e requires exponential size tree-resolution proofs. It is shown that the class of tautologies which are hard (for tree-resolution) is identical to the class of tautologies which are based on combinatorial principles which are violated for infinite sets. Actually it is shown that the gap-phenomena is valid for tautologies based on infinite mathematical theories (i.e. not just based on a single proposition).

We clarify the link between translating combinatorial principles (or more general statements from predicate logic) and the recent idea of using the symmetrical group to generate problems of propositional logic.

Finally, we show that is undecidable whether a sequence ψ_n (of the kind we consider) has polynomial size tree-resolution proofs or requires exponential size tree-resolution proofs. Also we show that the degree of the polynomial in the polynomial size (in case it exists) is non-recursive, but semi-decidable.

Keywords: Logical aspects of Complexity, Propositional proof complexity, Resolution proofs.

1 Outline

In this paper we introduce a new kind of result for propositional logic. It is shown for a large class of uniform families of unsatisfiability problems $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_j, \ldots$ that the family either has polynomial size tree-resolution refutations or requires full exponential size tree-resolution refutations. For non-uniform families (where, for example, each $\mathcal{C}_j$ might express a different

[1] Basic Research in Computer Science, Department of Computer Science, University of Aarhus, Ny Munkegade, Building 540, 8000 Aarhus C, Denmark. Email: smriis@daimi.au.dk Phone: +45 89 42 32 85

combinatorial principle) there is no complexity gap and any super-polynomial but sub-exponential growth-rate can appear. Somewhat informally our main result states that if the sequence $\mathcal{C}_j$ express the *same* combinatorial principle for each j then there is a complexity gap for tree-resolution.

In the section *further perspectives* we show how it is possible to assign a mathematical theory $T_P(f)$ to any given propositional proof system P and any complexity f. This idea is new and places our main result in a larger perspective. For any propositional proof system P one can ask about the behaviour of $T_P(f)$ when the resources f increase. This question is completely well defined and is closely linked to the complexity gap phenomena.

I hope the expert in propositional proof lower bound will seriously consider this approach. The paper raises a number of questions related to the theory T_P. The paper also raises a number of problems which seem to lie just outside the scope of current techniques.

The paper is however not only aimed at the expert. The major part of the paper is intended for a broad audience with a primary interest in complexity theory. We all know that a few complexities appear again and again, while other complexities virtually never appear. This is folklore knowledge which I think puzzles many of us from time to time. The present paper is partly motivated by this phenomena. The paper shows that there are contexts where it is possible to have a general complexity gap theorem. In the paper we focus almost entirely on tree-resolution and the most basic version of the complexity gap phenomena. This way we avoid any serious technical complications. More ambitious gap-theorems ([29], [30] joint work with Meera Sitharam) lead to highly interesting but also very serious technical problems. In the present paper we consider the base case which undoubtly also has the greatest general interest.

The reader who is interested in the resolution method (perhaps mostly for predicate logic) might find some of our proofs interesting and stimulating. To achieve our upper bound we show how it is possible to bring a given resolution refutation (for predicate logic) on a special but very natural normal form. The normal form allows one to read-off the unification directly from the abstract proof form.

Finally I think that the method of generating unsatisfiability problems by use of the symmetric group will be of general interest. This idea was introduced in [28], but in the present paper we develop the idea somewhat further.

An important motivation for studying propositional proof systems is tied up with the following basic question: Given a true statement (tautology) what is the length of the shortest proof of the statement. Here the answer of course depends on which axiomatic proof system is being used. From a Computer Science perspective the question is particularly fundamental for propositional logic. As formalised by Cook and Reckhow [13], there exists a propositional proof system in which any tautology ψ has a proof of size bounded by $p(|\psi|)$ for a fixed polynomial p if and only if NP=co-NP. This question is far beyond current techniques. However Cook and Reckhow proposed a program of research which systematically tries to obtain non-polynomial lower bounds for stronger and stronger propositional proof systems. The hope is that this eventually will lead to a separation of NP from co-NP. In the section *further perspectives* we discuss this approach in the context of our results.

Tautologies expressing simple graph theoretic properties have been important test cases for obtaining bounds for the length of propositional resolution refutations. The first super-polynomial lower bound for resolution (satisfying a restriction called regularity) was obtained by Tseitin [32]. Subsequent work simplified Tseitin's proof and improved the lower bounds for regular resolution [14], [33]. However great difficulty was experienced in extending Tseitin's arguments to unrestricted resolution (=dag-resolution). In [16] Haken managed to give a super-polynomial lower bound for the pigeon-hole principle for dag-resolution. Later this result was improved considerably by Ajtai [1], [2] to a super-polynomial lower bound on bounded depth Frege proofs. Ajtai also used his approach to show independence results from Bounded Arithmetic. These results were later improved in various ways and generalities [3], [5], [8], [25], [26].

Informally we can state our result as follows: Let ψ_n denote a sequence of tautologies which expresses the validity of a fixed combinatorial principle $\mathcal{P}_{\text{com}}$. Let $\mathcal{C}_n$ denote the negation of ψ_n on Conjunctive Normal Form. Our main result states that for any such sequence $\mathcal{C}_n$ either the sequence has polynomial size tree-resolution refutations or the sequence requires truly exponentially sizeed tree-resolution refutations. Furthermore, exponential size is required exactly when $\mathcal{P}_{\text{com}}$ is false as a principle of infinitary combinatorics.

The reason we consider tree-resolution, rather than dag-resolution is mainly technical. Ideally we would have preferred to have proved our results for dag-resolution. Actually even stronger propositional systems might have complexity gaps, but for most propositional systems any proof of such a complexity gap would solve open problems which are beyond current techniques. In the case of tree-resolution we avoid any serious technical complications.

As already pointed out, we consider uniform sequences $\mathcal{C}_n$ of unsatisfiability problems. More specifically we consider uniformly S_n-generated sequences $\mathcal{C}_n$ of unsatisfiable clauses (here S_n denotes the symmetric group consisting of the permutations of $\{1, 2, \ldots, n\}$). This approach was introduced in [28] (see also below for more details). The idea is to select a finite collection of generating clauses and then obtain $\mathcal{C}_n$ as the S_n-closure of the generating clauses. This method is interesting in its own right because it provides a very easy and feasible method for generating test problems for proof systems for propositional logic. Also it is easy to organise and classify the test problems.

The class of S_n-generated unsatisfiability problems consists of highly uniform sequences of unsatisfiability problems. How restrictive is this S_n-generated uniformity?

The class of S_n-generated sequences of $\mathcal{C}_n$ is quite rich and wide. The class is so rich that the decision problem of deciding whether a given S_n-generated sequence $\mathcal{C}_n$ has polynomial size tree-resolutions, is undecidable. Also the degree of the polynomial bounding the size might be tremendously large. Actually for any fast-growing total recursive function F, e.g. the Ackerman function, F_{ϵ_0}, F_{Γ_0} etc. (see for example [31] for a survey on fast growing functions), there exists a (small) finite list $\mathcal{C}_{\text{gen}}$ of generating clauses such that the sequence $\mathcal{C}_n$ has polynomial size tree-refutations, but the degree of the polynomial needed to bound the size of the smallest tree-refutations, is larger than $F(|\mathcal{C}_{\text{gen}}|)$ (where $|\mathcal{C}_{\text{gen}}|$ denotes the number of symbols in $\mathcal{C}_{\text{gen}}$).

Another question we have to address is to what extent the class of S_n-generated unsatisfiability problems is relevant. It is certainly powerful enough to generate the hardest unsatisfiability problems which are known. Actually (assuming NEXP $\neq$ co-NEXP) the method is rich enough to generate a universally difficult sequence $C_{n_1}, C_{n_2}, \ldots$ of unsatisfiable collections of clauses which requires non-polynomial size refutations for any given propositional proof system [28].

The proof complexity of S_n-generated sequences $\mathcal{C}_n$, which are unsatisfiable for all values of n, is open. We do not know whether there are propositional proof systems which have polynomial size refutations of any such S_n-generated sequence. We will return to this question in the section *further perspectives* towards the end of the paper.

Let us briefly compare the S_n-generation method with the most commonly used method of generating satisfiability problems. This method (which is outside the scope of this paper) is to consider randomly chosen 3-satisfiability problems and to consider the case where the ratio c of clauses and variables is kept constant, while the number of variables tends to infinity. Experiments suggest that there is a phase transition near $c = c_{phase} \approx 4.23...$. Experimentally it is found that virtually all problems with $c > c_{phase}$ are unsatisfiable, while virtually all problems with $c < c_{phase}$ are satisfiable. Given a propositional refutation system P it seems to be possibile that there is a phase transition (for some constant c_P) in the following sense: For $c_{phase} < c < c_P$ almost certainly long (e.g. exponential size) refutations are required, while for $c_P < c$ there are almost certainly short (e.g. polynomial size) refutation size proofs. In such a case, where the threshold is sharp, it seems fair to say that a complexity gap occurs. Of course the situation could be much more complicated with various phase transitions and thresholds corresponding to different complexity classes, etc.

The only propositional refutation system for which the situation is well understood is the tree-resolution refutation system. It turns out that for this system there is no sharp phase transition (see [6]) and that the expected refutation complexity tails off very slowly as a function of c. This result does not contradict the complexity gap we show in this paper. This is because the S_n-generated test problems are very far from being random. I believe S_n-generated test problems are much superior to random test problems when it comes to discussing and analysing specific weaknesses of a given propositional proof system. It is in my opinion not a coincidence that the strongest known lower bounds - including Haken's [16] (for resolution) and Beame et.al. [7] (for bounded depth frege proofs) - can be achieved by S_n-generated problems, rather than by random generated unsatisfiability problems.

Instead of considering randomly chosen unsatisfiability problems we consider uniformly S_n-generated sequences $\mathcal{C}_n$ of unsatisfiable clauses. Later in this paper we will notice how it is possible to assign a mathematical first-order theory T to each uniformly S_n-generated sequence $\mathcal{C}_n$ of satisfiability problems. We also have the converse (which also follows from [28]) that for any first order theory (which might not be finitely axiomatisable and which might be highly non-recursive) there is a natural translation procedure which translates the question of whether T has a model of size n into a satisfiability problem $\mathrm{SAT}_{T,n}$. This satisfiability problem is uniformly S_n-generated.

This shows that uniformly S_n-generated satisfiability problems can be viewed as being satisfiability problems (in propositional logic) which arise from translating satisfiability problems in predicate logic. The idea is (in its most general form) to take as input any first-order theory T, which then is used to generate a sequence of propositional formulas $\psi_1, \psi_2, \ldots$ in which ψ_n expresses that T does not have a model of size n. As already pointed out, this method of generating tautologies (even when T only consists of a single sentence) is very general. It covers a large and important class of sequences of tautologies. Many natural sequences of tautologies which express a general combinatorial principle belong to this class. The class also includes the tautologies defined in [28]. Let k_T^{rel} denote the maximal arity of a relation symbol in the language for T while k_T^{fun} denote the maximal arity of a function symbol in the language for T. For a given propositional proof system (refutation system) P it is natural to try to understand which mathematical theories T lead to difficult unsatisfiability problems. In the paper we show:

Theorem: (informal version) *Let T be a first order theory (which might not be finitely axiomatisable and which might be highly non-recursive). There is a natural translation procedure which translates the question of whether T has a model of size n into a satisfiability problem $\mathrm{SAT}_{T,n}$. There are two possibilities:*

(1) *For each value of n for which $\mathrm{SAT}_{T,n}$ is unsatisfiable, the smallest*

tree-resolution refutations has size at least $2^{n/\max(k_T^{\mathrm{rel}},1+k_T^{\mathrm{fun}})}$

(2) Asymptotically (i.e when n *tends to infinity)* $\mathrm{SAT}_{T,n}$ *has polynomial size (in* n*) tree-resolution refutations.*

Possibility (1) happens if and only if T *has an infinite model. The lower bound in (1) also holds if* $\mathrm{SAT}_{T,n'}$ *is satisfiable for some* $n' > n$*.*

In general - even when T *only consists of a single sentence* ψ *- it is undecidable whether* $\mathrm{SAT}_{\psi,n}$ *has polynomial size tree-refutations or require exponential size tree-refutations. The collection of* ψ *which have polynomial size tree-refutations is recursively enumerable (but not recursive). There is no total recursive function which given input* ψ *outputs* $u \in N$ *such that if* $\mathrm{SAT}_{\psi,n}$ *has polynomial size tree-refutations then it has* $\leq n^u$*-size tree-refutations.*

The theorem gives a complete classification of the theories T for which $\mathrm{SAT}_{T,n}$ requires large tree-resolution refutations (if there are any at all - $\mathrm{SAT}_{T,n}$ could be satisfiable). More specifically, a theory T leads to hard (for tree-resolution) tautologies if and only if T has an infinite model.

Let me point out that the philosophy behind this result first was articulated in [23] where it was shown (in the context of Bounded Arithmetic) that combinatorial principles which fail as infinitary combinatorics in a sense (which can be made precise) are harder (to prove) than combinatorial principles which also are valid as part of infinitary combinatorics. More specifically in [23] we showed that combinatorial principles which fails for infinite sets never can be proved on the first tree levels $S_2^1(\alpha) \subseteq T_2^1(\alpha) \subseteq S_2^2(\alpha)$ of Sam Buss hierarchy of Bounded Arithmetic, while such combinatorial principle in certain cases can be proved on the fourth level $T_2^2(\alpha)$. It is well known that provability in fragments of Bounded Arithmetic is closely related to propositional proof complexity (for more details see [17]). The results in the present paper are, however, technically unrelated to the results in [23]. The proof technique in the current paper is different from the rudimentary forcing technique which was employed in [23]. Jan Krajicek has pointed out (personal communication) that our exponential lower bound follows by a modification of his proof of Theorem 11.3.2 in [17] (which essentially is the main result in [23]). See also Lemma 9.5.2 in [17] where this is stated explicitly.

I am aware of only one other result which gives a complexity gap between polynomial complexity and exponential complexity. A beautiful result [15] which relates the Vapnik-Chervonenkis (VC) dimension to the growth rate of the complexity of learning the concept class C. It states that this growth rate is either polynomial or exponential. Furthermore, it is polynomial if and only if the VC-dimension of C is finite. The underlying mathematics in this result are completely different from ours. It is however remarkable that the dichotomy of finite versus infinite plays a crucial role in both the VC-complexity gap theorem as well as in our complexity gap theorem.

2 Background and Notation

A *literal* is a propositional variable or the negation of a propositional variable. A clause $C := \{l_1, l_2, \ldots, l_u\}$ is a collection of literals, and it is satisfied if $l_1 \vee l_2 \vee \ldots \vee l_u$ holds. In the famous NP-complete problem 3-SAT, the decision problem is to decide if a given collection of clauses (which each contain at most 3 literals) is satisfiable.

Resolution is a refutation system designed to provide certificates (i.e. proofs) that a system of clauses is unsatisfiable. A given formula is shown to be a tautology by showing that its negation, put into conjunctive normal form (i.e. clausal form) is unsatisfiable. This is done by means of the resolution rule

$$\textit{Resolution rule}: \qquad \frac{C_1 \cup \{p\} \quad C_2 \cup \{\neg p\}}{C_1 \cup C_2}$$

The given clauses are often referred to as *axioms*, and the task it to derive the empty clause (the contradiction) from the axioms. In tree-resolution the proof is organised as a binary tree with the axioms in the leaves and the empty clause in the root.

As comparison in unrestricted resolution (dag-resolution) the derived clauses are listed in a linear fashion $C_1, C_2, \ldots, C_u$, and any clause C_l is either an axiom or appears by resolving two clauses $C_i, C_j, i, j < l$. Such a derivation can also be represented as a dag which explains the terminology dag-resolution. In dag-resolution a derived clause can be reused, while this is not the case in tree-resolution.

As already noted, Haken considered a sequence of tautologies expressing the so-called pigeonhole principle. It can be shown that Haken's tautologies require tree-resolution proofs of size $\geq n2^n$ [11]. In this paper we will show that an exponential lower bound actually follows from the simple fact that the pigeon-hole principle fails as a principle of *infinite* combinatorics.

Haken's tautologies (Γ_n) can be written as follows:

$\cup_{j=1}^{n} \{a_{ij}\}$, where $i = 1, 2, \ldots, n$

$\{\bar{a}_{ij}, \bar{a}_{ik}\}$, where $i, j, k = 1, 2, \ldots, n$ and $j \neq k$

$\cup_{j=1}^{n} \{a_{0j}\}$, $\{\bar{a}_{0i}, \bar{a}_{0j}\}$, where $i, j = 1, 2, \ldots, n$ and $i \neq j$

$\{\bar{a}_{ik}, \bar{a}_{jk}\}$, where $i, j, k = 1, 2, \ldots, n$

$\{\bar{a}_{ij}, \bar{a}_{0j}\}$, where $i, j = 1, 2, \ldots, n$

This collection is, in a rather obvious way, finitely generated by the symmetric group S_n. More specifically each Γ_n is generated by taking the S_n-closure of the clauses:

$\cup_j \{a_{1j}\}$, $\{\bar{a}_{12}, \bar{a}_{13}\}$, $\{\bar{a}_{11}, \bar{a}_{12}\}$, $\cup_j \{a_{0j}\}$, $\{\bar{a}_{01}, \bar{a}_{02}\}$, $\{\bar{a}_{13}, \bar{a}_{23}\}$, $\{\bar{a}_{12}, \bar{a}_{02}\}$, $\{\bar{a}_{12}, \bar{a}_{22}\}$, $\{\bar{a}_{11}, \bar{a}_{01}\}$

For future reference let us denote this collection of generators by Γ_{Haken}. For any n let $\mathcal{C}_n$ denote the collection of clauses which appear by closing Γ_{Haken} under the natural action of the symmetrical group S_n (permuting $\{1, 2, 3, \ldots, n\}$ while keeping 0 fixed). This system of clauses is equivalent to the system for which Haken obtained his famous super-polynomial lower bound.

2.1 S_n-generated unsatisfiability problems

The translation of many combinatorial principles into a system of unsatisfiable clauses naturally leads to clauses which are generated by applying the symmetric group S_n to a collection of generators. The S_n-symmetry arises naturally when the combinatorial problem is independent from the underlying representation. Consider, for example, a combinatorial principle K which is valid for some graph G. The principle K is also valid when the enumeration of the vertices is permuted by an element $\pi \in S_n$. It turns out that this S_n-symmetry survives (as will become clear) when we reformulate the combinatorial principle in terms of an (un)satisfiability problem.

Before we move on we will be slightly more general and consider hypergraphs; For fixed $r = 0, 1, 2, \ldots$ we can consider the collection $a_{n_1,n_2,\ldots,n_r}$ of boolean variables for which $n_1, n_2, \ldots, n_r \in \{1, 2, \ldots\} = \mathrm{N}$. Actually we might also have other boolean variables $b_{n_1,n_2,\ldots,n_r}$ for which $n_1, n_2, \ldots, n_r \in \{1, 2, \ldots, \}$, or more generally we might fix a collection $a^i_{n_1,n_2,\ldots,n_{r_i}}$ $i = 1, 2, \ldots$ of boolean variables of different variable types (one for each i). The *support* of a boolean variable $a^i_{n_1,n_2,\ldots,n_{r_i}}$ is $\{n_1, n_2, \ldots, n_{r_i}\}$. We consider two kind of clauses. An ordinary clause is a collection $\{p_1, p_2, \ldots, p_l\}$ of literals (i.e. boolean variables or negations of boolean variables). An abstract clause is a formal expression of the form $\cup_j \ \{a^i_{n_1,n_2,\ldots,n_{r_i-1},j}\}$. In the case of Γ_{Haken} the generators $\cup_j \ \{a_{1j}\}$ and $\cup_j \ \{a_{0j}\}$ were the only abstract clauses - all other clauses were ordinary clauses. In the example of Γ_{Haken} we can view boolean variables which contain a zero (e.g. a_{02}) as variables of a different variable type than variables which do not contain a zero (e.g. a_{12}).

Now let Γ_{gen} be a collection of clauses (normal clauses as well as abstract clauses). Assume that all boolean variables which appear in Γ_{gen} have support contained in $\{1, 2, \ldots, l\}$. For each $n \geq l$ we get a collection Γ_n of clauses by taking the S_n-closure of the clauses in Γ_{gen} in the obvious fashion. The sequence Γ_n is S_n-generated if there exists a collection Γ_{gen} (not necessarily finite) such that Γ_n is the S_n-closure of Γ_{gen}. The sequence Γ_n is finitely S_n-generated if there exists a finite collection Γ_{gen} which generates the sequence Γ_n. Notice that Γ must involve infinitely many variable types in the case where Γ is infinite.

In [28] we showed how one could obtain a finite collection of generators (like the list above) whenever given an existential second order sentence Ψ. If

Ψ is second order existential which is on prenex normal form with its first order part purely universal, then the translation into propositional logic does not involve the introduction of skolem functions. In this case we simply translate the question of whether a purely universal first order sentence Ψ' has a model (of size n) into a satisfiability problem $\Gamma_{\Psi,n}$. As we showed in [28] there is a one-to-one correspondence between the satisfying assignments of $\Gamma_{\Psi,n}$ and the models of size n of Ψ'.

The converse is essentially (see later for the exact result) also true: *For any collection Γ of generators there exists a universal first order theory T such that there is a one-to-one correspondence between satisfying assignments of Γ_n (the S_n-closure of Γ) and the models M_n of T which have size n. Furthermore, if Γ is finite, then T can be replaced by a single universal first order sentence Ψ.*

Again there is a one-to-one correspondence between the satisfying assignments of Γ_n and the models of size n of T (Ψ).

2.2 Link to predicate logic

Consider the system $\Gamma_{\text{gen}} := \Gamma_{\text{Haken}}$. We claim (and it is essentially just a matter of changing notation) that the satisfiability problem Γ_n is equivalent to the question of whether a sentence (theory) in predicate logic has a model of size n. The sentence is the following predicate formula expressing the negation of the pigeon-hole principle:

$$\forall x \; f(x) \neq c \quad \wedge \quad \forall x, y, z \; (f(x) = y \wedge x \neq z) \rightarrow (f(z) \neq y)$$

We can write this sentence in clausal form as a satisfiability problem in predicate logic. This problem consists of the clauses

$$\{\neg f(x) = c\}, \quad \text{and} \quad \{\neg f(x) = z, \neg f(y) = z, x = y\}$$

as well as the usual clauses for axioms of equality (see [19] or below for more details). To these clauses we add the clauses $\{\neg c_i = c_j\}$ for $i \neq j, i, j \leq n$ as well as the clause $\{x = c_1, x = c_2, \ldots, x = c_n\}$ (see below for a discussion of this choice). The collection these clauses gives us a system $\mathcal{C}_n$ of clauses in predicate logic. The satisfiability problems Γ_n and $\mathcal{C}_n$ are (not surprisingly) closely related.

We will now focus on the case of translating satisfiability problems in predicate logic into a satisfiability problem in propositional logic (*Our translation should NOT be confused with the usual Herbrand-style (or Henkin-style) translation in which sentences are viewed as propositional variables).*

Let $\mathcal{C}$ be a collection of clauses for predicate logic over some fixed language L in which function symbols and relation symbols have arities bound by fixed constants k_C^{rel} and k_C^{fun}. The collection $\mathcal{C}$ might be infinite. Any universal theory can be written as a collection of clauses. In general any theory T' can be replaced by a logical equivalent universal theory T (this process of introducing skolem-functions is not unique).

Let $\mathcal{C}_{\text{eq}}$ denote the collection $\mathcal{C}$ extended with clauses expressing the axioms of equality. More specifically let $\mathcal{C}_{\text{eq}}$ consists of the clauses in $\mathcal{C}$ together with the clauses $\{x = x\}$,
$\{\neg x = y, y = x\}, \{\neg x = y, \neg y = z, x = z\}$ and a clause $\{\neg x_1 = y_1, \neg x_2 = y_2, \ldots, \neg x_k = y_k, \neg R(x_1, x_2, \ldots, x_k), R(y_1, y_2, \ldots, y_k)\}$ for each k-ary relation symbol R ($k = 1, 2, \ldots, k_C^{\text{rel}}$) and a clause $\{\neg x_1 = y_1, \neg x_2 = y_2, \ldots, \neg x_k = y_k, f(x_1, x_2, \ldots, x_k) = f(y_1, y_2, \ldots, y_k)\}$ for each k-ary function symbol f ($k = 1, 2, \ldots, k_C^{\text{fun}}$). Now let $n \in \mathbf{N}$ be given. Let $c_1, c_2, \ldots, c_n$ be new constants which does not appear in $\mathcal{C}$. Consider the following collection (of clauses):

$\mathcal{C}_{\geq n} := \{\{\neg c_1 = c_2\}, \{\neg c_1 = c_3\}, \ldots, \{\neg c_{n-1} = c_n\}\}$

If we add these clauses to $\mathcal{C}_{\text{eq}}$, any model which satisfies the clauses must have size $\geq n$. A little care is needed when we add clauses expressing that there are at most n elements in the domain. One could, for example, add the clause, $\{x_1 = x_2, x_1 = x_3, \ldots, x_n = x_{n+1}\}$. The presence of this clause, however, smuggles in a version of the pigeon-hole principle which is not available as a rule in propositional resolution proofs. Instead we chose the collection:

$\mathcal{C}_{\leq n} := \{\{x = c_1, x = c_2, \ldots, x = c_n\}\}$

The collection of the clauses in $\mathcal{C}_{\text{eq}}$ together with the clauses axiomatising n-ness is denoted $\mathcal{C}_n$ (i.e. $\mathcal{C}_n := \mathcal{C}_{\text{eq}} \cup \mathcal{C}_{\geq n} \cup \mathcal{C}_{\leq n}$). We also introduce a slightly weaker axiomatisation of n-ness. In this axiomatisation we replace the clause $\{x = c_1, x = c_2, \ldots, x = c_n\}$ by the schema $\mathcal{C}_{\leq n}^{\text{weak}}$ which consists of the clauses $\{f(c_{i_1}, c_{i_2}, \ldots, c_{i_k}) = c_1, f(c_{i_1}, c_{i_2}, \ldots, c_{i_k}) = c_2, \ldots, f(c_{i_1}, c_{i_2}, \ldots, c_{i_k}) = c_n\}$. There is one such clause for each function symbol and for each $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}$. For each constant symbol c the schema contain the clause $\{c = c_1, c = c_2, \ldots, c = c_n\}$. This system of clauses is denoted $\mathcal{C}_n^{\text{weak}}$ (i.e. $\mathcal{C}_n^{\text{weak}} := \mathcal{C}_{\text{eq}} \cup \mathcal{C}_{\geq n} \cup \mathcal{C}_{\leq n}^{\text{weak}}$).

To get propositional tautologies like the ones which have already been extensively examined in the literature ([2], [11], [13], [16], [28]) we proceed as follows: We are given a system $\mathcal{C}_n$ (or $\mathcal{C}_n^{\text{weak}}$). We want to ensure that each literal (=atomic formula or negation of atomic formula) is of the form: $R(x_1, x_2, \ldots, x_k)$ or $f(x_1, x_2, \ldots, x_k) = x_{k+1}$ or $c = x_1$. To achieve this we rewrite the clauses in the obvious way. Assume, for example, that we want to rewrite the clause: $\{R(x, S(S(x))), f(S(x), y) = x\}$. We do this in steps, getting $\{R(x, z), \neg z = S(S(x)), \neg w = S(x), f(w, y) = x\}$, $\{R(x, z), \neg z = S(u), \neg u = S(x), \neg w = S(x), f(w, y) = x\}$, and then finally $\{R(x, z), \neg S(u) = z, \neg S(x) = u, \neg S(x) = w, f(w, y) = x\}$. The resulting system is denoted $\mathcal{C}_n^*$ ($\mathcal{C}_n^{\text{weak},*}$ ($\mathcal{C}_{\text{eq},n}^*$ in the case of $\mathcal{C}_{\text{eq}}$)). Finally consider all clauses which can appear by replacing each of its variables by a constant from $c_1, c_2, \ldots, c_n$. We denote the resulting system of clauses $\mathcal{C}_n^{\text{prop}}$ ($\mathcal{C}_n^{\text{prop,weak}}$).

Notice that each clause in $\mathcal{C}_{\leq n}^{\text{weak}}$ is a substitution instance of the clause $\{x = c_1, x = c_2, \ldots, x = c_n\}$ in $\mathcal{C}_{\leq n}$. On the other hand $\{fff(c_1) = c_1, fff(c_1) = c_2, \ldots, fff(c_1) = c_n\}$ (which is a substitution instance of $\{x = c_1, x = c_2, \ldots, x = c_n\}$) becomes $\{\neg f(c_1) = x, \neg f(x) = y, f(y) = c_1, f(y) = c_2, \ldots, f(y) = c_n\}$ and thus when constants c_i, c_j are substituted for x and y $\{\neg f(c_1) = c_i, \neg f(c_i) = c_j, f(c_j) = c_1, f(c_j) = c_2, \ldots, f(c_j) = c_n\}$. But clauses of this form are just weakenings of the clauses in $\mathcal{C}_{\leq n}^{\text{weak}}$. Actually it is not difficult to see that this always is the case. Thus from now on we do not distinguish between $\mathcal{C}_n^{\text{prop,weak}}$ and $\mathcal{C}_n^{\text{prop}}$. From now on $\mathcal{C}_n^{\text{prop}}$ denotes the same system as $\mathcal{C}_n^{\text{prop,weak}}$ (except we might be allowed to include weakenings of the clauses to be refuted to the unsatisfiability problem).

Notice that the size of $\mathcal{C}_n^{\text{prop}}$ is bounded by a polynomial in n. It should, however, be emphasised that our translation procedure - naively speaking - typically translates combinatorial principles (like the pigeon-hole principle) into an infinite system of clauses. This is because, besides the "usual" clauses, we also get clauses which, for example, express properties and behaviour of the terms (including skolem-functions). In the case of the pigeonhole principle we have, for example, clauses expressing properties which involve the iteration of the function symbol. We have already seen that this is irrelevant when it occurs in $\{x = c_1, x = c_2, \ldots, x = c_n\}$ and that we essentially get the same clauses whether we allow iterations of terms or only allow atomic terms to be substituted. This is also (trivially) the case for a general clause in $\mathcal{C}$. This is because substitution in a clause (e.g. $x \leftarrow f(x), y \leftarrow g(y)$ in $\{R(x, y)\}$) always leads to a weakening of the original clause ($\{R(f(x), g(y))\}$ which turns into the form $\{R(z, u), \neg f(x) = z, \neg g(y) = u\}$).

These considerations show that the translation $\mathcal{C}_n^{\text{prop}}$ (after having discarded irrelevant weakenings) always contains only polynomially (in n) many clauses. The translation corresponds (except from the treatment of constants in the original language for $\mathcal{C}$) to the informal procedure which seems to have been used when considering a principle like the pigeonhole principle [13] or the parity principle [3]. This translation also agrees with (and extends) the procedure defined in [28].

Let $S_C(n)$ denote the size of the smallest tree-resolution refutation of $\mathcal{C}_n^{\text{prop}}$. If there is no such refutation, we let $S_C(n) = \infty$. Usually proof complexity is measured in the size of the satisfiability problem, however we get cleaner results if we use n as input parameter. Also there exist polynomials p_1, p_2 such that $p_1(n) \leq |\mathcal{C}_n| \leq p_2(n)$ so our complexity gap agrees with the usual conventions. For our purposes it is most sensible to use the model size n as the relevant parameter.

Finally let me briefly mention the treatment of constants. In the case where L has finitely many constants $c^{(1)}, c^{(2)}, \ldots, c^{(u)}$ and where we assume that these are distinct, it is natural (mostly for cosmetic reasons) to replace the clauses $\{c^{(i)} = c_1, c^{(i)} = c_2, \ldots, c^{(i)} = c_n\}$ by the clauses $\{c^{(1)} = c_1\}, \{c^{(2)} = c_2\}, \ldots, \{c^{(u)} = c_u\}$ and to let the symmetric group S_{n-u} act on $\{u+1, u+2, \ldots, n\}$. Notice that this modification of $\mathcal{C}_n^{\text{prop}}$ only affects our lower bounds mildly (at most by a polynomial factor n^u).

3 Main Results

Theorem 1: (Gap theorem) *The following are equivalent:*

(1) *For any polynomial $P(n)$, there exists n such that $S_C(n) > P(n)$.*

(2) *For each n, $S_C(n) \geq 2^{n/\max(k_C^{\text{rel}}, 1+k_C^{\text{fun}})}$.*

(3) *$\mathcal{C}$ is satisfied in an infinite model.*

The decision problem of deciding whether $\mathcal{C}$ satisfies (1),(2) and (3) is undecidable. The collection of finite collections $\mathcal{C}$ of clauses for which there exists a polynomial P such that $P(n) > S_C(n)$ for all $n \in N$ is recursively enumerable (but not recursive). There is no total recursive function, which given input $\mathcal{C}$, outputs $u \in N$ such that $n^u > S_C(n)$ whenever (1),(2) or (3) fails.

Now let Ψ be a universal sentence in first order logic. In the previous section we showed that we can express the claim that Ψ does not have a model of size n as a boolean tautology Ψ_n (see also [28]). We may even consider the case where Ψ is a Π_2-sentence (i.e. not just a Π_1-sentence). In this case we can translate the sentence into propositional logic by means of abstract clauses very similar to the abstract clauses described earlier (the existential quantifier get translated into clauses like $\cup_{j_1, j_2, \ldots, j_r} \{a_{i_1, i_2, \ldots, i_{s_1}-1, j_1, i_{s_1}+1, \ldots, i_{s_k}-1, j_k, i_{s_k}+1, \ldots, i_m}\}$. In general (for arbitrary first order formulas) we can introduce skolem functions (see [28] for details) to rewrite the sentence as a Π_1-sentence (or just a Π_2-sentence) which then can be translated into a satisfiability problem in propositional logic. One way which leads to the same result is to translate Ψ into a satisfiability problem $\mathcal{C}_\Psi$ (the usual way by introducing skolem functions etc. see for example [19]) and then proceed as described the the previous section. The collection of clauses $\mathcal{C}_{\Psi,\text{eq}}$ (which contains the clauses for equality) is satisfiable if and only if Ψ has a model. If Ψ does not have a model, there exists (by Herbrands Theorem) a finite collection of clauses (from $\mathcal{C}_{\Psi,\text{eq}}$) together with a suitable unification, such that the resulting system is unsatisfiable in the sense of propositional logic. Let Ψ_n denote the tautology which express the unsatisfiability of $\mathcal{C}_{\Psi,n}^{\text{prop}}$. With this notation we get:

Corollary 1: *Assume Ψ is a sentence of first order logic. Assume Ψ does not have models of size n for infinitely many values $n_1, n_2, n_3, \ldots$ of n. Then the* *following are equivalent:*

(1) Ψ_n *has a tree-resolution refutation of sub-exponential size.*

(2) Ψ_n *has a tree-resolution refutation of polynomial size.*

(3) Ψ *has no infinite model.*

Furthermore if (1), (2) or (3) holds, there exists n_0 *such that* Ψ_n *is a tautology for all* $n \geq n_0$.

Corollary 2: *If* Ψ_n *has a tree-resolution refutation of size* $< 2^{n_0/\max(k_T^{\text{rel}}, 1+k_T^{\text{fun}})}$ *just for one value* $n = n_0$*, then there exists a polynomial* $P(n)$ *such that each* Ψ_n*, for* $n \geq n_0$ *have tree-refutations of size* $\leq P(n)$.

Given our main result there is nothing mystical in this Corollary. A tree-resolution of size $< 2^{n/\max(k_T^{\text{rel}}, 1+k_T^{\text{fun}})}$ witness the fact that $\mathcal{C}_{\text{eq}} \cap \{\{c_1 \neq c_2\}, \ldots \{c_{n-1} \neq c_n\}\}$ is unsatisfiable, and that $\mathcal{C}$ is not satisfied in any infinite model. It is well known that predicate logic is decidable in an oracle which provide an upper bound on the Herbrand Complexity for a logical valid formula. This shows that there is no general computable method for computing the degree of the polynomial P in Corollary 1.

As a by-product and somewhat related we get a complexity gap for the Herbrand complexity:

Theorem 3: (Complexity gap for Herbrand Complexity) *Let* $\mathcal{C}$ *be a satisfiability problem (for predicate logic). Assume that the underlying language has all functions and relations of arity bounded by a constant. Consider* $\mathcal{C}_n := \mathcal{C}_{\text{eq}} \cup \mathcal{C}_{\geq n} \cup \mathcal{C}_{\leq n}$ *and let us only consider the values of* n *for which* $\mathcal{C}_n$ *is unsatisfiable.*

Then either $\mathcal{C}_n$ *has a Herbrand Complexity bounded by a constant, or* $\mathcal{C}_n$ *has Herbrand Complexity which is linear in* n.

Furthermore, the first case appears exactly when $\mathcal{C}$ *is unsatisfiable.*

We noticed (after Corollary 2) that there is no general computable method for computing the degree of the polynomial P in Corollary 1. The same observation shows that there is no general computable method for bounding the constant in Theorem 3. Also there is, given input $\mathcal{C}$, no computational method which can decide whether the sequence $\mathcal{C}_n^{\text{prop}}$ ($\mathcal{C}_n$) requires super-polynomial tree-resolution refutations (has non-constant Herband Complexity) or has polynomial size tree-resolution refutations (have constant Herband Complexity)

The argument for the polynomial upper bound can be broadened somewhat further. We present these results in the section *further perspectives* .

4 Examples

Now let us illustrate the main ideas in this paper by a few examples:

Example 1: Let $\Theta \equiv \forall x \exists y R(x,y) \vee \exists x \forall y \neg R(x,y)$. This sentence is logically valid. To show this (by the resolution method for predicate logic) we first consider $\Psi \equiv \neg\Theta \equiv \exists x \forall y \neg R(x,y) \wedge \forall x \exists y \neg R(x,y)$ and rephrase this (by introducing skolem-functions) as $\forall y \neg R(c,y) \wedge \forall x R(x, f(x))$. This translation method is standard and is, for example, described in [19]. To show Θ is equivalent to showing that the system of clauses $\{\{\neg R(c,y)\}_y, \{R(x,f(x))\}_x\}$ is unsatisfiable. The unsatisfiability follows from the fact that we have a unification of $(R(c,y), R(x,f(x))$ by $x \to c, y \to f(c)$ which leads to the refutation $\frac{\{\neg R(c,f(c))\} \quad \{R(c,f(c))\}}{\emptyset}$.

For a given n, the clauses in $\mathcal{C}_n$ consist of the clauses $\{\neg R(c,y)\}_y, \{R(x,f(x))\}_x$ together with the clauses for equality and the clauses $\{\neg c_1 = c_2\}, \ldots, \{\neg c_{n-1} = c_n\}$, as well as the schema: $\{c = c_1, c = c_2, \ldots, c = c_n\}$, and $\{f(c_i) = c_1, f(c_i) = c_2, \ldots, f(c_i) = c_n\}$. Now to get the system $\mathcal{C}_n^{\text{prop}}$ we rewrite $\{\neg R(c,y)\}_y$ and $\{R(x,f(x))\}_x$ as $\{\neg c = x, \neg R(x,y)\}_{x,y}$ and $\{R(x,y), \neg f(x) = y\}_{x,y}$.

Finally, after taking the union of all clauses which appear by replacing free variables by constants $c_1, c_2, \ldots, c_n$, we arrive at $\mathcal{C}_n^{\text{prop}}$ the following clauses, where f_{ij} is shorthand for $f(c_i) = c_j$ and where d_i is shorthand for $c = c_i$:

$\{\neg r_{ik}, \neg d_i\}$ for $i, k \in \{1, 2, \ldots, n\}$, $\quad \{r_{ik}, \neg f_{ik}\}$ for $i, k \in \{1, 2, \ldots, n\}$,

$\{d_1, d_2, \ldots, d_n\}$, $\quad \{f_{i1}, f_{i2}, \ldots, f_{in}\}$ for $i \in \{1, 2, \ldots, n\}$.

The system $\mathcal{C}_n^{\text{prop}}$ has the following tree-resolution refutation proof:

$$\cfrac{\cfrac{B_n \quad \cfrac{.... \quad \cfrac{B_4 \quad \cfrac{B_3 \quad \cfrac{B_2 \quad \cfrac{B_1 \quad \{d_1, d_2, \ldots, d_n\}}{\{d_2, d_3, \ldots, d_n\}}}{\{d_3, \ldots, d_n\}}}{....}}{........}}{\{d_n\}}}{\emptyset}$$

where

$$B_i := \cfrac{A_{in} \quad \cfrac{.... \quad \cfrac{A_{i4} \quad \cfrac{A_{i3} \quad \cfrac{A_{i2} \quad \cfrac{A_{i1} \quad \{f_{i1}, f_{i2}, \ldots, f_{in}\}}{\{\neg d_i, f_{i2}, f_{i3}, \ldots, f_{in}\}}}{\{\neg d_i, f_{i3}, \ldots, f_{in}\}}}{....}}{........}}{\cfrac{\{\neg d_i, f_{in}\}}{\{\neg d_i\}}}$$

and where

$$A_{ik} := \frac{\{\neg r_{ik}, \neg d_i\} \quad \{r_{ik}, \neg f_{ik}\}}{\{\neg d_i, \neg f_{ik}\}}$$

It is not hard to verify that this proof consists of $4n^2+2n+1$ clauses (which is optimal because any tree-resolution refutation must have each of the $2n^2+n+1$ clauses appearing in the some leaf). ♣

This example illustrate how the fact that Ψ has no models (of size n) can be translated into a "test-case" unsatisfiability problem for propositional logic. In the example $\neg\Psi$ is logically valid so Ψ has no infinite model. According to our main result this implies (what we just verified) that $\mathcal{C}_n^{\text{prop}}$ has polynomial size tree-resolution refutations.

Example 2: Let T denote the first order theory axiomatised by the single axiom $\psi := \forall i, j(\ i \neq j \rightarrow\ \ s(i) \neq s(j)) \wedge \forall j\ s(j) \neq c$. The theory T is an axiomatisation of the first order theory of a constant and a successor function. The clauses in $\mathcal{C}$ consist of $\{x = y, \neg S(x) = S(y)\}_{x,y}$, and $\{\neg S(x) = c\}_x$. To make the translation into propositional logic we rewrite these clauses as $\{x = y, \neg S(x) = z, \neg S(y) = z\}_{x,y,z}$ and $\{\neg S(x) = y, \neg c = y\}_{x,y}$. To simplify the readability we abbreviate $S(c_i) = c_j$ as s_{ij}, and $c = c_j$ as d_j. This gives us a satisfiability problem $\mathcal{C}_n^{\text{prop}}$ in the boolean variables $s_{11}, s_{12}, \ldots, s_{nn}$, $d_1, d_2, \ldots, d_n$ and with the following clauses:

(1) $\{s_{i1}, s_{i2}, \ldots, s_{in}\}$ for $i = 1, 2, \ldots, n$.

(2) $\{\bar{s}_{ik}, \bar{s}_{jk}\}$ for $i \neq j,$ where $i, j, k \in \{1, 2, \ldots, n\}$

(3) $\{d_1, d_2, \ldots, d_n\}$

(4) $\{\bar{d}_i, \bar{d}_j\}$ for $i \neq j,$ where $i, j \in \{1, 2, \ldots, n\}$

(5) $\{\bar{d}_i, \bar{s}_{ji}\}$ for $i, j \in \{1, 2, \ldots, n\}$.

The theory T has infinite models so, according to our main result, $\mathcal{C}_n^{\text{prop}}$ requires tree-resolution refutations of size $\geq 2^{n/\max(k_T^{\text{rel}}, 1+k_T^{\text{fun}})} = 2^{n/2}$.

♣

Example 3: Let T be the theory which is axiomatised by a single axiom stating that there exists a injective map from the universe onto the universe minus one point. More specifically let T be axiomatised by the sentence:

$$\forall x, y\ x \neq y \rightarrow f(x) \neq f(y) \wedge \forall x\ f(x) \neq c.$$

In clausal form we have $\mathcal{C} := \{\{x = y, \neg f(x) = f(y)\}_{x,y}, \{\neg f(x) = c\}_x\}$. Then $\mathcal{C}^* := \{\{x = y, \neg f(x) = z, \neg f(y) = z\}_{x,y,z}, \{\neg f(x) = z, c = z\}_{x,z}\}$. Let e_{ij} be short hand for $f(i) = j$ and let d_j be short hand $c = j$. Let $\mathcal{C}_n^{\text{prop}'} := \{\{\bar{e}_{ik}, \bar{e}_{jk}\}$ for $i \neq j, \{\bar{e}_{ik}, d_k\}, \{e_{i1}, e_{i2}, \ldots, e_{in}\}, i = 1, 2, \ldots, n\}$.

The clauses in $\mathcal{C}_n^{\text{prop}}$ involve other substitution instances of $\mathcal{C}$. As we already noticed these substitutions play no role. For example substituting $f(x) \rightarrow x$ and $f(y) \rightarrow y$ in $\{x = y, \neg f(x) = f(y)\}_{x,y}$ gives us the clause $\{f(x) = f(y), \neg f(f(x)) = f(f(y))\}_{x,y}$. If we rewrite this on the (*)-form we get the clause $\{\neg f(y) = z, f(x) = z, \neg f(x) = u, \neg f(y) = v, \neg f(u) = w, \neg f(v) = w\}$. Thus $\mathcal{C}_n^{\text{prop}}$ includes the clause $\{\neg f(c_j) = c_k, f(c_i) = c_k, \neg f(c_i) = c_u, \neg f(c_j) = c_v, \neg f(c_u) = c_w, \neg f(c_v) = c_w\}$ for $i \neq j$ and $i, j, k, v, w, z \in \{1, 2, \dots, n\}$ with $i \neq j$. In our shorthand notation these clauses are of the form $\{\bar{e}_{jk}, e_{ik}, \bar{e}_{iu}, \bar{e}_{jv}, \bar{e}_{uw}, \bar{e}_{vw}\}$ where $i \neq j$. These clause are weakenings of $\{\bar{e}_{uw}, \bar{e}_{vw}\}$ which is already (when $u \neq v$) in $\mathcal{C}_n^{prop}$. When $u = v$, the clause is a weakening of $\{\bar{e}_{iu}, \bar{e}_{jv}\}$ which is already (when $i \neq j$) in $\mathcal{C}_n^{prop}$. Thus these clause are redundant. Other substitution instances produce even more clauses, but these clauses are redundant. Thus $\mathcal{C}_n^{\text{prop}} = \mathcal{C}_n^{\text{prop}'}$.

There exists an infinite model in which T is valid. Thus $\mathcal{C}_n^{prop}$ require tree-resolution refutations of size $\geq 2^{\frac{n}{2}}$. The usual version of the pigeon-hole principle is obtained by treating c as a fixed element (for example c_1). In this case we get the clauses
$\mathcal{C}_n^{\text{prop}^*} := \{\{\bar{e}_{ik}, \bar{e}_{jk}\} \text{ for } i \neq j, \{\bar{e}_{i1}, d\}, \{e_{i1}, e_{i2}, \dots, e_{in}\}, i = 1, 2, \dots, n\}$. Clearly this give a lower bound of $2^{\frac{n}{2}}/n$ on refuting $\mathcal{C}_n^{\text{prop}^*}$. Actually it is not hard to see that we can modify our main theorem to the case where the symmetrical group S_{n-1} acts on $\{c_2, c_3, \dots, c_n\}$ (and where we let $c = c_1$). This gives a lower bound of $2^{\frac{(n-1)}{2}}$. The best known lower bound on this propositional version of the pigeon-hole principle is $(n-1)2^{n-1}$ (see [11]).

♣

Example 4: It is well known that each finite division ring is a field. There are various ways to rephrase the statement that *the universe is a division ring which is not a field* as a satisfiability problem (for predicate logic). Let $\mathcal{C}$ be one of these satisfiability problems. Then for each n, $\mathcal{C}_n^{\text{prop}}$ is an unsatisfiable set of clauses expressing the fact that there are no non-commutative division rings with n elements.

There are infinite divisions rings which not are fields (for example Hamilton's famous quaternions). Thus, according to our main result the sequence $\mathcal{C}_n^{\text{prop}}$; $n = 1, 2, ...$ requires exponential size tree-resolution refutations. It is not clear if $\mathcal{C}_n^{\text{prop}}$ is hard for stronger propositional proof systems like for example the extended Frege proof system. ♣

The final example is somewhat curious. It shows a case where the so-called weak propositional pigeonhole principle appears somewhat unexpected:

Example 5: Let T be the first order theory in the language $L(=, d_i; i \in \mathbf{N})$ of equality plus infinitely many constants. Assume that the theory T is axiomatised by a conjunction of the axioms $\{\neg d_i = d_j\}$ where $i \neq j, i, j \in \mathbf{N}$. Notice that T not is finitely axiomatisable. Our main theorem include this situation. Let $\mathcal{C}_T = \{\{\neg d_i = d_j\} : i \neq j, i, j \in \mathbf{N}\}\}$. The system $\mathcal{C}_n^{\text{weak}} = \mathcal{C}_n$ consists of the clauses: $\{\neg d_i = d_j\}$ for $i \neq j, i, j \in \mathbf{N}$ and $\{\neg c_i = c_j\}$ and $\{d_i = c_1, \ldots, d_i = c_n\}$, $i \in \mathbf{N}$ as well as the clauses for equality. Thus $\mathcal{C}_n^{\text{prop}}$ consists of the clauses $\{\bar{e}_k^{(i)}, \bar{e}_k^{(j)}\}$ and $\{e_1^{(i)}, e_2^{(i)}, \ldots, e_n^{(i)}\}$, where $e_k^{(i)}$ is shorthand for $d_i = c_k$. So $\mathcal{C}_n^{\text{prop}}$ is (essentially) a boolean version of the pigeonhole principle (stating that there is no injective map from an infinite set onto a set with n elements). A priori T has an infinite model, so according to our main result, $\mathcal{C}_n^{\text{prop}}$ requires exponential size tree-refutations. ♣

5 The polynomial size upper bound

As before let $\mathcal{C}$ be a collection of clauses (in the sense of predicate logic). The clauses in $\mathcal{C}$ consists of atomic formulas and negations of atomic formulas.

The system $\mathcal{C}$ can be rewritten on a normal form $\mathcal{C}^*$ where all (negations of) atomic formulas are (negation of) basic atomic formulas i.e. each atomic formula is either of the form $R(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ where R is a r-ary ($r \in \{1, 2, \ldots\}$) relation symbol or is of the form $f(x_{i_1}, x_{i_2}, \ldots, x_{i_r}) = x_{i_{r+1}}$ where f is a r-ary ($r \in \{1, 2, \ldots\}$) function symbol. Constants are treated as 0-ary function symbols. Here are some examples of basic atomic formulas: $R(x, z, x), f(x, y, z) = z$ and $U(x, y)$. On the other hand the atomic formulas $R(g(y), x)$ and $f(x, g(y), z) = h(z, x)$ are not basic.

Now it is clear we can rewrite any clause $\mathcal{C}$, containing atomic formulas, as an equivalent clause $\mathcal{C}^*$ containing only basic atomic formulas. In the case where $\mathcal{C} = \{\{\neg R(c, y)\}_y, \{R(x, f(x))\}_x\}$ we can rewrite this as $\mathcal{C}^* := \{\neg R(x, y), \neg c = x\}_{x,y}, \{R(x, y), \neg f(x) = y\}_{x,y}\}$.

From the usual completeness theorem it follows that $\mathcal{C}$ has a model if and only if $\mathcal{C}_{\text{eq}}$ is satisfiable. And this only holds if and only if $\mathcal{C}_{\text{eq}}^*$ is satisfiable (has a model).

Assume $\mathcal{C}$ has no model. Then $\mathcal{C}_{\text{eq}}$ (and $\mathcal{C}_{\text{eq}}^*$) is unsatisfiable and according to Herbrands Theorem there exists a unification τ and a finite subset $\mathcal{C}' \subseteq \mathcal{C}_{\text{eq}}$ ($\mathcal{C}' \subseteq \mathcal{C}_{\text{eq}}^*$) such that $\tau(\mathcal{C}')$ is unsatisfiable in the sense of propositional logic.

Actually in the case of $\mathcal{C}_{\text{eq}}^*$ we can write the refutation on a special normal form. We will use this normal form in the proof of our upper bound.

Let me illustrate this by an example. Consider the "proof-form"

$$\frac{\{\neg R(c, y)\}_y \quad \{R(x, f(x))\}_x}{\emptyset}$$

The substitution $x \to c, y \to f(c)$ turns this into a valid resolution. Now we can rewrite the proof-form as

$$\frac{\{\neg R(x, y), \neg c = x\}_{x,y} \quad \{R(x, y), \neg f(x) = y\}_{x,y}}{\{\neg c = x, \neg f(x) = y\}_{x,y}}$$

This is a valid proof (though in many formalisms only closed terms, and not variables are allowed). The final clause $\{\neg c = x, \neg f(x) = y\}_{x,y}$ can be refuted by the unification $x \to c, y \to f(c)$.

This leads to the proof:

$$\dfrac{\{f(c) = f(c)\} \qquad \dfrac{\{c = c\} \qquad \dfrac{\dfrac{\{\neg R(x,y), \neg c = x\}_{x,y} \quad \{R(x,y), \neg f(x) = y\}_{x,y}}{\{\neg c = x, \neg f(x) = y\}_{x,y} \qquad (= D)}}{\{\neg c = c, \neg f(c) = f(c)\}}}{\{\neg f(c) = f(c)\}}}{\emptyset}$$

The refutation consists of two parts. The first part ends with the clause $= D$. The unification is uniquely determined from this clause, and the next step (leading to the clause $\{\neg c = c, \neg f(c) = f(c)\}$) is the only place in the refutation where a substitution takes place. The final part of the refutation only uses clauses expressing the axioms of equality.

This idea works in general. For any refutation of $\mathcal{C}_{\text{eq}}$, we can rewrite this as a tree-refutation of $\mathcal{C}^*_{\text{eq}}$ which is of a normal form such that:

(1) The first part of the refutation consists solely of basic atomic formulas and negation of basic atomic formulas.

(2) The last clause D in this part uniquely determines the unification which is required to refute the clause.

(3) The clause D consists solely of negations of atomic equations.

(4) The last part of the proof refutes the unification (of D) using only applications of substitutions of equality axioms.

To illustrate these ideas further let
$\mathcal{C} := \{\{P(0)\}, \{\neg P(x), P(S(x))\}_x, \{\neg P(S(S(0)))\}\}$. This statement is written in the language $L = L(0, P, S)$ of predicate logic which contains a unary relation symbol P, a constant 0 and a unary function symbol S. Clearly $\mathcal{C}_{\text{eq}}$ is unsatisfiable. Any refutation requires Herbrand complexity HC $= 3$ (if $\{\neg P(S(S(0)))\}$ is replaced by $\{\neg P(S^{(k)}(0)))\}$ it requires Herbrand Complexity HC $= k$). Clearly $\mathcal{C}^* = \{\{P(x), \neg 0 = x\}_x, \{\neg P(x), P(y), \neg S(x) = y\}_{x,y}, \{\neg P(z), \neg S(z_1) = z_2, \neg S(z_2) = z, \neg 0 = z_1\}_{z,z_1,z_2}\}$. The corresponding propositional unsatisfiability problem $\mathcal{C}_n^{\text{prop}}$ consists of the clauses:

(1) A clause $\{p_i\}$ for $i = 1, 2, \ldots, n$.

(2) A clause $\{\neg p_i, p_j, \neg s_{ij}\}$ for each $i, j \in \{1, 2, \ldots, n\}$.

(3) A clause $\{\neg p_i, \neg s_{jk}, \neg s_{ki}, \neg o_j\}$ for each $i, j, k \in \{1, 2, \ldots, n\}$.

(4) The clause $\{o_1, o_2, \ldots, o_n\}$.

(5) A clause $\{s_{i1}, s_{i2}, \ldots, s_{in}\}$ for each $i \in \{1, 2, \ldots, n\}$.

This system is (a priori as we know) finitely S_n-generated. The generators are $\{p_1\}, \{\neg p_1, p_2, \neg s_{12}\}, \{\neg p_1, \neg s_{23}, \neg s_{31}, \neg o_2\}$ as well as the degenerate versions of these
(i.e. $\{\neg p_1, \neg s_{21}, \neg s_{11}, \neg o_2\}$ etc.). The axioms for $\mathcal{C}_{\leq n}$ are generated by the two abstract generators $\vee_j\{o_j\}$ and $\vee_j\{s_{1j}\}$.

The clauses $\mathcal{C}$ have no model so we want to show (this is the issue in this section) that the sequence $n \to \mathcal{C}_n^{\text{prop}}$ has polynomial size tree-resolution refutations. The clauses in $\mathcal{C}_{\text{eq}}$ have the following refutation:

$$\dfrac{\{\neg P(S(S(0)))\} \qquad \dfrac{\{\neg P(S(0)), P(S(S(0)))\} \qquad \dfrac{\{P(0)\} \quad \{\neg P(0), P(S(0))\}}{\{P(S(0))\}}}{\{P(S(S(0)))\}}}{\emptyset}$$

This can be translated into the normal form (which is a refutation of $\mathcal{C}^*_{\text{eq}}$). The first part of this refutation is

$$\dfrac{\{P(z), \neg 0 = x, \neg S(x) = y, \neg S(y) = z\} \quad U}{\{\neg 0 = x, \neg S(x) = y, \neg S(y) = z\} \quad (= D)}$$

where

$$U := \dfrac{\{\neg P(y), P(z), \neg 0 = x, \neg S(x) = y, \neg S(y) = z\} \quad V}{\{P(z), \neg 0 = x, \neg S(x) = y, \neg S(y) = z\}}$$

and where

$$V := \dfrac{\{P(x), \neg 0 = x\} \quad \{\neg P(x), P(y), \neg 0 = x, \neg S(x) = y\}}{\{P(y), \neg 0 = x, \neg S(x) = y\}}$$

The last clause D determines the unification $x \leftarrow 0, y \leftarrow S(0)$ and $z \leftarrow S(S(0))$. The last part of the refutation consists of the obvious refutation of the clause $\{\neg 0 = 0, \neg S(0) = S(0), \neg S(S(0)) = S(S(0))\}$.

Now let us show how we can turn the abstract derivation of D into propositional refutations of $\mathcal{C}_n^{\text{prop}}$. For any i, j, k we have a derivation $A_{i,j,k}$:

$$\dfrac{\{\neg p_k, \neg o_i, \neg s_{ij}, \neg s_{jk}\} \qquad \dfrac{\{\neg p_j, p_k, \neg s_{ij}, \neg o_i, \neg s_{jk}\} \qquad \dfrac{\{p_i, \neg o_i\} \quad \{\neg p_i, p_j, \neg o_i, \neg s_{ij}\}}{\{p_j, \neg o_i . \neg s_{ij}\}}}{\{p_k, \neg o_i, \neg s_{ij}, \neg s_{jk}\}}}{\{\neg o_i, \neg s_{ij}, \neg s_{jk}\} \quad (= D_{ijk})}$$

These derivations can now be resolved using the axioms in (4) and (5). First we derive $\{\neg o_i, \neg s_{ij}\}$ from the clauses $\{\neg o_i, \neg s_{ij}, \neg s_{jk}\} \quad k = 1, 2, \ldots, n$ and $\{s_{j1}, s_{j2}, \ldots, s_{jn}\}$. Then we derive the clauses $\{\neg o_i\}$ from $\{\neg o_i, \neg s_{ij}\}$ $j = 1, 2, \ldots, n$ and $\{s_{i1}, s_{i2}, \ldots, s_{in}\}$. Finally we derive the empty clause from $\{\neg o_i\}$ and $\{o_1, o_2, \ldots, o_n\}$.

In this derivation each clause in (1)-(5) was used exactly once so $\mathcal{C}_n^{\text{prop}}$ has refutation which has $n^3 + n^2 + 2n + 1$ leaves. The derivation tree is a binary tree so the derivation uses $2n^3 + 2n^2 + 4n + 1$ clause.

These considerations leads to

Lemma 1: *Assume $\mathcal{C}_{\text{eq}}$ is unsatisfiable. Then there exists $l \in \mathbf{N}$ such that for each $n \geq 2$, there exists a tree-resolution refutation of $\mathcal{C}_n^{\text{prop}}$ of size $\leq n^l$*

Proof: Assume $\mathcal{C}_{\text{eq}}$ is unsatisfiable . Then $\mathcal{C}_{\text{eq}}^*$ is unsatisfiable. According to Herbrand's Theorem (see for example [19]) there exists a finite refutation of a suitable substitution instance. Actually there exists a finite tree-refutation on the normal form (described above). The first part of this refutation leads to a clause D of the form;

$$\{\neg h_1(z_{11}, \ldots, z_{1r_1}) = z_{1,r_1+1}, \neg h_2(z_{21}, \ldots, z_{2r_2}) = z_{2,r_2+1}, \ldots,$$

$$\ldots, \neg h_u(z_{u1}, \ldots, z_{ur_u}) = z_{u,r_u+1}\}$$

where each h_j denotes one of the function symbols $f_1, f_2, \ldots, f_l$ and where each z_{ij} denotes a variable (i.e. $x, y, z, x_{ij}, y_{i_1,i_2,i_3,\ldots,i_r}$ etc). All the variables which are not among the variables on the right hand sides (i.e. the variables not among $z_{1,r_1+1}, z_{2,r_2+1}, \ldots, z_{u,r_u+1}$) can be substituted with anything (for example c_1).

Consider all substitutions where the variables on the right hand sides (i.e. the variables $z_{1,r_1+1}, z_{2,r_2+1}, \ldots, z_{u,r_u+1}$) are replaced by for constants in $\{c_1, c_2, \ldots, c_n\}$. If there are u different atomic term equations in D, there are exactly n^u different derivations $A_{i_1,i_2,\ldots,i_u}$. All together these derivations use at most $\text{HC}(\mathcal{C}_{\text{eq}}^*)\ n^u$ different axioms.

The derivation $A_{i_1,i_2,\ldots,i_u}$ derives some clause $D_{i_1,i_2,\ldots,i_u}$. Each clause $D_{i_1,i_2,\ldots,i_u}$ expresses the propositional version of a list of negations of basic atomic formulas. And each of these basic equations has all free variables replaced by constants from $\{c_1, c_2, \ldots, c_n\}$.

Now we use the fact that D has an unification. The unification problem is to unify a list of the form

$$h_1(z_{11}, z_{12}, \ldots, z_{1r_1}) = z_{1,r_1+1}, h_2(z_{21}, z_{22}, \ldots, z_{2r_2}) = z_{2,r_2+1}, \ldots,$$

$$\ldots, h_u(z_{u1}, z_{u2}, \ldots, z_{ur_u}) = z_{u,r_u+1}$$

All variables z_{ij} which do not appear among the variables $z_{1,r_1+1}, z_{2,r_2+1}, \ldots, z_{u,r_u+1}$ are substituted with c_1 (or any other constant). Next notice that there is a natural partial ordering on the remaining variables. We define the ordering by defining z_{i,r_1+1} to dominate the variable z_{ij} if z_{ij} appears in $h_i(z_{i1}, z_{i2}, \ldots, z_{ir_i})$ (as well as on the right hand side somewhere).

The fact that D has an unification ensures this is a well defined ordering $\prec$. Without loss of generality we can assume that the u variables in the notation of $A_{i_1,i_2,\ldots,i_u}$ have been displayed according to some total ordering which extends $\prec$.

The clause $D_{i_1,i_2,\ldots,i_u}$ (in the propositional notation) is of the form

$$\{\neg h^1_{i_{11},i_{12},\ldots,i_{1r_1},i_1}, \neg h^2_{i_{21},i_{22},\ldots,i_{2r_2},i_2}, \ldots, \neg h^u_{i_{u1},i_{u2},\ldots,i_{ur_u},i_u}\}$$

Resolving with the clause $\{h^1_{i_{11},i_{12},\ldots,i_{1r_1},1}, h^1_{i_{11},i_{12},\ldots,i_{1r_1},2}, \ldots, h^1_{i_{11},i_{12},\ldots,i_{1r_1},n}\}$ we derive the clauses $D'_{i_2,\ldots,i_u}$.

$$\{\neg h^2_{i_{21},i_{22},\ldots,i_{2r_2},i_2}, \ldots, \neg h^u_{i_{u1},i_{u2},\ldots,i_{ur_u},i_u}\}$$

Notice that the variable i_1 does not appear elsewhere because the variable z_{1,r_1+1} was dominating all other terms. Now we precede and resolve the clauses $D'_{i_2,\ldots,i_u}$ for $i_2 = 1, 2, 3, \ldots, n$ with the clause $\{h^2_{i_{21},i_{22},\ldots,i_{2r_2},1}, h^2_{i_{21},i_{22},\ldots,i_{2r_2},2}, \ldots, h^2_{i_{21},i_{22},\ldots,i_{2r_2},n}\}$ and derive the clauses $D''_{i_3,\ldots,i_u}$

$$\{\neg h^3_{i_{31},i_{32},\ldots,i_{3r_3},i_3}, \ldots, \neg h^u_{i_{u1},i_{u2},\ldots,i_{ur_u},i_u}\}$$

We continue this procedure. Eventually we derive the empty clause. The number of clauses used in the leaves is at most HC($\mathcal{C}$) n^u where u is the number of atomic term equations in the clause D. The number u is bound by the number of function symbols which appear in the unification needed to refute $\mathcal{C}_{\text{eq}}$. □

Please notice that it is the non circular relationship between the variables on the right hand sides of D which allows this procedure to go through. If D, for example, happens to be $\{\neg S(x) = y, \neg S(y) = x\}$ it would be impossible to translate $\{\neg s_{ij}, \neg s_{ji}\}_{ij}$ (as well as $\{s_{i1}, s_{i2}, \ldots, s_{in}\}_i$) into a (polynomial size) resolution refutation. Well, the point of course is, that D can never be of the form $\{\neg S(x) = y, \neg S(y) = x\}$ because $S(x) = y$ and $S(y) = x$ (and thus $S(S(x)) = x$) have no unification.

Lemma 2: *Assume $\mathcal{C}$ is not satisfied in any infinite model. Then there exists $l \in \mathbf{N}$ such that for each $n \geq 2$, there exists a tree-refutation of $\mathcal{C}_n^{\text{prop}}$ of size $\leq n^l$*

Proof: If $\mathcal{C}$ has no infinite model according to the compactness theorem, it cannot have arbitrary large finite models. Assume $\mathcal{C}$ has no models of size

$\geq n_0$. Introduce constants $c_1, c_2, \ldots, c_{n_0}$ together with axioms $\{\neg c_i = c_j\}$ for $i \neq j$. It is clear that the system $\mathcal{C}_{n_0}^{\text{prop}}$ is unsatisfiable. Now consider a general $n \geq n_0$. According to Lemma 1 (with the obvious treatment of the constants $c_1, c_2, \ldots, c_{n_0}$) there exists a tree-refutation of $\mathcal{C}_n^{\text{prop}}$ of size $\leq n^l$. □

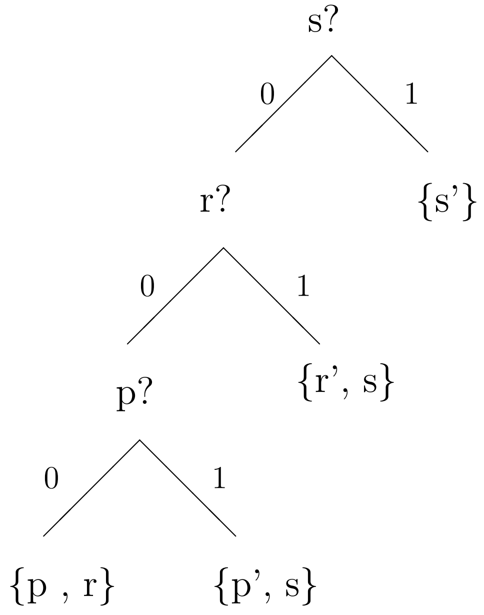
Corollary: *In theorem 1, (1) implies (3) and (2) implies (3).*

6 The exponential size lower bounds

For the lower bound it is convenient to view a tree-resolution refutation as a decision tree. This is essentially done by turning the refutation tree on its head (see for example [20]). On an input (i.e. a truth assignment) the decision tree outputs an unsatisfied clause. To illustrate the idea consider for example the tree-refutation:

$$\dfrac{\dfrac{\dfrac{\{p,r\} \quad \{\bar{p},s\}}{\{r,s\}} \quad \{\bar{r},s\}}{\{s\}} \quad \{\bar{s}\}}{\emptyset}$$

The refutation shows that $\{\{\bar{s}\},\{p,r\},\{\bar{p},s\},\{\bar{r},s\}\}$ is unsatisfiable. If we turn the refutation on its head we get the binary decision tree.



Decision tree refuting {{s'},{r', s},{p', s},{p , r}}

For any truth assignment of the variables (s, p and r) the decision tree uniquely determines a clause which is unsatisfiable.

We want to give a lower bound on the number of clauses needed to refute $\mathcal{C}_n^{\text{prop}}$ in a tree-refutation. This is equivalent of giving a lower bound of the number of nodes required by a decision tree which output an unsatisfied clause.

Now the variables in the clauses in $\mathcal{C}_n^{\text{prop}}$ are basic atomic sentences where the variables are substituted by concrete constants chosen from $\{c_1, c_2, \ldots, c_n\}$. Thus, we can view the decision tree as a decision tree which as input takes a model M of the language of $\mathcal{C}_{\text{eq}}$ extended by the constants $\{c_1, c_2, \ldots, c_n\}$. The decision tree outputs a clause which is unsatisfied by M.

If we restrict the inputs to models M which satisfy the clauses in $\mathcal{C}$, the decision tree never needs to output the clauses appearing from $\mathcal{C}_{\text{eq}}$. Only clauses expressing $\leq n$-ness and $\geq n$-ness are needed because all other clauses are automatically satisfied. If we further restrict the input to only include such models M for which the constants $c_1, c_2, \ldots, c_n$ are interpreted as different elements, the decision tree only needs to output clauses expressing $\leq n$-ness. These are clauses of the form $\{f(a_1, a_2, \ldots, a_r) = c_1, \ldots, f(a_1, a_2, \ldots, a_r) = c_n\}$.

All output nodes which do not output a clause expressing $\leq n$-ness can be cancelled and the decision tree can be simplified until it is a binary decision tree $\mathcal{T}$ where all leaves have been assigned an axiom of $\leq n$-ness.

Now we approach the key idea for the lower bound. We restrict the inputs further. We only consider inputs which are models M which have size $\geq n' \ \ (n' > n)$, which satisfy the clauses in $\mathcal{C}$, and which have the constants $c_1, c_2, \ldots, c_n, c_{n+1}, \ldots$ interpreted as different elements. Also we require that each element in M has a name c_j for some j. This extra requirement is permitted because if there is a model of $\mathcal{C}$ which has size n' then there is also such a model for which each of the constants $c_1, c_2, ...$ denotes its elements (skolem-lowenheims theorem is needed if we only want to consider at most countable many constants $c_\alpha, \ \alpha \in I$).

This restriction on the inputs allows us to simplify the decision tree T further. All nodes which correspond to outputs which can never be achieved can be removed. This leads to a decision tree T'.

Now the lower bound is almost trivial. Suppose that $\mathcal{C}$ has a model of size $\geq n'$. The model might be finite or infinite. We want to see what impact this has on the size of tree-resolution refutations of $\mathcal{C}_n^{\text{prop}}$.

The key observation is that each constant $c_1, c_2, \ldots, c_n$ must appear along each computational path through T'.

Lemma 3: *Assume there are models M of $\mathcal{C}$ which have size $> n$. Then each computational path through T' must involves each of the constants $c_1, c_2, \ldots, c_n$.*

Proof: Assume some branch does not involve the constant $c' \in \{c_1, c_2, \ldots, c_n\}$. The branch leads to a leaf l which has been assigned a clause on the form $\{f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c_1, f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c_2, \ldots, f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c_n\}$.

None of the inputs M which lead to the leaf l will have $f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c'$. The decision tree T' does not involve any of the constants $c_{n+1}, c_{n+2}, \ldots, c_{n'}$ (or $c_{n+1}, c_{n+2}, \ldots$ if $n' = \infty$) so none of the inputs M which lead to the leaf l will have $f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c_j$ for some $j = n+1, n+2, \ldots, n'$ (if this was the case, we get a contradiction by replacing the input M with a model M' in which the interpretation of c' and c_j is interchanged). But then all models M which lead to the leaf l fail to satisfy $f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = c$ for any constant c. Now we only considered inputs being models M for which each element has a name. Thus each model M which leads to the leaf l must have $\forall x\ \neg f(c_{i_1}, c_{i_2}, \ldots, c_{i_r}) = x$. This sentence is a logical contradiction. This is a contradiction to the fact that T' was constructed in such a way that there actually was some permissible input M leading to l. □

From this we get:

Lemma 4: *Assume n is given, and that there is a model of $\mathcal{C}$ which has size n' for some $n' > n$. Here $n' = \infty$ if $\mathcal{C}$ has an infinite model. Then there is no tree-refutation of $\mathcal{C}_n^{\text{prop}}$ which has size $< 2^{n/\max(k_C^{\text{rel}}, 1+k_C^{\text{fun}})}$.*

Thus (3) implies (1) and (2) in Theorem 1. This, in conjunction with the upper bounds, complete the proof of the first part of Theorem 1. The remaining part follows from the fact that the set of finite satisfiability problem $\mathcal{C}$ (for predicate logic) which is satisfiable in an infinite model is non-recursive. Actually the complement - the set of finite satisfiability problems $\mathcal{C}$ which has no infinite model - is r.e. complete.

Finally - to complete the proof of the Gap Theorem - we have to show that there is no total recursive function which given input $\mathcal{C}$, outputs $u \in N$ such that $n^u > S_C(n)$ whenever (1),(2) or (3) fails. If such total recursive function existed, we could use property (2) to provide a bound on smallest n_0 with $\mathcal{C}_n^{\text{prop}}$ unsatisfiable for each $n \geq n_0$. This would give us a decision procedure to decide whether $\mathcal{C}$ has an infinite model. It is well known that the existence of such a decision procedure leads to a contradiction. This completes the proof of the Gap Theorem.

7 Further perspectives

Up to this point we have considered uniform sequences $\mathcal{C}_n^{\text{prop}}$ of unsatisfiability problems. Notice however that our polynomial upper bound was achieved by highly uniform families of tree-refutations. This raises a crucial question. Given a propositional proof system P. When does P have the property that uniform generated (i.e. S_n-generated) sequences of tautologies which have short P-proofs have short uniformly (here used in an informal sense) generated P-proofs? In this paper we have seen that tree-resolution has this property.

Let T_{total} denote the mathematical theory which is axiomatised by all existential sentences ψ which are valid in all finite models (the sentences are written in an infinite language with an arbitrary number of function and relation symbols of each arity). This theory consists of the class of first order formulas which hold in all finite models. Notice that T_{total} is a well-defined theory because the property of being valid in all finite models is closed under logical deduction. The list of sentences in T_{total} forms a complete co-recursively enumerable set. Thus T_{total} is not recursively axiomatisable.

Given a propositional proof system P. In general we let $T_P \subseteq T_{\text{total}}$ denote the collection of existential sentences $\psi \in T_{\text{total}}$ for which the sequence ψ_n has polynomial size (or for example $n^{\log(\mathrm{n})^{o(1)}}$-size) P-proofs.

Let ψ be a sentence (formula) in predicate logic. Assume for example $\psi :\equiv \forall x \exists y \forall z R(x,y,z)$. The sentence (formula) $\psi' :\equiv \exists y R(c,y,f(y))$ is a *skolemisation* of ψ. Clearly ψ implies ψ' while ψ' does not implies ψ in general. In our context, things are very nice because we consider validity in a class of models (rather than validity in a single model). Notice that ψ holds in the class of all finite models if and only if ψ' holds in the class of all finite models. Often a sentence has more than one skolemisation. Using [4] it is possible to show that there exists a sequence of sentences ψ_u (in predicate logic) as well as two sequences ψ'_u and ψ''_u such that ψ'_u and ψ''_u both are skolemisations of ψ_u and such that the term-complexity as well as the Herbrand complexity of ψ''_u is 2_u-times larger than the term complexity as well as the Herbrand complexity of ψ'_u (here $2_{l+1} := 2^{2_l}$ and $2_0 = 1$). The function 2_u has a non-elementary growth rate.

Theorem 4: *For any reasonable (see proof for details) propositional system P, T_p is a well defined mathematical theory closed under logical deduction. More specifically T_p is closed under logical deduction in the following sense: Let ψ be an arbitrary sentence (not necessarily an existential sentence) which is a logical consequence of T_p. Then for any skolemisation of ψ turning it into an equivalent existential sentence ψ', the sequence ψ'_n have polynomial size P-proofs (i.e. $\psi' \in T_p$). If ψ'' is another skolemisation there are examples which show that the polynomial tree-resolution refutation complexity of $(\psi'')_n$ might have a degree which is non-elementarily larger than the degree of the tree-resolution refutation complexity of $(\psi')_n$. Thus the polynomial complexity of the sequence tautologies arising from different skolemisations can vary grossly. Still the translation procedure leads to robust complexity results: Either all translations produce sequences which all have polynomial size tree-resolution refutations or none of the translations produce sequences which have polynomial size tree-resolution refutations.*

Proof:(outline) The property of having polynomial size propositional proofs is closed under logical deduction. We showed this for tree-resolution, but for stronger systems (which polynomially simulate tree-resolution) this property follows as a corollary. More specifically the skolemisation ψ' is a logical consequence of ψ, and ψ is by assumption a logical consequence of T_P. Thus there exists a finite collection of axioms $\theta_1, \theta_2, \ldots, \theta_u \in T_P$ such that η, a skolemisation of $(\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_u \rightarrow \psi')$, is a logical tautology as well as an appropriate existential sentence of the form we consider.

Our polynomial upper bound ensures that the sequence η_n has polynomial size P-proofs. By assumption, the sequences $(\theta_1)_n$, $(\theta_2)_n \ldots (\theta_k)_n$ each have polynomial size P-proofs. The intuition tells us that in a reasonable propositional proof system this implies $(\eta)_n \equiv (\theta_1)_n \wedge (\theta_2)_n \wedge \ldots \wedge (\theta_k)_n \rightarrow (\psi')_n$ and thus the sequence ψ'_n has polynomial size P-proofs.

Some care is however needed at this stage, and we need to extend our upper bound result slightly. Let us assume that P is a refutation system which is at least as efficient as tree-resolution. The situation can then be described as follows: By assumption we have polynomial size refutations of the translation of the clauses
$\{\{l_{111}, l_{112}, \ldots, l_{11k}\}, \{l_{121}, l_{122}, \ldots, l_{12k}\}, \ldots, \{l_{1k1}, l_{1k2}, \ldots, l_{1kk}\}\}$ (the refutations of $(\neg\theta_1)_n$). Here each literal l_{ijm} denotes a basic atomic sentence or the negation of a basic atomic sentence. To simplify the notation we have assumed (and we can clearly do this without loss of generality) that all index run between 1 and k. We also assume we have polynomial size P-refutations of the clauses $\{\{l_{i11}, l_{i12}, \ldots, l_{i1k}\}, \{l_{i21}, l_{i22}, \ldots, l_{i2k}\}, \ldots, \{l_{ik1}, l_{ik2}, \ldots, l_{ikk}\}\}$ when $i = 2, 3, \ldots, k$ (refutations of $(\neg\theta_i)_n$ where $i = 2, 3, \ldots, k$). By assumption we can refute the CNF formula
$\theta :\equiv \wedge_{j=1}^{k} \wedge_{\pi \in S_k} \{\neg l_{j1\pi(1)}, \neg l_{j2\pi(2)}, \ldots, \neg l_{jk\pi(k)}\} \wedge \wedge_{i=1}^{k} \{m_{i1}, m_{i2}, \ldots, m_{ik}\}$ (θ is a CNF sentence (in predicate logic) which is logically equivalent to $\neg\eta$). According to our main result (the upper bound) there exists polynomial size tree-refutations of the propositional translations of θ. We want to show we can obtain polynomial size P-refutations of the CNF $\wedge_{i=1}^{k} \{m_{i1}, m_{i2}, \ldots, m_{ik}\}$ (i.e. a refutation of $\neg(\psi')_n$). To show this we need to assume P satisfies the following natural condition:
Assumption (slightly informal): If $A_1 \wedge A_2 \ldots A_k \rightarrow B$ has a derivation of size s, then the CNF of $A_1 \wedge A_2 \ldots A_{k-1} \rightarrow (B \vee \neg A_k)$ has a derivation of size $s^{O(1)}$.

This assumption suffices (we leave the rather tedious details to the reader) and ensures that it is possible to construct polynomial size P-refutation proofs of the propositional translations of $\wedge_{i=1}^{k} \{m_{i1}, m_{i2}, \ldots, m_{ik}\}$. This construction is very similar to the construction in our polynomial upper bound. We call a propositional system P as *reasonable* if it satisfies the polynomial upper bound for translations of tautologies (this happens, in particular, for all propositional proof systems which are stronger than tree-resolution) and satisfies the assumption above. This concludes the proof that T_P is a well defined and a well-behaved theory.

There exists a sequences of sentences ψ which have skolemisations ψ' and ψ'' which leads to unsatisfiability problems $\mathcal{C}_{\psi'}$ and $\mathcal{C}_{\psi''}$ (of predicate logic) such that the Herbrand Complexity of the two systems differ with a non-elementary function (see [4] for details). Also the term complexity might differ by a non-elementary function. The canonical translation (we used for our upper bound) thus leads to tree-resolution refutations in which the degree in the polynomial complexity differs by a non-elementary function.

The final part of the theorem follows from the extension of our upper bound we already discussed. □

Notice that the assumption that P is a *reasonable* propositional system is a harmless assumption. Only reasonable propositional proof systems are relevant for complexity theory. An unreasonable propositional system can always be strengthened (by adding a few extra deduction rules) such that it becomes reasonable. All standard propositional systems are reasonable.

To show a super-polynomial lower bound for the propositional system P it suffices to show that $T_P \neq T_{\text{total}}$. In this paper we showed that when P is tree-resolution then T_P is the minimal theory i.e. just predicate logic. For tree-resolution T_P is recursively axiomatisable (axiomatised by the empty set (!) of axioms over predicate logic). In general for stronger propositional proof systems T_P need not be recursively axiomatisable. It is a mistake to equate T_P with $I\Delta_0(\alpha)$ (resp. V_1^1 or U_1^1) when P is polynomial size bounded depth Frege proofs (resp. polynomial size extended Frege proofs or $n^{\log(n)^{o(1)}}$-size Frege proofs) [remark: for readers who are not familiar with the theories $I\Delta_0(\alpha)$, V_1^1 or U_1^1 it suffices to know that these are all recursively axiomatisable by universal axioms (see [17])]. The theory T_P is axiomatised by purely existential axioms and thus it behaves different from the theories ($I\Delta_0(\alpha)$, V_1^1 or U_1^1) which not are axiomatisable by purely existential axioms.

It is possible to formulate various interesting conjectures [27] concerning general properties of the axiomatisations of T_P. Here we only state the most general of these conjectures:

Conjecture: *For any propositional proof system* $T_P \neq T_{\text{total}}$.

Using Cook's and Reckhow's reformulation of the NP $\neq$ co-NP question [13] we get:

Proposition: *The conjecture implies NP $\neq$ co-NP.*

The main aim in this section was to define the theory T_P (where P is an arbitrary propositional proof system). Our main result can now be stated as follows:

Theorem 5: *Let P denote tree-resolution and let $T_P(f) \subseteq T_{\text{total}}$ denote the theory of principles $\psi \in T_{\text{total}}$ which have $f(n)$-size tree proofs (refutations).*

The theory $T_P(f)$ is well-defined as well as well-behaved (Theorem 4). Furthermore $T_P(f)$ is the same theory (namely predicate logic) for any superpolynomial but sub-exponential function $f(n)$.

8 General considerations

Consider the theory $T_P(f)$ defined in the previous section. When the growth rate of f changes the theory $T_P(f)$ might of course also change. The question is whether this change happens in a continuous manner or - as we have shown for tree-resolution - happens in jumps.

This touches a fundamental question in complexity theory. It is an empirical fact that only a relative small number of complexities appears in practice (e.g. $\theta(1)$, $\theta(\sqrt{\log(n)/\mathrm{loglog}(n)})$, $\theta(\log(n)/\mathrm{loglog}(n))$, $\theta(n)$, $\theta(nlogn)$, $\theta(n^2)$, $2^{n^{\theta(1)}}$). This phenomena is folklore, but to my knowledge there is at present no heuristic explanations of this.

Theoretically, virtually any complexity is possible, so why do so few complexities appear in practice? One feature of real world computational problems is that they in some sense involve the *same* general computational problem. It would, for example, be highly unnatural to consider a computational problem where certain lists have to be sorted for some values of n while certain bin-packing problems have to be solved for other values of n. Uniformity is clearly a feature of real world problems as we meet them in theoretical computer science. I hope the reader will forgive me these pure speculations, but perhaps *uniformly* (here used informally) given computational problems always have worst case complexities belonging to a finite list (including e.g. $\theta(1)$, $\theta(\sqrt{\log(n)/\mathrm{loglog}(n)})$, $\theta(\log(n)/\mathrm{loglog}(n))$, $\theta(n)$, $\theta(nlogn)$, $\theta(n^2)$, $2^{n^{\theta(1)}}$) of possibilities? Perhaps the fact that so few complexities appear in complexity theory are the shadows of a master theorem - a theorem which states that uniform complexity questions only can have certain possible answers (among finitely many possibilities). Or does this phenomena only reflect our limitations in showing matching upper and lower bounds? In the setting of propositional logic perhaps $T_P(f)$ always makes at most finitely many jumps. Clearly (using Cook's and Reckhow's result [13]) if there is a complexity gap for any propositional system P we must have NP $\neq$ co-NP.

It is, of course, also possible that our gap-theorem is a rather singular phenomena which only occur for weak propositional proof systems (like tree-resolution) where certain versions of the so-called weak pigeon-hole principle fails.

Let me point out that it is possible to extend the complexity gap theorem to a wider class of tautologies. In the present paper we solely focused on the propositional versions of *the first finitisation principle* [23]. This was mainly done to keep the proofs as simple as possible. It is possible to achieve a polynomial ↔ exponential complexity gap related to the propositional version of *the second finitisation principle* [23]. In order to keep the application of mathematical logic to a pleasant minimum we will not discuss this extension in the present paper.

Let me point out that there also is a complexity gap for the so-called Nullstellensatz proof system as well as for the Polynomial calculus. The exact size of this gap is still not completely settled. From [18] and [29] it is however clear that we essentially have a complexity jump from polynomial size NS-proof (PS-proof) to at least $n^{\log(n)}$. We suggest that the actual jump is from polynomial size to size $2^{o(n)}$. This was claimed (with incomplete proof) in [30].

A very interesting question is to classify the collection of propositions Ψ which have polynomial size refutations for different propositional proof systems (or equivalently to characterise the theory T_P). Consider for example the NS-proof system (over fields of characteristic 0). This is a very interesting propositional proof system which has been studied intensively in the recent years. The system was first introduced in [5] and has many nice features [12]. We finish the paper by showing that the Nullstellensatz proof system proves the following version of the pigeon-hole principle.

For fixed $n \in \mathbf{N}$ consider the class Poly_n of polynomials in the variables x_{ij} where $i \in \{0, 1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, n\}$.

Consider the following polynomial equations:

$$(\sum_{j=1, j\neq i}^{n} x_{ij}) + x_{ii} - 1 = 0 \text{ for } i = 1, 2, \ldots, n, \quad (\sum_{j=1}^{n} x_{0j}) - 1 = 0.$$

$$(\sum_{j=1, j\neq i}^{n} x_{ji}) + x_{ii} + x_{0i} - 1 = 0 \text{ for } i = 1, 2, \ldots, n.$$

These equations have no 0/1-solution as such a solution would define a bijection from $\{0, 1, \ldots, n\}$ onto $\{1, 2, \ldots, n\}$. Actually we show:

Proposition: *The equations do not have any solutions over any ring.*

Proof: Notice that

$$\sum_{i=1}^{n}((\sum_{j=1, j\neq i}^{n} x_{ij}) + x_{ii} + x_{0i} - 1) - \sum_{i=1}^{n}((\sum_{j=1, j\neq i}^{n} x_{ji}) + x_{ii} - 1) - (\sum_{j=1}^{n} x_{0j}) - 1 = 1$$

and that 1 thus can be written as a linear combination of the polynomials which appear in the polynomial equations. □

The tautologies for the usual pigeonhole principle is an extension of the above equations. Besides the equations above, they include:

$x_{ij}^2 - x_{ij} = 0$, where $i \in \{0, 1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, n\}$.

$x_{ij}x_{ik} = 0$ for $i \in \{0, 1, 2, \ldots, n\}$ and $j, k \in \{1, 2, \ldots, n\}$ with $j \neq k$.

$x_{ji}x_{ki} = 0$ for $i \in \{1, 2, \ldots, n\}$ and $j, k \in \{0, 1, 2, \ldots, n\}$ with $j \neq k$.

It is well known that the bijective pigeon-hole principle requires exponential size bounded depth Frege proofs [7]. Thus we have:

Theorem 6: *There exists a sequence of tautologies which has linear size Nullstellensatz proofs, but require exponential size bounded depth Frege Proofs.*

This shows that the strength of the NS-proof system is incompatible to Bounded Depth Frege. It also shows that the complexity gap for the NS-proof system does not take place at the same place as for tree-resolution (i.e. $T_{\text{TR}} \subset T_{\text{NS}}$ with $T_{\text{TR}} \neq T_{\text{NS}}$). An exact characterisation of T_{NS}, i.e. an exact characterisation of the class of $\mathcal{C}_n$'s which have polynomial size NS-proofs, is open. Is the theory T_{NS} recursively axiomatisable? Also the size of the complexity jump for the NS-proof system is open although some special cases has been settled [29]. The same questions for bounded depth Frege are open, and any answer has to involve ideas beyond current techniques. Does the cutting plane propositional proof system have a complexity gap? Does unrestricted resolution? Does the unrestricted resolution system have a gap-theorem similar to that for tree-resolution?

To end I would like to thank Peter Bro Miltersen and Ulrich Kohlenbach for their useful comments related to this paper.

References

[1] Ajtai, M.: The complexity of the pigeonhole principle. In 29th Annual Symposium on Foundations of Computer Science, IEEE (1988) pp. 346-355.

[2] Ajtai, M.: The complexity of the pigeonhole principle. Combinatorica, 14(4) (1994) pp. 417-433.

[3] Ajtai, M.: The independence of the modulo p counting principles. In Proceedings of the 26th ACM STOC, (1994) 402-411

[4] Baaz, M.,Leitsch, A.: On Slolemization and proof complexity. Fund. Inform. 20 no. 4 (1994) 353-379

[5] Beame, P., Impagliazzo, R., Krajicek, J., Pitassi, T., Pudlak, P.: Lower bounds on Hilbert's Nullstellensatz and propositional proofs. Proceedings of the London Mathematical Society **73(3)** 1-26 (1996)

[6] Beame, P.,Karp R., Pitassi T., Saks M.: On the complexity of Unsatisfiability Proofs for Random k-CNF Formulas. STOC 98. The 30th ACM STOC, (1998)

[7] Beame, P.,Impagliazzo, R.,Krajicek, J., Pitassi,T.,Pudlak,P., Woods,A.: Exponential lower bounds for the pigeonhole principle, In the proceedings of the 24th ACM STOC 200-221 (1992)

[8] Beame, P., Riis, S.: More on the relative strength of counting principles. In: Proceedings of the DIMACS workshop on Feasible Arithmetic and Complexity of Proofs, (1996)

[9] Buss, S.: The propositional pigeonhole principle has polynomial size Frege proofs, J. of Symbolic Logic, 52 (1987) 916-927

[10] Buss, S.: Propositional consistency proofs, Annals of Pure and Applied Logic 52 (1991) 3-29

[11] Buss, S., Pitassi, T.: Resolution and the weak Pigeonhole Principle (notes).

[12] Buss, S., Krajicek, j,. Pitassi, T., Razborov, A., Sergal, J.: Polynomial bound on Nullstellensatz for counting principles. To appear in Computational Complexity (1997)

[13] Cook, S., Reckhow, R.: The relative efficiency of propositional proof systems, Journal of Symbolic Logic, 44 (1979) 36-50.

[14] Galil, Z.: On the complexity of regular resolution and the Davis-Putnam procedure. Theoretical Computer Science, 4 (1977) 23-46

[15] Kearns, Vazirani: An Introduction to Computational Learning Theory, MIT Press, (1994)

[16] Haken, A.: The intractability of resolution, Theoretical Computer Science, 39 (1985), 297-308.

[17] Krajicek, J.:Bounded Arithmetic, propositional logic, and complexity theory, Encyclopedia of Mathematics and Its Applications, Vol. 60, Cambridge University Press (1995)

[18] Krajicek, J.: On the degree of ideal membership proofs from uniform families of polynomials over a finite field (manuscript)

[19] Leitsch, A.: The resolution Calculus. Book in the series of Texts in Theoretical Computer Science, Ed. Brauer, W., Rozenberg, G., Salomaa, A. Springer / Heidelberg (1996)

[20] Lovasz, L. Naor,M., Newman,I.,Wigderson,A. Search problems in the decision tree model. In Proceedings of the 32th ACM STOC, (1991) 576-585

[21] Pudlak, P.: On the length of proofs of finitistic consistency statements in first order theories, in: J.B. Paris et al., eds, Logic Colloquium '84 North-Holland, Amsterdam (1986) 165-196

[22] Pudlak, P.: Improved bounds to the lengths of proofs of finitistic consistency statements, in: Logic and combinatorics, Contemporary Math. 65 (Amer. Math. Soc., Providence, RI (1987) 309-331.

[23] Riis, S.: Making infinite structures finite in models of Second Order Bounded Arithmetic. In: Arithmetic, proof theory and computorial complexity, 289-319, Oxford: Oxford University Press 1993

[24] Riis, S.: Independence in Bounded Arithmetic. DPhil dissertation, Oxford University (1993)

[25] Riis, S.: Count(q) does not imply Count(p) Annals of Pure and Applied Logic, 90(1-3) (1997) 1-56

[26] Riis, S.: Count(q) versus the pigeon-hole principle. Archive for Mathematical Logic **36** (1997) 157-188

[27] Riis, S.: (manuscript in preparation)

[28] Riis, S., Sitharam, M: Generating hard tautologies using logic and the symmetric group. BRICS RS-98-19.

[29] Riis, S., Sitharam, M: Uniformly Generated Submodules of Permutation Modules. BRICS RS-98-20.

[30] Riis, S., Sitharam, M: (manuscript in preparation). Incomplete version appear as: Non-constant Degree Lower Bounds imply Linear Degree Lower Bounds, Technical report TR97-048 of the *Electronic Colloquium on Computational Complexity,* http://www.eccc.uni-trier.de/pub/eccc (1997)

[31] Simpson,S.: Unprovable theorem and fast growing functions. Contempora Mathematics Vol 65 (1987) 359-394

[32] Tseitin, G.S.: On the complexity of derivations in the propositional Calculus. In A.O. Slisenko, editor, Studies in Constructive Mathematics and Mathematical Logic, Part II (1968)

[33] Urquhart, A.: Hard examples for resolution. Journal of the ACM, 34(1) (1987) 209-219.

[34] Urquhart, A.: The Complexity of Propositional Proofs. The Bulletin of Symbolic Logic, 1 (1995) 425-467

Recent BRICS Report Series Publications

RS-99-29 Søren Riis. *A Complexity Gap for Tree-Resolution*. September 1999. 33 pp.

RS-99-28 Thomas Troels Hildebrandt. *A Fully Abstract Presheaf Semantics of SCCS with Finite Delay*. September 1999. 37 pp. To appear in *Category Theory and Computer Science: 8th International Conference,* CTCS '99 Proceedings, ENTCS, 1999.

RS-99-27 Olivier Danvy and Ulrik P. Schultz. *Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure*. September 1999. 57 pp. To appear in the November 2000 issue of *Theoretical Computer Science*. This revised report supersedes the earlier BRICS report RS-98-54.

RS-99-26 Jesper G. Henriksen. *An Expressive Extension of TLC*. September 1999. 20 pp. To appear in Thiagarajan and Yap, editors, *Fifth Asian Computing Science Conference,* ASIAN '99 Proceedings, LNCS, 1999.

RS-99-25 Gerth Stølting Brodal and Christian N. S. Pedersen. *Finding Maximal Quasiperiodicities in Strings*. September 1999. 20 pp.

RS-99-24 Luca Aceto, Willem Jan Fokkink, and Chris Verhoef. *Conservative Extension in Structural Operational Semantics*. September 1999. 23 pp. To appear in the *Bulletin of the EATCS*.

RS-99-23 Olivier Danvy, Belmina Dzafic, and Frank Pfenning. *On proving syntactic properties of CPS programs*. August 1999. 14 pp. To appear in Gordon and Pitts, editors, *3rd Workshop on Higher Order Operational Techniques in Semantics,* HOOTS '99 Proceedings, ENTCS, 1999.

RS-99-22 Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir. *On the Two-Variable Fragment of the Equational Theory of the Max-Sum Algebra of the Natural Numbers*. August 1999. 22 pp.

RS-99-21 Olivier Danvy. *An Extensional Characterization of Lambda-Lifting and Lambda-Dropping*. August 1999. 13 pp. Extended version of an article to appear in *Fourth Fuji International Symposium on Functional and Logic Programming,* FLOPS '99 Proceedings (Tsukuba, Japan, November 11–13, 1999). This report supersedes the earlier BRICS report RS-98-2.