

Julius-Maximilians-Universität Würzburg  
Institut für Informatik  
Lehrstuhl für Informatik IV  
Theoretische Informatik

**Bachelor Thesis**

**simulation of proof systems**

Nils Wisiol

submitted on May 11, 2012

supervisor:  
Dr. Christian Glaßer

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Preliminaries</b>	<b>4</b>
<b>3</b>	<b>A set in <math>\text{co-NEXP} \setminus \text{OPT}</math></b>	<b>5</b>
<b>4</b>	<b>Conclusion and future work</b>	<b>6</b>
	<b>Bibliography</b>	<b>8</b>

# 1 Introduction

Welcome to my bachelor's thesis. Based upon work of Messner (1999).

## 2 Preliminaries

Let  $x$  be something.

**Definition 1** (Proof system). *A function  $f \in \mathcal{FP}$  is called proof system for a language  $L$  if the range of  $f$  is  $L$ .*

**Definition 2** (OPT). *Let OPT be the complexity class of all sets that have a  $p$ -optimal proof system.*

### 3 A set in $\text{co-NEXP} \setminus \text{OPT}$

In this section, we will show that there are sets without optimal proof systems.

**Theorem 1.** *Let  $t : \mathbb{N} \rightarrow \mathbb{N}$  be a time-constructible function such that for every polynomial  $p$  there is a number  $n$  with  $p(n) \leq t(n)$ . Then there is a language  $L \in \text{NTIME}(t(n))$  that has no  $p$ -optimal proof system.*

Jochen Messner showed in Messner (1999) that under the same presumptions as in our theorem, there is a language  $L \in \text{NTIME}(t(n))$  without an optimal acceptor. He also proofed that the existence of a optimal acceptor is equivalent to the existence of a optimal proof system for every  $p$ -cylinder  $L$ .

*Proof.* Let  $f_1, f_2, \dots$  be a enumeration of all  $\mathcal{FP}$ -functions with  $\text{time}(f_i) \leq n^i + i$ . For any  $i > 0$ , let  $L_i$  be the regular language described by the expression  $0^i 10^*$ . Define

How is that obtained?

$$L'_i = \{x \in L_i \mid \forall y \in \Sigma^* \mid y| \leq t(|x|) \implies f_i(y) \neq x\}.$$

That is, as long as you put proofs of polynomial length (relative to  $x$ ) into  $f_i$ , you will not get  $x$  out. Let  $L = \bigcup_{i>0} L'_i$ .

First, we obtain  $L \in \text{NTIME}(t(n))$ . For any given  $x$ , one can simulate for every  $y \in \Sigma^*$  up to  $t(|x|)$  steps of a deterministic machine calculating  $f_i(y)$ .

For a proof system  $f_i$  with  $f_i(\Sigma^*) = L$ , we observe that  $L'_i = L_i$ . Assume there is an  $L_i \ni x = 0^i 1z \notin L'_i$ . Then there is an  $y$  with  $|y| \leq t(|x|)$  and  $f_i(y) = x$ . Since  $f_i$  is a proof system for  $L$ , this yields  $x = 0^i 1z \in L$  and so  $x \in L'_i$ , which contradicts the assumption. Therefore, for any  $y$  with  $f_i(y) = x \in L_i$  we know that  $|y| > t(|x|)$ . Speaking informally, every proof system  $f_i$  for  $L$  is “slow” on  $L'_i \subset L$ .

Assume now, for contradiction, that  $f_i$  is a optimal proof system for  $L$ . Let  $g$  be a function defined as

$$g(bx) = \begin{cases} f(x) & (b = 0), \\ x & (b = 1 \text{ and } x = 0^i 10^* \in L_i). \end{cases}$$

$g$  is a proof system for  $L$  with polynomial length-bounded proofs for all  $x \in L_i$ . As  $f_i$  is optimal, there is a function  $p \in \mathcal{FP}$  such that for all  $x \in L'_i$ ,  $f_i(p(x)) = g(x)$ . But as  $p \in \mathcal{FP}$ ,  $|p(x)| =: y \leq q(|x|) \leq t(|x|)$  for a polynomial  $q$ . According to the definition of  $L'_i$ , this yields  $f_i(y) \neq x$ . Therefore,  $f_i$  is not optimal on  $L'_i$ , which contradicts the assumption that  $f_i$  is optimal on  $L$ .  $\square$

## 4 Conclusion and future work

What a great work!

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen Hilfsmittel und Quellen als die angegebenen benutzt habe. Weiterhin versichere ich, die Arbeit weder bisher noch gleichzeitig einer anderen Prüfungsbehörde vorgelegt zu haben.

Würzburg, den \_\_\_\_\_, \_\_\_\_\_  
(Nils Wisiol)

# Bibliography

Messner, J. (1999). On optimal algorithms and optimal proof systems. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science*, STACS'99, pages 541–550, Berlin, Heidelberg. Springer-Verlag.