Filosofická fakulta
Universita Karlova v Praze

Weak Formal Systems

diplomová práce
Praha, 2003

studijní obor: logika-informatika
vypracoval: Jan Henzl
vedoucí práce: Doc. RNDr. Jan Krajíček, DrSc.

Prohlašuji, že jsem tuto práci vypracoval samostatně a použil výhradně citovaných pramenů.

Contents

1 Abstract

In this work we study the calculi for first-order logic as propositional proof systems. It is easy to see (and discussed in this thesis formally) that predicate calculi can serve as propositional proof systems. The question is whether they can allow for some shorter proofs.
The motivation for this comes from a problem whether there exists a polynomially bounded proof system. We will shortly review this problem (first stated by S.A.Cook and R.A.Reckhow [CR]) in the introduction to this thesis.
We interpret calculi for predicate logic and some first-order theories as propositional proof system in the sense of Cook-Reckhow [CR] and we study their efficiency in terms of polynomial simulations.
We prove that predicate calculus is polynomially equivalent to Frege systems while a theory saying that there are at least two different elements is polynomially equivalent to the Quantified propositional calculus. We prove analogous results also for some stronger theories. Further we define the notion of a "weak theory" and show that weak theries can be polynomially simulated by the Quantified propositional logic too. We conclude with some negative examples and some open problems.

2 Introduction

By the well known result of S.A. Cook [C] the set of propositional tautologies (TAUT) in DeMorgan language regarded as strings over a finite alphabet is $coNP$-complete. Thus the problem whether the set NP is closed under the complementation is equivalent to the problem if TAUT is in NP. This problem is further equivalent to a question whether there exists a propositional proof system in which all propositional tautologies would have a proof of length polynomial in the length of the tautology. This question was first studied in [CR]. In that paper a general definition of a propositional proof system was introduced, allowing to study the length of a shortest proof of propositional tautology in various proof systems as a function of the length of the tautology. The smallest upper bound known for this function is exponential for any proof system. The difficult question is whether there is a polynomial bound on this function for some proof system.

The only known results for stronger proof systems are relative. They say that some proof systems are about equivalent. The paper [CR] comes with several such results. It compares the standard proof systems called Frege systems and natural deduction systems on different classes of connective sets and states that they are equivalent up to a translation of proofs by a polynomial-time function. The paper alse introduces the so called Extended Frege systems that allow for shortening of the formulas in proofs in some cases.

The question is how can we create other propositional proof systems that might be stronger than the traditional systems. One of the natural ways that can be used is to extend the calculi of Frege systems into Predicate calculi and use the nulary predicate symbols as propositional variables. Such systems are sound and complete because they are an extension of a Frege system. The question we are interested in is if such systems can allow for some shorter proofs. Similarly we can further extended the predicate calculi into first-order theories and study such systems as propositional proof systems. This provides an unlimited number of ways of creating new propositional proof systems.

To help us study and compare different proof systems we will introduce in Chapter 3 a more general definition of a propositional proof system as well as ways of measuring and comparing the lengths of proofs. Although it is not known except for some special cases if some proof systems are more powerful then others there are some results stating that some basic proof

systems have the same strength. We will introduce in Chapter 4 the basic proof systems that were studied by Cook and others such as Frege systems, extended Frege systems and the Quantified propositional calculi. In the next Chapter 5 we will formally define the predicate calculi as propositional proof system to fit the general definition. Further we will show that such systems are polynomially equivalent to Frege systems up to an application of a polynomial. The next natural step is to extend predicate calculi by other axioms and to study the resulted theories. An interesting result we prove is that some theories are as powerful as Extended Frege systems or the Quantified propositional calculi. These results are developed in Chapter 6. In Chapter 7 we will try to find some very general description of theories equivalent to Quantified propositional calculus while in the last chapter we will try to examine some theories about which we do not know if they are not stronger. Chapter 9 will state some open problems pointing towards possible further research.

3 Propositional proof systems

The symbols Σ, Σ_1,... will denote in the following text a finite alphabet of cardinality at least 2. The symbol Σ^* will denote the set of all finite words over alphabet Σ.
The *DeMorgan language* for propositional logic consists of $\{0, 1, \wedge, \vee, \neg\}$ plus the auxiliary symbols like brackets, commas, etc. and symbols for variables p, q, r, p_0, p_1,.... In a finite alphabet, the variables and constants are strings (say a letter p followed by a string over $0, 1$) in order to have an unlimited supply of them.
A *formula* refers to a propositional formula built up in the usual way from atoms (propositional variables) and connectives from DeMorgan language, using infix notation.
If $A_1, \ldots, A_n, B$ are formulas, then we write $A_1, \ldots, A_n \models B$ if B is a tautological consequence of $A_1, \ldots, A_n$. A *derivation* (from zero or more formulas called *hypotheses*) in such a system is a particular finite sequence of formulas ending in the formula proved. Each formula must be eather a hypotheses, or must follow from earlier lines by a rule of inference. If derivation has no hypotheses, it is called a *proof*.
The notation $A_1, \ldots, A_k \vdash_P^\pi B$ means that π is a derivation of B from hypotheses $A_1, \ldots, A_k$ in the proof system P. (The notation $\vdash_P$ means that there is some derivation π in system P.)
Thus to specify a propositional proof system for our purposes, it is only necessary to specify a finite system of rules of inference.
We use the following notation for various complexity measures of proofs and formulas according to [CR]:
$|w|$ will denote the length of a word w in DeMorgan language as a string. If encoded in a finite alphabet Σ, the length of a variable p_i is proportional to $\log(i)$. In the case of formulas, it is more natural to count the size by the number of occurrences of atoms and of constants in the formula A. We denote this as $l(A)$. Notice that every atom ocuring in A has the length (after possible renaming of atoms) $|p_i| \leq O(\log(l(A))$. And so $l(A) \leq |A| \leq O(l(A) \cdot \log(l(A))$. Therefore we will measure the length by $l(A)$: It is an insignificant change and the l-measure is more natural in our context. For a derivation π, $l(\pi)$ is the sum of $l(A)$s for all formulas A in π.
In following text we will use a more general definition of a proof system than in the example above:

Definition. 3.1 [CR]

1. *TAUT will denote the set of propositional tautologies in DeMorgan language.*

2. *If $L \subseteq \Sigma^*$, a* proof system *for the language L is a polynomial-time function $f : \Sigma_1^* \to L$ for some alphabet Σ_1 such that f is onto. If $y = f(x)$, then we will say that x is a* f-proof *of y.*

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

4. *A* propositional proof system *is any proof system for the set TAUT.*

It is easy to see (and argued for in [CR]) that any conventional proof system for tautologies naturally fits this general definition of a proof system.
Although it is doubtful that every propositional proof system fitting this general definition is natural, the following proposition will explain the motivation for the general definition and also one of the motivations for the research on the lengths of the proofs. In the following NP and $coNP$ are the well known classes in computational complexity.

THEOREM. 3.2 [CR] *TAUT has a polynomially bounded proof system iff $NP = coNP$.*

The following definition allows us to compare the strength of two propositional proof systems, with respect to the lengths of the proofs.

DEFINITION. 3.3 *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 provided there is a polynomial-time function $g : \Sigma_1^* \to \Sigma_2^*$ such that $f_1(x) = f_2(g(x))$ for all $x \in \Sigma_1^*$.*

The following proposition is obvious:

PROPOSITION. 3.4 *If a proof system f_2 for L p-simulates a polynomially bounded proof system f_1 for L, then f_2 is also polynomially bounded.*

We will use the notation $f_1 \leq_p f_2$ to denote that f_2 p-simulates f_1. It can be easily seen that the relation $\leq_p$ is reflexive and transitive, i.e. it is a quasi-ordering. If two systems p-simulate each other, they would be considered as *p-equal* ($f_1 \approx_p f_2$).

If f_1, f_2, g are as in the Definition 3.3 and there is a constant c such that $c \cdot l(x) \geq l(g(x))$, then we would say that f_2 *linearly simulate* f_1 (also *l-simulate*, $f_1 \leq_l f_2$). If both systems linearly simulate each other, they will be considered as *linearly equal* (*l-equal*, $\approx_l$).
Note that in l-simulation the function g translating the proofs can still be polynomial-time computable and need not to be linear-time computable.

4 Examples of propositional proof systems

Frege systems. In the most usual propositional proof systems, the rules of inference are formula schemes and an instance of the scheme is obtained by applying a substitution to the scheme. Such systems are being called *Frege systems*.

Definition. 4.1 [CR]
If $D_1, \ldots, D_k$ are formulas and $p_1, \ldots, p_k$ are distinct atoms, then $\sigma = (p_1, \ldots, p_k) \leftarrow (D_1, \ldots, D_k)$ is a substitution*, and σA is a formula which results by simultaneously replacing p_i by D_i, $i = 1 \ldots k$, in the formula A.*
A Frege rule *is a tuple of formulas $(C_1, \ldots, C_n)/D$, where D is a tautological consequence of $C_1, \ldots, C_n$ ($C_1, \ldots, C_n \models D$ in symbols), i.e. every truth assignment satisfying all $C_1, \ldots, C_n$ satisfies also D.*
If $n = 0$, the rule is an axiom scheme*.*
For any substitution σ we say that σD follows from *$\sigma C_1, \ldots, \sigma C_n$ by the rule $(C_1, \ldots, C_n)/D$.*
An inference system *F is a finite set of Frege rules. The notions of a* derivation *and the symbol $\vdash_F$ for F are defined as in the previous chapter. By our condition on the definition of a Frege rule, it is clear that if $A_1, \ldots, A_n \vdash_F B$ then $A_1, \ldots, A_n \models B$.*

Definition. 4.2 [CR]
An inference system F is implicationaly complete *if $A_1, \ldots, A_n \vdash_F B$ whenever $A_1, \ldots, A_n \models B$. A* Frege system *is an implicationaly complete inference system in a complete language.*

Example. 4.3 *An example of a Frege system could be one which has connectives $\neg$, $\rightarrow$, the inference rule is Modus Pones (MP): $(A, A \rightarrow B)/B$, and which has additional six axiom schemes:*

$F1$: $A \rightarrow (B \rightarrow A)$

$F2$: $(C \rightarrow (B \rightarrow A)) \rightarrow ((C \rightarrow B) \rightarrow (C \rightarrow A))$

$F3$: $(D \rightarrow (B \rightarrow A)) \rightarrow (B \rightarrow (D \rightarrow A))$

$F4$: $(B \rightarrow A) \rightarrow (\neg A \rightarrow \neg B)$

$F5$: $\neg\neg A \rightarrow A$

$F6 \ : A \to \neg\neg A$

THEOREM. 4.4 [CR]
All Frege systems over any language containing the DeMorgan language p-simulate each other and thus they are p-equal. This also holds for the so called natural deduction systems and Gentzen sequent calculus with cut.

Extended Frege systems. [CR] introduces a proof system that might allow for shorter proofs than Frege proof systems. They show a way of shortening proofs on an example of a proof of the "pigeon-hole principle". The device, by which these systems shorten formulas in proofs of Frege systems, is by introducing new atoms, which serve as abbreviations of subformulas in formulas of the proof.

DEFINITION. 4.5 [CR] *An* extended Frege system *is a proof system which consist of a Frege system F together with the* extension rule *which allows formulas of the form $p \equiv A$ to be added to the derivation, where A is any formula and p is any "new" atom. Atom p must not occur in A, in any lines preceding $p \equiv A$, or in any hypotheses to the derivation. p can occur in the later lines, but not in the last line.*
The extended Frege system based on F is denoted EF.

The following propositions are proved for extended Frege systems in [CR]:

PROPOSITION. 4.6 (SOUNDNESS OF EF) *If $A_1, \ldots, A_n \vdash_{EF} B$ then $A_1, \ldots, A_n \models B$.*

PROPOSITION. 4.7 *A given extended Frege system EF is polynomially bounded if and only if all extended Frege systems over all languages are polynomially bounded.*
Also, an extended Frege system EF is polynomially bounded if and only if there is a polynomial bound on the number of lines in proofs in EF.

Quantified propositional calculus. For an easier application in subsequent proofs, we will take the definition of Quantified propositional calculus (G) based on Gentzen sequent calculi. However, by the p-equivalence of Frege systems and sequent calculus showed in [CR], this system is p-equal to quantified propositional calculi based on Frege systems from Example 4.3.

In sequent calculi, the derivation is not made up from formulas, but from sequents of formulas. In following definition, the symbols Γ, Π, Δ, Λ denote (possible empty) sets of formulas.

DEFINITION. 4.8 ([KP]) *Calculus G is based on classical propositional Gentzen sequent calculus:*

$A: \quad / \quad \Gamma, \varphi \Rightarrow \Delta, \varphi;$ $\qquad / \quad \Rightarrow 1;$ $\qquad / \quad 0 \Rightarrow$

$W: \quad \Gamma \Rightarrow \Delta \ / \ \Gamma \Rightarrow \Delta, \varphi;$ $\qquad \Gamma \Rightarrow \Delta \ / \ \Gamma, \varphi \Rightarrow \Delta$

$\vee r: \quad \Gamma \Rightarrow \Delta, \varphi \ / \ \Gamma \Rightarrow \Delta, \varphi \vee \psi;$ $\qquad \Gamma \Rightarrow \Delta, \varphi \ / \ \Gamma \Rightarrow \Delta, \psi \vee \varphi$

$\wedge l: \quad \Gamma, \varphi \Rightarrow \Delta \ / \ \Gamma, \varphi \wedge \psi \Rightarrow \Delta;$ $\qquad \Gamma, \varphi \Rightarrow \Delta \ / \ \Gamma, \psi \wedge \varphi \Rightarrow \Delta$

$\wedge r: \quad \langle \Gamma \Rightarrow \Delta, \varphi \rangle, \langle \Gamma \Rightarrow \Delta, \psi \rangle \ / \ \Gamma \Rightarrow \Delta, \varphi \wedge \psi$

$\vee l: \quad \langle \Gamma, \varphi \Rightarrow \Delta \rangle, \langle \Gamma, \psi \Rightarrow \Delta \rangle \ / \ \Gamma, \varphi \vee \psi \Rightarrow \Delta$

$\neg l: \quad \Gamma \Rightarrow \Delta, \varphi \ / \ \Gamma \neg \varphi \Rightarrow \Delta$

$\neg r: \quad \Gamma, \varphi \Rightarrow \Delta \ / \ \Gamma \Rightarrow \Delta, \neg \varphi$

$\rightarrow r: \quad \Gamma \varphi \Rightarrow \Delta, \psi \ / \ \Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$

$\rightarrow l: \quad \langle \Gamma, \varphi \Rightarrow \Delta \rangle, \langle \Pi, \psi \Rightarrow \Lambda \rangle \ / \ \Gamma, \Pi, \varphi \rightarrow \psi \Rightarrow \Delta, \Lambda$

$Cut: \quad \langle \Gamma, \Rightarrow \Delta, \varphi \rangle, \langle \Pi, \varphi \Rightarrow \Lambda \rangle \ / \ \Gamma, \Pi \Rightarrow \Delta, \Lambda$

and contains also the quantifier rules:

$\exists r: \quad \Gamma \Rightarrow \Delta, \varphi(\psi) \ / \ \Gamma \Rightarrow \Delta, \exists p \varphi(p)$

$\forall l: \quad \varphi(\psi), \Gamma \Rightarrow \Delta \ / \ \forall p \varphi(p), \Gamma \Rightarrow \Delta$

$\exists l: \quad \varphi(p), \Gamma \Rightarrow \Delta \ / \ \exists x \varphi(x), \Gamma \Rightarrow \Delta$

$\forall r: \quad \Gamma \Rightarrow \Delta, \varphi(p) \ / \ \Gamma \Rightarrow \Delta, \forall x \varphi(x)$

with the proviso that p does not occur in the lower sequents of $\forall r$ and $\exists l$.

THEOREM. 4.9 $G \geq_p EF \geq_p F$
PROOF. Naturally $EF \geq_p F$, as EF is an extension of F, so a proof in F is a special case of a proof in EF.
It is also well known that $G \geq_p EF$, as shown for example in [K]. The p-simulation of EF by G also follows from our results Lemma 6.7, Proposition 6.6 and Proposition 6.2. □

On the other hand it is known neither if $F \geq_p EF$ nor if $EF \geq_p G$. It is expected that none of these p-simulations are true. In the first case it is because there is no hint how to simulate the extension rule in F, while in the second case it seems difficult to simulate quantifiers in EF.
Notice that a propositional formula with quantifiers can be easily converted into an equivalent formula without quantifiers (because the quantified propositional variables can only have two values: 0-false, 1-true), according to the schemes:

$$\begin{aligned} \exists p\varphi(p) &\equiv \varphi(0) \vee \varphi(1) \\ \forall p\varphi(p) &\equiv \varphi(0) \wedge \varphi(1) \end{aligned}$$

However, by a such translation the length of the formulas doubles, and by translating a block of quantifiers, the length of the translated formula might grow exponentially.

5 Predicate calculus as a propositional proof system

In this section we will show how the predicate calculi can serve as propositional proof systems. This is trivial, as predicate calculi extend propositional calculi, and the nulary predicates can be seen as propositional variables. The more important result will be, that classical predicate calculi are p-equivalent to Frege systems.
The symbol L_0 will denote the extension of the DeMorgan language for predicate calculi by allowing arbitrary many but only nulary predicate symbols.
A general language L will contain n-ary predicate symbols, for any natural number n, but no function symbols. Later on, we will allow also constants.
The standard calculi for first-order logic (predicate calculi) are based on some propositional calculi. They contain the propositional axioms and rules, with the difference, that the formulas being substituted are now first-order formulas. In addition they contain some set of quantifier rules depending on the particular predicate calculus used.
For a convenience we will use two predicate calculi in the following text. The Hilbert style calculus based on the Frege system from Example 4.3 and Gentzen sequent calculus based on the non-quantifier part of Gentzen calculus from Definition 4.8. These are p-equivalent as propositional proof systems, similarly as their propositional parts are by Theorem 4.4 .

DEFINITION. 5.1 Hilbert predicate calculus *will have the axioms* $F1$ - $F6$ *and the rule* MP *from Example 4.3, where now the formulas* A, B, C *are first-order formulas in some language of predicate logic. It also contains the following axiom schemes for quantifiers:*

specification:	$\forall x\varphi \to \varphi(x/t)$,	*where* t *is term substitutable in* φ
distribution:	$\forall x(\varphi \to \psi) \to (\varphi \to \forall x\psi)$	φ *does not contain free* x,

and the rule of generalization: $\varphi/\forall x(\varphi)$.
The quantifier $\exists$ *is defined as* $\neg\forall\neg$.

Gentzen sequent predicate calculus *has the rules from Definition 4.8 except of the propositional quantifier rules. In addition it contains the following predicate quantifier rules:*

$\exists r: \quad \Gamma \Rightarrow \Delta, \varphi(t) \;\; / \;\; \Gamma \Rightarrow \Delta, \exists x\varphi(x)$

$\forall l: \quad \varphi(t), \Gamma \Rightarrow \Delta \ / \ \forall x \varphi(x), \Gamma \Rightarrow \Delta$

$\exists l: \quad \varphi(y), \Gamma \Rightarrow \Delta \ / \ \exists x \varphi(x), \Gamma \Rightarrow \Delta$

$\forall r: \quad \Gamma \Rightarrow \Delta, \varphi(y) \ / \ \Gamma \Rightarrow \Delta, \forall x \varphi(x)$

where in the $\exists r$, $\forall l$ term t is substitutable for x in φ, and in case of $\exists l$, $\forall r$ y is substitutable for x in φ and does not have any free occurrences in Γ,Δ or in $\exists x \varphi$ (resp. $\forall x \varphi$). See [SV].

Any predicate calculus with language $L \supseteq L_0$ can be naturally interpreted as a propositional proof system fitting the general Definition 3.1. Just view the nulary predicates of L_0 as propositional variables. If the last line of a derivation in predicate calculus contains only nulary predicates of L_0 and no quantifiers, it can be seen as a propositional formula. By correctness of the calculus it is a tautology, and by its completeness any propositional tautology can be proved in this way.
Further we assume that L is encoded in a finite alphabet Σ. Also, a theory means a set of axioms and not the set of their consequences. That is, we consider two different sets of axioms having the same consequences as different. That is irrelevant when studying provability, but important when studying lengths of proofs.

Definition. 5.2 *Propositional proof system based on a predicate calculus will be a predicate calculus with a language containing L_0, with the difference that the derivations do not have any premises, and the last line of a derivation can be only a formula that contain only the nulary predicates from L_0 and no quantifiers. In the case of sequent calculus we assume the last line of the derivation has the form $\Rightarrow \varphi$, where φ is a formula that contain only the nulary predicates from L_0 and no quantifiers.*

Lemma. 5.3 *Let P be a sound predicate calculus in a relational language $L \supseteq L_0$. Assume that there is a polynomial-time algorithm deciding if a string in the alphabet Σ is an axiom of P and if a k-tuple of strings is an instance of a rule of P.*
Then P, interpreted as a proof system for propositional logic as above, is a propositional proof system in the sense of Definition 3.1.

Proof. We have verified the completeness and soundness already. It remains to verify that it is decidable in polynomial time if a string is a proof in P of a formula. But that follows from the hypotheses. □

Whether we use Hilbert or Gentzen calculi, the propositional proof system based on it will be denoted as PL_0, if it has language only L_0, and PL, if it has general language L.
We also extend the definition of the length of a formula to first-order formula φ: $l(\varphi)$ is the number of occurences of first-order variables, constants and of nulary predicates in φ.

DEFINITION. 5.4 *$PL_=$ will denote a proof system containing PL, the binary predicate "=", and the axioms of identity:*

$E1: \quad \forall x(x = x)$

$E2: \quad \forall x \forall y(x = y \rightarrow y = x)$

$E3: \quad \forall x \forall y \forall z(x = y \wedge y = z \rightarrow x = z)$

$E4: \quad \forall \bar{x} \forall \bar{y}(x_1 = y_1 \wedge \ldots \wedge x_n = y_n \rightarrow R(x_1, \ldots, x_n) \equiv R(y_1, \ldots, y_n))$

where $E4$ is an axiom scheme for every n-ary predicate symbol R from L.

Because predicate calculi contain also the propositional part (axioms and rules), the proof systems based on it have at least the same strength as the propositional calculi, with respect to the length of the proofs. On the other hand it also has some new tools (predicate axioms, rules), so it is a question whether it can allow for some shorter proofs. The rest of this section will try to compare Frege systems with PL_0, PL, and $PL_=$.

In the following text many of the propositions will be of the form: "proof system f_2 p-simulates proof system f_1" and their proofs will have the style that any derivation in f_1 can be transformed into a derivation in f_2 with the same last line. In such case we will denote the derivation in f_1 as π and the translated derivation in f_2 as π'. Similarly the formulas φ in π will have their translations in π' denoted as φ'.

LEMMA. 5.5 *PL_0 is linearly equal to its underlying Frege system, and hence it is polynomially equal to all Frege systems.*
PROOF. We shall show that both systems can simulate each other in maximum linear increase of the lengths of the proofs. Let F be the underlying Frege system.
$F \leq_l PL_0$ - Derivation in the Frege system is a special case of a derivation in PL_0 with no quantifiers. So the proof can remain unchanged except

that the symbols for propositional variables will be replaced everywhere by symbols for nulary predicates.
The opposite direction:
$F \geq_l PL_0$ - Because system PL_0 contains only nulary predicates, it's formulas do not contain any predicate variables and therefore the quantifiers are useless. Any formula in L_0 containing quantifiers has the same logical value as the same formula with the quantifiers removed. Therefore we remove all quantifiers from all formulas in the derivation π in PL_0 and change the nulary predicates to propositional variables. We need to verify only that we obtain in this way a derivation π' in F.
Let us consider translations of several types of inferences in it. In the case that formula φ was obtained from previous formulas $\psi, \psi \to \varphi$ by Modus Pones (MP), in π' will be obtained in the same way φ' from $\psi', \psi' \to \varphi'$. In the case $\forall x\varphi$ was obtained by the generalization rule from φ, we have $(\forall x\varphi)' = \varphi'$: in the new derivation the inference is void.
The last possibility is that a formula with a quantifier appeared in the derivation as an axiom of the specification $\forall x\varphi \to \varphi(x/t)$ or as the distribution axiom $\forall x(\varphi \to \psi) \to (\varphi \to \ \forall x\psi)$. In such a case formulas $(\varphi' \to \varphi')$ resp. $(\varphi' \to \psi') \to (\varphi' \to \psi')$ will appear in the derivation π'. These formulas can be easily proved in a Frege system by a proof of the length proportional to the length of the formula. □

PL_0 can be clearly lineary simulated by both PL and $PL_=$, as it is included in them. But more holds:

LEMMA. 5.6 *PL_0 linearly simulates PL and $PL_=$.*
PROOF. Observe that in a derivation of a tautology each line is also a tautology. Because a tautology must be true in any structure, it must be true also in the particular structure with only one element. In an evaluation for such a structure, each predicate P returns always the same value, depending only on the name of predicate and not on the terms or variables taking place in it. Further we postulate the fact, that the binary predicate "$\_=\_$" returns always true, because of the axiom $\forall x(x = x)$.
This suggests that we can define a transformation of formulas φ of $L_=$ to equivalent formulas φ' in L_0, thinking about φ' as the evaluation in a one element model.
More formally: For φ from π define φ' by induction on the complexity of φ:

1. $(R(x_1, \ldots, x_n)' := 0$, if R is not nulary and different from $=$.

2. $(x = y)' := 1$

3. $R' := R$ if R is nulary

4. the translation commutes with $\wedge$, $\vee$, $\neg$

5. $(\forall x\varphi)' = (\exists x\varphi)' = \varphi'$

By such a replacement in the whole derivation in PL or in $PL_=$ we obtain an instance of a proof in the system PL_0. □

COROLLARY. 5.7 *F, PL_0, PL, $PL_=$ l-simulate each other.*
PROOF. Using the last two lemmas and the transitivity of the linear simulation. □

6 Weak first order theories

Now we start adding axioms to PL_0 or $PL_=$ and we shall investigate how strong will the resulting theories be as propositional proof systems. In order to be a propositional proof system, axioms of a theory must not be contradictory and may not say anything about the predicates from L_0. It will be clear that all the theories have polynomial-time sets of axioms.
The first thing to investigate is the strength of axioms rulling out the one element model used in the proof of Theorem 5.6.
While we still do not allow function symbols of arity at least 1, we shall allow now constants in the language.

DEFINITION. 6.1 *The following theories extend the predicate logic with identity $PL_=$:*

$T_{0\neq 1}$ - *is a theory of structures of size at least 2 and two 'named' elements. It has a language containing constants* 0, 1 *and an axiom* $0 \neq 1$.

$T_{\geq n}$ - *with axiom*

$$\exists x_1 \ldots \exists x_n (\bigwedge_{1 \leq i < j \leq n} (x_i \neq x_j))$$

is a theory of structures of size at least n.

T_∞ - *defined as*

$$\bigcup_{n \in N} T_{\geq n}$$

is a theory of infinite structures.

PROPOSITION. 6.2 *$T_{0\neq 1}$ p-simulates EF.*
PROOF. To simulate the extension rule we would like for any formula φ find a predicate that will be equal to it. The important difference of $T_{0\neq 1}$ from PL is, that $T_{0\neq 1}$ allows to derive sequents

$$\Rightarrow \exists x(x = 1) \equiv \varphi \tag{1}$$

for any formula φ.
To simulate a derivation π in EF, for every its extension axiom $p \equiv \varphi$, replace the extension atom p by an atomic formula $(x_p = 1)$, where x_p is a new variable. Formulas ψ after such a substitution will be denoted as ψ'.

Then add formula $(x_p = 1) \equiv \varphi'$ to the antecedent of every sequent in the derivation. Thus the extension axioms will transform into rule (A) of Gentzen sequent calculus.
By such a change we obtain a correct derivation in $T_{0\neq 1}$. The derivations will be ending with a line of the form

$$(x_{p_1} = 1) \equiv \varphi'_1, \ldots, (x_{p_n} = 1) \equiv \varphi'_n \Longrightarrow A$$

where $p_1, \ldots, p_n$ are the defined atoms in order in which they were introduced by extension axioms $p_1 \equiv \varphi_1, \ldots, p_n \equiv \varphi_n$ in π.
From the definition of EF it follows that for every $i = 1 \ldots n$, p_i does not appear in any of the formulas $A, \varphi_1, \ldots \varphi_i$ and therefore x_{p_i} does not appear in any of the formulas $A, \varphi'_1, \ldots \varphi'_i$. So the rule $\exists l$ can be always applied on the highest x_{p_i}, starting with x_{p_n}. Then the formula, on which the $\exists l$ rule was applied, can be removed by CUT with formula (1). By repeating n-times this step we obtain the wanted sequent $\Longrightarrow A$.
The first part of derivation π' contains the same number of lines as π. The formulas are maximum 2-times longer than in π, plus every line contains the formulas $(x_{p_i} = 1) \equiv \varphi'_i$ for every extension axiom of π. Because the total length of the extension axioms is not bigger than $l(\pi)$, this first part is of length $O(l(\pi)^2)$. The second part is n-times using rule $\exists l$, a proof of formula (1) for the formula φ'_i, which is linearly equal to φ'_i and CUT. So it has also length maximum $O(l(\pi)^2)$. □

For the next proposition we will first prove some useful lemmas:

LEMMA. 6.3 *If a sequent $\varphi(P^0), \Gamma \Rightarrow \Delta$ (resp. $\Gamma \Rightarrow \varphi(P^0), \Delta$), where P^0 is a nulary predicate not appearing in φ, Γ, Δ, is derivable in derivation π, then any sequent of the form $\varphi(\psi), \Gamma \Rightarrow \Delta$ (resp. $\Gamma \Rightarrow \varphi(\psi), \Delta$) is derivable by derivation of the length proportional to $l(\pi) \cdot l(\psi)$.*
PROOF. The proof is trivial. By replacing predicates P^0 by ψ in derivation of $\varphi(P^0), \Gamma \Rightarrow \Delta$, receive a derivation of sequent $\varphi(\psi), \Gamma \Rightarrow \Delta$ of the same number of lines and formulas maximum $l(\varphi)$-times longer. □

LEMMA. 6.4 *The sequent $\psi_1 \equiv \psi_2, \varphi(\psi_1) \Rightarrow \varphi(\psi_2)$ has a proof in G polynomial in the lengths of ψ_1, ψ_2 and φ.*
PROOF. We omit the proof as a stronger version of this Lemma will be introduced in Lemma 7.3. □

Now we are ready to improve Proposition 6.2.

PROPOSITION. 6.5 *$T_{0\neq 1}$ polynomially simulates G.*
PROOF. If π is a derivation of a tautology in G then it can be considered as a derivation π' in $T_{0\neq 1}$ (with the difference that propositional variables (p, q, r) are now being a nulary predicates (P^0, Q^0, R^0)) until the quantifier rules are used.
A quantifier rule applied in π on a propositional variable p cannot be simply applied on nulary predicate P^0 in derivation π'. Thus, in order to simulate a quantifier rule, we would have to substitute the nulary predicate by some formula containing a predicate variable x_p, which can be quantified. The value of the formula must depend only on the evaluation of x_p.
In $T_{0\neq 1}$, the only predicate that can contain a variable, is the binary predicate "=". Thus we can use, for example, the formula $(x_p = 1)$ for substituting for the predicate symbol P^0. Because the models of $T_{0\neq 1}$ are of size at least two, it can then be either true or false depending only on the evaluation of the variable x_p.
Now we discuss in detail the simulation of the G quantifier rules in π':
The rules $\exists l$, $\forall r$ will be simulated differently than $\exists r$, $\forall l$. The $\exists l$ in G is

$$\frac{\varphi(p), \Gamma \Rightarrow \Delta}{\exists q \varphi(q), \Gamma \Rightarrow \Delta}$$

where formulas $\exists q \varphi(q), \Gamma, \Delta$ do not contain p. Now we would like to substitute the variable p by the formula $(x_p = 1)$, where x_p is a new first-order variable.
Now, by theorem from [K], we may assume that π is in a tree form. So $\varphi(p), \Gamma \Rightarrow \Delta$ has a separate derivation in π. And this derivation can be, by Lemma 6.3, changed into a derivation of

$$\varphi(x_p = 1), \Gamma \Rightarrow \Delta$$

of proportional length. Then apply the rule $\exists l$ of PL to it

$$\exists x \varphi(x = 1), \Gamma \Rightarrow \Delta$$

to get the wanted sequent. The rule $\forall r$ can be simulated in the same way.
In the case of $\exists r$ or $\forall l$ the situation is different. The $\exists r$ in G be

$$\frac{\Gamma \Rightarrow \Delta, \varphi(\psi)}{\Gamma \Rightarrow \Delta, \exists p \varphi(p)}$$

We cannot so easily substitute a formula ψ, because it can be a tautology, or a negation of a tautology, or some variables free in it might be also free in Γ or Δ. But according to Lemma 6.4, from $\Gamma \Rightarrow \Delta, \varphi(\psi)$ we can derive a sequent

$$(x_\psi = 1) \equiv \psi, \Gamma \Rightarrow \Delta, \varphi(x_\psi = 1)$$

where x_ψ is a new variable, in the length of a derivation depending polynomially on the lengths of φ, ψ. And from that it can be derived

$$\begin{aligned} (x_\psi = 1) \equiv \psi, \Gamma &\Rightarrow \Delta, \exists x \varphi(x = 1) \\ \exists x((x = 1) \equiv \psi), \Gamma &\Rightarrow \Delta, \exists x \varphi(x = 1) \end{aligned}$$

by predicate rules $\exists r$, $\exists l$. And because $\Rightarrow \exists x((x = 1) \equiv \psi)$ is derivable from axioms $0 \neq 1$ and $E1$ for any formula ψ, by CUT from that we derive the wanted sequent

$$\Gamma \Rightarrow \Delta, \exists x \varphi(x = 1)$$

Similarly in simulation of $\forall l$, from the sequent

$$\Gamma, \varphi(\psi) \Rightarrow \Delta$$

we can derive

$$(x_\psi = 1) \equiv \psi, \varphi(x_\psi = 1), \Gamma \Rightarrow \Delta$$

by Lemma 6.4 and by CUT. The rest is the same.
Because the last line of π by definition does not contain any quantifiers, the last line of π' will not contain any predicates "=" (and first-order variables).
In the case of $\exists l$, $\forall r$, altogether the simulations could only prolong the derivation by doubling the size of the formulas. And in the case of $\exists r$, $\forall l$ the simulation is polynomial in the length of the line on which the rule was applied.
So there is a polynomial $p(x)$ such that for any proof π in G it holds that $l(\pi') \leq p(l(\pi))$. □

PROPOSITION. 6.6 *$T_\infty \geq_l (T_{\geq n}) \geq_l T_{0\neq 1}$, for $n \geq 2$.*
PROOF. First we show a special case of the second simulation $T_{\geq 2} \geq_l T_{0\neq 1}$: The theory $T_{\geq 2}$ can prove the same formulas as $T_{0\neq 1}$ which does not contain the constants 0, 1. But these constants can be substituted by variables v_0, v_1 if the premise $v_0 \neq v_1$ is included. So every line of π will be

changed into a line of π' by replacing the constants 0, 1 by variables v_0, v_1, and by adding the formula $v_0 \neq v_1$ into the antecedent of the sequent.
In proofs of propositional tautologies, the last line of the derivation would contain only one propositional formula A in the succedent (with no first-order variables or constants). Thus the last line of π' will have the form:

$$v_0 \neq v_1 \Rightarrow A$$

From that, by twice using $\exists l$ (A does not contain v_0, v_1), derive

$$\exists v_0 \exists v_1 (v_0 \neq v_1) \Rightarrow A$$

But the left part of this sequent is an axiom of $T_{\geq 2}$, so by CUT the formula A is derivable.
The length of π' is linearly longer then the length of π, with the multiplicative constant ($l(v_0 \neq v_1)$) on the number of lines of π, plus it has two lines being proportional to A at the end. So π' is being proportional to π.

$T_{\geq n} \geq_l T_{\geq m}$ for $n \geq m$:
The specific axiom of $T_{\geq m}$ is provable in $T_{\geq n}$ in number of lines depending only on n and m.

$T_\infty \geq_l T_{\geq n}$:
By definition T_∞ contains the axiom of $T_{\geq n}$ for any n. So the derivation in $T_{\geq n}$ is automaticaly a derivation in T_∞. □

In the rest of this section we will discuss the power of G. The lemmas will grow in their strength, so we could have only stated and proved the last one. But we will proceed gradually because the parts proved will be later used anyway, and also it will help to keep the text well structured. In fact, the first two lemmas are slightly stronger (they show a linear simulation) then it is necessary for the final result.

Lemma. 6.7 *G l-simulates $T_{\geq 2}$.*
Proof. The main idea of this proof is that predicate variables x of π will have assigned their propositional variables p_x in π'. The formulas φ' in π' will be obtained from formulas φ of π by substituting all their subformulas $(x = y)$ by formulas $(p_x \equiv p_y)$ and quantifiers $\forall x$ (resp. $\exists x$) by quantifiers $\forall p_x$ (resp. $\exists p_x$).

The lines obtained by PL-rules in the derivation π can be simulated by the same G-rules in π'. Further observe that axioms of identity $E1$, $E2$, $E3$ and the special axiom of $T_{\geq 2}$ are provable in G after such a translation. By the completeness they have some proofs in G of a constant length. So by adding these proofs at the beginning of the derivation obtained from π by translating the formulas, we obtain a correct derivation π' in G. □

LEMMA. 6.8 *G l-simulates $T_{\geq n}$ where n is any natural number bigger then 2.*

PROOF. The main idea of translating π in T_n into π' in G is in an encoding of predicate variables x into a systems of k propositional variables where $k = \lceil \log_2(n) \rceil$.

Every predicate variable x in π will be encoded by a tuple of propositional variables $p_1^x, \ldots, p_k^x$ denoted as $\bar{p}^x$. Thus every formula φ in π will be transformed into a formula φ' in π' in the following way: Every quantifier Qx in φ will be replaced by the string

$$Qp_1^x \ldots, Qp_k^x$$

denoted by $Q\bar{p}^x$, and every formula of form $x = y$ will be replaced by

$$((p_1^x \equiv p_1^y) \wedge \ldots \wedge (p_k^x \equiv p_k^y))$$

denoted as $\bar{p}^x = \bar{p}^y$. This abbreviated notation will be used through the rest of this thesis.

Because this new subformula $\bar{p}^x = \bar{p}^y$ can be satisfied only if all pairs p_i^x, p_i^y are of the same value, and there is at least n possible different evaluations of the system of k propositional variables, the formula φ' is a tautology if φ follows from $T_{\geq n}$.

Now let us discuss the formulas φ appearing in π and their translations φ' in π'.

If φ appeared in π as in propositional axiom $\varphi \Rightarrow \varphi$, then $\varphi' \Rightarrow \varphi'$ is an instance of the same axiom schema, and therefore does not have to be especially proved. Similarly, if φ was derived by some propositional rule from formulas ψ_1, ψ_2 in previous sequents, observe that φ' can be derived from formulas ψ_1', ψ_2' in the translated sequents.

If φ was derived by a quantifier rule of PL from ψ, then φ' can be derived by k times using the equivalent rule of G from ψ'. We will show the simulation of the rule $\exists r$. The other cases ($\forall r$, $\exists l$, $\forall l$) would be similar. $\exists r$ in PL is:

$$\frac{\Gamma \Longrightarrow \Delta, \psi(t)}{\Gamma \Longrightarrow \Delta, \exists x \psi(x)}$$

Now, because the term t in language of $T_{\geq n}$ could be only some variable y appearing in the predicate $=$, in $\psi(t)'$ term t will be some system of variables $\bar{p}$ in a formula translating $=$. So this rule will be simulated by k-times using $\exists r$ in G on the variables $\bar{p}$:

$$\begin{array}{rcl} \Gamma' & \Longrightarrow & \Delta', \psi'(\bar{p}) \\ \Gamma' & \Longrightarrow & \Delta', \exists x_k \psi'(p_1, \ldots, p_{k-1}, x_k) \\ & \vdots & \\ \Gamma' & \Longrightarrow & \Delta', \exists x_2 \ldots \exists x_k \psi'(p_1, x_2, \ldots, x_k) \\ \Gamma' & \Longrightarrow & \Delta', \exists \bar{x} \psi'(\bar{x}) \end{array}$$

Finally, if φ is the axiom of identity or the axiom specific to $T_{\geq n}$, then φ' has a proof in G of a length c, where c depends only on n.
Because every line in π is by such a translation represented by c_1 lines maximum c_2-times longer then the original line, where c_1, c_2 are constants depending only on k, $l(\pi') \leq c_1 c_2 l(\pi)$. $\square$

PROPOSITION. 6.9 $G \geq_p T_\infty$.
PROOF. Infinite number of all possible x's cannot be encoded by a finite number of proposional variables. However, any particular derivation π can use only a finite number of axioms of theory T_∞. Therefore there is some maximum n such that the axiom $\exists x_1 \ldots \exists x_n (\bigwedge_{1 \leq i < j \leq n} (x_i \neq x_j))$ appears in π. Note that n is bounded by the length of π. So π is also a derivation of $T_{\geq n}$ (or can be polynomially simulated, if shorter axioms of T_∞ appeared in π also).
Every predicate variable in π can be again encoded by a system of k propositional variables, like in Lemma 6.8, where now

$$k = \lceil \log_2(n) \rceil \leq \lceil \log_2(l(\pi)) \rceil$$

The main difficulty now is that n, (and k) depend on π and not only on the theory like in the previous proof. Theory T_∞ has infinitely many axioms, and it is not known which of them will appear in π. So we have to show that the simulation of axioms and their derivations in π' does not depend exponentially on n. It can be of the length $O(n^c)$ or $O(c^k)$, because $k = \lceil \log_2(n) \rceil$), but not of the length $O(n^k)$ or more because then it would grow faster than any polynomial.

If φ was derived by a propositional rule in π, φ' can be derived by the same rule, like in the last argument. Also, if it was derived by some quantifier rule, φ' has a derivation of k steps in π' as shown in the last proof.
But if φ is an axiom of identity or an axiom specific to T_∞, we have to show that φ' has a short proof in G. The axioms of identity here are $E1$, $E2$ and $E3$. We will show the proof of $E3'$, the other two cases are similar:

First, for every $i = 1 \dots k$, derive

$$p_i \equiv q_i \wedge q_i \equiv r_i \implies p_i \equiv r_i$$

by a proof of length c_1. From these sequents derive

$$\bar{p} = \bar{q} \wedge \bar{q} = \bar{r} \implies p_i \equiv r_i$$

by $2k$-times using $(\wedge l)$, and then connect these sequents into

$$\bar{p} = \bar{q} \wedge \bar{q} = \bar{r} \Longrightarrow \bar{p} = \bar{r}$$

by k-times using $(\wedge r)$. This sequent has length $6k$. Now derive

$$\Longrightarrow \bar{p} = \bar{q} \wedge \bar{q} = \bar{r} \to \bar{p} = \bar{r}$$

by $(\to r)$, and by $\leq 3k$-times using $(\forall r)$ derive

$$\Longrightarrow \forall \bar{p}, \bar{q}, \bar{r}(\bar{p} = \bar{q} \wedge \bar{q} = \bar{r} \to \bar{p} = \bar{r})$$

the required sequent, $9k$ long. So this proof has length
$\leq ck + 2k(4k + 2) + k6k + 6k + 3k9k = O(k^2)$.

If φ in π was an axiom of T_∞, then a proof of φ' in G will look like this: From $(0 \equiv 1) \Rightarrow$ or from $(1 \equiv 0) \Rightarrow$ using twice $(\wedge l)$ derive

$$\tilde{c_i} = \tilde{c_j} \Longrightarrow$$

for every i,j, $1 \leq i < j \leq n$, where $\tilde{c_i}$'s are n distinct k-tuples of 0,1, each $\tilde{c_i}$ representing the binary code of the natural number i. There is $\frac{n^2}{2} - n = O(n^2)$ of such pairs. From them derive, using the rule $(\neg r)$, sequents:

$$\Longrightarrow \neg(\tilde{c_i} = \tilde{c_j})$$

and by using maximum $O(n^2)$-times inferences ($\wedge r$) connect them into one sequent

$$\Longrightarrow \bigwedge_{i<j} \neg(\tilde{c_i} = \tilde{c_j})$$

which is $O(kn^2)$ long. In the last step use kn-times the rule ($\exists r$) to derive the final formula

$$\Longrightarrow \exists \bar{p_1} \ldots \exists \bar{p_n} (\bigwedge_{i \leq j} \neg(\bar{p_i} = \bar{p_j}))$$

which is $kn + O(kn^2) = O(kn^2)$ long.

So the length of the whole proof is less then
$O(kn^2) + O(kn^2) + O(n^2)O(kn^2) + knO(kn^2) \leq O(n^5)$ long. □

COROLLARY. 6.10 *G, $T_{0\neq 1}$, $T_{\geq n}$, T_∞ are p-euivalent.*
PROOF. By putting together Propositions 6.5, 6.6, and 6.9, and using the transitivity of the p-simulation. □

7 Theories extending T_∞

In this chapter we will search for theories extending $T_{\leq n}$ and T_∞ but that are still p-equivalent to these. Our aim is to find a very general description of some such theories.

DEFINITION. 7.1 *Theory of exactly n elements - T_n is an extension of $PL_=$. In addition it has constants $c_1, \ldots, c_n$ and two axioms:*

$$\bigwedge_{1 \leq i < j \leq n} (c_i \neq c_j)$$

$$\forall x((x = c_1) \vee \ldots \vee (x = c_n))$$

PROPOSITION. 7.2 *If T is a first-order theory with a finite number of axioms, and has a model of size n, then $T_n \geq_l T$.*

PROOF. In a concrete model M of T of size n, any predicate P of theory T can be represented by a disjunctive normal form formula table listing all possible k-tuples of elements satysfying P in M:

$$\begin{aligned} P(x_1, \ldots, x_k) \quad \equiv \quad & ((x_1 = c_1^1 \wedge \ldots \wedge x_k = c_k^1) \vee \\ & \vdots \\ & (x_1 = c_1^l \wedge \ldots \wedge x_k = c_k^l)) \end{aligned}$$

where $l \leq n^k$. Because the axioms of T contain only a finite number of predicates, these predicates are of arity at most k, for some natural number k. Thus the tables representing them are maximum of size $O(n^k)$. The predicates that do not appear in the axioms of T can be represented by some fixed small table.
If we substitute the predicates in axioms of T by their tables determined by some concrete model M of size n, we will receive sentences true in T_n: This just restates the fact that M models T. By the completeness, these sentences have in T_n proofs of constant lengths (depending on n but n is fixed). Let c be the total length of these proofs. And if we make such a substitution in a derivation π in T and add the proofs of the translated axioms at the beginning, we will receive a derivation π' simulating π in T_n. Because the formulas after the substitution are maximum $O(n^k)$ times longer, $l(\pi') \leq l(\pi)O(n^k) + c \leq O(l(\pi))$. □

The following Lemmas will be used in the proof of Proposition 7.9 which is a stronger version of Proposition 7.2.

LEMMA. 7.3 *The sequent* $x_1 = y_1, \ldots, x_n = y_n, \varphi(\bar{x}) \Longrightarrow \varphi(\bar{y})$ *has a proof in* $PL_=$ *with the length polynomially equal to* $l(\varphi)$. *Simillary* $p_1 \equiv q_1, \ldots, p_n \equiv q_n, \varphi(\bar{p}) \Longrightarrow \varphi(\bar{q})$ *has a proof of length polynomially equal to* $l(\varphi)$ *in* G.

PROOF. In this proof the string $\bar{x} = \bar{y}$ denotes the conjunction $x_1 = y_1 \wedge \ldots \wedge x_n = y_n$. However the same proof would work also for the calculus G, where the string $\bar{p} = \bar{q}$ would denote the formula $p_1 \equiv q_1 \wedge \ldots \wedge p_n \equiv q_n$.
The derivation is constructed by the induction according to the complexity of the formula φ:
If φ is an atomic formula made of predicate P, then

$$\begin{aligned} \bar{x} = \bar{y}, P(\bar{x}) &\Longrightarrow P(\bar{y}) \\ \bar{x} = \bar{y}, P(\bar{y}) &\Longrightarrow P(\bar{x}) \end{aligned}$$

are simply derivable sequents from the axiom of identity $(E4)$.
If φ is of the form $\neg\psi$, by induction assumption the lemma is true for ψ. By $\neg r$, $\neg l$ derive:

$$\begin{aligned} \bar{x} = \bar{y}, \psi(\bar{x}) &\Longrightarrow \psi(\bar{y}) \\ \bar{x} = \bar{y}, \neg\psi(\bar{x}) &\Longrightarrow \neg\psi(\bar{y}) \end{aligned}$$

If φ is of the form $\psi_1 \wedge \psi_2$, then apply rules $\wedge l$ and $\wedge r$ as follows:

$$\begin{aligned} \bar{x} = \bar{y}, \psi_1(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \\ \bar{x} = \bar{y}, \psi_2(\bar{x}) &\Longrightarrow \psi_2(\bar{y}) \\ \bar{x} = \bar{y}, \psi_1(\bar{x}) \wedge \psi_2(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \\ \bar{x} = \bar{y}, \psi_1(\bar{x}) \wedge \psi_2(\bar{x}) &\Longrightarrow \psi_2(\bar{y}) \\ \bar{x} = \bar{y}, \psi_1(\bar{x}) \wedge \psi_2(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \wedge \psi_2(\bar{y}) \end{aligned}$$

Similarly, if φ is of the form $\psi_1 \vee \psi_2$, use $\vee r$ and $\vee l$:

$$\begin{aligned} \bar{x} = \bar{y}, \psi_1(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \\ \bar{x} = \bar{y}, \psi_2(\bar{x}) &\Longrightarrow \psi_2(\bar{y}) \\ \bar{x} = \bar{y}, \psi_1(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \vee \psi_2(\bar{y}) \\ \bar{x} = \bar{y}, \psi_2(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \vee \psi_2(\bar{y}) \\ \bar{x} = \bar{y}, \psi_1(\bar{x}) \vee \psi_2(\bar{x}) &\Longrightarrow \psi_1(\bar{y}) \vee \psi_2(\bar{y}) \end{aligned}$$

If φ begins with the existential quantifier $(\exists z\psi)$, use the quantifier rules $\exists r$ and $\exists l$:

$$\begin{aligned} \bar{x} = \bar{y}, \psi(\bar{x}) &\implies \psi(\bar{y}) \\ \bar{x} = \bar{y}, \psi(\bar{x}) &\implies \exists z\psi(\bar{y}) \\ \bar{x} = \bar{y}, \exists z\psi(\bar{x}) &\implies \exists z\psi(\bar{y}) \end{aligned}$$

For $\varphi = \forall z\psi$ similarly first use the rule $\forall l$, and then $\forall r$.
Every induction step is done by a constant number of lines and the induction proceeds for $O(l(\varphi))$ steps, so the derivation is done in $O(l(\varphi))$ lines. Because the lengths of the lines are also proportional to $l(\varphi)$ the derivation has the length $O(l(\varphi)^2)$. □

LEMMA. 7.4 *If φ does not have quantifiers and all atomic subformulas of φ can be proved or (dis)proved in a derivation of maximum length k, then φ can be proved or (dis)proved in a derivation proportional to $O(l(\varphi)^2)$.*

PROOF. By induction on the complexity of φ construct the derivation of the formula from its atomic subformulas. While all the induction steps take a constant number of lines of length $\leq l(\varphi)$, the whole derivation is proportional to $O(l(\varphi)^2)$. □

LEMMA. 7.5 *If for all choices $c_{i_1}, \ldots, c_{i_k}$ of constants of T_n formula $\varphi(c_{i_1}, \ldots, c_{i_k})$ can be proved in T_n by a derivation of length l in T_n, then $\forall x_1 \ldots \forall x_k \varphi(x_1, \ldots, x_k)$ can be proved in T_n by a derivation of length $O(p(l(\varphi)) \cdot n^k)$.*
This also holds for the case that there are $\exists$-quantifiers for some other variables of φ in between the $\forall$-quantifiers.

PROOF. First derive n^k sequents, one for each k-tuple of constants:

$$\begin{gathered} x_1 = c_1, \ldots, x_k = c_1 \Longrightarrow \varphi(x_1, \ldots, x_k) \\ \vdots \\ x_1 = c_n, \ldots, x_k = c_n \Longrightarrow \varphi(x_1, \ldots, x_k) \end{gathered}$$

each in $p(l(\varphi))$ steps by Lemma 7.3, for some polynomial p. In this we use the proofs of all $\varphi(c_{i_1}, \ldots, c_{i_k})$ that exists by hypotheses of the Lemma. Then connect them by $\vee l$ into one sequent

$$x_1 = c_1 \vee \ldots \vee x_1 = c_n, \ \cdots \ , x_k = c_1 \vee \ldots \vee x_k = c_n \implies \varphi(x_1, \ldots, x_k)$$

which is $kn + l(\varphi)$ long. Then by k-times $\forall l$, CUT with the axiom of T_n and k-times $\forall r$, derive

$$\begin{aligned} \forall x(x = c_1 \vee \ldots \vee x = c_n) &\implies \varphi(x_1, \ldots, x_k) \\ &\implies \varphi(x_1, \ldots, x_k) \\ &\implies \forall x_1, \ldots, x_k \varphi(x_1, \ldots, x_k) \end{aligned}$$

If $\exists$-quantifiers are also included, derive the initial sequents with the right constants. Then in the second step do the connecting by $\vee l$ in order from x_k to x_1 and where needed, use the $\forall l$, CUT, $\forall r$ rules followed by $\exists r$ rule on the right constants to put $\exists$-quantifiers on the right places.
Counting all the lines together, this derivation is $O(p(l(\varphi)) \cdot n^k)$ long. □

LEMMA. 7.6 *There is a polynomial $p(x)$, such that for all n, m, $n \leq m$, there is a p-simulation $\pi \to \pi'$ of T_n proofs by T_m proofs such that $l(\pi') \leq p(l(\pi))$.*

PROOF. From a derivation π in T_n we will obtain a derivation π' in T_m by substituting atomic subformulas of form $x = y$ by the formulas $x \sim y$:

$$x = y \vee ((x = c_n \vee \ldots \vee x = c_m) \wedge (y = c_n \vee \ldots \vee y = c_m))$$

The two axioms of T_n and first three axioms of identity ($E1$,$E2$,$E3$) translated in this way are true in T_m and by Lemma 7.5 can be proved in a derivation polynomially equal to m. Thus by applying such substitution on the whole derivation π and adding the proofs of the translated axioms, we will obtain polynomially simulating derivation π' in T_m. □

LEMMA. 7.7 *There is a polynomial $p(x)$, such that for all $n = 2^k$, where k is a natural number, there is a p-simulation $\pi \to \pi'$ of T_n by G such that $l(\pi') \leq p(l(\pi))$.*

PROOF. This proof is an extension of a proof of Proposition 6.9 ($G \geq_p T_\infty$). The encoding of predicate $=$ and proving the translation of axiom

$$\bigwedge_{1 \leq i < j \leq n} (c_i \neq c_j)$$

would be done in the same way. Now we only have to show that the proof of the translation of the second axiom

$$\forall x((x = c_1) \vee \ldots \vee (x = c_n))$$

is of the length polynomial in n:
The notation $\tilde{n}$ will stand for the binary code of natural number n. From simply derivable sequents

$$\begin{aligned} p_i &\implies p_i \equiv 1 \\ \neg p_i &\implies p_i \equiv 0 \end{aligned}$$

for $i = 1 \dots k$, derive n sequents

$$\begin{aligned} \neg p_1, \dots, \neg p_k &\implies \bar{p} = \tilde{0} \\ &\vdots \\ p_1, \dots, p_k &\implies \bar{p} = \tilde{n} \end{aligned}$$

by k-times using $\wedge r$. Then by applying $\vee r$ on each of them derive

$$\begin{aligned} \neg p_1, \dots, \neg p_k &\implies \bar{p} = \tilde{0} \vee \dots \vee \bar{p} = \tilde{n} \\ &\vdots \\ p_1, \dots, p_k &\implies \bar{p} = \tilde{0} \vee \dots \vee \bar{p} = \tilde{n} \end{aligned}$$

In these sequents there are couples that differ only in one of the literals p_i in the antecedent of the sequents. Connect all such couples by rule $\vee l$ applied on the different literals into $n/2$ sequents. The resulting sequents again contain such couples differing in only one literal. So apply this method again and again until you get one sequent

$$(p_1 \vee \neg p_1), \dots, (p_k \vee \neg p_k) \implies \bar{p} = \tilde{0} \vee \dots \vee \bar{p} = \tilde{n}$$

which is $2k + 2nk = O(nk)$ long. This can be done in $n - 1 = O(n)$ lines. The last step is n CUTs with the simply derivable sequents $\implies p_i \vee \neg p_i$ and k-times $\forall r$ to get the final sequent

$$\begin{aligned} &\implies \bar{p} = \tilde{0} \vee \dots \vee \bar{p} = \tilde{n} \\ &\implies \forall \bar{p}(\bar{p} = \tilde{0} \vee \dots \vee \bar{p} = \tilde{n}) \end{aligned}$$

Counting all the steps together this proof is $O(kn^2)$ long. □

Definition. 7.8 *Theory T is* weak *iff there exists constants k, m and an integer polynomial p such that:*
- T has predicates of arity at most k in its language

- every axiom of T has at most m universal quantifiers when written in prenex form
- every finite part F of T has a model of size at most $p(l(F))$ that can be found by a polynomial time algorithm (receiving F as the input).

All theories that were defined in the previous text ($T_{\leq n}$, T_n, T_∞) fit this definition.

PROPOSITION. 7.9 *Let T be a weak theory, then $G \geq_p T$.*

PROOF. Let T, k, m, p be as in Definition 7.8. Let π be a derivation in T. Then π contains a finite number of axioms, which, according to the definition of a weak theory, have a model of size $n \leq p(|\pi|)$. Thus, similary as in the proof of Proposition 7.2, the predicates in formulas in π can be substituted for by tables of size $O(n^k)$.
Further, every axiom φ of T translated in this way, can be proved in T_n in a derivation of the length maximum $O(l(\varphi')n^m)$ by Lemma 7.5. Such a derivation has the length at most $O(p(l(\pi))^{k+m})$. Further, by Lemma 7.6, the derivation can be p-simulated in $T_{n'}$ for the first $n' \geq n$ that is some power of 2. Finally, a derivation in $T_{n'}$ can be polynomially simulated by G by Lemma 7.7 □

8 Theories with "fast growing models"

In this chapter we will examine theories that do not satisfy the premises of Proposition 7.9, i.e are not weak. However, some of them can still be p-simulated by G using different ways than the table method.
In the following example we will introduce a theory T_{exp} which has the property that the smallest models of its finite subtheories are growing exponentially with the size of the subtheories. This theory cannot be p-simulated by G using the table method, because the formula tables, used for substituting for the predicate symbols, would be too large. However, it can still be p-simulated by G by using a more sophisticated substitution for the predicate symbols.
Let us now give an example of a theory with exponentialy growing models of its subtheories. The theory defines a set P by an unary predicate symbol P and has an infinite system of axioms. Every finite part F of these axioms enforces that the set P has size proportional to $l(F)$, similarly to the axioms of T_∞. Further it has a finite number of axioms implying, that the size of the whole model is at least exponentially greater then the set P. This is done using three binary relations S, Q and R, schematically illustrated on Figure 1.

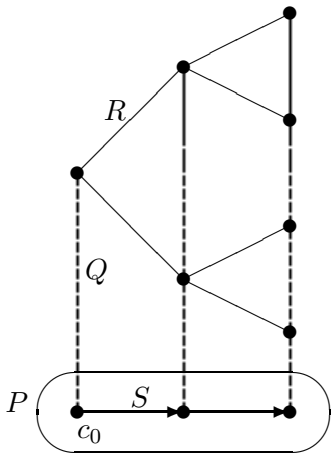

Figure 1: Model of subtheory of T_{exp}

Relation S is creating a linearly ordered string on the whole set P starting in some c_0 of P. Relation Q is mapping elements of the model to the

elements of the set P. There is at least one element mapped by Q to c_0. About the relation R the theory says, that it is injective and if an element c_i of P has a successor c_{i+1} in the relation S on P, then every other element, mapped to c_i by relation Q, has at least two different successor in the relation R and these successor are mapped by Q to c_{i+1}. So for every c_i of P, if c_i is being successor by n elements of P in the string S, then every other element mapped to c_i by Q has at least $2^n - 1$ followers in the relation R (R creates a binary tree of depth n). And because the string S starts at c_0 and it is covering the whole set P, the whole model must be of size at least exponential size of the size of P, i.e. of $l(F)$.
Now formally:

DEFINITION. 8.1 *Theory T_{exp} is an extension of $PL_=$. It has three binary predicates S, Q, R and one constant c_0.*
A binary relation $S(x, y)$ is defined on the set P by an infinite system of axioms:

$$s_n: \quad \exists x_1 \ldots \exists x_n \quad (x_1 = c_0 \ \wedge \bigwedge_{i=1\ldots n} P(x_i) \ \wedge \bigwedge_{1 \le i < j \le n} (x_i \neq x_j) \ \wedge \\ S(x_1, x_2) \wedge \ldots \wedge S(x_{n-1}, x_n))$$

For every natural number n, s_n is saying that S creates a string on P starting with c_0 and having length n.
The axiom

$$r1: \ \exists y Q(c_0, y)$$

is saying that there is at least one element mapped by the relation Q to c_0. Then there are axioms for the relation R, implying that R is injective:

$$r2: \ \forall x_1, x_2, y (R(x_1, y) \wedge R(x_2, y) \longrightarrow x_1 = x_2)$$

and that if x_1 has a successor x_2 in the relation S, then every y mapped to x_1 by Q has at least two different successors (y_1,y_2) in the relation R that are mapped to x_2 by Q:

$$r3: \quad \forall x_1, x_2, y (\, P(x_1) \wedge P(x_2) \wedge (S(x_1, x_2) \wedge Q(x_1, y)) \ \longrightarrow \\ \exists y_1, y_2 (R(y, y_1) \wedge R(y, y_2) \wedge Q(x_2, y_1) \wedge Q(x_2, y_2) \wedge y_1 \neq y_2) \,)$$

From these axioms it follows that elements of the string S, starting with c_0 and of the length n, have together at least $2^n - 1$ elements mapped to them by Q. So the size of the model of subtheory F of T containing axioms $r1$, $r2$ and $r3$, is at least $2^n - 1$, where n is the maximum from axioms s_n appearing in the subtheory.

Similarly, simply by changing the axiom $r3$, we can define theories with the models of their subtheories growing by c^n, for any fixed natural number $c \geq 2$. Such theories cannot be p-simulated in G by using the table method, because it would have to use exponentially many constants representing the elements in the formula tables. However, in G we can represent the elements by binary codes, using only $\lceil \log_2 c^n \rceil \leq n \lceil \log_2 c \rceil$ propositional variables.
To show a p-simulation, we will only have to find 'short' formulas for substituting the predicates and 'short' proofs of formulas resulting from the axioms after the substitution. This method of p-simulation is illustrated by the following proposition.

PROPOSITION. 8.2 $G \geq_p T_{exp}$

PROOF. Let π be a derivation in T_{exp}. Let k be the maximal n from the s_n axioms appearing in π. Then π is also a derivation in the subtheory F of T_{exp}, containing the axioms $r1$, $r2$, $r3$ and the axioms s_i for $i = 1 \dots k$. As argued above, F has a model M of size 2^k. To p-simulate π in G, we will encode the elements of this model by binary codes into k-tuples of propositional constants. Then we will find formulas P', S', Q' and R' in G representing the relations P, S, Q and R in M using these binary codes. These formulas will be used for substituting the predicates in π similarly as the formula tables were substituting predicates in the proof of Proposition 7.9. The important thing will be that the length of these formulas will not grow exponentially with k and further that the formulas obtained by thr translation of the axioms of T_{exp} will have proofs of the length polynomial in k. So the encoding of the elements of M should not be done just randomly, but in a way helpful for writing simple formulas P', S', Q' and R'.
One possible encoding, that we will use in this proof, is illustrated by the Figure 2.
The constant c_0 will be substituted by the code $\tilde{1}$ (i.e. 001 for $k = 3$). The

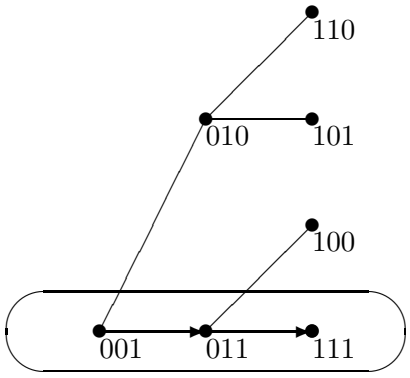

Figure 2: Encoding of elements of a model of subtheory of T_{exp} for $k = 3$. (Code 001 means: $p_3 = 0$, $p_2 = 0$, $p_1 = 1$)

formulas substituting for the predicates will have the following form:

$$\begin{aligned} P'(\bar{p}) &\equiv \bigwedge_{i=2\ldots k} (p_i \to p_{i-1}) \\ S'(\bar{p},\bar{q}) &\equiv \bigwedge_{i=2\ldots k} (p_{i-1} \equiv q_i) \wedge \neg p_k \wedge q_1 \\ Q'(\bar{p},\bar{q}) &\equiv \bigwedge_{i=1\ldots k} (q_i \to p_i) \\ R'(\bar{p},\bar{q}) &\equiv \bigwedge_{i=2\ldots k} (p_{i-1} \equiv q_i) \wedge \neg p_k \end{aligned}$$

It is easy to see that these formulas are of the length proportional to k. The last thing to show is that the formulas, obtained by translating the axioms, have G-proofs of the length polynomially equal to k.
(It would be nice to introduce some general proposition for a simulation by binary encoding, similar to Proposition 7.9. But although Lemmas 7.4 and 7.5 can be generalized also for the calculus G, it does not help much, because the number of universal quantifiers in the axioms, translated by binary encoding to G, grows proportionally with k. So the length of the general proofs, introduced by Lemma 7.5, would grow exponentially with k. Thus we have to find short proofs for the specific translations of axioms "manually".)

The proofs for translated axioms of identity $E1$, $E2$ and $E3$ where shown in Proposition 6.9. The axioms of identity for predicates ($E4$) can be simulated in polynomial length according to Lemma 7.3, which is true also for the calculus G.
The formulas s'_i $(i = 1 \dots k)$ do not contain any universal quantifiers or free variables and thus can be proved by a derivation of the length proportional to $l(s'_i)$. The same applies to the formula $r1'$.
For proving $r2'$, first derive k simple sequents

$$\begin{array}{rcl} p_1^1 \equiv q_2, p_1^2 \equiv q_2 & \Longrightarrow & p_1^1 \equiv p_1^2 \\ & \vdots & \\ p_{k-1}^1 \equiv q_k, p_{k-1}^2 \equiv q_k & \Longrightarrow & p_{k-1}^1 \equiv p_{k-1}^2 \\ \neg p_k^1, \neg p_k^2 & \Longrightarrow & p_k^1, \equiv p_k^2 \end{array}$$

and connect them using $\wedge l$, $\wedge r$, and use $\rightarrow r$ and $3k$-times $\forall r$ to get:

$$\begin{array}{rcl} S'(\overline{p}^1, \bar{q}) \wedge S'(\overline{p}^2, \bar{q}) & \Longrightarrow & \bar{p}^1 = \bar{p}^2 \\ & \Longrightarrow & (S'(\overline{p}^1, \bar{q}) \wedge S'(\overline{p}^2, \bar{q})) \rightarrow \bar{p}^1 = \bar{p}^2 \\ & \Longrightarrow & \forall \overline{p}^1 \forall \overline{p}^2 \forall \overline{q}((S'(\overline{p}^1, \bar{q}) \wedge S'(\overline{p}^2, \bar{q})) \rightarrow \bar{p}^1 = \bar{p}^2) \end{array}$$

This derivations takes $O(k)$ lines of the length at most $O(k)$, so together it is $O(k^2)$ long.
Finally for the longest formula $r3'$:
From the conjunction of formulas $P'(\bar{p}^1)$, $P'(\bar{p}^2)$, $S'(\bar{p}^1, \bar{p}^2)$ and $Q'(\bar{p}^1, \bar{q})$ we will derive formulas $Q'(\bar{p}^2, \bar{A})$, $Q'(\bar{p}^2, \bar{B})$, $R'(\bar{q}, \bar{A})$, $R'(\bar{q}, \bar{B})$ and $\bar{A} \neq \bar{B}$, where $\bar{A}$, $\bar{B}$ are tuples of some specific formulas which will be in the last step of the derivation substituted for by variables $\bar{r}^1$, $\bar{r}^2$ by applying the rule $\exists r$.
One way how to choose simply these formulas, as it is seen from the picture, is by using the variables $\bar{q}$ and the definition of the formula R':

$$\begin{array}{rl} \bar{A}: & q_{k-1}, \dots, q_1, 0 \\ \bar{B}: & q_{k-1}, \dots, q_1, 1 \end{array}$$

To derive $\bar{A} \neq \bar{B}$, apply to sequent $0 \equiv 1 \Longrightarrow$ rules $\wedge l$ and $\neg r$:

$(q_{k-1} \equiv q_{k-1} \wedge \dots \wedge q_1 \equiv q_1 \wedge 0 \equiv 1) \quad \Longrightarrow$

$$\begin{aligned} &\Longrightarrow \quad \neg(q_{k-1} \equiv q_{k-1} \wedge \ldots \wedge q_1 \equiv q_1 \wedge 0 \equiv 1) \\ &\Longrightarrow \quad \neg(A_k \equiv B_k \wedge \ldots \wedge A_1 \equiv B_1) \qquad (2) \end{aligned}$$

Deriving formulas $R(\bar{q}, \bar{A})$, $R(\bar{q}, \bar{B})$ is also simple, because it follows from the definition:

$$\begin{aligned} & \Longrightarrow \quad q_{k-1} \equiv q_{k-1} \wedge \ldots \wedge q_1 \equiv q_1 \\ \neg p_k^1, q_k \to p_k^1 \quad & \Longrightarrow \quad \neg q_k \\ \neg p_k^1, q_k \to p_k^1 \quad & \Longrightarrow \quad q_{k-1} \equiv A_k \wedge \ldots \wedge q_1 \equiv A_2 \wedge \neg q_k \qquad (3) \\ \neg p_k^1, q_k \to p_k^1 \quad & \Longrightarrow \quad q_{k-1} \equiv B_k \wedge \ldots \wedge q_1 \equiv B_2 \wedge \neg q_k \qquad (4) \end{aligned}$$

The formulas in the antecedent ($\neg p_k^1$, $q_k \to p_k^1$) are part of the conjunctions $S(\bar{p}^1, \bar{p}^2)$ and $Q(\bar{p}^1, \bar{q})$.
For deriving $Q(\bar{p}^2, \bar{A})$, $Q(\bar{p}^2, \bar{B})$, derive two simple sequents

$$\begin{aligned} p_1^2 \quad & \Longrightarrow \quad A_1 \to p_1^2 \\ p_1^2 \quad & \Longrightarrow \quad B_1 \to p_1^2 \end{aligned}$$

where p_1^2 is from $S(\bar{p}^1, \bar{p}^2)$, and for $i = 2, \ldots, k$ derive sequents

$$\begin{aligned} A_i \equiv q_{i-1}, q_{i-1} \to p_{i-1}^1, p_{i-1}^1 \equiv p_i^2 \quad & \Longrightarrow \quad A_i \to p_i^2 \\ B_i \equiv q_{i-1}, q_{i-1} \to p_{i-1}^1, p_{i-1}^1 \equiv p_i^2 \quad & \Longrightarrow \quad B_i \to p_i^2 \end{aligned}$$

where $A_i \equiv q_{i-1}$ (or $B_i \equiv q_{i-1}$) is by definition of A_i (or B_i), $q_{i-1} \to p_{i-1}^1$ is from $Q(\bar{p}^1, \bar{q})$ and $p_{i-1}^1 \equiv p_i^2$ from $S(\bar{p}^1, \bar{p}^2)$, and connect them into

$$\begin{aligned} Q'(\bar{p}^1, \bar{q}), S'(\bar{p}^1, \bar{p}^2) \quad & \Longrightarrow \quad A_1 \to p_1^2 \wedge \ldots \wedge A_k \to p_k^2 \qquad (5) \\ Q'(\bar{p}^1, \bar{q}), S'(\bar{p}^1, \bar{p}^2) \quad & \Longrightarrow \quad B_1 \to p_1^2 \wedge \ldots \wedge B_k \to p_k^2 \qquad (6) \end{aligned}$$

Then connect sequents $(1), \ldots, (6)$ into

$$S'(\bar{p}^1, \bar{p}^2) \wedge Q'(\bar{p}^1, \bar{q}) \quad \Longrightarrow \quad Q'(\bar{p}^2, \bar{A}) \wedge Q'(\bar{p}^2, \bar{B}) \wedge R(\bar{q}, \bar{A}) \wedge R(\bar{q}, \bar{B}) \wedge \bar{A} \neq \bar{B}$$

and use $2k$-times $\exists r$ on the subformulas $\bar{A}$, $\bar{B}$:

$$\begin{aligned} &P'(\bar{p}^1) \wedge P'(\bar{p}^2) \wedge S'(\bar{p}^1, \bar{p}^2) \wedge Q'(\bar{p}^1, \bar{q}) \quad \Longrightarrow \\ &\exists \bar{r}^1 \exists \bar{r}^2 Q'(\bar{p}^2, \bar{r}^1) \wedge Q'(\bar{p}^2, \bar{r}^2) \wedge R'(\bar{q}, \bar{r}^1) \wedge R'(\bar{q}, \bar{r}^2) \wedge \bar{r}^1 \neq \bar{r}^2 \end{aligned}$$

Finally use $\to r$ and $3k$-times $\forall r$ to get the wanted sequent.
Again there are $O(k)$ lines of the length at most $O(k)$, so the derivation is $O(k^2)$ long. $\Box$

Theory T_{exp} has exponentially growing smallest models of its finite subtheories. But it can still be p-simulated by G using a binary encoding of its variables. The next example will introduce a theory in which the smallest models of its finite subtheories are growing by c^{c^n}, where n is the size of the subtheory and c is some constant.
We will create such a theory by extending theory T_{exp}. Axioms of this new theory will put all the elements of a model of T_{exp} into a set P_2. On the set P_2 will be created a string S_2. To make S_2 cover the whole P_2, we will first define a linear ordering by a binary relation $<$ on P_2 and than define S_2 on $<$. Further there will be axioms maping new elements of the model to the elements of S_2 by relations Q_2, R_2 analogously as in T_{exp}.

EXAMPLE. 8.3 Theory T_{2exp} is an extension of T_{exp}. Further it contains new predicate symbols: unary P_2 and binary $<$, S_2, Q_2, R_2 and axioms describing them. First axiom sais that all elements y, that have been maped to some x by relation Q (in model of T_{exp}, belong to set P_2:

$$\forall y(\exists x Q(x,y) \rightarrow P_2(y)$$

On P_2 exists a linear ordering starting in c_0 and all pairs of adjoining elements in the ordering are in relation S_2. So the relation S_2 creates a string starting with c_0 and covering the whole P_2:

$\forall P_2(x), P_2(y)(x < y \rightarrow \neg(y < x))$	antisymmetry of $<$
$\forall P_2(x), P_2(y), P_2(z)(x < y \wedge y < z \rightarrow \neg(x < z))$	transitivity of $<$
$\forall P_2(x), P_2(y)(x < y \vee y < x)$	linearity of $<$
$\forall x, y(S_2(x,y) \equiv (x < y \wedge \neg\exists P_2(z)(x < z < y)))$	S_2 is a string
$P_2(c_0) \wedge \forall P_2(x)\neg x < c_0$	c_0 is first in the string

The last three axioms are analogous to axioms $r1$,$r2$ and $r3$ of T_{exp}, but with the predicates P_2, S_2, Q_2, R_2 in the place of P, S, Q, R.

Now let F be a subtheory of T_{2exp}, containing axioms $r1$, $r2$ and $r3$ of T_{exp}, and all the special axioms of T_{2exp}. Let k be the maximum n of the s_n axioms appearing in F. Any model of F contains a set P_2 of size at least 2^n, by the same argument as with T_{exp} (In fact it is $2^n - 1$ but we can assume adding of one more element for convinience in the counting). And by the same argument the whole model of F is at least exponential to P_2, that is 2^{2^n}.

Similarly, by changing axiom $r3$ and its analogies, we can get theories with models of subtheories growing by c^{c^n} to the size of the subtheories, where c is a natural number. And by repeating the same method we can increase this growing to c^{c^n}, $c^{c^{c^n}}$, ...
These theories cannot be p-simulated by G using the binary coding method. Even T_{2exp} would need at least $\lceil \log_2(c^{c^n}) \rceil = O(c^n)$ propositional variables in binary encoding. So the translated formulas in the simulating derivations would grow exponentially in their length (with the size of the subtheories).

9 Further research

In the whole thesis we were stating theorems saying that some proof systems are p-equivalent (or in some cases we were able to show the p-simulation only one way). On the other hand we were not able to proof any theorem saying that some proof system cannot p-simulate some other proof system. Such problems are considered as difficult and there are no results saying that some proof systems are even stronger then Frege systems.
One way the research can continue is by examining stronger first-order theories as propositional proof system. From Chapter 6 we were examining theories with an infinite model, but with the finite subtheories having finite models. The next step would be to examine a theory with an infinite model implied by a finite number of axioms. For example, the theory of dense linear ordering.
Another natural way of creating possible strong proof systems would be by extending the system PL into a second-order logic.

10 Symbol index

p, q, r, p_0, p_1, ...	- propositional variables
x, y, z, x_0, x_1, ...	- first-order variables
A, B, C, ...	- propositional formulas
φ, ψ, ...	- first-order formulas or quantified propositional formulas
Γ, Δ, Λ, Π, ...	- sets of formulas
F, EF, G, $PL_{=}$, ...	- proof systems
π, π' ...	- derivations
t, s	- terms
c, c_0, ...	- constants

References

[CR] S. A. Cook and R. A. Reckhow, The relative efficiency of propositional proof systems, The Jornal of Symbolic Logic, vol. 44, pp. 36-50 (1977).

[C] S. A. Cook, The complexity of theorem proving procedures, Proceedings of the Third Annual ACM Symposium on the Theory of Computing, pp. 151-158 (1971).

[K] J. Krajíček, Cambridge University Press, Bounded Arithmetic, Propositional Logic, and Complexity Theory (1995).

[KP] J. Krajíček and P. Pudlák, Quantified propositional calculi and fragments of bounded arithmetics, Zeitschr. f. math. Logic und Grundlagen d. Math. Bd. 36., pp. 29-46 (1990).

[SV] V. Švejdar, Logika, neúplnost, složitost a nutnost (Logic: Incompleteness, Complexity, and Necessity). Academia Praha, 2002.