Technische Universität München
Department of Computer Science

Joint Advanced Students School 2009
Propositional Proof Complexity

May 2009

Frege Systems

Michael Herrmann

Contents

1. Introduction

Proof Theory plays a central role in mathematics. It studies the concepts of mathematical proofs and more generally provability. Proofs are very important to convince other people of the correctness of a theorem. They are usually done by natural language with the help of some symbols and figures. They are by no means explicit and different people would create very different proofs for the same problem. This proofs are also far away from being understandable by a computer. In contrast to this there are formal proofs. A formal proof is a string, which satisfies some precisely stated set of rules. Everybody who knows this rules is able to verify the proof or is able to create a proof.

Of course our motivation in studying formal proof systems is to develop a formal proof system. In propositional proof complexity we ask for a proof system, that is able to proof all formulas which are always true, so called tautologies. But there is also a greater impact on proof systems to complexity theory, which we show in Section 1.1.

In Section 1.1 we cover some basic principles of proof systems and in Section 2 we introduce the so called Frege and extended Frege proof systems. To illustrate the difference between normal and extended Frege systems we introduce in Section 3 for both systems a proof for the pigeonhole principle.

1.1 Basics

In this section we will introduce the basics of proof systems and show their impact on the complexity theory. Before we start we want to give a short repetition of the complexity classes $\mathcal{NP}$ and co$\mathcal{NP}$.

First of all in complexity theory there is the class $\mathcal{P}$, which is the set of all decision problems, that can be solved by a deterministic Turing machine in polynomial time. Decision problems for that the *yes answer* has simple proofs if the answer is indeed yes are in the class $\mathcal{NP}$. Finally we introduce the class co$\mathcal{NP}$. A element $\mathcal{X}$ is in co$\mathcal{NP}$ if and only if the negation of $\mathcal{X}$, $\bar{\mathcal{X}}$ is in $\mathcal{NP}$. A important question is, whether $\mathcal{NP}$ is closed under complementation, i.e. $\Sigma^* - L$ is in $\mathcal{NP}$ whenever L is in $\mathcal{NP}$. For further and more formal informations about complexity theory and complexity classes in particular, the reader is referred to [1].

To be able to consider the basics, we need a definition of a set of effective computable functions.

Definition 1.1. $\mathcal{F}$ is a set of functions $f : \Sigma_1^* \rightarrow \Sigma_2^*$, with Σ_1, Σ_2 are any finite alphabets, such that f can be computed by a deterministic Turing machine, time bounded by a polynomial in the length of the input string.

A proof system is a function f so that the proof is given as parameter and the to be proven proposition is the output of the function. We need f

to be onto, to assure that every proposition is covered by the function.

Definition 1.2. Let $L \subseteq \Sigma^*$, a *proof system* for L is a function $f : \Sigma_1^* \rightarrow L$ for some alphabet Σ_1 and $f \in \mathcal{F}$ such that f is onto.

We are especially in such proof systems interested, which are able to deliver short proofs. With short we mean, that the proof length is polynomially bounded in the length of the proposition.

Definition 1.3. A proof system is *polynomially bounded* if and only if there is a polynomial $p(n)$ such that for all $y \in L$ there is $x \in \Sigma_1^*$ such that $y = f(x)$ and $|x| \leq p(|y|)$.

Our first result is that $\mathcal{NP}$ is closed under complementation if and only if all the decision problem whether a formula is a tautology is in $\mathcal{NP}$.

Proposition 1.4. *$\mathcal{NP}$ is closed under complementation if and only if $TAUT$ is in $\mathcal{NP}$.*

For the proof we need the result of Theorem 2 of [4], that there is a function $f \in \mathcal{F}$ with that every set L is reducible to the complement of the tautologies.

Proof. Assume $\mathcal{NP}$ is closed under complementation. To verify that a formula is not a tautology one can guess a truth assignment and verify that it falsifies the formula. Because we assumed, that $\mathcal{NP}$ is closed under complementation, the set of tautologies is also in $\mathcal{NP}$.

Assume that the set of tautologies is in $\mathcal{NP}$. So a non deterministic procedure for accepting the complement of L would be : On input x, compute $f(x)$ (the result of [4]) and accept if x is a tautology. Hence the complement of L is in $\mathcal{NP}$. □

With the help of polynomially bounded proof systems we can decide whether a set L is in $\mathcal{NP}$.

Proposition 1.5. *A set L is in $\mathcal{NP}$ if and only if $L = \emptyset$ or L has a polynomially bounded proof system.*

Proof. Assume $L \in \mathcal{NP}$. That means. there is a non deterministic Turing Machine M, that accepts L in polynomial time. If $L \neq \emptyset$, the proof system calculates for the case that M accepts y, $f(x) = y$. Where x is the calculation on an output tape of M on input y. Otherwise it sets $f(x) = y_0$, for a fixed y_0.

Let f be a polynomially bounded proof system for L. On input y guess an proof for x and accept if $f(x) = y$. □

Proposition 1.4 says that $\mathcal{NP}$ is closed under complementation if and only if TAUT $\in \mathcal{NP}$ and Proposition 1.5 says that any Language $L \in \mathcal{NP}$ if and only if L has an polynomially bounded proof system. If we are putting this two things together, we are getting directly the following result:

Proposition 1.6. *$\mathcal{NP}$ is closed under complementation if and only if TAUT has a polynomially bounded proof system.*

With that we are able to get a further result. But at first we need to know something about the polynomially dependence of proof systems. Namely if we have two proof systems f_1, f_2 and we are able to map every proof of f_1 to a proof of f_2 and if this mapping can be done in polynomial time, then we say that f_2 p-simulates f_1.

Definition 1.7. If $f_1 : \Sigma_1^* \to L$ and $f_2 : \Sigma_2^* \to L$ are proof systems for L, then f_2 p-simulates f_1 if there is a function $g_1 : \Sigma_1^* \to \Sigma_2^*$ such that g is in $\mathcal{F}$ and $f_2(g(x)) = f_1(x)$ for all x.

The next proposition gives us the result, that all proof systems, that are p-simulated by a polynomially bounded proof system, are also polynomially bounded.

Proposition 1.8. *If a proof system f_2 for L p-simulates a polynomially bounded proof system f_1, then is f_2 also polynomially bounded.*

Proof. Since p-simulation means that there is a $g \in \mathcal{F}$, and polynomially bounded means that $|x| \leq p(|y|)$ the equation $|g(x)| \leq p(|f(g(x))|)$ still holds. □

2. Frege Systems

Frege systems are a special type of proof systems. They are named after Friedrich Ludwig Gottlob Frege (8 November 1848 – 26 July 1925), who was a great mathematician and philosopher. He was the first who made a successful step into how to formalize proofs. In this section we will cover the normal *Frege system* (or just Frege systems) and a extended type called *extended Frege systems*. In Section 3 we will prove the propositional pigeonhole principle in both proof systems. First to give a example of the systems and second, more important, to show the difference of both systems and the advantage of extended Frege systems versus the normal ones.

2.1 Normal Frege Systems

First we define a Frege system. Formally we define a Frege system as follows:

Definition 2.1. A *Frege System* $\mathcal{F}$ is a three tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A})$. Where

- $\mathcal{L}$ is the propositional language.
- $\mathcal{R}$ is a finite set of rules.
- $\mathcal{A}$ is a finite set of axioms.

The propositional language $\mathcal{L}$ is the set of possible formulas, which are defined by a set of connectives κ. Connectives are stated in infix notation and connect parts of the whole formula. An example for such a set of connectives is $\{\neg, \wedge, \vee\}$, which is called the *standard basis*. The least possible formula is a *atom*, which is just a variable, a for example, which can be equal to the propositional values 0 or 1. A connective can be used to inductively connect formulas to new formulas. For example if we have the propositional formulas ϕ and ψ, $\phi \wedge \psi$ is also a propositional formula. We say a Frege System is *propositionally complete* if every formula ϕ over the *standard basis* has an equivalent formula ϕ' over $\mathcal{L}$.

Rules are used to combine some true formulas to other true formulas. General a rule is a system of formulas $(C_1, C_2, ..., C_n)/D$, where $(C_1, C_2, \ldots, C_n) \models D$. It means, that if we have the true formulas $(C_1, C_2, ..., C_n)$ then we can conclude D. If n is 0, which means that we have no assumptions, then the rule is called an axiom.

Example. We are introducing now an example Frege system. This Frege system is also used in [3].

Language with the connectives $\kappa = \{\neg, \wedge, \vee, \rightarrow\}$. Possible formulas could be: $((x \vee y) \wedge z)$ or $((\neg a \wedge b) \vee (c \rightarrow b))$

Rule of inference: $$\frac{P \qquad (P \rightarrow Q)}{Q}\ \text{MP}$$

Axioms:
$(P \wedge Q) \rightarrow P$
$(P \wedge Q) \rightarrow Q$
$P \rightarrow (P \vee Q)$
$Q \rightarrow (P \vee Q)$
$(P \rightarrow Q) \rightarrow ((P \rightarrow \neg Q) \rightarrow \neg P)$
$(\neg\neg P) \rightarrow P$
$P \rightarrow (Q \rightarrow P \wedge Q)$
$(P \rightarrow R) \rightarrow ((Q \rightarrow R) \rightarrow (P \vee Q \rightarrow R))$
$P \rightarrow (Q \rightarrow P)$
$(P \rightarrow Q) \rightarrow (P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow R)$

As in every other proof system type we have two important conditions, namely *soundness* and *completeness*. Soundness means that every theorem, proved in the system, is true and completeness means, that every valid formula has a proof. There are the related notation of *implicational soundness* and *implicational completeness*. For this we need the following notation. We write $\phi \vdash \psi$ whenever there is a derivation from ϕ to ψ and we write $\phi \models \psi$ when ψ follows from ϕ. A Frege system is implicationally sound whenever $\phi \vdash \psi$ then $\phi \models \psi$ is also true. Furthermore is a Frege system implicationally complete whenever $\phi \models \psi$ then $\phi \vdash \psi$ holds

We will now define what a Frege proof is. Intuitively it is a sequence of formulas, for that every formula is either an axiom or can be inferred by a rule from previous ones.

Definition 2.2. A Frege proof Π in a Frege system $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A})$ is a sequence $A = (A_1, ..., A_m)$ such that for all i, either:

- A_i is an instance of an axiom.
- It exists $j_1, ..., j_k$ with $k < i$ and with a k-ary rule $R \in \mathcal{R}$ such that $A_i = R(A_{j_1}, ..., A_{j_k})$.

Then Π is a proof of the theorem A_m. We may write then $\vdash A_m$. Or $\mathcal{F} \vdash A_m$, respectively $\vdash_{\mathcal{F}} A_m$, to state that A_m has a proof in the Frege system $\mathcal{F}$.

To regard the complexity of a Frege proof, in the scope of this work, we see the complexity of a proof as the symbol length of the proof, namely:

$$n = |\Pi| = \sum_{i=1}^{m} |A_i|$$

Before we get started with the first important theorem we need some more terminology:

- $A_1, ..., A_n \vdash^{\pi}_{\mathcal{F}} B$ means that there is the derivation π in the system $\mathcal{F}$ from $A_1, ..., A_n$ to B.
- $A_1, ..., A_n \vdash_{\mathcal{F}} B$ means that there is some derivation in the system $\mathcal{F}$ from $A_1, ..., A_n$ to B.
- $l(A)$ is the number of atoms (variables) in the formula or sequence A.
- $\lambda(\pi)$ is the number of lines in the derivation.
- $\rho(\pi) = \max_i l(A_i)$, if π is $A_1, ..., A_n$.
- $|\pi|$ or $|A|$ is the length as string.

Furthermore we need the result from the following lemma. It says, that any correct proof stays correct, if we apply a substitution on it.

Lemma 2.3. *Let $A_1, \dots, A_k$ be some formulas and π is the derivation of B from these formulas, then $\sigma(\pi)$ is a derivation of σB from $\sigma A_1, \dots, \sigma A_k$ for any substitution σ.*

The correctness of this lemma can be shown like in [5] by induction over the length of π. The sketch of the proof is to take a π is a single variable as induction start and show in the induction step, that two formulas connected by connective from the set of κ does not violate the lemma.

We will now introduce and prove a theorem, which says something about the linear dependency of two Frege systems with the same set of connectives.

Theorem 2.4. *For any two Frege systems $\mathcal{F}_1$ and $\mathcal{F}_2$ over κ there is a function $f \in \mathcal{F}$ and constant c such that for all formulas $A_1, ..., A_n, B$ and derivations π, if $(A_1, ..., A_n) \vdash^{\pi}_{\mathcal{F}_1} B$ then $(A_1, ..., A_n) \vdash^{f(\pi)}_{\mathcal{F}_2} B$, and $\lambda(f(\pi)) \leq c_1\lambda(\pi)$ and $\rho(f(\lambda)) \leq c_2\rho(\pi)$.*

The proof idea is to find a way how to change every derivation from $\mathcal{F}_1$ into a derivation of $\mathcal{F}_2$ and then to analyze this changes.

Proof. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two complete and implicationally complete Frege systems over κ. Then there is for every rule $R = (C_1, ..., C_m)/D$ in $\mathcal{F}_1$ a derivation π_r of D from $C_1, ..., C_m$ in $\mathcal{F}_2$.

Now let π be a derivation of B from $A_1, ..., A_n$ in $\mathcal{F}_1$ and suppose $\pi = (B_1, ..., B_k)$. To construct the $\mathcal{F}_2$-derivation $f(\pi)$ from π do the following: If B_i follows from earlier B_j's by the $\mathcal{F}_1$ rule R_i and substitution σ_i, simply replace B_i by the derivation $\sigma_i(\pi_{R_i})$. According to Lemma 2.3 $\sigma_i(\pi_{R_i})$ is the derivation of B_i from the same earlier B_j's.

The condition $\lambda(f(\pi)) \leq c_1\lambda(\pi)$ holds if c_1 is the number of lines in the longest derivation π_R over all rules R.

The condition $\rho(f(\pi)) \leq c_2\rho(\pi)$ holds too, with c_2 is an upper bound on $l(A)$, with A are all formulas in the derivations π_R. □

An immediate result of Proposition 1.8 and Theorem 2.4 is the following corollary, which shows the importance of having a polynomially bounded proof system for a certain κ.

Corollary 2.5. *Any two Frege systems over κ p-simulate each other. Hence one Frege system over κ is polynomially bounded if and only if all Frege systems over κ are.*

The rest to show is, that our Frege system introduced in Example 2.1 is sound and complete.

Theorem 2.6. *The in Example 2.1 introduced Frege system is both sound and implicationally sound.*

The proof idea is very straightforward. One notes, that all axioms are valid and show, that the modus ponens rule preserves the property of an formula being valid. This can be done by a truth table and noting, that whenever P and $P \rightarrow Q$ are true, Q is also true. Hence every proved theorem must be valid.

Theorem 2.7. *The in Example 2.1 introduced Frege system is both complete and implicationally complete.*

1. *If ϕ is a tautology, then $\vdash \phi$.*
2. *If $\psi \models \phi$, then $\psi \vdash \phi$*

The proof idea for this is to see, that part (2) of the Theorem can be reduced to part (1) and to prove part (1). Since the proof is very lengthy we skip here and refer the reader to [2].

2.2 Extended Frege Systems

We introduce now extended Frege systems. This kind of proof systems are an ordinary Frege system with one additional proof rule.

Definition 2.8. A extended Frege proof is a sequence of formulas $A_1, ..., A_n$ such that for all i:

- A_i is an instance of an axiom.
- It exists $j_1, ..., j_k$ with $k < i$ and with a k-ary rule $R \in \mathcal{R}$ such that $A_i = R(A_{j_1}, ..., A_{j_k})$.
- A_i is an *extension formula* of the form $P_i \equiv \phi$

Where ϕ is any formula and P_i is a fresh extension variable. We say that P_i is a defined atom and $P_i \equiv \phi$ is it's defining formula.

The idea of the extension rule is, that P_i can be used as an abbreviation for ϕ in all subsequent steps of the proof. This can reduce the proof complexity (number of used symbols) greatly.

We show now, that the in Example 2.1 introduced Frege system is also a sound extended Frege system.

Proposition 2.9. *If π is a derivation of B from $A_1, ..., A_n$ in a extended Frege system $e\mathcal{F}$, then $A_1, ..., A_m \models B$.*

Proof. Let τ be any truth assignment to the atoms of $A_1, ..., A_n$ and B, which satisfies $A_1, ..., A_n$ (normal Frege). Now we extend τ to make each line in the derivation true. In particular, if $P_i \equiv \phi$ is a defining formula, then P has not occurred earlier in the derivation.

That means that we are able to extend τ so $\tau(P_i) = \tau(\phi_i)$. Hence $\tau(B)$ is true since B is the last line of derivation. □

3. The propositional Pigeonhole Principle PHP

The propositional pigeonhole principle is a very well studied problem for proof systems. It states that, given two natural numbers n and m with $n > m$, if n pigeons are put into m pigeonholes, then at least one pigeonhole must contain more than one pigeon.

A not very surprisingly result is that in a family with three children, there must be at least two children with the same gender. Another, in the first moment unexpected result, is that in a city with population over 1 million, there must be at least two inhabitants with the same number of hairs. If we assume, that a typical head has 150000 hairs and no one has more then 1000000 hairs. That means we have $m = 1000000$ and $n > 1000000$. There are many more of such examples.

3.1 PHP Model for Frege Proof

Before we introduce the Frege proof for the PHP we have to develop a propositional model.

Let P_{ij} with $1 \leq i \leq n, 1 \leq j \leq n-1$ be a set of axioms. Whose meaning is that i is mapped to j.

Let $\mathcal{S}_n$ be the set:
$\{P_{i1} \vee ... \vee P_{i,n-1} | 1 \leq i \leq n\} \cup \{\neg P_{ik} \vee \neg P_{jk} | 1 \leq i < j \leq n, 1 \leq k \leq n-1\}$

To illustrate this model, we give an example for $n = 3$.

Example. $\{P_{i1} \vee P_{i,n-1} | 1 \leq i \leq n\} = \{(P_{11} \vee P_{12}), (P_{21} \vee P_{22}), (P_{31} \vee P_{32})\}$

The interpretation for this is, that for the pigeon i every hole is allowed in the first place.

$\{\neg P_{ik} \vee \neg P_{jk} | 1 \leq i < j \leq n, 1 \leq k \leq n-1\} = \{(\neg P_{11} \vee \neg P_{21}), (\neg P_{21} \vee \neg P_{31}), (\neg P_{12} \vee \neg P_{22}), (\neg P_{22} \vee \neg P_{32}), \}$

With that we say, that it is only allowed to put one pigeon in one hole.

We are able to represent this conditions also a function f. For this function we need two properties. The first one is, that f is defined over $\{0, 1, ..., n\}$ and maps to $\{0, 1, ..., n-1\}$, which addresses the need for putting n pigeons into $n-1$ holes. The second is, that f is injective. We need this, to assure, that every pigeon goes into a different hole.

We will proof the pigeon-hole principle by induction, where we remove in every step one pigeon and one hole, we need a inductive definition of f. Let us assume, that $f : \{0, 1, ..., n\} \rightarrow \{0, 1, ..., n-1\}$ is an injective function. Then is f' defined by:

$$f'(i) = \begin{cases} f(i) & f(i) \neq n-1 \\ f(n) & \text{else} \end{cases}$$

$f'(i)$ is then a new assignment for the next inductive step, by assigning every pigeon a hole for the next step. We can do this, by differing between two cases. In the first case, we assume, that for the current pigeon either the pigeon nor the hole will be removed. That means, that this assignment can be left unchanged. In the second case the hole of the actual pigeon will be removed. Because we will also remove one pigeon, we know, that there must also be a empty hole. And this hole becomes the new hole of the current pigeon.

Our goal is to show that $\neg\mathcal{S}_n$ is a tautology, which means that $\mathcal{S}_n$ is unsatisfiable.

3.2 Frege Proof

Now we are able to prove the pigeonhole principle with a Frege system.

To do the proof, we try to deduce $\mathcal{S}_{n-1}$ from $\mathcal{S}_n$. For each i, j we introduce a formula B_{ij}, which means $f'(i) = j$ and is defined by $B_{ij} = P_{ij} \vee (P_{i,n-1} \wedge P_{nj})$ with $1 \leq i \leq n-1, 1 \leq j \leq n-2$. Let σ_{n-1} be the substitution, which replaces P_{ij} by B_{ij}, formally $\sigma_{ij} = B_{ij}/P_{ij}$. Because f is injective implies f' is also injective we get $\mathcal{S}_n \models \sigma_{n-1}(\mathcal{S}_{n-1})$. Since our Frege system is complete we have $\mathcal{S}_n \vdash \sigma_{n-1}(\mathcal{S}_{n-1})$. The same holds for $n-1$: $\mathcal{S}_{n-1} \vdash \sigma_{n-2}(\mathcal{S}_{n-2})$. And so, by Lemma 2.3, we know, that there is a derivation, with the same number of lines showing, $\sigma_{n-1}(\mathcal{S}_{n-1}) \vdash \sigma_{n-1}\sigma_{n-2}(\mathcal{S}_{n-2})$, so $\mathcal{S}_n \vdash \sigma_{n-1}\sigma_{n-2}(\mathcal{S}_{n-2})$. Proceeding this way, we finally obtain a derivation showing $\mathcal{S}_n \vdash \sigma_{n-1}...\sigma_2(\mathcal{S}_2)$. But $\mathcal{S}_2 = \{P_{11}, P_{21}, \neg P_{11} \vee \neg P_{21}\}$. For which we can't find a truth assignment, which makes the formula true. So we can conclude, that $\vdash \neg\mathcal{S}_n$.

To analyze the proof Cook and Reckhow showed in [5], that one can choose the rules of a Frege system, so that the derivation of $\sigma_{n-1}(\mathcal{S}_{n-1})$ from $\mathcal{S}_n$ can be done in $\mathcal{O}(n^3)$. Because we have n derivations we come to an upper bound of $\mathcal{O}(n^4)$. The problem is, that every application of the substitution triples the length of a formula, so the longest formula in the proof grows exponentially in n.

3.3 Extended Frege Proof

We are using the possibility to use abbreviation formulas to reduce the proof length significantly. We define a atom $\mathcal{Q}^1_{ij} \equiv (P_{ij} \vee (P_{i,n-1} \wedge P_{n,j}))$ with $1 \leq i \leq n, 1 \leq j \leq n-2$. With that formula and with $\mathcal{S}_n$ the formula $\tau_{n-1}(\mathcal{S}_{n-1})$ can be derived, where τ_{n-1} is the substitution $\mathcal{Q}^1_{ij}/P_{ij}$.

In general we set: $\mathcal{Q}^{k+1}_{ij} \equiv (\mathcal{Q}^k_{ij} \vee (\mathcal{Q}^k_{i,n-k-1} \wedge \mathcal{Q}^k_{n-k,j}))$. With that the formulas $\tau_{n-k-1}(\mathcal{S}_{n-k-1})$ can be derived from $\tau_{n-k}(\mathcal{S}_{n-k})$ where τ_{n-k} is the substitution Q^k_{ij}/Pij. With that we get a contradiction in $\mathcal{O}(n^4)$, which is $\mathcal{O})(n)$ times more because of the new replacement step. So we come to an totally upper bound of $\mathcal{O}(n^5)$, but do not have the problem of an exponentially growth like in a normal Frege system.

References

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach (Draft)*. January 2007.

[2] Samuel R. Buss. An introduction to proof theory. *Elsevier Amsterdam*, (1):1–78, 1998.

[3] Samuel R. Buss. Propositional proof complexity: An introduction. (1), 1999.

[4] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM.

[5] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.