Linear lower bounds and simulations in Frege systems with substitutions*

Maria Luisa Bonet** Nicola Galesi***

Abstract. We investigate the complexity of proofs in Frege ($\mathcal{F}$), Substitution Frege ($s\mathcal{F}$) and Renaming Frege ($r\mathcal{F}$) systems. Starting from a recent work of Urquhart and using Kolmogorov Complexity we give a more general framework to obtain superlogarithmic lower bounds for the number of lines in both tree-like and dag-like $s\mathcal{F}$. We show the previous known lower bound, extend it to the tree-like case and, for another class of tautologies, we give new lower bounds that in the dag-like case slightly improve the previous one. Also we show optimality of Urquhart's lower bounds giving optimal proofs. Finally we give the following two simulation results: (1) tree-like $s\mathcal{F}$ p-simulates dag-like $s\mathcal{F}$; (2) Tree-like $\mathcal{F}$ p-simulates tree-like $r\mathcal{F}$.

1 Introduction

Since the work of Cook and Reckhow [CR], the study of complexity of proofs in propositional logic is viewed as related to main questions like $NP \neq \text{co}NP$ in Complexity Theory. The main open problem is whether for all propositional proof systems there exists a class of tautologies requiring superpolynomial size proofs.

Frege ($\mathcal{F}$), Substitution Frege ($s\mathcal{F}$) and Extended Frege ($e\mathcal{F}$) systems are fundamental propositional proof systems for which this kind of question is far away from being answered. Actually the best known lower bounds are only linear for the number of lines and quadratic for the number of symbols in $\mathcal{F}$ and $e\mathcal{F}$ ([Bu1]). In the same work, Buss posed the open question of finding superlogarithmic lower bounds for the number of lines in $s\mathcal{F}$ and Urquhart in [Ur] showed how to obtain a $\Omega(\frac{n}{\log n})$ lower bound for the number of lines needed for the proofs of some tautologies in the class $\top_x$, where $\top_x$ is a tautology associated with a binary string x.

Here we reformulate Urquhart's proof in terms of Kolmogorov Complexity giving a more general and intuitive approach to his technique. This way we obtain his lower bound for the class of tautologies $\top_x$ and also we show that $\Omega(n)$ lines are needed in the tree-like case (a result that can also be proved with

* Authors' Address: Universitat Politecnica de Catalunya, Dept. de Llenguatges i Sistemes Informatics, Jordi Girona Salgado 1-3, 08034 Barcelona, Spain. e-mail: {bonet, galesi} @lsi.upc.es

** Research partly supported by NSF grant number CCR-9403447.

*** Supported by European Community grant under the project Training and Mobility of Researchers.

another technique discussed in the paper). Moreover, with the same method, we obtain lower bounds for $s\mathcal{F}$ that, in the dag-like case slightly improve Urquhart's one, using a class of tautologies known as permutation tautologies introduced in [Or1]. Also we give proofs for the tautologies $\top_x$ in both tree-like and dag-like $s\mathcal{F}$ obtaining as a consequence that the lower bounds given are optimal.

Since there is a logarithmic speed-up between tree-like and dag-like $s\mathcal{F}$ proofs of $\top_x$ and since the given bounds are optimal, we approach the question of whether tree-like $s\mathcal{F}$ simulates dag-like $s\mathcal{F}$ with a logarithmic factor of increment in the number of lines. This result holds for $\mathcal{F}$ ([BB]) and we prove it for $e\mathcal{F}$. At his point is still open if the same holds for $s\mathcal{F}$. Nevertheless, we are able to show a polynomial simulation result, obtaining it in an indirect way by the following steps: (1) $e\mathcal{F}$ p-simulates $s\mathcal{F}$; (2) tree-like $e\mathcal{F}$ p-simulates $e\mathcal{F}$; (3) tree-like $s\mathcal{F}$ p-simulates tree-like $e\mathcal{F}$. The first simulation is work of Krajìček and Pudlàk [KP], and the third is a result of [CR] that we improve to $O(n \log n)$ lines instead of $O(n^2)$.

Finally we prove that $\mathcal{F}$ linearly simulates tree-like $s\mathcal{F}$ only in the number of lines (whether this holds or not also for the number of symbols is a big open problem). This result has as a consequence that $\mathcal{F}$ p-simulates tree-like Renaming Frege $(r\mathcal{F})$[4] - a restricted version of $s\mathcal{F}$ introduced in [Bu1]. This is quite surprising if we consider Buss' result that dag-like $r\mathcal{F}$ p-simulates $s\mathcal{F}$ and that for $\mathcal{F}$, $e\mathcal{F}$ and $s\mathcal{F}$ the tree-like systems p-simulate the dag-like ones. So tree-like $r\mathcal{F}$ p-simulates dag-like $r\mathcal{F}$ implies that $\mathcal{F}$ and $s\mathcal{F}$ are polynomially equivalent.

The paper is organized as follows. In section 2 we give basic definitions. In section 3 we prove lower bounds using the Kolmogorov Complexity approach and explain how this is a reformulation of Urquhart's technique. In section 4 we prove upper bounds for Urquhart's tautologies showing the optimality of the lower bounds. In section 5 we prove that tree-like $s\mathcal{F}$ p-simulate $s\mathcal{F}$ and that tree-like $\mathcal{F}$ p-simulates tree-like $r\mathcal{F}$.

2 Preliminaries on propositional Hilbert-style proof systems

A *Frege system* $\mathcal{F}$ is an inference system for propositional logic based on (1) a language of well-formed formulas obtained from a numerable set of propositional variables and any finite propositionally complete set of connectives; (2) a finite set of axiom schemes; and (3) the rule of *Modus Ponens* ($\frac{A \quad A \rightarrow B}{B}$). A *proof* P of the formula A in a Frege system is a sequence $A_1, \ldots, A_n$ of formulas such that A_n is A and every A_i is either an *instance* of an axiom scheme or it is obtained by the application of the Modus Ponens from premises A_j and A_k with $j, k < i$. We will call A a *theorem* and we write $\vdash A$. A proof P is said to be *tree-like* if every A_i is used only once as premise of a rule in P. Any Frege system must

[4] While we were writing the paper, we have learned from T. Pitassi that the same result has been obtained with a similar technique in [IP].

be *sound* and *complete*. We write $A \models B$ if every truth assignment satisfying A also satisfies B, and $A \vdash B$ if adding the formula A among the axioms we can give a proof of B. A Frege proof system is *implicationally complete* if whenever $A \models B$, then $A \vdash B$, and *implicationally sound* if whenever $A \vdash B$, then $A \models B$. The main notions of complexity of proofs are: (1) the *number of lines* of a proof P for $P = A_1, \ldots, A_n$ is equal to n; (2) the *number of symbols* or *size* $|P|$ defined as $\sum_{i=1}^{n} |A_i|$ where $|A_i|$ is the number of symbols in A_i. A Frege system $\mathcal{F}_1$ *p-simulates* another Frege system $\mathcal{F}_2$ if whenever a formula A has a proof P in $\mathcal{F}_2$ of size m, there is a proof P' of A in $\mathcal{F}_1$ of size $p(m)$, where p is a polynomial. $T_1 \mathrel{\big|\!\!\frac{f(n)}{\quad}} T_2$ denotes that the proof system T_1 simulates the proof system T_2 with a $f(n)$ factor of increment in the number of lines.

Frege systems can be extended with some extra rules. An *extended Frege system* $e\mathcal{F}$ is a Frege system augmented with the *extension rule*. This rule allows to write for any formula B, the formula $p \leftrightarrow B$ under the restrictions that p does not occur in B, in previous lines or in the last line of the proof. A *substitution* σ is a mapping from a finite set of propositional variables $\mathrm{Dom}(\sigma)$ to a set of well-formed formulas $\mathrm{Rng}(\sigma)$. It will be called *renaming* whenever $\mathrm{Rng}(\sigma)$ is a set of propositional variables and $\top/\bot$-*substitution* whenever $\mathrm{Rng}(\sigma) \subseteq \{\top, \bot\}$. By $A\sigma$ we denote the result of *simultaneously* replacing in A any propositional variable p_i in A with $\sigma(p_i)$. A *substitution Frege system* is a Frege system augmented with the substitution rule $\frac{A}{A\sigma}$ that from a formula A allows to infer the formula $A\sigma$ for any σ. It will be called *Renaming Frege system* ($r\mathcal{F}$) or $\top/\bot$-*Frege system* ($\top/\bot$-$\mathcal{F}$) whenever the substitutions are respectively renaming or $\top/\bot$. Some properties can be lost by adding extra rules. For example $s\mathcal{F}$, $r\mathcal{F}$ are implicationally complete but not implicationally sound. The following Theorem holds for any two implicationally complete systems.

Theorem 2.1 *([CR]) Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two systems differing at most in the set of connectives used. There exists a polynomial p such that if P is a proof of A in $\mathcal{F}_2$ of n lines and m symbols, there is a proof of A in $\mathcal{F}_1$ of $p(n)$ lines and $p(m)$ symbols. Moreover if $\mathcal{F}_1$ and $\mathcal{F}_2$ have the same set of connectives, then the simulations are linear.*

Since these linear simulations do not affect our work, we don't worry about the set of axiom schemes we use. We fix the set of the connectives as $\{\wedge, \rightarrow, \vee, \neg\}$. Also, we add to our language the constants $\top$ and $\bot$ whose intended meanings are TRUE and FALSE and the extra axiom $\top$.
Let σ and λ be two substitutions such that $\mathrm{Rng}(\lambda) \subseteq \mathrm{Dom}(\sigma)$, by $\lambda\sigma$ we denote the new substitutions θ such that $\mathrm{Dom}(\theta) = \mathrm{Dom}(\lambda)$, $\mathrm{Rng}(\theta) = \mathrm{Rng}(\sigma)$ and if $\lambda(p_i) = A_i$, then $\theta(p_i) = A_i\sigma$. Two formulas A and B are isomorphic if there is a bijective renaming (or relettering) σ such that $A = B\sigma$.

Let $\Sigma = \{A_1, \ldots, A_k\}$ be a set of formulas and $\mathrm{Var}(\Sigma) = \bigcup_{i=1}^{k} \mathrm{Var}(A_i)$. A substitution σ unifies Σ if $A_1\sigma = \ldots = A_k\sigma$ and in this case Σ is said to be *unifiable*. A substitution σ is a *most general unifier* (mgu) for Σ if and only if it unifies Σ and for any other unifier θ there is a substitution λ with $\mathrm{Dom}(\lambda) \subseteq \mathrm{Var}(\mathrm{Rng}(\sigma))$ such that $\theta = \lambda\sigma$. This means that a mgu of a set of formulas Σ is essentially unique since for any two mgu's σ_1 and σ_2 there is a bijective renaming λ such that $\sigma_1\lambda = \sigma_2$.

The following Theorem is due to Robinson (see [G] cap.8 for its proof).

Theorem 2.2 *There is a deterministic algorithm $\mathcal{A}$, that always halts, such that for any set Σ of formulas, if Σ is unifiable, then $\mathcal{A}$ outputs its most general unifier.*

Let A and $B \to C$ be two formulas. The rule of *condensed detachment* with premises A and $B \to C$ and conclusions D, denoted by $\mathbf{CD}(A, B \to C)$ was introduced originally in [Pr] and used in [HM, Ur]. It is defined as follows:

1. find an isomorphic formulas A' of A such that $\mathrm{Var}(A') \cap \mathrm{Var}(B \to C) = \emptyset$;
2. if $\mathcal{A}(\{A', B\}) = \sigma$, then change by a renaming σ to σ^*, in such a way that $[\mathrm{Var}(B\sigma^*) - \mathrm{Var}(B)] \cap \mathrm{Var}(C) = \emptyset$ and define $D = C\sigma^*$. If $\{A', B\}$ is not unifiable then $\mathbf{CD}(A, B \to C)$ is undefined.

Intuitively we can think of CD as a single rule merging together the Modus Ponens and the substitution rules. A *Condensed Detachment Frege system* $CD(\mathcal{F})$ is a Frege system whose only rule is the condensed detachment and where we use a finite number of axioms instead of axiom schemes. We suppose that the axioms used in $CD(\mathcal{F})$are indexed by an order and that in the proofs all the axioms are introduced in the first lines. It is easy to see that any proof in $CD(\mathcal{F})$can be transformed into such an equivalent one. We denote by $i = CD(j, k)$, with $j, k < i$, the fact that in a proof the formula in the line i has been obtained by the CD rule applied to formulas on line j and on line k.

The following are *Urquhart's tautologies*. Let x be a binary string, then

$$\top_x = \begin{cases} \top & \text{if } x = \epsilon \\ (\top \to \top_y) & \text{if } x = 1y \\ (\bot \vee \top_y) & \text{if } x = 0y \end{cases}$$

The *permutation tautologies* are defined as

$$\Pi_n = (p_1 \wedge (p_2 \wedge (\ldots \wedge (p_{n-1} \wedge p_n) \ldots))) \to (p_{\pi(1)} \wedge (p_{\pi(2)} \wedge (\ldots \wedge (p_{n-1} \wedge p_{\pi(n)}) \ldots)))$$

where p_i are propositional variables and π is a permutation function of $[1 \ldots n]$.

All log functions are in base two and $\mathbf{j}$ denotes the binary representation of the number j.

3 Lower bounds for $s\mathcal{F}$ using Kolmogorov Complexity

In [Ur], the following theorem is proven:

Theorem 3.1 *If P is a dag-like proof in $s\mathcal{F}$, then there is a dag-like proof P' in $CD(\mathcal{F})$ such that: (1) every step in P is a substitution instance of a step in P': (2) the number of lines of P' is less than or equal to the number of lines of P.*

It is easy to see that the same theorem also holds for the tree-like case. Therefore, for classes of tautologies that are not substitution instances of shorter tautologies, we can obtain lower bounds for the number of lines in dag-like (resp. tree-like) $s\mathcal{F}$, giving lower bounds for the number of lines in dag-like (resp. tree-like) $CD(\mathcal{F})$. So, in what follows we will only work with $CD(\mathcal{F})$.
Following [Ur] we define the *succinct representation* for a dag-like $CD(\mathcal{F})$ proof P of m lines as a string G_P over the alphabet $\{0, 1, \#\}$. For each line i in P we define $G_P(i)$ as follows: (1) if i corresponds to the axiom j, then $G_P(i) = \mathbf{j}$; (2) if i is $CD(j, k)$, then $G_P(i) = \mathbf{j\#k}$. G_P is defined as $G_P(1)\#\#G_P(2)\#\# \dots \#\#G_P(m)$ and it is easy to see that $|G_P| = O(m \log m)$. Algorithm $\mathcal{A}$ of Theorem 2.2 allows us to recover uniquely the conclusion of a CD rule from the two premises, so that a proof in $CD(\mathcal{F})$ can be recovered uniquely from its succinct representation.

Theorem 3.2 *([Ur]) From G_P we can recover P uniquely (up to a relettering of the variables).*

In the tree-like case we modify the definition of succinct representation, improving its size by a logarithmic factor, but preserving the previous Theorem.

The succinct representation S_P of a tree-like $CD(\mathcal{F})$ proof P of m lines is a string over the alphabet $\{0, 1, [,], \#\}$. For each line i we define $S_P(i)$ as follows: (1) if i is the axiom j, then $S_P(i) = \mathbf{j}$; (2) if i is $CD(j, k)$, then $SP(i) = [S_P(j)\#S_P(k)]$. S_P is defined as $S_P(m)$, and it is easy to see that $|S_P| = m(3 + \log d) = O(m)$, where d is the number of axiom in $CD(\mathcal{F})$.

Theorem 3.3 *([PB, Or]) If P is a tree-like $CD(\mathcal{F})$ proof of A, then from S_P we can recover uniquely a tree-like proof P' of A with the same number of lines of P.*

Proof. First from S_P we recover the skeleton (i.e. the tree structure) of the proof. Then starting from the leaves of the skeleton, we first recover the axioms from their index numbers, then proceeding by depth, for each internal node we recover uniquely, using the algorithm $\mathcal{A}$ of Theorem 2.2 the formula associated with that node. Since the number of the nodes in the skeleton is the number of lines of P, we have recovered a proof P' with the same number of lines. The last formula of P' is A since the root node of the skeleton corresponds to the last formula in P and it is the last one to be recovered. $\square$

3.1 Lower bounds for Urquhart's tautologies

We will give an extremely brief introduction to Kolmogorov Complexity quoted from [B-YGN] and outline the idea of our lower bound proof.

The original goal of Kolmogorov Complexity was to have a quantitative measure of the complexity of a finite object. Kolmogorov and others had the following idea: the regularities of an object can be used to give a short description of it; on the other hand, if an object is highly non-regular, or random, there should be no way of describing it that is much shorter than giving the full object itself. To formalize this notion, we first encode discrete objects as strings. Second, we want to have descriptions that can be handled algorithmically, so we identify descriptions with "programs for a sufficiently powerful model of computation".

Fix a Universal Turing Machine U whose input alphabet is $\{0,1\}$ and output alphabet is Σ. The Kolmogorov complexity of a string $x \in \Sigma^*$ is the minimum length of a program that makes U generate x and stops. Observe that this notion seems to depend on the choice of the Universal Turing Machine. However, it can be shown that changing the machine only affects this measure of complexity by an additive constant. Strings whose Kolmogorov complexity is equal, or close to, their length are called Kolmogorov random, or *incompressible.* These are strings that cannot be compressed algorithmically. As there are at most $2^n - 1$ binary "programs" of length $n-1$ or less, clearly there is some string of length n whose Kolmogorov complexity is at least n. For c even a small constant, this amounts to say that most strings of length n, all but a fraction of 2^{-c}, have Kolmogorov complexity $\geq n-c$, or are almost random (see [LV] pp. 96). Many combinatorial properties have simple proofs via this prepackaged counting argument. Suppose you want to show that property $P(x)$ holds for some string x. Take a Kolmogorov random string x. Assume that $P(x)$ is false; show that this gives a way to describe x concisely. This is a contradiction. In fact, this argument usually gives proofs that $P(x)$ holds with high probability as the majority of strings are Kolmogorov random up to small constant. In our case the property $P(x)$ will be "for the binary string x of size n, any proof of $\top_x$ in $CD(\mathcal{F})$ requires $\Omega(f(n))$ lines". We take x a random string (i.e. its shortest description is of length close to n) and suppose that $P(x)$ is false. Then we use the succinct representation of a proof of $\top_x$ to describe x succinctly, and this will give us a contradiction.
The next Theorem shows a $\Omega(n/\log n)$ lower bound for the number of lines required in $CD(\mathcal{F})$to prove tautologies in the class $\top_x$. Our result is an improvement of the same result of [Ur] since we count how many tautologies in $\top_x$are hard and also provide a trade-off between this number and the number of lines required.

Theorem 3.4 *([Ur]) For any $c > 0$ there are at least $2^n(1-2^{-c})+1$ binary string x of size n such that any dag-like proof of $\top_x$ in $CD(\mathcal{F})$ must have more than $\lfloor \frac{(n-c)}{d\log(n-c)} \rfloor = \Omega(\frac{n}{\log n})$ lines for some constant $d > 1$.*

Proof. Fix $c > 0$. Recall that for any $CD(\mathcal{F})$ proof of n lines the size of its succinct representation is $\leq bn\log n$ for some constant b. Let $n > c+2$ and assume that for all x of size n there exists a dag-like proof P in $CD(\mathcal{F})$ of $< \frac{(n-c)}{d\log(n-c)}$ lines with $d > 2b\log 3 > 1$. Let G_P be the succinct representation of P. So

$$\begin{aligned} |G_P| &< \tfrac{b(n-c)}{d\log(n-c)} \cdot \log(\tfrac{(n-c)}{d\log(n-c)}) \\ &= \tfrac{b(n-c)}{d\log(n-c)} \cdot [\log(n-c) - \log(d\log(n-c))] \end{aligned}$$

Since $d > 1$ and $n > c+2$ we have that $\log(d\log(n-c)) > 0$ and therefore

$$|G_P| < \frac{b(n-c)}{d}$$

Fix a string x of size n whose Kolmogorov complexity is $\geq n-c$. Observe that, once the axioms of $CD(\mathcal{F})$ are fixed, x can be reconstructed from G_P and from a program $\mathcal{P}$ that, on input G_P: (1) recovers the proof P, (2) takes the last formula $\top_x$ and (3) rebuilds x. $|\mathcal{P}| = O(1)$ since it is independent from x. This means that the Kolmogorov complexity of x is $\leq |G_P| \log 3 + O(1) = \frac{b(n-c)}{d} \log 3 + O(1)$ (the factor $\log 3$ is due to the fact that G_P is defined over a three symbols alphabet). Since $d > 2b \log 3$ we have that the Kolmogorov complexity of x is $< \frac{(n-c)}{2} + O(1)$. Given that x is a c-incompressible binary string, its Kolmogorov Complexity is $\geq n - c$, therefore we have that $\frac{n-c}{2} + O(1) > n - c$ or $O(1) > \frac{n-c}{2}$, which is a contradiction for n sufficiently large. The result then follows since there are at least $2^n(1 - 2^{-c}) + 1$ binary strings of size n whose Kolmogorov Complexity is greater than $n - c$ (see [LV] pp. 96). □

The next Theorem extends the previous result to tree-like $CD(\mathcal{F})$ giving a linear lower bound. Observe that another way to obtain this result for $s\mathcal{F}$ is using the result of Buss in [Bu1] that $\Omega(n)$ lines are required in $\mathcal{F}$ for proof of tautologies that are not substitution instance of any shorter tautology and Theorem 5.3. In order to simplify the proofs we do not specify the trade-off between c and the number of lines in the next Theorems of this section. It is easy to see that this trade-off can also be obtained in a similar way as in the previous Theorem.

Theorem 3.5 *For any $c > 0$ there are at least $2^n(1 - 2^{-c}) + 1$ binary string x of size n such that any tree-like proof of $\top_x$ in $CD(\mathcal{F})$ must have $\Omega(n)$ lines.*

Proof. Fix $c > 0$ and observe that for tree-like $CD(\mathcal{F})$ proof P of m lines the size of its succinct representation S_P is $O(m)$ and apply Theorem 3.3 to recover the last formula of the proof. □

3.2 Lower bounds for permutation tautologies

Permutation tautologies Π_n were introduced in [Or1]. It was been shown in [Or1] and [PB] that $\Omega(n \log n)$ lines are required for tree-like $\mathcal{F}$ proofs of some of them. Here we show that $\Omega(n)$ lines are required in dag-like $s\mathcal{F}$ and $\Omega(n \log n)$ in tree-like $s\mathcal{F}$. Observe that, if we count the size of indices of variables, then the size of any Π_n is $O(n \log n)$. Therefore this last lower bound is not really an improvement of Theorem 3.4.

Theorem 3.6 *For any $c > 0$, there are at least $2^{O(n \log n)}(1 - 2^{-c}) + 1$ permutations π over $[1 \ldots n]$ such that any dag-like $CD(\mathcal{F})$ proof of Π_n requires $\Omega(n)$ lines.*

Proof. Fix $c > 0$. Assume that for any permutation π over $[1 \ldots n]$ there is a dag-like $CD(\mathcal{F})$ proof for the tautology associated with π with $< bn$ lines, for some constant b. The $n!$ possible tautologies obtained from the associated permutations can be encoded in $\log(n!)$ bits. By Stirling formula we have that $n! \approx \frac{n^n\sqrt{2\pi n}}{e^n}$. So $\log(n!) \approx n \log n - O(n) + O(\log \sqrt{n})$ that is $O(n \log n)$. This

means that we can encode any permutation tautology with a string of $dn \log n$ bits for some constant d. Now consider a binary string $x \in \{0,1\}^{dn \log n}$ with Kolmogorov complexity $\geq |x| - c$. Let P be the proof of the permutation tautology associated with x with less of bn lines. The succinct representation G_P of P has size $O(n \log n)$. Applying the same method of Theorem 3.4 and noting that the length of x is $O(n \log n)$ we obtain the desired lower bound.
□

The bound for the tree-like case is obtained from the previous Theorem as in the case of Urquhart's tautologies.

Theorem 3.7 *For any $c > 0$, there are at least $2^{O(n \log n)}(1 - 2^{-c}) + 1$ permutations π over $[1 \dots n]$ such that any tree-like $CD(\mathcal{F})$ proof of Π_n requires $\Omega(n \log n)$ lines.*

4 Upper bound for Urquhart's tautologies

We know that some tautologies $\top_x$ associated with binary strings of size n require $\Omega(\frac{n}{\log n})$ lines $s\mathcal{F}$. Here we show how to obtain a dag-like $s\mathcal{F}$ proof in $O(\frac{n}{\log n})$ lines. Moreover we show that also the tree-like lower bound for T_x's is optimal since we give a $O(n)$ line tree-like $s\mathcal{F}$ proof for them.

4.1 Dag-like proofs of the class T_x

Consider the following formulas associated with a binary string x:

$$\top_x^p = \begin{cases} p & \text{if } x = \epsilon \\ (\top \rightarrow \top_y^p) & \text{if } x = 1y \\ (\bot \vee \top_y^p) & \text{if } x = 0y \end{cases}$$

Lemma 4.1 *All tautologies $p \rightarrow \top_x^p$ for any $x \in \{0,1\}^n$ with $n > 0$ can be obtained in a $s\mathcal{F}$ proof of $O(2^n)$ lines.*

Proof. First observe that $p \rightarrow \top_1^p = p \rightarrow (\top \rightarrow p)$ and $p \rightarrow \top_0^p = p \rightarrow (\bot \vee p)$ can be obtained in a constant number of steps. Now suppose we have in the proof all the tautologies $p \rightarrow \top_y^p$ for $|y| = n - 1$. To obtain $p \rightarrow \top_x^p$ for $x = y1$ (resp. $x = y0$) and $|x| = n$ we do following steps:

$p \rightarrow \top_y^p$	by ind. hyp.
$(\top \rightarrow p) \rightarrow \top_{y1}^p$	subst. p with $(\top \rightarrow p)$
$p \rightarrow \top_{y1}^p$	cut with $p \rightarrow (\top \rightarrow p)$ (resp. $p \rightarrow (\bot \vee p)$)

Each one of the $p \rightarrow \top_x^p$'s can be obtained in a constant number of lines from one of the $p \rightarrow \top_y^p$'s and one between the first tautologies, where $x = y1$. So the total numbers of lines to obtain all the $\top_x^p$ is $c \sum_{i=1}^n 2^i = O(2^n)$. □

Theorem 4.1 *Given $x \in \{0,1\}^n$ there is a $s\mathcal{F}$ proof of $\top_x$ in $O(\frac{n}{\log n})$ lines.*

Proof. The proof is divided in two parts. First, by previous Lemma, we obtain a proof of all tautologies $p \to \top_y^p$ for all $|y| = \log(\frac{n}{\log n})$. This part requires $O(\frac{n}{\log n})$ lines.
Second, to obtain $p \to \top_x^p$ divide x in $\frac{n}{\log(\frac{n}{\log n})} = \frac{n}{\log n - \log\log n}$ substrings $x_1, \ldots, x_k$ (with $k = \frac{n}{\log n - \log\log n}$) of size $\log(\frac{n}{\log n})$. In the first part we have already proved the tautologies $p \to \top_{x_i}^p$ for all i. We put them together to form $p \to \top_x^p$ starting from the innermost one in the following way: consider the sequence $p \to \top_{x_1}^p, \ldots, p \to \top_{x_k}^p$. Let $\overline{\top}_{x_{k-1}}^p$ be the formula obtained substituting p by $\top_{x_k}^p$ in $\top_{x_{k-1}}^p$. In a constant number of lines we obtain the formula $p \to \overline{\top}_{x_{k-1}}^p$ the following way:

$$
\begin{array}{ll}
p \to \top_{x_{k-1}}^p & \\
\top_{x_k}^p \to \overline{\top}_{x_{k-1}}^p & \text{subst. } p \text{ with } \top_{x_k}^p \\
p \to \overline{\top}_{x_{k-1}}^p & \text{cut with } p \to \top_{x_k}^p
\end{array}
$$

Now consider the sequence $p \to \top_{x_1}^p, \ldots, p \to \top_{x_{k-2}}^p, p \to \overline{\top}_{x_{k-1}}^p$ and iterate the previous steps. After $k = \frac{n}{\log n - \log\log n}$ iterations we obtain $p \to \top_x^p$ and then by replacing p by $\top$ and one instance of Modus Ponens we obtain $\top_x$. This second part requires $O(\frac{n}{\log n - \log\log n}) = O(\frac{n}{\log n})$ lines. The total number of lines is $O(\frac{n}{\log n})$. $\Box$

4.2 Tree-like proofs for the class $\top_x$

We give a tree-like $\mathcal{F}$ proof of $O(n)$ lines for any of the 2^n formulas $\top_x$ associated with binary strings x of size n.

Theorem 4.2 *For any $x \in \{0,1\}^n$, there is a tree-like proof of $\top_x$ of $O(n)$ lines.*

Proof. By induction on x. The base case is trivial. Let x be $1y$. Then we have a proof of $\top_y$ in $c(n-1)$ lines; introduce by an axiom, the line $\top_y \to (\top \to \top_y)$ and cut with $\top_y$. If x is $0y$, then use, in the same way, the axiom $\top_y \to (\top \vee \top_y)$. The proof of $\top_x$ is $c(n-1)+2$, and for $c \geq 2$ is less than or equal to cn. $\Box$

5 Simulations for Frege systems with substitutions

Optimal lower and upper bounds for $s\mathcal{F}$ proofs of $\top_x$ seem to suggest that tree-like $s\mathcal{F} \overset{O(n\log n)}{\vdash}$ dag-like $s\mathcal{F}$. This is also supported by the fact that the same result holds for $\mathcal{F}$ [BB] and for $e\mathcal{F}$ (see 5.1). But the technique used in [BB], which can also be applied to $e\mathcal{F}$, does not work for $s\mathcal{F}$. Moreover, note that until now it was not known whether tree-like $s\mathcal{F}$ p-simulates dag-like $s\mathcal{F}$. In this section we solve this problem.

5.1 Tree-like $s\mathcal{F}$ simulation of $s\mathcal{F}$

The task of obtaining a tree-like $s\mathcal{F}$ p-simulation of $s\mathcal{F}$ can be divided in the following steps: (1) $e\mathcal{F}$ p-simulates $s\mathcal{F}$; (2) tree-like $e\mathcal{F}$ p-simulates $e\mathcal{F}$; (3) tree-like $s\mathcal{F}$ p-simulates tree-like $e\mathcal{F}$. In the last two simulations it is possible to give a bound for the number of lines in the simulation proof as a function of the number of lines of the simulated proof. Namely we obtain

1. tree-like $e\mathcal{F} \mid\!\frac{O(n\log n)}{\quad}$ dag-like $e\mathcal{F}$;
2. tree-like $s\mathcal{F} \mid\!\frac{O(n\log n)}{\quad}$ tree-like $e\mathcal{F}$.

Consider the following formulas introduced in [Bo1]:

Definition 5.1 *Let $A_1, \ldots, A_n$ be formulas with n a power of 2. The Balanced conjunction $\bigwedge\!\!\!\bigwedge_{i=1}^{n} A_i$ of $A_1, \ldots, A_n$ is defined inductively by*

- *if $n = 1$, then $\bigwedge\!\!\!\bigwedge_{i=1}^{n} A_i$ is A_1;*
- *otherwise, $(\bigwedge\!\!\!\bigwedge_{i=1}^{n/2} A_i) \wedge (\bigwedge\!\!\!\bigwedge_{i=1}^{n/2} A_{n/2+i})$.*

Definition 5.2 *Let $A_1, \ldots, A_n$ be formulas with $n = 2^m + s$ with $0 < s \leq 2^m$. The Psuedobalanced conjunction $\bigwedge\!\!\!\bigwedge_{i=1}^{n} A_i$ of $A_1, \ldots, A_n$ is defined inductively by*

- *if $n = 1$,then $\bigwedge\!\!\!\bigwedge_{i=1}^{n} A_i$ is A_1;*
- *otherwise, $(\bigwedge\!\!\!\bigwedge_{i=1}^{2^m} A_i) \wedge (\bigwedge\!\!\!\bigwedge_{i=1}^{s} A_{(2^s+i)})$ where the first conjunct is balanced and the second pseudobalenced.*

From this point on all conjuncts $\bigwedge\!\!\!\bigwedge$ are intended to be pseudobalanced. The following Lemma is from [BB]:

Lemma 5.1 ([BB]) *The formula $(\bigwedge\!\!\!\bigwedge_{i=1}^{k-1} A_i) \wedge A_k \rightarrow (\bigwedge\!\!\!\bigwedge_{i=1}^{k} A_i)$, where the conjunctions are associated in a pseudobalanced way, has a tree-like $\mathcal{F}$proof of $O(\log k)$ lines.*

Tree-like $e\mathcal{F}$ simulation of $e\mathcal{F}$. This proof is similar to the analogous theorem for $\mathcal{F}$ proved in [BB]. We only sketch it.

Theorem 5.1 *Tree-like $e\mathcal{F}$ p-simulates dag-like $e\mathcal{F}$. Moreover the following result holds: tree-like $e\mathcal{F}$ $\mid\!\frac{O(n\log n)}{\quad}$ dag-like $e\mathcal{F}$.*

Proof. Let $P = A_1, \ldots A_n$ be a proof in $e\mathcal{F}$. Let $B_i = \bigwedge\!\!\!\bigwedge_{j=i}^{n} A_j$ for $i = 1, \ldots, n$ and $B_0 = \top$. The technique is that of obtaining separate tree-like proofs of $B_i \rightarrow B_{i+1}$ for all $i = 1 \ldots n-1$ in $O(\log i)$ lines depending on how A_{i+1} is inferred in P. Then prove in a tree-like way $B_n \rightarrow A_n$ and finally get A_n performing cuts between the previous proof. We treat only the case in which A_{i+1} is inferred by the extension rule. Assume that A_{i+1} is $p_k \leftrightarrow C_k$. Then obtain $B_i \rightarrow B_i$ in a constant number of steps, introduce $p_k \leftrightarrow C_k$ (this is correct since p_k never occurs in $B_0, \ldots, B_i$) and in a constant number of steps obtain $B_i \rightarrow B_i \wedge p_k \leftrightarrow C_k$. By previous Lemma we have $B_i \wedge p_k \leftrightarrow C_k \rightarrow B_{i+1}$ in $O(\log i)$ lines and finally $B_i \rightarrow B_{i+1}$ cutting with the previously obtained formula. $\Box$.

Tree-like $s\mathcal{F}$ simulation of tree-like $e\mathcal{F}$. Cook and Reckhow show in [CR] that $s\mathcal{F}$ p-simulates $e\mathcal{F}$. Here we show that this simulation is preserved also in the tree-like case and improve from $O(n^2)$ to $O(n \log n)$ the number of lines used.

Theorem 5.2 *Tree-like $s\mathcal{F}$ p-simulates tree-like $e\mathcal{F}$. Moreover the following result holds: tree-like $s\mathcal{F}$ $\vdash^{O(n \log n)}$ tree-like $e\mathcal{F}$.*

Proof. Let P be the $e\mathcal{F}$ tree-like proof $A_1, \ldots, A_n$, and suppose that k many of the A_i's are formulas of the form $p_j \leftrightarrow B_j$ introduced by the extension rule. Consider the formula B defined by $\bigwedge_{j=1}^{k}(p_j \leftrightarrow B_j)$ following the order of introduction of the extension rules. First we give a tree-like $\mathcal{F}$proof of $B \rightarrow A$. This is obtained by showing that for all $i = 1, \ldots, n$ there is a proof of $B \rightarrow A_i$.

case 1: A_i is an axiom, then there is a tree-like $\mathcal{F}$proof of $B \rightarrow A_i$ in a constant number of lines;

case 2: A_i is obtained by *Modus Ponens* from A_j and A_k with $j, k < i$. We can therefore assume there are tree-like $\mathcal{F}$proofs of $B \rightarrow A_j$ and $B \rightarrow (A_j \rightarrow A_i)$. A tree-like proof of $B \rightarrow A_i$ can be easy obtained in a constant number of lines.

case 3: It is easy to see that $B \rightarrow (p_l \leftrightarrow B_l)$ in $O(\log k)$ lines.

This first part requires $O(n \log k)$ lines. Now we obtain A by the following steps:

1. from Lemma 5.1 obtain $\bigwedge_{j=1}^{k-1}(p_j \leftrightarrow B_j) \wedge (p_k \leftrightarrow B_k) \rightarrow B$ in $O(\log k)$ lines;
2. Cut this formula with $B \rightarrow A$ and obtain $\bigwedge_{j=1}^{k-1}(p_j \leftrightarrow B_j) \wedge (p_k \leftrightarrow B_k) \rightarrow A$;
3. substitute B_k for p_k in the last formula, derive the axiom $B_k \rightarrow B_k$ and then obtain $\bigwedge_{j=1}^{k-1}(p_j \leftrightarrow B_j) \rightarrow A$;
4. iterate this process substituting the p_i in the reverse order respect to their introduction in P.

This second part requires $O(k \log k)$ lines. Since $k \leq n$, the total number of lines is $O(n \log n)$. □

5.2 Tree-like $\mathcal{F}$ simulations of tree-like $r\mathcal{F}$ and tree-like $\top/\bot$-$\mathcal{F}$

In this subsection we first show that tree-like $\mathcal{F} \vdash^{O(n)}$ tree-like $s\mathcal{F}$ and then we discuss differences between $s\mathcal{F}$, $r\mathcal{F}$ and $\top/\bot$-$\mathcal{F}$. The technique used is that of pushing substitution lines up above Modus Ponens lines, to obtain different instances of the axioms. This is also stated in Lemma 1.11 of [HM]. Let P be a tree-like $s\mathcal{F}$ proof. At each formula $A\sigma$ obtained by a substitution, we associate its *degree* d_σ as the depth of the formula $A\sigma$ in the tree associate to the proof P, and define the degree d_P of the proof as the maximal d_σ. Note that a *degree*-0 proof is a tree-like Frege proof.

Lemma 5.2 *Given a degree d tree-like $s\mathcal{F}$ proof P of the formula A, there is a tree-like $s\mathcal{F}$ proof P' of the formula A with degree strictly less than d.*

Proof. Without loss of generality we assume that substitutions are applied only to formulas that are conclusions of Modus Ponens. Let P be a tree-like $s\mathcal{F}$ proof and let $A\sigma$ be the last substitution in the proof. A is obtained by Modus Ponens from B and $B \to A$. Then P' is obtained from P by the following transformation:

$$\cfrac{\cfrac{B \quad B \to A}{A}}{A\sigma} \qquad \text{to} \qquad \cfrac{\cfrac{B}{B\sigma} \quad \cfrac{B \to A}{B\sigma \to A\sigma}}{A\sigma}$$

P' is a valid $d-1$ degree proof. □

Theorem 5.3 *Tree-like* $\mathcal{F} \vdash^{O(n)}$ *tree-like* $s\mathcal{F}$.

Proof. Let P a tree-like $s\mathcal{F}$ proof and assume that substitutions are only applied to conclusions of Modus Ponens. We construct, by induction on d_P a proof P' of degree 0. At each inductive step we copy the proof tree if no substitution rule is used; when a substitution is used we apply Lemma 5.2 and eliminate eventually substitutions of formulas introduced by axiom schemes and from another substitution. At the base case $d_{P'}$ is 0 and so we have done. Observe that in P' we eliminate all the lines obtained by substitution and maintain all the other lines, so the number of lines of P' is $n-k$. □

Let m be the size of P. It is easily seen that the formulas introduced by the axioms in P' could have $O(m^{k+1})$ number of symbols and so we cannot conclude that $\mathcal{F}$ p-simulates tree-like $s\mathcal{F}$. But, an immediate corollary of this last Theorem is that any lower bound for the number of lines for tree-like $\mathcal{F}$ must hold also for tree-like $s\mathcal{F}$.

We can restrict the substitutions to be renamings or $\top/\bot$-substitutions. In this case, indeed, every substitution in the proof does not add any new symbols. Therefore we have:

Theorem 5.4 *([IP])* $\mathcal{F}$ *p-simulates tree-like* $r\mathcal{F}$ *and tree-like* $\top/\bot$-$\mathcal{F}$.

We see that tree-like $r\mathcal{F}$ is p-equivalent to $\mathcal{F}$, and by Buss [Bu1] dag-like $r\mathcal{F}$ is p-equivalent to $s\mathcal{F}$. This means that tree-like $r\mathcal{F}$ p-simulates dag-like $r\mathcal{F}$ implies $\mathcal{F}$ and $s\mathcal{F}$ are p-equivalent. Since an exponential separation is conjectured between $\mathcal{F}$ and $e\mathcal{F}$, this must lie between tree-like and dag-like $r\mathcal{F}$ or $\top/\bot$-$\mathcal{F}$. This is very surprising since for $\mathcal{F}$, $s\mathcal{F}$ and $e\mathcal{F}$ the tree-like system and the dag-like one are p-equivalent. Observe that the known $r\mathcal{F}$ simulation of $s\mathcal{F}$ does not preserve the tree-like property (see [Bu1] Lemma 17) and the Cook-Reckhow proof that tree-like $s\mathcal{F}$ p-simulates tree-like $e\mathcal{F}$ (Theorem 5.2) cannot be extended to an analogous proof for $r\mathcal{F}$ or $\top/\bot$-$\mathcal{F}$.

6 Conclusions

We have shown how to obtain some quasi-linear lower bounds for the number of lines in $s\mathcal{F}$ using Kolmogorov Complexity. We think that what we have proven is the best that can be done using this particular application of Kolmogorov Complexity in the sense that no even quadratic lower bounds can be proved this way. This is because to recover a random string x we are actually recovering the entire proof of the tautology associated with x and this means that the succinct representation encodes more information than we really need. In our case the lower bounds for $\top_x$ are optimal in $s\mathcal{F}$, but this will in general not be true.

Aknowledgement

We would thank Ricard Gavaldá for some discussions on this paper and Sam Buss for reading a preliminary version and making some comments.

References

[B-YGN] R. A. Baeza-Yates, R. Gavaldá, G. Navarro. Bounding the Expected Length of Longest Common Subsequences and Forest. Invited talk at the *Third South American Workshop on String Processing* (WSP'96), Recife (Brazil), Aug. 8-9 1996.

[Bo1] M. Bonet. Number of symbols in Frege proofs with and without the deduction rule. In *Arithmetic, proof theory and computational complexity.* Oxford University Press Eds. P. Clote and J. Krajìček -1992.

[BB] M. Bonet, S. Buss. The deduction rule and linear and near-linear proof simulations. *Journal of Symbol Logic*, **58** (1993) pp. 688-709.

[Bu1] S. Buss. Some remarks on length of propositional proofs. *Archive for Mathematical Logic.* **34** (1995) pp. 377-394.

[Bu2] S. Buss, *Lectures on proof theory.* TR SOCS-96.1 School of C.S. - Mc Gill University 1996.

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

[G] J. H. Gallier. *Logic for Computer Science - Foundations of Automatic Theorem Proving.* J. Wiley & Sons. 1987

[HM] J. R. Hindley, D. Meredith. Principal type schemes and condensed detachment. *Journal of Symbolic Logic*, **55** (1990) pp. 90-105.

[IP] K. Iwana, T. Pitassi. Exponential Lower bounds for the tree-like Hajós Calculus. *Manuscript* 1997.

[Kr1] J. Krajìček. Speed-up for propositional Frege systems via generalizations of proofs. *Commentationes Mathematicae Universiatatis Carolinae*, **30** (1989) pp. 137-140.

[Kr2] J. Krajìček. On the number of steps in proof. *Annals of Pure and Applied Logic*, **41** (1989) pp. 153-178.

[KP] J. Krajìček, P. Pudlàk. Propositional proof systems, the consistency of first order theories and the complexity of computation. *Journal of Symbolic Logic*, **54**(1989) pp. 1063-1079.

[LV] M.Li, P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Application.* Springer-Verlag 1993.

[Or] V.P. Orevkov. Reconstruction of a proof from its scheme. *Soviet Mathematics Doklady* **35**(1987) pp. 326-329.

[Or1] V.P. Orevkov. On lower bounds on the lengths of proofs in propositional logic.(In Russian), *in Proc. of All Union Conf. Metody Matem. v Problemach iskusstvennogo intellekta i sistematicheskoje programmirovanie*, Vilnius , Vol I. 1980. pp. 142-144.
[P] R. Parikh Some results on the length of proofs. *Trans. A.M.S.*, **177**(1973) pp. 29-36.
[Pr] A. N. Prior. *Formal Logic*. Oxford, second edition, 1960.
[PB] P. Pudlak, S. Buss. How to lie without being (easily) convicted and the length of proofs in propositional calculus. *8th Workshop on CSL, Kazimierz, Poland, September 1994*, Springer Verlag LNCS n.995, 1995, pp. 151-162.
[Ur] A. Urquhart. The number of lines in Frege proof with substitution. *Archive for Mathematical Logic*. To appear.