Electronic Colloquium on Computational Complexity, Report No. 18 (2012)

Some definitorial suggestions for parameterized proof complexity

Jörg Flum[1] and Moritz Müller[2]

[1] Universität Freiburg, Germany, joerg.flum@math.uni-freiburg.de
[2] Kurt Gödel Research Center, Universität Wien, Austria, moritz.mueller@univie.ac.at

Abstract. We introduce a (new) notion of parameterized proof system. For parameterized versions of standard proof systems such as Extended Frege and Substitution Frege we compare their complexity with respect to parameterized simulations.

1. Introduction

Consider the following problems for graphs: the vertex cover problem VC, the clique problem CLIQUE, and the dominating set problem DS; they ask, given a graph G and a natural number k, whether G contains a cardinality k vertex cover, clique, and dominating set, respectively. All three problems are NP-complete and hence, from the point of view of polynomial reductions any two of them have the same computational complexity.

Taking in each case the natural number k as the parameter of an instance we get the parameterized problems p-VC, p-CLIQUE, and p-DS. In parameterized complexity there is not only a new notion of tractability, namely fixed-parameter tractability, but also the notion of reducibility has been adapted so that it preserves fixed-parameter tractability; the new notion being that of fpt-reduction. One knows that p-VC $\leq_{\text{fpt}}$ p-CLIQUE (that is, p-VC is fpt-reducible to p-CLIQUE) and p-CLIQUE $\leq_{\text{fpt}}$ p-DS. However, accepting the hypotheses FPT $\neq$ W[1] and W[1] $\neq$ W[2] (which are fundamental hypotheses of parameterized complexity and each of them implies P $\neq$ NP) neither p-CLIQUE $\leq_{\text{fpt}}$ p-VC nor p-DS $\leq_{\text{fpt}}$ p-CLIQUE. As Downey and Fellows write in [7]:

> *Parameterized reductions tend to be much more* structure preserving *than classical reductions, and certainly most classical reductions ... are definitely not parameterized reductions. ... Parameterized reductions are sufficientlly refined that instead of one large class of naturally intractable problems all of the same complexity, there seem to be many sets of natural combinatorial problems, all intractable in the parameterized sense, and yet of differing parameterized complexity*

In proof theory among the proof systems best studied there are Frege systems, Extended Frege systems, and Substitution Frege systems. Classically, they are compared via polynomial simulations. It is known that there are polynomial simulations between any Extended Frege system and any Substitution Frege system, while it is not known

ISSN 1433-8092

whether Extended Frege systems and Substitution Frege systems may be simulated by Frege systems. The question arises whether also in this context parameterized complexity yields new insights or even allows a more fine-grained analysis. In this note we want to lay down the conceptual framework for such an analysis. Furthermore, we give some positive and some negative answers and state some open problems.

What are natural parameterizations of proof systems? Recall that the definitions of parameterized complexity are tailored to address complexity issues in situations where we know that the parameter is relatively small. We believe that for Extended Frege systems the number of extension axioms used in a proof could be a natural parameter. At least, if we start with an arbitrary, say, random tautology it doesn't seem plausible that many extension axioms can be used in a proof with advantage. We should emphasize the word "random" here. For example, in a standard example often mentioned to motivate the use of extension axioms, namely the formalization of the pigeon-principle in propositional logic, the number of extension axioms used to derive the n pigeonhole principle by a straightforward induction on n is $\Omega(n^3)$ and hence, by no means, relatively small. [3] Similarly the number of applications of the substitution rule seems to be a natural parameter for Substitution Frege Systems.

As proof systems are functions, simulations between them should be value-preserving functions (as are the standard polynomial simulations). We believe that this fact has not been taken into account appropriately in the approaches to proof theory using parameterized complexity. Taking this fact seriously, we define the notion of fpt-simulation. When we realized that our notion coincides with the notion of parsimonious reduction between parameterized counting functions, we were confirmed in our believe that this is the appropriate definition.

We show that under fpt-simulations the parameterized versions of Extended Frege and Substitution Frege are both equivalent to Frege. In this sense, the notion of fpt-simulation does not offer a more fine-grained complexity analysis of these proof systems; or, expressing it in positive terms, we gain the insight that there is a simulation, say, of an Extended Frege system in a Frege system whose superpolynomial running time is confined to a factor depending only on the number of extension axioms used in the original proof. Similarly, we see that there is a simulation of Substitution Frege in Extended Frege where the number of extension axioms is bounded in terms of the number of applications of the substitution rule.

Having in mind the goal of a more refined analysis, we propose to study the relationship between these proofs systems under parameterized polynomial simulations, a notion that in some sense refines both, polynomial simulations and fpt-simulations: such a simulation is a polynomial simulation with the additional property that it increases the parameter at most polynomially. We do not see any way to simulate Substitution Frege in Extended Frege in this sense (while conversely this is easy). However, we construct a parameterized polynomial simulation of treelike Substitution Frege in treelike Extended Frege.

Related work A different approach to introduce parameterizations into proof complexity has been initiated by Dantchev et al. [6]. They introduced parameterized proof sys-

[3] It is well-known that Buss [3] gave polynomial proofs of the pigeon-principle in Frege systems.

tems for *parameterized* problems. They considered the following parameterized problem: given a pair (α, k) of a CNF α and $k \in \mathbb{N}$, where k is the parameter, decide whether α has no satisfying assignment of Hamming weight at most k. The proof systems they had in mind are classical refutation systems such as Resolution that may freely use additional clauses expressing the constraint on the Hamming weight. The goal of this approach is to strengthen lower bounds of classical refutation systems by showing that their parameterized counterparts are not *fpt bounded*[4]. It can be understood as a parameterized analogue of Cook's program, here trying to prove coW[2] $\not\subseteq$ paraNP. For this approach Beyersdorff et al. [1] lack an interpretation of the parameterization of the proof system and argue that it can be dispensed with.

2. Preliminaries

In this section we fix some notations and recall some definitions and results, in the first part of parameterized complexity theory and in the second part of proof theory.

2.1. Parameterized Complexity. Formally, a *parameterized problem* is a pair (Q, κ) consisting of a (classical) problem $Q \subseteq \{0,1\}^*$ and a polynomial time computable *parameterization* $\kappa : \{0,1\}^* \to \mathbb{N}$ that maps any input $x \in \{0,1\}^*$ to its *parameter* $\kappa(x) \in \mathbb{N}$. A parameterized problem (Q, κ) is *fixed-parameter tractable*, that is, tractable from the point of view of parameterized complexity, if there is an algorithm solving $x \in Q$ in $\leq f(\kappa(x)) \cdot |x|^{O(1)}$ steps for some computable $f : \mathbb{N} \to \mathbb{N}$.

A function $R : \{0,1\}^* \to \{0,1\}^*$ is *fpt-computable* with respect to a parameterization κ if $R(x)$ can be computed in time $f(\kappa(x)) \cdot |x|^{O(1)}$, where again $f : \mathbb{N} \to \mathbb{N}$ is computable.

Also the notion of polynomial reduction, that is, the natural notion of reduction preserving classical tractability, has to be adapted so that it preserves fixed-parameter tractability. An *fpt-reduction* R from a parameterized problem (Q, κ) to another (Q', κ') is an fpt-computable (with respect to κ) reduction from Q to Q' such that $\kappa'(R(x)) \leq g(\kappa(x))$ for some computable $g : \mathbb{N} \to \mathbb{N}$ and all $x \in \{0,1\}^*$. We write $(Q, \kappa) \leq_{\text{fpt}} (Q', \kappa')$ if there is an fpt-reduction from (Q, κ) to (Q', κ').

2.2. Proof Theory. A *proof system* for a problem $Q \subseteq \{0,1\}^*$ is a polynomial time computable surjection P from $\{0,1\}^*$ onto Q. If $P(w) = x$, then w is a P-*proof* of x. In case $Q = \text{TAUT}$, we call P *propositional*. A proof system P is *p-bounded* if any $x \in Q$ has a P-proof of size $|x|^{O(1)}$. Cook and Reckhow [5] observed that a p-bounded propositional proof system exists if and only if NP = coNP. Cook's program asks to prove that natural propositional proof systems are not p-bounded.

Proof systems for a problem Q are compared in strength via p-simulations: a *p-simulation* of a proof system P' in a proof system P is a polynomial time computable function R such that $P(R(w')) = P'(w')$ for all $w' \in \{0,1\}^*$; in case such an R exists, we say P *p-simulates* P' and write $P' \leq_{\text{pol}} P$; if additionally, P' *p*-simulates P, we call P and P' *p-equivalent*.

A *Frege system* F is a propositional proof system given by finitely many axiom schemes (in the de Morgan language) and finitely many rules including, for simplicity, modus ponens. An F-*proof* of a (propositional) formula α from a set of formulas Γ is

[4] As pointed out in [1] one should restrict attention to instances (α, k) with contradictory α

a sequence of formulas such that each of them is either a member of Γ or a substitution instance of an axiom scheme or follows from earlier formulas in the sequence by one of the rules of F; furthermore, the last formula of the sequence is α. An F-proof of α is an F-proof of α from the empty set of formulas. Frege systems are assumed to be *implicationally complete*, that is, whenever a set of formulas Γ logically implies α, then there exists an F-proof of α from Γ.

For a Frege system F we denote by F^* the proof system *treelike* F: an F-proof π is *treelike* if every occurrence of a formula in π is used as an hypothesis in an application of a rule at most once; equivalently, π is treelike if it can be written as a tree labeled by the formulas in π such that the leaves are labeled by the substitution instances of the axiom schemes and the labels of inner nodes are obtained by one of the rules from their immediate predecessors.

The following are well-known [10, 5].

Theorem 1. *(1) (Cook, Reckhoff) Any two Frege systems are p-equivalent.*
(2) (Krajíček) F and F^* are p-equivalent for every Frege system F.

By part (1) of this theorem we get that, instead of (2), we could claim

F_1 and F_2^* are p-equivalent for Frege systems F_1 and F_2.

The same observation applies to all equivalences mentioned in this paper (not only to p-equivalences but also to fpt-equivalences and pp-equivalences introduced later).

There are two well-studied extensions of a Frege system F:

Extension Frege. Let F be a Frege system. The *Extension Frege system EF* adds to F the *extension rule*: It allows to add in a proof of α (without any hypotheses) an *extension axiom* $(r \leftrightarrow \sigma)$ where σ is a propositional formula and the *extension variable* r neither occurs in σ nor in α nor in any earlier line of the proof.

Equivalently, an *EF*-proof of α is an F-proof of α from an extension sequence whose extension variables do not occur in α. Here, an *extension sequence* (for α) of length k is a sequence of the form

$$(r_1 \leftrightarrow \sigma_1), \dots, (r_k \leftrightarrow \sigma_k)$$

with pairwise distinct *extension variables* $r_1, \dots, r_k$ such that r_i does not occur in σ_j for $1 \leq j \leq i$.

By *EF** we denote the treelike version of *EF*.

Substitution Frege. Let F be a Frege system. The *Substitution Frege system SF* adds to F the *substitution rule* that allows to derive from the formula α the formula $\alpha[x/\sigma]$ where $\alpha[x/\sigma]$ is obtained from α by substituting the variable x by the formula σ. By *SF** we denote the treelike version of *SF*.

In [2] Buss introduces two restrictions of *SF*:

– *Boolean Substitution Frege BSF* requires that in any application of the substitution rule the formula σ to be the Boolean constant $\top$ (TRUE) or $\bot$ (FALSE);
– *Renaming Frege RF* requires σ to be a variable.

Again, *BSF** and *RF** denote the treelike versions of these systems.

Natural simulations of *EF* and *SF* in F roughly proceed as follows:

- Let π be an *EF*-proof. To delete the first extension axiom $(r \leftrightarrow \sigma)$ substitute everywhere in π the formula σ for r; this transforms the extension axiom into the tautology $(\sigma \leftrightarrow \sigma)$ for which we add a linear size F-proof. Proceed like this with the second extension axiom and so on. If π contains k extension axioms, the resulting F-proof has size $|\pi|^{O(k)}$.
- Let π be an *SF*-proof. Let the first application in π of the substitution rule yield $\alpha[x/\sigma]$ from α. Replace it by a proof of $\alpha[x/\sigma]$ obtained by applying the substitution x/σ to the initial segment of π up to α. If π contains k substitution inferences, the resulting F-proof has size $|\pi|^{O(k)}$.

Hence, both simulations are not polynomial ones. In fact, it is open whether *EF* $\leq_{\text{pol}} F$ and whether *SF* $\leq_{\text{pol}} F$. However, the following is known [12, 2]:

Theorem 2. *(1) EF, EF*, SF, SF*, RF, BSF are p-equivalent for every Frege system F.*
(2) RF*, BSF* and F are p-equivalent for every Frege system F.

Comparing their status with that of *RF** and of *BSF** we see that perhaps *RF* and *BSF* are proof systems where the ability to reuse already derived lines adds power. We shall see a similar phenomenon for *SF* in the parameterized setting.

3. Parameterized proof systems and fpt-simulations

In this section we introduce the main new concepts of this paper, parameterized proof systems and simulations between them.

Definition 3. A *parameterized proof system for Q* is a pair (P, κ) such that P is a proof system for Q and κ a parameterization.

Having in mind, as we do, to compare Frege systems, Extended Frege systems, and Substitution Frege systems, it seems not natural to consider a more general notion of parameterized proof systems where P is only required to be an fpt-computable (with respect to κ) function from $\{0,1\}^*$ onto Q instead of a polynomial time computable one.

We identify a (classical) proof system P for Q with the parameterized proof system $(P, 0)$, i.e., P with the parameterization that is constantly 0.

For an Extended Frege systems *EF* we denote by κ_{EF} the parameterization

$$\kappa_{EF}(w) := \text{number of extension axioms in } w.$$

Similarly, for a Substitution Frege systems *SF* we denote by κ_{SF} the parameterization

$$\kappa_{SF}(w) := \text{number of applications of the substitution rule in } w.$$

We consider the restriction *EF** of *EF* with the parameterization κ_{EF} and the restrictions *SF**, *BSF*$^{(*)}$, and *RF*$^{(*)}$ of *SF* with the parameterization κ_{SF}. We denote the resulting parameterized proof systems by *p-EF*, *p-EF**, *p-SF*, *p-RF*, *p-BSF*, *p-SF**, *p-RF** and *p-BSF**.

In order to compare parameterized proof systems in strength we use the following notion of simulation. We already mentioned that for parameterized counting problems the notion coincides with that of fpt parsimonious reduction introduced in [8, Definition 14.10].

Definition 4. Let (P, κ) and (P', κ') be parameterized proof systems for $Q \subseteq \{0, 1\}^*$. An *fpt-simulation* of (P', κ') in (P, κ) is a function $R : \{0, 1\}^* \to \{0, 1\}^*$ such that
(a) R is *fpt*-computable with repect to κ';
(b) $P'(w') = P(R(w'))$ for all $w' \in \{0, 1\}^*$;
(c) $\kappa(R(w')) \leq g(\kappa'(w'))$ for some computable $g : \mathbb{N} \to \mathbb{N}$ and all $w' \in \{0, 1\}^*$.
In case such an R exists, we say that (P, κ) *fpt-simulates* (P', κ') and write $(P', \kappa') \leq_{\text{fpt}} (P, \kappa)$. The problems (P, κ) and (P', κ') are *fpt-equivalent*, written $(P, \kappa) \equiv_{\text{fpt}} (P, \kappa)$, if $(P, \kappa) \leq_{\text{fpt}} (P', \kappa')$ and $(P', \kappa') \leq_{\text{fpt}} (P, \kappa)$.

Note that if P and P' are classical proof systems for a problem Q, then P fpt-simulates P' if and only if P p-simulates P'. However, in general, neither $(P, \kappa) \leq_{\text{fpt}} (P', \kappa')$ implies $P \leq_{\text{pol}} P'$ nor $P \leq_{\text{pol}} P'$ implies $(P, \kappa) \leq_{\text{fpt}} (P', \kappa')$.

Lemma 5. *If* $(P, \kappa) \leq_{\text{fpt}} (P', \kappa')$ *and* $(P', \kappa') \leq_{\text{fpt}} (P'', \kappa'')$*, then* $(P, \kappa) \leq_{\text{fpt}} (P'', \kappa'')$.

4. Comparing proof systems via fpt-simulations

By the following result all parameterized proof systems introduced so far are fpt-equivalent.

Theorem 6. *p-EF*, *p-SF, and* F *are pairwise fpt-equivalent.* [5]

As $F \leq_{\text{fpt}}$ *p-EF*, the theorem follows from the following three propositions showing (among others):

$$p\text{-}EF \leq_{\text{fpt}} p\text{-}SF \leq_{\text{fpt}} p\text{-}BSF \leq_{\text{fpt}} F.$$

In Proposition 7 and Proposition 8 we obtain the first two 'inequalities' by merely observing that known p-simulations already are fpt-simulations.

Proposition 7. *p-EF* $\leq_{\text{fpt}}$ *p-SF and p-EF** $\leq_{\text{fpt}}$ *p-SF**.

Proof. Cook and Reckhow's original p-simulation [5] of *EF* in *SF* is an fpt-simulation of *p-EF* in *p-SF*; this yields the first assertion.

We turn to the second claim. An *EF**-proof π of α is an F^*-proof of α from an extension sequence $(r_1 \leftrightarrow \sigma_1), \ldots, (r_k \leftrightarrow \sigma_k)$ (recall that the r_i have to be paiwise

[5] The second author gave a talk at the workshop *Proof complexity* (11w5103, Banff International Research Station) on this subject mentioning that at that time we didn't know whether *p-EF* $\leq_{\text{fpt}}$ F. Kaveh Ghasemloo pointed out that he was convinced that such a simulation could be constructed via the system G_1^* (cf. [4, p.179]).

distinct and that r_i neither occurs in σ_j for $1 \leq j \leq i$ nor in α). By the Deduction Theorem for F (see [11, Lemma 4.4.10]) there is an F-proof π' of

$$(r_k \leftrightarrow \sigma_k) \rightarrow (r_{k-1} \leftrightarrow \sigma_{k-1}) \rightarrow \cdots \rightarrow (r_1 \leftrightarrow \sigma_1) \rightarrow \alpha \tag{1}$$

(where the iterated implications are associated to the right) of size $|\pi|^{O(1)}$. By part (2) of Theorem 1 we can assume that π' is treelike.

By our assumption on the extension variables, the variable r_k occurs exactly once in (1). We apply the substitution rule and substitute σ_k for r_k in (1); hence we get the formula obtained from (1) by replacing the equivalence $(r_k \leftrightarrow \sigma_k)$ by $(\sigma_k \leftrightarrow \sigma_k)$. We add a short F^*-proof of $(\sigma_k \leftrightarrow \sigma_k)$ and apply modus ponens to arrive at formula (1) with $k-1$ instead of k. Repeating this process gives an *SF**-proof of α of size $O(k \cdot |\pi'|)$. We observe that in this simulation k extension axioms are simulated in *SF** by k applications of the substitution rule. Therefore, this is an fpt-simulation. ☐

Proposition 8. *p-SF $\leq_{\text{fpt}}$ p-BSF.*

Proof. Buss [2] simulates an application of the substitution rule $\frac{\alpha}{\alpha[x/\sigma]}$ as follows: first, he applies twice the *BSF*-substitution rule to get

$$\alpha[x/\top] \qquad \text{and} \qquad \alpha[x/\bot]$$

from α; then he adds short proofs of

$$((\sigma \wedge \alpha[x/\top]) \rightarrow \alpha[x/\sigma]) \qquad \text{and} \qquad ((\neg\sigma \wedge \alpha[x/\bot]) \rightarrow \alpha[x/\sigma]).$$

Finally, he derives $\alpha[x/\sigma]$ from these four formulas.

In this way, an *SF*-proof with k applications of the substitution rule is transformed in polynomial time into an *BSF*-proof with $2k$ applications of the *BSF*-substitution rule. Hence, this is an fpt-simulation. ☐

Proposition 9. *p-BSF $\leq_{\text{fpt}}$ F.*

Proof. Let π be an *BSF*-proof of β with k applications of the *BSF*-substitution rule. Let π_1 be the initial segment of π that ends in the premise α of the first application $\frac{\alpha}{\alpha[x/\sigma]}$with $\sigma \in \{\top, \bot\}$ of this rule. We obtain the F-proof π_1' of $\alpha[x/\sigma]$ by applying the substitution x/σ to every line of π_1. Furthermore, delete all occurrences of $\alpha[x/\sigma]$in π, thus getting π'. Then π_1', π' is a *BSF*-proof of β with $(k-1)$ applications of the *BSF*-substitution rule and of size at most $2|\pi|$. Repeating this process we finally obtain an F-proof of β of size $2^k \cdot |\pi|$. ☐

Standard p-simulations of *SF* in *EF* (e.g., see [12]) map an *SF*-proof π of a formula $\alpha(\bar{x})$ (where $\bar{x}$ are the propositional variables in α) with k applications of the substitution rule and ℓ lines to an *EF*-proof with $\ell \cdot |\bar{x}|$ extension axioms. They are not fpt-simulations. By the previous theorem there is an fpt-simulation of *p-SF* in *p-EF*. We encourage the reader to give a 'direct' one.

5. Comparing proof systems via parameterized polynomial simulations

In the previous section we have seen that fpt-simulations are too coarse in the sense that they do not distinguish any two of the parameterized proof system considered so far. In this section therefore we analyze these proof systems under a notion of simulation which strengthens both, the notion of p-simulation and that of fpt-simulation. For parameterized decision problems this concept was introduced in [9].

Definition 10. Let (P, κ) and (P', κ') be parameterized proof systems for $Q \subseteq \{0,1\}^*$. A *pp-simulation* (or, *parameterized polynomial simulation*) of (P', κ') in (P, κ) is a p-simulation R of P' in P such that

$$\kappa'(R(w')) \leq q(\kappa(w')) \text{ for some polynomial } q \text{ and all } w' \in \{0,1\}^*.$$

In case such an R exists, we say that (P, κ) *pp-simulates* (P', κ') and write $(P', \kappa') \leq_{\text{pp}} (P, \kappa)$. The problems (P, κ) and (P', κ') are *pp-equivalent*, written $(P, \kappa) \equiv_{\text{pp}} (P, \kappa)$, if $(P, \kappa) \leq_{\text{pp}} (P', \kappa')$ and $(P', \kappa') \leq_{\text{pp}} (P, \kappa)$.

Clearly, if $(P', \kappa') \leq_{\text{pp}} (P, \kappa)$, then $P' \leq_{\text{pol}} P$ and $(P', \kappa') \leq_{\text{fpt}} (P, \kappa)$.

As the proofs of Proposition 7 and of Proposition 8 show, we get:

Proposition 11. *p-EF* $\leq_{\text{pp}}$ *p-SF, p-EF** $\leq_{\text{pp}}$ *p-SF**, *and p-SF* $\leq_{\text{pp}}$ *p-BSF.*

Example 12. The p-simulation of *BSF* in *RF* from [2] maps a *BSF*-proof with k substitution inferences of a formula with m variables to an *RF*-proof with $k \cdot (m-1)$ substitution inferences. This is not a pp-simulation (not even an fpt-simulation).

By the results of the previous section there is an fpt-simulation of p-*SF* in p-*EF* even though (as mentioned at the end of that section) standard p-simulations of *SF* in *EF* are not fpt-simulations. We do not know whether p-*SF* $\leq_{\text{pp}}$ p-*EF*. However, this holds for the tree-like versions of these proof systems:

Theorem 13. p-*SF** $\leq_{\text{pp}}$ p-*EF**.

Proof. We say that an *SF**-proof of β from an extension sequence (for β) is an *ESF**-proof of β if every application of the substitution rule has the form

$$\frac{\alpha}{\alpha[x/\sigma]}$$

where the formula $x \wedge \sigma$ does not contain any extension variable.

Clearly, an *EF**-proof of β is an *ESF**-proof of β without applications of the substitution rules.

We now describe how to stepwise eliminate applications of the substitution rule in *ESF**-proofs. So, let π be an *ESF**-proof of β with k applications of the substitution rule. We depict π as a labeled tree T with β at the root; for any node t of T labeled by γ the subtree T_t rooted at this node (and consisting of the predecessors of this node) constitutes an *ESF**-proof of γ. Consider a node t such that

– t is labeled by a formula $\alpha[x/\sigma]$ obtained from its predecessor t^- labeled by α by an application of the substitution rule (via the substitution x/σ);
– no further applications of the substitution rule occur in T_t.

Let r be a variable not occuring in π and obtain $T_{t^-}(x/r)$ by substituting x by r in all formulas of T_{t^-}. By the proviso on the applications of the substitution rule in an *ESF**-proof, the variable x is not a substitution variable and hence extension axioms of T are transformed into extension axioms in $T_{t^-}(x/r)$. Hence, $T_{t^-}(x/r)$ is an F^*-proof of $\alpha[x/r]$ from a set of extension axioms.

Let π' be a short F^*-proof of

$$(\alpha[x/r] \to ((r \leftrightarrow \sigma) \to \underbrace{\alpha[x/r][r/\sigma]}_{=\alpha[x/\sigma]}))$$

Using the new extension axiom $(r \leftrightarrow \sigma)$ (and adding some applications of modus ponens) we merge this F^*-proof with $T_{t^-}(x/r)$ to get a F^*-proof of $\alpha[x/\sigma]$ from an extension sequence.

$$\dfrac{\dfrac{\begin{matrix}\vdots\ T_{t^-}(x/r)\\ \alpha[x/r]\end{matrix} \qquad \begin{matrix}\vdots\ \pi'\\ (\alpha[x/r] \to ((r \leftrightarrow \sigma) \to \alpha[x/\sigma])\end{matrix}}{((r \leftrightarrow \sigma) \to \alpha[x/\sigma])} \qquad (r \leftrightarrow \sigma)}{\alpha[x/\sigma]}$$

Replace in the original proof π the subtree $T_t(x/r)$ by this new proof, thus obtaining a proof π''. It should be clear that π'' is an *ESF**-proof of β with $k-1$ applications of the substitution rule.

Iterating this process k times we finally get an F^*-proof π^* of β from an extension sequence (for β) consisting of k extension axioms. As π^* is obtained from π in polynomial time the mapping $\pi \mapsto \pi^*$ is the desired pp-simulation of *p-SF** in *p-EF**. □

Note that in the previous proof we have used that the *SF*-proof we start with is treelike: the simulation replaces all predecessors of a formula obtained by a substitution rule. In an arbitrary *SF*-proof some later inferences may be based on some formulas not further available.

We prove the following result by standard means:

Proposition 14. *p-EF* $\leq_{\text{pp}}$ *p-EF**.

Proof. Let $\pi = \alpha_1, \ldots, \alpha_s$ be an *EF*-proof with k extension axioms. For $1 \leq i \leq s$ we set $\gamma_i := \bigwedge_{j=1}^{i} \alpha_j$. We construct for $i = 1, \ldots, s$ successively *EF**-proofs π_i of γ_i such that the variables in π_i are precisely those in $\alpha_1, \ldots, \alpha_i$ and the extension axioms in π_i are the same as in $\alpha_1, \ldots, \alpha_i$.

The tree π_1 just consists of the root labeled by α_1. Assume that we have already constructed the *EF**-proof π_i of γ_i. To construct π_{i+1} we first consider the case where α_{i+1} is an extension axiom or a substitution instance of an axiom of F. Let π^1 be a short F^*-proof of $(u \to (v \to (u \wedge v)))$. Then $\pi^1[u/\gamma_i, v/\alpha_{i+1}]$ is an F^*-proof of $(\gamma_i \to (\alpha_{i+1} \to \gamma_{i+1}))$ of size $O(|\gamma_{i+1}|)$. As an intermediate step we get an F^*-proof π^2 of $(\alpha_{i+1} \to \gamma_{i+1})$ from the F^*-proofs π_i and $\pi^1[u/\gamma_i, v/\alpha_{i+1}]$ by an application of modus ponens. A further modus ponens inference yields from π^2 and the 'leaf' α_{i+1} the desired F^*-proof π_{i+1} of γ_{i+1}.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ \pi^1[u/\gamma_i, v/\alpha_{i+1}] \\ (\gamma_i \to (\alpha_{i+1} \to \gamma_{i+1}))\end{array} \qquad \begin{array}{c}\vdots\ \pi_i \\ \gamma_i\end{array}}{(\alpha_{i+1} \to \gamma_{i+1})} \qquad \alpha_{i+1}}{\gamma_{i+1}}$$

Now assume that α_{i+1} is obtained by one of the rules of F. The general case being analogous, we treat the case where this rule is modus ponens. So assume α_{i+1} is obtained from α_k and α_ℓ (where $1 \le k, \ell \le i$) by modus ponens. Let π^1 be an F^*-proof of $(\bigwedge_{j=1}^{i} u_j \to (u_k \wedge u_\ell))$ of size polynomial in i. Substituting in π^1 the u_js by the α_js yields an F^*-proof π^2 of $(\gamma_i \to (\alpha_k \wedge \alpha_\ell))$ of size polynomial in $|\gamma_i|$.

To a short F^*-proof of $((u \to v) \to ((v \to w) \to (u \to (u \wedge w))))$ we apply the substitution $[u/\gamma_i, v/(\alpha_k \wedge \alpha_\ell), w/\alpha_{i+1}]$ obtaining an F^*-proof π^3 of size $O(|\gamma_{i+1}|)$ of

$$((\gamma_i \to (\alpha_k \wedge \alpha_\ell)) \to ((\alpha_k \wedge \alpha_\ell) \to \alpha_{i+1}) \to (\gamma_i \to \gamma_{i+1}))).$$

Finally, let π^4 be an F^*-proof of $((\alpha_k \wedge \alpha_\ell) \to \alpha_{i+1})$ of size $O(|\alpha_k| + |\alpha_\ell| + |\alpha_{i+1}|)$ (recall that α_{i+1} was obtained from α_k and α_ℓ by modus ponens). Now it is easy to merge π_i, π^1, π^2, π^3, and π^4 to an F^*-proof π_{i+1} of γ_{i+1}.

It is easy to construct a treelike proof π^* of α_s from π_s. It is clear that π^* can be computed from π in polynomial time. □

Theorem 15. *$F \equiv_{\mathrm{pp}}$ p-BSF* $\equiv_{\mathrm{pp}}$ p-RF* $\le_{\mathrm{pp}}$ p-EF $\equiv_{\mathrm{pp}}$ p-EF* $\equiv_{\mathrm{pp}}$ SF* $\le_{\mathrm{pp}}$ p-SF $\equiv_{\mathrm{pp}}$ p-BSF.*

Proof. The first two equivalences are easy to see. The third equivalence follows from the preceding proposition. The equivalence *p-EF** $\equiv_{\mathrm{pp}}$ *p-SF** follows from Proposition 11 and Theorem 13. The last equivalence follows from Proposition 11, too. □

Hence, the proof systems mentioned in the previous theorem belong to at most three distinct pp-degrees. Are these degrees distinct? Note that this theorem doesn't mention *p-RF*. Does it belong to any of these degrees? Of course, F $\le_{\mathrm{pp}}$ *p-RF* $\le_{\mathrm{pp}}$ *p-SF*. Furthermore, we can show the following:

Proposition 16. *If p-RF $\le_{\mathrm{pp}}$ p-EF, then p-SF $\le_{\mathrm{pp}}$ p-EF.*

Proof. Assume *p-RF* $\le_{\mathrm{pp}}$ *p-EF*. By Proposition 11 it suffices to show *p-BSF* $\le_{\mathrm{pp}}$ *p-EF*. So let $\pi = \alpha_1, \ldots, \alpha_s$ be a *BSF*-proof with k substitution inferences (substituting a variable by $\bot$ or by $\top$). Let $y_1, \ldots, y_k$ and $z_1, \ldots, z_k$ be new variables (not occurring in π) and let

$$\delta := \bigwedge_{i=1}^{k} \neg y_i \wedge \bigwedge_{i=1}^{k} z_i.$$

Consider the sequence

$$(\delta \rightarrow \alpha_1), \ldots, (\delta \rightarrow \alpha_s).$$

This sequence can be "filled up" to an *RF*-proof with k substitution inferences (substituting a variable by another variable): if α_i in π is a substitution instance of an axiom, replace $(\delta \rightarrow \alpha_i)$ by a short F-proof of $(\delta \rightarrow \alpha_i)$. If α_i is obtained by modus ponens from $\alpha_j, \alpha_{j'}$ with $j, j' < i$, then replace $(\delta \rightarrow \alpha_i)$ by a short F-proof of $(\delta \rightarrow \alpha_i)$ from $(\delta \rightarrow \alpha_j)$ and $(\delta \rightarrow \alpha_{j'})$. Finally, if α_i is obtained by a substitution inference, then there is $j < i$ such that $\alpha_i = \alpha_j[x/\bot]$ or $\alpha_i = \alpha_j[x/\top]$ for some variable x. Assume this is the ℓth substitution inference ($1 \leq \ell \leq k$) in π and that $\alpha_i = \alpha_j[x/\bot]$ (the other case $\alpha_i = \alpha_j[x/\top]$ is similar). Replace $(\delta \rightarrow \alpha_i)$ by the following *RF*-proof: give a short F-proof of $(\delta \wedge \alpha_j[x/y_\ell] \rightarrow \alpha_i)$ (note that $\neg y_\ell$ is a conjunct of δ) and derive $\alpha_j[x/y_\ell]$ from α_j by an *RF* substitution inference; from these two formulas it is easy to derive $(\delta \rightarrow \alpha_i)$.

Clearly, this *RF*-proof can be computed from π in polynomial time. By assumption we can in polynomial time compute from this *RF*-proof an *EF*-proof π' of $(\delta \rightarrow \alpha_s)$ with $k^{O(1)}$ extension axioms. Since the y_i's and the z_i's occur in δ, they are not used as extension variables in π'. Let π'' result from π' by substituting $\bot$ for all occurrences of the y_i's and $\top$ for all occurrences of the z_i's. Then (note the y_i's and the z_i's do not occur in α_s) π'' is an *EF*-proof of $(\delta' \rightarrow \alpha_s)$ where δ' is a true Boolean sentence (a true formula without variables). Adding a short proof of δ' and an application of modus ponens gives an *EF*-proof of α_s. □

Acknowledgements. The authors thank the John Templeton Foundation for its support through Grant #13152, *The Myriad Aspects of Infinity*. The second author thanks the FWF (Austrian Research Fund) for its support through Grant P 23989 - N13.

References

1. O. Beyersdorff, N. Galesi, M. Lauria and A. Razborov. Parameterized Bounded-Depth Frege is Not Optimal. Proceedings 38th International Colloquium on Automata, Languages and Programming (ICALP), pp. 630–641, Springer-Verlag, 2011.
2. S. Buss. Some remarks on the lengths of propositional proofs. Archive for Mathematical Logic 34:377–394, 1995.
3. S. Buss. Polynomial size proofs of the propositional pigeon principle. Journal of Symbolic Logic 52:916–927, 1987.
4. S. Cook and P. Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.
5. S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic* 44:36–50, 1979.
6. S. S. Dantchev, B. Martin, and S. Szeider. Parameterized proof complexity. *Computational Complexity* 20 (1):51–85, 2011.
7. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
8. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.
9. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *Journal of Computer and System Sciences* 77(1):91-106, 2011.
10. J. Krajíček. On the number of steps in proofs. *Annals of Pure and Applied Logic* 41:153–178, 1989.

11. J. Krajíček. *Bounded arithmetic, propositional logic, and complexity theory.* Cambridge University Press, 1995.
12. J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic* 54:1063–1088, 1989.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de