Nonuniform proof systems: a new framework to describe nonuniform and probabilistic complexity classes

Jürgen Kämper*

Universität Oldenburg, Fachbereich Informatik, Postfach 2503, D 2900 Oldenburg, Germany

Communicated by R. Book
Received March 1988
Revised December 1988, March 1989

Abstract


The concept of nonuniform proof systems is introduced. This notion allows a uniform description of nonuniform complexity classes, probabilistic classes and language classes defined by simultaneous nonuniform and nondeterministic time bounds. Nonuniform proof systems provide a better understanding of many results concerning these classes; particularly, their connections to uniform complexity measures. We give a uniform approach to lowness results for various complexity classes. For instance, we show that co-NP/Poly ∩ NP is contained in the third level of the low hierarchy and that, NP ⊆ (NP ∩ co-NP)/Poly implies that the polynomial-time hierarchy collapses to its second level. Finally, we show that, beginning at the third level, levels of the low hierarchy cannot be extended to higher levels by the use of nonuniform information such as advice strings.

0. Introduction

Motivated by the still unsolved P–NP question, various models of polynomial-time computations have been investigated. Our work is centered on three of the most successful ones.

(1) Nonuniform complexity theory deals with functions limiting the growth rate of descriptions of the finite initial segments of languages. To formalize this complexity measure in a polynomial setting various concepts have been developed: circuits of polynomial size, polynomially time-bounded Turing machines with polynomial advice strings, and polynomial-time oracle Turing machines using sparse sets as their oracle languages. All these models describe the same language class, called P/Poly [11].

* Present address: IBM Deutschland, ENC, Tiergartenstraße 8, 6900 Heidelberg, Germany
Nonuniform complexity has become an important tool in complexity theory because of many results establishing connections with uniform complexity classes. Of particular interest are connections of the form: “If a fixed set $A$ has small nonuniform complexity, then some property holds for uniform Turing-machine complexity”; e.g. If NP is included in $P$/Poly, then the polynomial-time hierarchy collapses to its second level [11]. This result gives evidence that the nonuniform complexity measure $P$/Poly describes only a fragment of NP.

(2) In recent years many attempts have been made to attack intractable problems, for which no deterministic polynomial time algorithm is known, by probabilistic algorithms or by approximation algorithms. Probabilistic algorithms lead to classes such as BPP and R (for a broad discussion see [9, 16, 28, 32, 33]). Related to probabilistic complexity measures is the class AM defined by Arthur-Merlin games (with 2 rounds) [2, 3]. Approximation algorithms can be divided into “almost correct” algorithms leading to the class P-close [31] and “almost fast” algorithms leading to the class APT [18]. It can easily be seen that P-close and APT are subclasses of the nonuniform complexity class P/Poly (see e.g. [23]). By iterating probabilistic algorithms and using quantifier simulation techniques, connections between probabilistic complexity classes and nonuniform measures have also been established (see e.g. [25]).

(3) Chandra et al. [8] introduced alternating Turing machines as a generalization of nondeterministic Turing machines. In the case of nondeterministic machines, a single configuration $a$ can reach several configurations $b_1, b_2, \ldots, b_k$. The configuration $a$ leads to acceptance if at least one successor $b_i$ leads to acceptance. In addition to these “existential branches”, alternating machines can also make “universal branches”. Then, $a$ leads to acceptance if all the successors $b_i$ lead to acceptance. Polynomially time-bounded, alternating Turing machines accept the class PSPACE that contains all sets recognized by deterministic Turing machines using polynomial space. The power of this computational model can be restricted by allowing only a constant number of alternations (compare with [8]). Thereby, we get a characterization of the polynomial-time hierarchy PH: the $k$th levels $\Sigma^P_k$ and $\Pi^P_k$ contain all languages recognized by polynomially time-bounded, alternating Turing machines with $k$ alternations starting with an existential or a universal branch, respectively. Originally, Stockmeyer [29] defined the polynomial-time hierarchy using the concepts of polynomially length-bounded quantifiers and nondeterministic, polynomial-time oracle machines.

The low and high hierarchies from Schöning [20] reflect the polynomial-time hierarchy on NP-sets. A set $L$ in NP belongs to the $k$th level $L^P_k$ of the low hierarchy iff $\Sigma^P_k(L) \subseteq \Sigma^P_k$. $L$ belongs to the $k$th level $H^P_k$ of the high hierarchy iff $\Sigma^P_{n+1} \subseteq \Sigma^P_k(L)$. Thus, if $L$ is low, then with respect to the operator $\Sigma^P_k$, $L$ does not encode any information, but if $L$ is high, then $L$ encodes the power of an additional, polynomially length-bounded quantifier. It is easy to see that the polynomial-time hierarchy collapses iff there is a set which is both low and high. It has been shown that the restrictions of nonuniform classes ($P$/Poly), probabilistic classes (R, BPP) and language classes
defined by approximation algorithms (APT, P-Close) to NP-sets are included in some levels of the low hierarchy (for an overview see [23]). Therefore, these concepts describe only fragments of the class NP unless the polynomial-time hierarchy collapses. Recently, Schöning [24] has shown that the problem “Graph isomorphism” belongs to the second level of the low hierarchy.

Nonuniform proof systems are a new model that can be used to obtain the main results of the above-mentioned areas in a homogeneous manner. To explain this model we first formalize nonuniform complexity measures using the notion of advice strings [11]: Let $\Sigma$ be the fixed alphabet $\{0, 1\}$ and let $\langle \ldots \rangle$ be a pairing function. For a set $I$ and a string $u$ over $\Sigma$, $I_u$ denotes the language $I_u = \{ x \in \Sigma^* | \langle x, u \rangle \in I \}$. $I^{\leq n}$ consists of all strings from $I$ which are bounded in their length by $n$. For a language class $C$, let $C/Poly$ be the class of all sets $A$, for which there is a set $I \subset C$ and a polynomial $p$ such that for all $n \in \mathbb{N}$ there exists an advice string $u$ of length $p(n)$ with $A^{\leq n} = I_u^{\leq n}$. In other words, using a polynomially length-bounded advice string which depends only on the length of the considered string, the set $A$ can very easily be reduced to the set $I$, namely, by the pairing function $\langle \ldots \rangle$. The set $I$ interpretes the advice strings as descriptions of the finite initial subsets of $A$. Therefore, we call $I$ an interpretation set. The fundamental idea of this paper is to investigate sets of advice strings that describe the language $A$ correctly with respect to the interpretation set. Note that for any $n \in \mathbb{N}$ advice sets contain at least one advice string of appropriate length; namely, of length $p(n)$, where $p$ is a polynomial. Advice sets are called proof sets, and we will use them to investigate the complexity of computing correct advice strings. Moreover, advice sets are helpful in describing various complexity classes and in obtaining, in a uniform way, known and new results concerning certain complexity measures.

Without making these ideas precise at this stage, we introduce the following notation. For two language classes $C_1$ and $C_2$, $C_1$-$C_2$-$PSL$ (proof system language) denotes the class of all sets $A$ for which there is a proof system with an interpretation set from $C_1$ and a proof set from $C_2$.

In Section 1 we establish our notation, whereas Section 2 contains the exact definition of nonuniform proof systems and some easy results concerning this concept. It turns out that nonuniform complexity classes can be described in our terms by using proof sets of unlimited complexity, e.g. $P/Poly = P-P_2$-$PSL$, where $P_2$ denotes the class of all languages over $\Sigma$.

Section 3 locates languages from NP, for which there are proof systems with an interpretation set from co-NP and a proof set from the polynomial-time hierarchy, inside the low hierarchy ($(co-NP)$-$\Sigma^p_2$-$PSL \cap NP \subset L^p_2$). This main theorem is strong enough to prove that the class $co-NP/Poly \cap NP$ is included in the third level of the low hierarchy. Thereby, we improve results from Balcázar et al. [4, 6] and from Yap [30]. Later we will illustrate that all known lowness results can, in fact, be obtained by this theorem.

Two interesting types of nonuniform proof systems are investigated in Section 4. In a secure proof system with an interpretation set $I$, an advice string $u$ not belonging to
the proof set can lead to errors only in one direction: strings not belonging to the
considered language \( A \) are rejected by \( I \) when using \( u \) as its advice string, i.e.
\( I u^e \subseteq A^{\leq e} \). \( C_1-C_2-\text{SPSL} \) (secure proof system language) contains a language \( A \) iff
there is a secure proof system for \( A \) with an interpretation set from \( C_1 \) and a proof set
from \( C_2 \).

Dense proof systems are characterized by the property that their proof sets include
a majority of all advice strings of appropriate length \( (1/2 + \delta \text{, where } \delta \text{ is independent of the length of the advice strings}) \). We denote the class of all languages having a dense
proof system with an interpretation set from \( C_1 \) and a proof set from \( C_2 \) by
\( C_1-C_2-\text{DPSL} \) (dense proof system language). Classes characterized by proof systems
which are both dense and secure are denoted by an expression of the form \( C_1-C_2-\text{SDPSL} \).

We show that for secure proof systems the complexity of computing correct advice
strings can be bounded in the complexity of interpreting these advice strings, i.e.
\( C-P_2-\text{SPSL}=C-P_2^1(C)-\text{SPSL} \). Similarly, for all \( k \geq 1 \), \( \Pi_k^e-P_2-\text{SPSL} \cap \Sigma_k^p =
\Pi_k^e-P_2^1-\text{PSL} \cap \Sigma_k^p \). This proposition and our main theorem from Section 3 prove that
\( P-P_2-\text{SPSL} \) and also \( (\text{co-NP})-P-\text{SPSL} \cap \text{NP} \) are included in the second level of the
low hierarchy. We characterize the class \( P-P_2-\text{SPSL} \) by a restricted Turing-reducibility
to sparse sets and show that this class includes many known complexity classes, e.g.
\( \text{APT} \cap \text{NP} \) and \( \text{P-SELECTIVE} \cap \text{NP} \). Since \( P-P_2-\text{SPSL} \) belongs to the second level of
the low hierarchy, we obtain the known lowness results for these classes in our
framework. Moreover, we show that proof systems for disjunctive self-reducible
languages can always be transformed into secure proof systems \( (C_1-C_2-
\text{PSL} \cap \text{DSR} \subseteq P(C_1)-C_2-\text{SPSL} \), where \( \text{DSR} \) is the class of all disjunctive self-reducible
sets). We conclude that the polynomial-time hierarchy collapses to its second level if
the class \( (\text{NP} \cap \text{co-NP})/\text{Poly} \) contains \( \text{NP} \). This result, which was independently
proved by Abadi et al. [1], extends a theorem by Karp and Liptons [1] that if \( P/\text{Poly} \)
includes \( \text{NP} \), then the polynomial-time hierarchy collapses to its second level.

Dense proof systems are used to describe probabilistic complexity classes, e.g.
\( \text{BPP} = P-P_2-\text{DPSL} \), \( \text{AM} = \text{NP}-P_2-\text{DPSL} \), and proof systems which are both dense
and secure are used for the description of probabilistic classes with one-sided error,
e.g. \( \text{R} = P-P_2-\text{SDPSL} \). These characterizations require only the well-known tech-
niques of iterating probabilistic algorithms and quantifier simulation. Using these
descriptions of probabilistic complexity classes, we obtain new results, e.g.

\[(\forall k \geq 1): \text{ BP}(\Sigma_k^P \cap \Pi_k^P) \subseteq \Sigma_{k+1}^p \cap \Pi_{k+1}^p,\]
\[(\forall k \geq 0): \text{ BP}(\Sigma_k^P \cap \Pi_k^P) \supseteq \text{NP} \rightarrow R(\Sigma_k^P \cap \Pi_k^P) \cap \text{NP} \].

Moreover, we show that in some cases dense proof systems can be transformed into
secure proof systems (if \( \text{co-C} \) is closed under \( \text{NP}_{\text{pos}} \)-reductions, then \( C-P_2-\text{DPSL} \subseteq C-
P_2-\text{SDPSL} \)). From this result the lowness of some probabilistic classes follows.

As shown in Sections 3 and 4 all language classes known to belong to the low
hierarchy can be characterized by proof systems with interpretation sets from \( \text{co-NP} \).
Moreover, in all cases these classes could be located in one of the first three levels of the low hierarchy. In Section 5 we show that for $k \geq 3$ proof systems with an interpretation set from the $k$th level of the low hierarchy capture only NP-sets from the same level, i.e. $L_k^P \cap PSL \cap NP = L_k^P$. This shows that polynomial advice does not augment the power of the higher levels of the low hierarchy. Hence, it seems that nonuniform complexity measures cannot be used to extend one level of the low hierarchy to the next level beyond the third level of this hierarchy.

1. Preliminaries

It is assumed that the reader is familiar with the basic concepts of complexity theory. Here, only our notation is established. For basic definitions and elementary results see e.g. [7,23].

All the sets in this work are languages over the fixed alphabet $\Sigma = \{0,1\}$. For a string $w \in \Sigma^*$, let $|w|$ be its length. $e$ denotes the empty string. For a set $A$ and an $n > 0$, define $A^{\leq n} = \{x \in A \mid |x| \leq n\}$, $A^{\leq n} = \{x \in A \mid |x| = n\}$, and $A^{-} = \{x \in \Sigma^* \mid x \notin A\}$. $|S|$ denotes the cardinality of the set $S$. Let $P_2$ be the class of all languages over $\Sigma$. The join of two sets $A$ and $B$ is $A \oplus B = \{0x \mid x \in A\} \cup \{1x \mid x \in B\}$, and the join of two classes $C_1$ and $C_2$ is $C_1 \oplus C_2 = \{L \in P \mid (\exists L_1 \in C_1)(\exists L_2 \in C_2) \colon L = L_1 \oplus L_2\}$. For a class $C$ let $co-C$ be the class of complements of sets in $C$, i.e. $co-C = \{A \in P \mid A^{-} \in C\}$.

Let $L(T)$ be the set accepted by Turing machine $T$, and let $L(M, A)$ be the set accepted by oracle Turing machine $M$ when using $A$ as its oracle language. The classes of the polynomial-time hierarchy and their relativized versions are denoted as usual. Some more particular classes that we use are SPARSE, TALLY, APT [18] and P-SELECTIVE [26].

Definition 1.1. (i) A set $L \in P_\Sigma$ is sparse iff for some polynomial $p$ and each $n \equiv L^{\leq n} \leq p(n)$. SPARSE denotes the class of all sparse sets.

(ii) TALLY is the class of all languages over the one-letter alphabet {0}.

(iii) A set $L \in P_\Sigma$ is P-selective iff there is a 2-placed function $f$, computable in polynomial time, such that

$$\forall x, y \in \Sigma^* \colon [f(x, y) \in \{x, y\}] \& [x \in L \text{ or } y \in L \Rightarrow f(x, y) \in L].$$

(iv) A set $L \in P_\Sigma$ is in the class APT iff there is a deterministic Turing machine $T$ that accepts $L$, a polynomial $p$, and a sparse set $S$ such that $T$ runs for at most $p(|x|)$ steps for all $x \in S$.

$\leq^P_m$ ($\leq^P_\Sigma$) denotes polynomial-time many-one (Turing) reducibility. A language class $C$ is closed under a certain reducibility such as $\leq^P_m$ iff for all sets $A \in C$ and $B \leq^P_m A$ it holds that $B \in C$. Let $Q_k$ denote the quantifier $\exists$ if $k$ is odd, and the quantifier $\forall$ if $k$ is even. Let $\langle, \rangle$ be a pairing function. This function and its inverses should be computable in polynomial time. For all $k > 2$ and for all $y_1, y_2, \ldots, y_k \in \Sigma^*$ let
\begin{align*}
\langle y_1, y_2, \ldots, y_k \rangle & \text{ denote the string } \langle y_1, \langle y_2, \ldots, y_k \rangle \rangle. \text{ If } x \text{ and } y \text{ are tally strings (over a one-letter alphabet), then also } \langle x, y \rangle \text{ is assumed to be tally. Poly denotes the class of all polynomials. Let SAT be the set of all satisfiable Boolean formulas.}
\end{align*}

2. Definitions and elementary results

In this section we formally define the concept of nonuniform proof systems and prove some easy results that will be helpful in later sections.

In the theory of nonuniform complexity measures, the decision whether a string \( x \) belongs to a language \( A \) is made relative to some additional information, called an advice string. This advice string depends only on the length of \( x \). To formalize this notion, we follow Karp and Lipton \[11\].

**Definition 2.1.** For a set \( I \in \mathbb{P}_\Sigma \) and a string \( u \in \Sigma^* \) let \( I_u = \{ x \in \Sigma^* | \langle x, u \rangle \in I \} \). For a class \( C \) of sets \( \Sigma^* \) and a class of functions \( F \) from \( \mathbb{N} \) to \( \mathbb{N} \) we define \( C/F \) as the class of sets \( A \in \mathbb{P}_\Sigma \) for which there is a set \( I \in C \) and a function \( f \in F \) such that

\[
(\forall n \in \mathbb{N}) (\exists u \in \Sigma^n) : A \leq^m_n = I_u^{f(n)}.
\]

Actually, the definition in \[11\] requires only that the length of the advice strings is bounded by the function \( f \). If the function class \( F \) contains only time-constructible functions \( f \) satisfying

\[
(\forall c \in \mathbb{N})(\exists f' \in F)(\forall n \in \mathbb{N}) : c \cdot f(n) \leq f'(n),
\]

and if the language class \( C \) is closed under \( \leq^p_m \)-reductions, then both definitions are equivalent. Observe that the class of all polynomials and all language classes from the polynomial-time hierarchy satisfy these conditions.

Nonuniform complexity measures can also be characterized by oracle Turing machines using sparse or tally sets as their oracle languages (see \[23\]), e.g.

\[
\begin{align*}
P/\text{Poly} &= P(\text{SPARSE}) = P(\text{TALLY}), \\
\text{NP/} \text{Poly} &= \text{NP}(\text{SPARSE}) = \text{NP}(\text{TALLY}), \\
\text{co-NP/} \text{Poly} &= \text{co-NP}(\text{SPARSE}) = \text{co-NP}(\text{TALLY}).
\end{align*}
\]

Pippenger \[19\] shows that \( P/\text{Poly} \) contains all languages having polynomially sized circuits, and Schöning \[23\] gives a characterization of \( \text{NP/} \text{Poly} \) by polynomial size generators. It is easy to see that an analogous result for \( \text{co-NP/} \text{Poly} \) can be obtained by equipping circuits with universal quantifiers instead of the existential quantifiers of generators.

The set \( I \) in Definition 2.1 interprets the advice strings as descriptions of finite initial segments of the language \( A \). Therefore, we call this set an interpretation set. Let \( B' \) be
the set of all advice strings which are correct with regard to the interpretation set I, i.e.
\[ B' = \{ \langle 0^n, u \rangle \mid n \in \mathbb{N}; u \in \Sigma^{=f(n)}; A \leq^u_n = I_u^{\leq n} \}. \]

Then, \( B' \) may contain many advice strings \( \langle 0^n, u \rangle \) for some \( n \in \mathbb{N} \). Definition 2.1 demands only that for all \( n \in \mathbb{N} \), \( B' \) includes at least one advice string \( \langle 0^n, u \rangle \). In order to recognize the set \( A \) using the interpretation set \( I \) this requirement suffices.

The fundamental idea of this paper is to consider arbitrary subsets of \( B' \) that contain at least one advice string for each \( n \in \mathbb{N} \). We call such a subset of \( B' \) a proof set. Note that recognition of \( A \) by the interpretation set \( I \) can be based on any proof set in place of \( B' \). Therefore, the notion of proof sets allows an investigation of the complexity of computing correct advice strings. Note that subsets of \( B' \) may be much easier to recognize than \( B' \).

Our notion of proof sets is a generalization of the CIR(\( A \))-notation from [14]. This is defined formally in the following.

**Definition 2.2.** (i) For a function \( f: \mathbb{N} \to \mathbb{N} \) a set \( B \subseteq \{ \langle 0^n, u \rangle \mid n \in \mathbb{N}; u \in \Sigma^{=f(n)} \} \) is called an \( f \)-set.

(ii) Let \( I \in P_\Sigma \) be a set, let \( f: \mathbb{N} \to \mathbb{N} \) be a function, and let \( B \) be an \( f \)-set. \( (I, B, f) \) is a (nonuniform) proof system for a language \( A \) iff

\begin{align*}
(2.1) & \quad (\forall n \in \mathbb{N})(\exists u \in \Sigma^{=f(n)}) : \langle 0^n, u \rangle \in B, \\
(2.2) & \quad (\forall n \in \mathbb{N})(\forall u \in \Sigma^{=f(n)}) : \langle 0^n, u \rangle \in B \Rightarrow A \leq u_n = I_u^{\leq n}.
\end{align*}

The first (second/third) component of a nonuniform proof system is called interpretation set (proof set/length function).

Note that the conditions (2.1) and (2.2) guarantee that the proof set contains an advice string for any length of strings, and that all elements of the proof set are correct with regard to the interpretation set.

**Remark.** The terms proof set and nonuniform proof system indicate that the advice strings of the proof set can be used as follows. Let \( (I, B, p) \) be a proof system for a language \( A \), let \( \langle 0^n, u \rangle \in B \), and let \( x \in \Sigma^{\leq n} \). Then, the advice string \( u \) represents (with regard to the interpretation set \( I \)) a proof for one of the facts "\( x \in A \)" or "\( x \notin A \)". The terms "advice set" and "advice system" would also be suitable.

The power of nonuniform proof systems depends on the complexities of the interpretation sets and the proof sets, and the rate of growth of the length functions. We limit ourselves to length functions growing polynomially. Note that for any language \( A \) there is a proof system with an interpretation set from \( P \) and a length function growing exponentially since the finite initial segments of \( A \) can be encoded very easily by advice strings of exponential length.
Definition 2.3. Let $C_1$ and $C_2$ be language classes. A proof system $(I, B, f)$ is called a $C_1$-$C_2$ proof system iff $I \in C_1$ & $B \in C_2$ & $f \in \text{Poly}$. $C_1$-$C_2$-$\text{PSL}$ denotes the class of languages for which there are $C_1$-$C_2$ proof systems.

This definition subsumes that of nonuniform complexity classes since for any language class $C$

$$C/\text{Poly} = C-P_z-\text{PSL}.$$ 

Therefore, nonuniform classes can be described by proof systems which do not limit the complexity of computing correct advice strings. Some instances of the above equation are of particular interest: $P/\text{Poly} = P-P_z-\text{PSL}$, $NP/\text{Poly} = NP-P_z-\text{PSL}$ and $\text{co-NP}/\text{Poly} = (\text{co-NP})-P_z-\text{PSL}$. The following relations among proof systems are easy consequences of our definitions. For all language classes $C_1$, $C_2$, $C_3$ and $C_4$

(2.3) $\text{co-(}C_1-C_2-\text{PSL}\text{)} = (\text{co-}C_1)-C_2-\text{PSL}$,

(2.4) $C_1 \subseteq C_3 \ & C_2 \subseteq C_4 \Rightarrow C_1$-$C_2$-$\text{PSL} \subseteq C_3$-$C_4$-$\text{PSL}$,

(2.5) $(C_1$-$C_2$-$\text{PSL}) \cup (C_3$-$C_2$-$\text{PSL}) = (C_1 \cup C_3)$-$C_2$-$\text{PSL}$.

Moreover, for $C_1$ closed under $\leq^p_m$-reductions and for $C_2$ containing any p-set $B$ (for any polynomial $p$) that satisfies (2.1)

(2.6) $C_1 \subseteq C_1$-$C_2$-$\text{PSL}$.

We next observe that in certain cases the complexity of proof sets can be decreased without reducing the power of the proof systems.

Proposition 2.4. For all language classes $C$ that are closed under $\leq^p_m$-reductions

$$(\forall k \geq 1): C-\Sigma^p_k$-PSL $= C-\Pi^p_{k-1}$-PSL.$$

Proof. For a proof of the nontrivial inclusion, let $(I, B, p)$ be a $C-\Sigma^p_k$ proof system for a given language $A$. Since $B \in \Sigma^p_k$ and $B$ is a p-set, there is a polynomial $q$ and a set $B' \in \Pi^p_{k-1}$ such that

$$B = \{ \langle 0^n, u \rangle \ | \ n \in \mathbb{N}; \ u \in \Sigma^{=q(n)} \}; \ (\exists y \in \Sigma^{=q(n)}): \langle 0^n, u, y \rangle \in B'. \}.$$ 

$B'$ can be extended to a $C-\Pi^p_{k-1}$ proof system $(I', B', p')$ for the language $A$ by letting $I' = \{ \langle x, u \rangle \ | \ x \in I \}$ and choosing $p'$ in an appropriate way. \[ \square \]

The proof of the above proposition actually shows that for any classes $C_1$ and $C_2$ which are closed under $\leq^p_m$-reductions, $C_1$-$C_2$-$\text{PSL} = C_1$-$\text{CL}_3(C_2)$-$\text{PSL}$, where $\text{CL}_3(C_2)$ denotes the closure of $C_2$ under polynomially bounded existential quantification. In particular, for $C_1$ closed under $\leq^p_m$-reductions,

$$C_1$-$\text{P}(C_2)$-$\text{PSL} = C_1$-$\text{NP}(C_2)$-$\text{PSL}.$$
Limiting the complexity of computing correct advice strings and of interpreting advice strings limits the complexity of the describable languages. The following proposition is based on this observation.

**Proposition 2.5.** For all language classes $C_1$ and $C_2$, $C_1\cap C_2 \subseteq \Sigma_p^n(C_1 \oplus C_2) \cap \Pi_p^n(C_1 \oplus C_2)$.

**Proof.** Let $C_1$ and $C_2$ be any language classes, and let $A \in C_1 \cap C_2$. Then, there is a $C_1 \cap C_2$ proof system $(I, B, p)$ for $A$. From our definitions, we get that the language $A$ can be characterized as follows:

1. $A = \{x \in \Sigma^* \mid (\exists u \in \Sigma^{\leq n \cdot l}) : \langle 0^{|x|}, u \rangle \in B$ and $\langle x, u \rangle \in I \}$,
2. $A = \{x \in \Sigma^* \mid (\forall u \in \Sigma^{\leq n \cdot l}) : \langle 0^{|x|}, u \rangle \notin B$ or $\langle x, u \rangle \in I \}$.

Hence, $A \in \Sigma_p^n(C_1 \oplus C_2) \cap \Pi_p^n(C_1 \oplus C_2)$. \ \(\square\)

In particular, sets belonging to the polynomial-time hierarchy are able to constitute proof systems only for languages of this hierarchy, i.e.

\begin{equation}
(\forall k, l \geq 0) : \Sigma^p_k \subseteq \Sigma^p_{\max\{k, l\}}.
\end{equation}

By (2.6), for all $l \geq 0$ and all $k \geq l$, also the inverse direction of this equation holds. The next proposition shows that the complexity of computing correct advice strings can be bounded in the complexity of interpreting these advice strings and the complexity of the described language.

**Proposition 2.6.** For all language classes $C_1$ and $C_2$,

\begin{equation}
C_1 \cap \Pi_p^n \cap C_2 = C_1 \cap \Pi_p^n(C_1 \oplus C_2) \cap \Pi_p^n(C_1 \oplus C_2).
\end{equation}

**Proof.** For a proof of the nontrivial inclusion, let $(I, B, p)$ be a $C_1 \cap \Pi_p^n$ proof system for a given language $A \in C_2$. The set $B'$ is constructed as a completion of $B$:

\begin{equation}
B' = \{ \langle 0^*, u \rangle \mid n \in \mathbb{N}_n; u \in \Sigma^{\leq n}; (\forall x \in \Sigma^{\leq n}) : \langle x, u \rangle \in I \Rightarrow x \in A \}.
\end{equation}

$(I, B', p)$ is a $C_1 \cap \Pi_p^n(C_1 \oplus C_2)$ proof system for the language $A$. \ \(\square\)

In the sequel the preceding proposition will be helpful for proving the lowness of several language classes. In particular, the following instances of this proposition will be required:

\begin{equation}
\Pi_p^n / \text{Polyn} \cap \text{NP} = \Pi_p^n \cap \Pi_p^n \cap \text{NP},
\end{equation}

and for all $C \subseteq \text{Poly}$,

\begin{equation}
P / \text{Polyn} \cap C = P \cap \Pi_p^n(C) \cap \text{Polyn} \cap C.
\end{equation}
Equation (2.9) was originally proved by Schöning using different terminology [23, Lemma 5.6].

3. Nonuniform proof systems and the low hierarchy

Schöning [20] introduced the low and the high hierarchy as follows.

**Definition 3.1.** For each $k \geq 0$, $L^p_k$ contains a language $A \in \text{NP}$ iff $\Sigma^p_k(A) \subseteq \Sigma^p_k$, and $H^p_k$ contains a language $A \in \text{NP}$ iff $\Sigma^p_{k+1} \subseteq \Sigma^p_k(A)$.

For a detailed discussion of these concepts see [15,20,21] where the following proposition is proved.

**Proposition 3.2.**
(i) $(\forall k \geq 0)$: $L^p_k \subseteq L^p_{k+1}$ & $H^p_k \subseteq H^p_{k+1}$;
(ii) $L^p_0 = \text{P}$ & $L^p_1 = \text{NP} \cap \text{co-NP}$;
(iii) $H^p_0 = \{ A \in \text{P} \mid A \text{ is } \leq^p_{\text{NP}} \text{-complete for } \text{NP} \}$ & $H^p_1 = \{ A \in \text{P} \mid A \text{ is } \leq^p_{\text{NP}} \text{-complete for } \text{NP} \}$;
(iv) $(\forall k \geq 0)$: $L^p_k \cap H^p_k \neq \emptyset \iff \Sigma^p_k = \Sigma^p_{k+1}$.

Here $\leq^p$ denotes the polynomial-time, strong nondeterministic Turing reducibility [17]. Many other notions of NP-completeness (using nondeterministic, probabilistic, or circuit reducibilities) are also included in various levels of the high hierarchy. This fact and part (iv) of the Proposition 3.2 give a handle on proving polynomial hierarchy collapsing results: if some set that is NP-complete with regard to one of the above-mentioned reducibilities were low, then the polynomial-time hierarchy collapses. On the other hand, there are many classes known to be included in the low hierarchy. They cannot contain an NP-complete language unless the polynomial-time hierarchy collapses.

We show now that sets from NP for which there are proof systems with an interpretation set from co-NP and a proof set from the polynomial-time hierarchy belong to the low hierarchy. In the proof we apply known techniques in a very general way.

**Theorem 3.3.** $(\forall k \geq 1)$: $\Pi^p_k \cap \Sigma^p_k \cap \text{PSL} \cap \text{NP} \subseteq L^p_k$.

**Proof.** By Proposition 2.4 we have to show that $(\forall k \geq 1)$: $\Pi^p_k \cap \Pi^p_{k-1} \cap \text{PSL} \cap \text{NP} \subseteq L^p_k$.

We distinguish three cases.

**Case 1.** $k = 1$: Suppose $A \in \Pi^p_1 \cap \text{PSL} \cap \text{NP}$. By (2.3) $A \in \text{co-(NP-P-PSL)} \cap \text{NP}$. With Proposition 2.5 we get $A \in \text{co-NP} \cap \text{NP}$. Hence, by Proposition 3.2 $A \in L^p_1$. 


Case 2. \(k \geq 2\) and \(k\) is even: Let \(A \in \Pi_2^P \cap \Pi_{k-1}^P \cap \text{NP}\), and let \(L \subseteq \Sigma_k^P(A)\). Then there is

1. A \(\Pi_2^P \cap \Pi_{k-1}^P\) proof system \((I, B, p)\) for \(A\);
2. A set \(D \subseteq P\), and a polynomial \(q\) such that
   \[A = \{x \in \Sigma^* | (\exists z \in \Sigma \subseteq q(|x|)) \langle x, z \rangle \in D\}\];
3. A set \(E \subseteq P\) and a polynomial \(q'\) such that \((\forall n \in \mathbb{N}) (\forall x \in \Sigma \subseteq n^q(n))\)
   \[\forall \langle x, u \rangle \in I \iff (\forall z' \in \Sigma \subseteq q'(n)) : \langle x, u, z' \rangle \in E\];
4. A deterministic, polynomial-time oracle machine \(M\), and a polynomial \(q''\) such that
   \[L = \{x \in \Sigma^* | (\exists y \in \Sigma \subseteq q''(|x|)) \forall y_2 \in \Sigma \subseteq q''(|x|)) \exists y_3 \in \Sigma \subseteq q''(|x|)) \exists y_4 \in \Sigma \subseteq q''(|x|)) \exists y_5 \in \Sigma \subseteq q''(|x|)) \langle x, y_1, y_2, y_3, y_4, y_5 \rangle \in L(M, A)\}\]

Let \(r(r')\) be a polynomial such that for all \(n \in \mathbb{N}\), \(r(n) (r'(n))\) bounds the number (length) of oracle queries of \(M\) on an input of the form \(\langle x, y_1, \ldots, y_k \rangle\), where \(x \in \Sigma \subseteq n\) and \(|y_i| \leq q''(n)\). We construct a deterministic polynomially time-bounded Turing machine \(T\) that, on an input of the form \(\langle x, y_1, \ldots, y_k, z_1, \ldots, z_{r(|x|)}, z'_1, \ldots, z'_{r(|x|)}, u \rangle\), operates like \(M\) on input \(\langle x, y_1, \ldots, y_k \rangle\), where the \(i\)th oracle query \(\langle s, z_i \rangle\) of \(M\) is simulated in such a way that

1. the answer “yes” is assumed if \(\langle s, z_i \rangle \in D\);
2. The answer “no” is assumed if \(\langle s, z_i \rangle \notin D\) and \(\langle s, u, z'_i \rangle \notin E\);
3. the machine \(T\) halts and accepts its input if \(\langle s, z_i \rangle \notin D\) and \(\langle s, u, z'_i \rangle \in E\). (In this case, \(T\) would not be determinable if \(s \in A\).)

Claim 3.4. For all \(n \in \mathbb{N}\), all \(x \in \Sigma \subseteq n\), all \(\langle 0^{r(n)}, u \rangle \in B\), and all \(y_1, \ldots, y_k \in \Sigma \subseteq q''(n)\),
\[\langle x, y_1, \ldots, y_k \rangle \in L(M, A) \iff (\forall z_1, \ldots, z_{r(n)} \in \Sigma \subseteq q'(r(n))) \langle x, y_1, \ldots, y_k, z_1, \ldots, z_{r(n)}, z'_1, \ldots, z'_{r(n)}, u \rangle \in L(T)\].

Proof. For fixed \(n \in \mathbb{N}\), \(x \in \Sigma \subseteq n\), \(\langle 0^{r(n)}, u \rangle \in B\), and \(y_1, \ldots, y_k \in \Sigma \subseteq q''(n)\), the claim follows from two observations:

1. For all \(z_1, \ldots, z_{r(n)} \in \Sigma \subseteq q'(r(n))\) and all \(z'_1, \ldots, z'_{r(n)} \in \Sigma \subseteq q'(r(n))\), on input \(\langle x, y_1, \ldots, y_k, z_1, \ldots, z_{r(n)}, z'_1, \ldots, z'_{r(n)}, u \rangle\) the machine \(T\) does not simulate the wrong answer to any oracle query \(\langle s \in A? \rangle\) of \(M\) on input \(\langle x, y_1, \ldots, y_k \rangle\) since, if the input does not allow \(T\) to determine whether \(s\) belongs to \(A\), then \(T\) halts and accepts its input. This proves the direction “\(\Rightarrow\)”.
2. There are strings \(z_1, \ldots, z_{r(n)} \in \Sigma \subseteq q'(r(n))\) and \(z'_1, \ldots, z'_{r(n)} \in \Sigma \subseteq q'(r(n))\) such that on input \(\langle x, y_1, \ldots, y_k, z_1, \ldots, z_{r(n)}, z'_1, \ldots, z'_{r(n)}, u \rangle\), the machine \(T\) is able to answer all oracle queries of \(M\) on input \(\langle x, y_1, \ldots, y_k \rangle\) correctly, and thus to complete the simulation of the machine \(M\). This proves the direction “\(\Leftarrow\)”.
\[\square\]
By this claim, 
\[ L = \{ x \in \Sigma^* | (\exists y_1 \in \Sigma^{q^r(|x|)})(\exists u \in \Sigma^{p(|x|)}) \} \]

1. \( \langle 0^{r(|x|)}, u \rangle \in B \);

2. \( \langle \forall y_2 \in \Sigma^{q^r(|x|)}(\forall y_3 \in \Sigma^{q^r(|x|)}) \ldots (\forall y_k \in \Sigma^{q^r(|x|)}) \}
\( (\forall z_1, \ldots, z_t(|x|) \in \Sigma^{q^r(|x|)})(\forall z_1', \ldots, z_t(|x|) \in \Sigma^{q^r(|x|)}) \}
\[ (x, y_1, \ldots, y_k, z', \ldots, z_t(|x|), u) \in L(T) \} \].

Since \( L(T) \in \mathbb{P} \) and \( B \in \Pi_{k-1}^\mathbb{P} \), it follows that \( L \in \Sigma_k^\mathbb{P} \), and that \( A \in L_k^\mathbb{P} \).

Case 3 \( k \geq 3 \) and \( k \) is odd: We only point out the main differences with the proof of the previous case:

1. If the machine \( T \) is not able to determine whether \( s \in A \), then \( T \) rejects the input instead of accepting it,

2. In the concluding characterization of the language \( L \) the strings \( z_1, \ldots, z_t(|x|) \) and \( z'_1, \ldots, z'_t(|x|) \) are guessed by existential quantifiers instead of universal quantifiers.

Consequently, all possible cases are proved. \( \square \)

Later we will see that this theorem is strong enough to prove most of the known results concerning the lowness of language classes. Using (2.8) we get a new result.

**Corollary 3.5.** \( \text{co-NP}/\text{Poly} \cap \text{NP} \subseteq L_3^\mathbb{P} \)

Therefore, if there is a high set in \( \text{co-NP}/\text{Poly} \cap \text{NP} \) then the polynomial-time hierarchy collapses. The above corollary strengthens two results in the literature: \( \text{P}/\text{Poly} \cap \text{NP} \subseteq L_3^\mathbb{P} \) [4, 6], and \( \text{co-NP}/\text{Poly} \supseteq \text{NP} \Rightarrow \text{PH} = \Sigma_3^\mathbb{P} \) [30].

## 4. Properties of nonuniform proof systems

### 4.1. Secure proof systems

Definition 2.2 demands the correctness of the interpretation set only for advice strings that belong to the proof set. In this section the behaviour of the interpretation set with regard to all possible advice strings is considered. The interpretation sets of secure proof systems make errors only in one direction (yes answers are always correct).

**Definition 4.1.** A proof system \( (I, B, p) \) for a set \( A \) is called **secure** iff

\( (\forall n \in \mathbb{N})(\forall u \in \Sigma^{p(n)}) : A \leq_n p \Rightarrow I_u \leq_n. \)

Let \( C_1 \) and \( C_2 \) be language classes. \( C_1 \cdot C_2 \cdot \text{SPSL} \) denotes the class of sets \( A \subseteq \Sigma^* \) for which there are secure \( C_1 \cdot C_2 \) proof systems.
The fact that secure proof systems only err in one direction allows us to show that for secure proof systems the complexity of computing correct advice strings can be bounded very sharply. This will imply that (co-NP)-P\Sigma P-SPLP∩NP is contained in the second level of the low hierarchy. Later we will see that (co-NP)-P\Sigma P-SPLP∩NP subsumes all of the other classes known to belong to this level of the low hierarchy. Moreover, we will show that proof systems for disjunctive self-reducible sets can always be transformed into secure proof systems. Thereby we will get the result that (co-NP∩NP)/Poly ⊆ NP implies PH = \Sigma_2^P.

The notion of secure proof systems will be helpful also for the descriptions of probabilistic complexity classes with one-sided error such as R, and of an interesting language class which we investigate in Section 4.2.

Observe that the statements (2.4), (2.5) and (2.6) hold for the classes of Definition 4.1 in an analogous way. For secure proof systems Eq. (2.7) can be improved.

**Proposition 4.2.** \((\forall k \geq 1): \Sigma^P_k \cdot P\Sigma P-SPLP = \Sigma^P_k .\)

**Proof.** For a proof of the nontrivial inclusion let \((I, B, p)\) be a secure \(\Sigma^P_k \cdot P\Sigma P\) proof system for a given language \(A\). The following characterization of \(A\) shows that \(A \in \Sigma^P_k \cdot P\Sigma P\).

\[A = \{x \in \mathbb{C}^* | (\exists u \in \Sigma^{=p}(x)) : \langle x, u \rangle \in I\}.\]

The characterization of the language \(A\) in the preceding proof shows that secure proof systems lead to language classes which are a combination of nonuniform and nondeterministic complexity classes: there are advice strings of polynomial length that describe the finite initial segments of \(A\), and, in order to reduce \(A\) to \(I\), these advice strings can be guessed by existential quantifiers.

Now, we will show that the class (co-NP)-P\Sigma P-SPLP∩NP is included in the second level of the low hierarchy. For this we have to improve Proposition 2.6 for secure proof systems.

**Proposition 4.3.**

(i) For each language class \(C\), \(C \cdot P\Sigma P-SPLP = C \cdot \Pi^P_k (C) \cdot SLP\).

(ii) \((\forall k \geq 1): \Pi^P_k \cdot P\Sigma P-SPLP \cap \Sigma^P_k = \Pi^P_k \cdot \Pi^P_k - SLP \cap \Sigma^P_k .\)

**Proof.** Let \((I, B, p)\) be a secure proof system for a language \(A\). Then, the completion of the proof set (compare with the proof of Proposition 2.6) can be characterized as follows:

(i) \(B' = \{\langle 0^n, u \rangle | n \in \mathbb{N}; u \in \Sigma^{=p(n)}; (\forall u' \in \Sigma^{=p(n)}) (\forall x \in \Sigma^{\leq n}): \langle x, u' \rangle \in I \Rightarrow \langle x, u \rangle \in I\};\)

(ii) \(B' = \{\langle 0^n, u \rangle | n \in \mathbb{N}; u \in \Sigma^{=p(n)}; (\forall x \in \Sigma^{\leq n}): x \notin A \text{ or } \langle x, u \rangle \in I\}.\)

Using these constructions, the inclusions of the left sides in the right sides follow easily. For part (i) the other direction is trivial. Therefore, it remains to prove the inclusion of the right-hand side in the left-hand side of part (ii). Let \((I, B, p)\) be a \(\Pi^P_k \cdot \Pi^P_k\) proof system for a language \(A\). We define the set \(I'\) by

\[I' = \{\langle x, u \rangle | x \in \mathbb{C}^* \& \langle x, u \rangle \in I \& [\exists n \geq 1 : u \in \Sigma^{=p(n)} \& \langle 0^n, u \rangle \in B]\}.\]
As one can easily check, \((I', B, p)\) is a secure proof system for \(A\) and \(I' \in \Pi^p_2\). Hence, \(A \in \Pi^p_2 \cap \Sigma^p_2\)-SPSL. □

Together with Theorem 3.3, part (ii) of this proposition shows that the class \((\text{co-NP}) \cap \Sigma^p_2\)-SPSL \(\cap \text{NP}\) is contained in the second level of the low hierarchy.

**Corollary 4.4.** \(\Pi^p_2 \cap \Sigma^p_2\)-SPSL \(\cap \text{NP} \subseteq L^p_2\).

Proof systems for languages being disjunctive self-reducible can always be transformed into secure proof systems. To formalize this result we apply a notion of disjunctive self-reducibility defined in [5].

**Definition 4.5.** A set \(A\) is **self-reducible** iff there exists a deterministic polynomial-time oracle machine \(M\) such that \(A = L(M, A)\), and on each input of length \(n\) every word queried to the oracle has length less than \(n\). \(A\) is **disjunctive self-reducible** iff it is self-reducible, and there is an oracle machine witnessing this fact that accepts its input whenever the oracle answers positively to any of the queries. DSR denotes the class of sets that are disjunctive self-reducible.

In [12] the inclusion \(\text{DSR} \subseteq \text{NP}\) is shown. SAT is a member of the class DSR. The required oracle machine is based on the following observation: a Boolean formula \(u\) that has a variable \(x\) is satisfiable iff at least one of the two formulas, that are obtained from \(u\) by assigning a value to \(x\), is satisfiable. Note that Boolean formulas can be encoded in such a way that assigning values to variables decreases the length of the formulas.

The following theorem holds also for certain generalizations of disjunctive self-reducibility (see e.g. [5, 12]), but the above definition suffices for our purposes.

**Theorem 4.6.** For all languages \(A \in \text{DSR}\) and all proof systems \((I, B, p)\) for \(A\), there is a set \(I' \in \text{P}(I)\) such that \((I', B, p)\) is a secure proof system for \(A\).

**Proof.** Let \(A\) be a disjunctive self-reducible language, say via machine \(M\), and let \((I, B, p)\) be a proof system for \(A\). We construct a deterministic oracle machine \(M'\) such that \((L(M', I), B, p)\) is a secure proof system for the language \(A\). On an input of the form \(\langle x, u \rangle\), \(M'\) operates as follows.

Simulate the machine \(M\) on input \(x\) replacing each oracle query of \(M\) for some string \(s\) by an oracle query for the string \(\langle s, u \rangle\) until one of the following cases occurs:

**Case 1.** The machine \(M\) stops: In this case accept the input \(\langle x, u \rangle\) iff the machine \(M\) accepts the input \(x\).

**Case 2.** An oracle query \(\langle s, u \rangle\) is answered positively: In this case accept the input \(\langle x, u \rangle\) iff the string \(\langle s, u \rangle\) is accepted by this algorithm.
It is easy to see that the machine $M'$ is polynomially time bounded. Therefore, it remains to show that $(L(M', I), B, p)$ is a secure proof system for the language $A$.

**Claim 4.7.** $(\forall x, u \in \Sigma^*) : \langle x, u \rangle \in L(M', I) \Rightarrow x \in A$.

**Proof of Claim 4.7.** For a fixed $u \in \Sigma^*$ we prove the statement by induction on the length of $x$.

$|x| = 0$: On inputs of length 0 (the empty string) the machine $M$ is not allowed to query the oracle. By our construction of the machine $M'$ it follows immediately that $\varepsilon \in L_u(M', I) \iff \varepsilon \in L(M, A) \iff \varepsilon \in A$.

$|x| > 0$: For a contradiction, assume that for some $m \in \mathbb{N}$, and some $x \in \Sigma^{=m+1}$, $\langle x, u \rangle \in L(M', I)$ and $x \notin A$. From Definition 4.5, and the fact that $x \notin A$ it follows that, on input $x$, the machine $M$ using $A$ as its oracle language queries its oracle only for strings not belonging to $A$. If all oracle queries of $M'$ (using $I$ as its oracle language) on input $\langle x, u \rangle$ are also answered negatively, then by our construction of $M'$, $\langle x, u \rangle \notin L(M', I)$.

In the other case let "$\langle s, u \rangle \in I$?" be the first oracle query of $M'$ on input $\langle x, u \rangle$ that is answered positively. Since all oracle queries of $M$ on input $x$ are answered negatively it follows that $s \notin A$. By the induction hypothesis, $\langle s, u \rangle \notin L(M', I)$, and so by our construction of $M'$, $\langle x, u \rangle \notin L(M', I)$. This contradiction completes the proof of claim 4.7. □

**Claim 4.8.** $(\forall n \in \mathbb{N})(\forall \langle 0^n, u \rangle \in B)(\forall x \in \Sigma^{\leq n}) : \langle x, u \rangle \in L(M', I) \iff x \in A$.

**Proof of Claim 4.8.** For a fixed $n \in \mathbb{N}$, and a fixed $\langle 0^n, u \rangle \in B$ we prove the statement by induction on the length of $x$.

$|x| = 0$: We have proved already that $\varepsilon \in L_u(M', I) \iff \varepsilon \in A$.

$|x| > 0$: Suppose for some $m < n$, $x \in \Sigma^{=m+1}$. By the fact that $\langle 0^n, u \rangle \in B$, it follows that, for all oracle queries of $M'$ on input $\langle x, u \rangle$ for some string $\langle s, u \rangle$, it must be that $\langle s, u \rangle \in I \iff s \in A$. We distinguish two cases:

1. If no oracle query of $M'$ on input $\langle x, u \rangle$ is answered positively, then, by the construction of $M'$, $\langle x, u \rangle \in L(M', I) \iff x \in L(M, A) \iff x \in A$.
2. If some oracle query "$\langle s, u \rangle \in I$?" of $M'$ on input $\langle x, u \rangle$ is answered positively, then:

   (i) from Definition 4.5 and the fact that the machine $M$ on input $x$ queries its oracle for the string $s \in A$ it follows that $x \in A$;

   (ii) by the induction hypothesis, it follows that $\langle s, u \rangle \in L(M', I)$, and so by our construction of $M'$, $\langle x, u \rangle \in L(M', I)$.

This completes the proof of Claim 4.8. □

**Proof of Theorem 4.6 (conclusion).** Using both claims, and the fact that $(I, B, p)$ is a proof system for the language $A$ it follows that $(L(M', I), B, p)$ is a secure proof system for $A$. □
Using Proposition 4.3 and the disjunctive self-reducibility of any NP-complete set, there are some interesting applications of the preceding theorem.

**Corollary 4.9.**

(i) $\text{DSR} \cap \text{P}/\text{Poly} \subseteq \text{P} \cdot \Pi^p_1 \cdot \text{SPSL}$;

(ii) $\text{DSR} \cap (\text{NP} \cap \text{co-NP})/\text{Poly} \subseteq \Pi^p_1 \cdot \Pi^p_2 \cdot \text{PSL}$;

(iii) $\text{NP} \subseteq (\text{NP} \cap \text{co-NP})/\text{Poly} \Rightarrow \text{PH} = \Sigma^p_2$.

Part (iii) improves the known result $\text{P}/\text{Poly} \supseteq \text{NP} \Rightarrow \text{PH} = \Sigma^p_2$ from Karp and Lipton [11] and was independently proved by Abadi et al. [1]. By part (ii) and Theorem 3.3 we get that the class $\text{DSR} \cap (\text{NP} \cap \text{co-NP})/\text{Poly}$ is included in the second level of the low hierarchy.

**Proof of Corollary 4.9.**

(i) This statement follows immediately from Theorem 4.6, the fact that the class P is closed under Turing-reductions and Proposition 4.3.

(ii) Suppose $A \in \text{DSR} \cap (\text{NP} \cap \text{co-NP})/\text{Poly}$. Then, $A \in (\text{NP} \cap \text{co-NP}) - \text{P} - \text{PSL}$. By Theorem 4.6, $A \in \text{P} \cdot (\text{NP} \cap \text{co-NP}) - \text{P} - \text{PSL}$. Since $\text{P} \cdot (\text{NP} \cap \text{co-NP}) \subseteq \text{co-NP}$ it holds that $A \in (\text{co-NP}) - \text{P} - \text{PSL}$. Using the inclusion $\text{DSR} \subseteq \text{NP}$, from Proposition 4.3 it follows that $A \in \Pi^p_1 \cdot \Pi^p_2 - \text{PSL}$.

(iii) By hypothesis, $\text{SAT} \in (\text{NP} \cap \text{co-NP})/\text{Poly}$. Hence by (ii), $\text{SAT} \in \Pi^p_1 \cdot \Pi^p_2 - \text{PSL}$. With Theorem 3.3 it follows that $\text{SAT} \in \text{L}^p_2$. Since SAT is $\leq^p_n$-complete for NP, by Proposition 3.2 we get that $\text{PH} = \Sigma^p_2$. $\Box$

### 4.2. Secure proof systems with interpretation sets from P

In this section we consider the class of all languages for which there are secure $\text{P} \cdot \Pi^p_2$ proof systems and outline connections between this class and other concepts.

**Definition 4.10.** $\text{MCPNP} := \text{P} \cdot \Pi^p_2 \cdot \text{SPSL}$.

Note that by Proposition 4.2, $\text{MCPNP} \subseteq \text{NP}$ and using Corollary 4.4, $\text{MCPNP} \subseteq \text{L}^p_2$. The class MCPNP can be characterized by nondeterministic, polynomial-time Turing machines that have, for all $n \in \mathbb{N}$, a “maximal” computation path for accepting strings of length at most $n$. This is a suitable interpretation of the following proposition.

**Proposition 4.11.** For each language $A$, the following are equivalent:

(i) $A \in \text{MCPNP};$

(ii) There exists a set $C \in \text{P}$ and a polynomial $p$ such that

\begin{align*}
(4.1) & \quad A = \{ x \in \Sigma^* | (\exists y \in \Sigma^*) \colon \langle x, y \rangle \in C \}, \\
(4.2) & \quad (\forall n \in \mathbb{N}) (\exists y \in \Sigma^{\leq p(n)}) : A^{\leq n} = C^{\leq n}.
\end{align*}
Proof (idea). "(i) ⇒ (ii)". Let \((I, B, p)\) be a secure P-PZ proof system for the language \(A\). Then the set \(C = \{ \langle x, y \rangle : (x, y) \in I \& (\exists n \geq |x|): |y| = p(n) \}\) and the polynomial \(p\) have the required properties.

"(ii) ⇒ (i)". Let \(C \in P\) be a set, let \(p\) be a polynomial, and let \(y(n) \in \Sigma^{\leq p(n)} (n \in \mathbb{N})\) be strings such that

1. \(A = \{ x \in \Sigma^* : (\exists y \in \Sigma^*): \langle x, y \rangle \in C \}\);
2. \((\forall n \in \mathbb{N}) : A \subseteq \Sigma^{\leq p(n)} \).

The sets \(I \in P\) and \(B \subseteq \Sigma^*\) are defined as follows:

\(I = \{ \langle x, y \rangle : x, y \in \Sigma^*; (\exists y', y'' \in \Sigma^*): y' y'' = y \& \langle x, y' \rangle \in C \}\),

\(B = \{ \langle 0^n, y \rangle : n \in \mathbb{N}; y \in \Sigma^{\leq p(n)}; (\exists y' \in \Sigma^{\leq p(n)}): y(n) y' = y \}\).

It is easy to see that \((I, B, p)\) is a secure proof system for \(A\). □

The above characterization motivates the name Maximal Computation Path NP for the introduced language class. Schöning [22] defined the class of languages with a short NP-description. This notion is just another characterization of MCPNP.

In the previous section (compare with Corollary 4.9) we have seen that MCPNP \(\supseteq (P/\text{Poly}) \cap \text{DSR}\). In Section 4.3 we will prove the inclusion MCPNP \(\subseteq \text{R}\). Schöning [23] showed that the classes SPARSE \(\cap\) NP, APT \(\cap\) NP and P-SELECTIVE \(\cap\) NP are included in the second level of the low hierarchy. His proofs can easily be modified to show that these classes are in fact included in MCPNP [10].

Theorem 4.12.

(i) SPARSE \(\cap\) NP \(\subseteq\) MCPNP.

(ii) APT \(\cap\) NP \(\subseteq\) MCPNP.

(iii) P-SELECTIVE \(\cap\) NP \(\subseteq\) MCPNP.

Proof (idea). (i) For a set \(A \in (\text{NP} \cap \text{SPARSE})\), there is a set \(B \in \text{P}\), and a polynomial \(p\) such that \(A = \{ x \in \Sigma^* : (\exists y \in \Sigma^{\leq p(|x|)}): \langle x, y \rangle \in B \}\), and a polynomial \(q\) such that, for all \(n \in \mathbb{N}\), \(|A| \leq q(n)\). We define the language \(C\) as follows.

\(C = \{ \langle x, y_1, y_2, \ldots, y_m \rangle : x, y_i \in \Sigma^*; m \geq 1; (\exists i \leq m): (\langle x, y_i \rangle \in B \& y_i \in \Sigma^{\leq p(|x|)}) \}\).

It follows that \(C \in \text{P}\). As one can easily check, for a suitable polynomial \(p'\), \(C\) and \(p'\) satisfy (4.1) and (4.2). Hence, \(A \in \text{MCPNP}\).

(ii) Let \(A \in \text{APT} \cap \text{NP}\). From Definition 1.1 it follows immediately that there exist sets \(A_1 \subseteq \text{P}\), and \(A_2 \subseteq \text{SPARSE}\) such that \(A = A_1 \cup A_2\). We get that \(A_2 \in \text{NP}\). By (i) it follows that \(A_2 \in \text{MCPNP}\). It is easy to see that the class MCPNP includes \(\text{P}\), and that MCPNP is closed under union. Hence, \(A \in \text{MCPNP}\).

(iii) For a set \(A \in (\text{P-SELECTIVE} \cap \text{NP})\) there is a set \(B \subseteq \text{P}\) and a polynomial \(p\) such that \(A = \{ x \in \Sigma^* : (\exists y \in \Sigma^{\leq p(|x|)}): \langle x, y \rangle \in B \}\), and (compare with Definition 1.1) a 2-placed function \(f\), computable in polynomial time, such that

\(\forall x, y \in \Sigma^* : (f(x, y) \in \{ x, y \}) \& (x \in A \text{ or } y \in A \Rightarrow f(x, y) \in A)\).
By Lemma 2.16 from [21], for all \( n \geq 0 \), there is a set \( C_n \subseteq A^{-n} \) such that

1. \( A^{-n} = \{x \in \Sigma^{-n} \mid (\exists y \in C_n): x \in \{f(x, y), f(y, x)\}\}; \)
2. \( \|C_n\| \leq n + 1. \)

Let \( C \in \mathbb{P} \) be the following language.

\[
C := \{\langle x, y_1, w(y_1), \ldots, y_k, w(y_k) \rangle \mid \langle x, y \rangle \in \Sigma^*; k \geq 1; w(y_i) \in \Sigma^{\leq p(|y_i|)} (1 \leq i \leq k); \\
(\exists i \leq k): \langle y_i, w(y_i) \rangle \in B \land x \in \{f(x, y_i), f(y_i, x)\}\}.
\]

Using the properties of the sets \( C_n \), we get the existence of a polynomial \( r \) such that \( C \) and \( r \) satisfy (4.1) and (4.2). Hence, \( A \in \text{MCPNP}. \)

Next, we introduce the notion of maximal Turing reducibility [10]. We have introduced and studied this notion independently from Ko [15] who defined and investigated the equivalent concept of one-sided helping.

**Definition 4.13.** A set \( A \subseteq \Sigma^* \) is (polynomially) maximal Turing reducible to a set \( B \subseteq \Sigma^* \) \( (A \preceq_{mT} B) \) iff there exists a deterministic, polynomial-time oracle machine \( M \) such that

1. \( A = L(M, B); \)
2. \( (\forall B' \subseteq \Sigma^*): A \supseteq L(M, B'). \)

For a language class \( C \), \( \preceq_{mT}^L(C) \) denotes the class of all languages \( L \) for which there is a set \( A \in C \) such that \( L \preceq_{mT} A \).

This notion can be interpreted as follows. By (1) the set \( A \) is Turing-reducible to the set \( B \) using the deterministic, polynomial-time machine \( M \). Moreover, by (2) the machine \( M \) can be transformed into a nondeterministic Turing machine \( T \) for \( A \) as follows: \( T \) just replaces any oracle query made by \( M \) by a guess of the answer, i.e. any computation path of \( T \) corresponds to a computation of \( M \) for some fixed oracle language \( B' \).

The classes MCPNP and NP can be characterized by \( \preceq_{mT}^P \)-reductions as in the following theorem.

**Theorem 4.14.**

(i) \( \text{MCPNP} = \preceq_{mT}^P(\text{TALLY}) = \preceq_{mT}^P(\text{P/Poly}), \)

(ii) \( \text{NP} = \preceq_{mT}^P(\text{NP}) = \preceq_{mT}^P(\text{P/Poly}). \)

**Remark.** Since there are sets in \( \text{P/Poly} \) not belonging to \( \text{NP} \), part (ii) of Theorem 4.14 shows that \( \preceq_{mT}^P \) is not reflexive. Hence, \( \preceq_{mT}^P \) is not a reducibility notion in the strict sense.

**Proof of Theorem 4.14.** (idea) (i) Let \( A \subseteq \Sigma^* \) be any language. We have to prove that the following statements are equivalent:

1. \( A \in \text{MCPNP}; \)

(ii) \( \text{NP} = \preceq_{mT}^P(\text{NP}) = \preceq_{mT}^P(\text{P/Poly}). \)
(2) \((\exists S \in \text{TALLY})\): \(A \leq_{mT}^P S\);
(3) \((\exists S \in \text{P/Pol})\): \(A \leq_{mT}^P S\).

"(1)\(\Rightarrow\)(3)"\: Let \((I, B, p)\) be a secure \(P-P_X\) proof system for \(A\). For all \(n \in \mathbb{N}\), let \(u_n\) be a string of length \(p(n)\) such that \(\langle 0^n, u_n \rangle \in B\). We define the set \(S \in \text{P/Pol}\) as follows:

\[ S := \{ \langle 0^n, u \rangle | n \in \mathbb{N}; \; u \in \Sigma^{\leq p(n)}; \; (\exists u' \in \Sigma^{p(n) - |u|}; \; u_n = uu' \}. \]

It is easy to see that there exists a deterministic, polynomially time-bounded oracle machine \(M\) that witnesses the fact \(A \leq_{mT}^P S\).

"(2)\(\Rightarrow\)(1)"\: Let \(S\) be any tally set, and let \(M\) be a deterministic, polynomially time bounded oracle machine that witnesses the fact \(A \leq_{mT}^P S\). Using proposition 4.11 it is easy to see that the set

\[ C := \{ \langle x, y_1, y_2, \ldots, y_m \rangle | x, y_i \in \Sigma^*; m \geq 0; \; x \in L(M, \{y_1, \ldots, y_m\}) \} \]

and a suitable polynomial \(p'\) imply that \(A \in \text{MCPNP}\).

"(3)\(\Rightarrow\)(2)"\: This direction follows by the facts that

(1) \(\text{P/Pol} \cap \text{TALLY})\), and
(2) \((\forall A, B, C \subseteq \Sigma^*)\): \(A \leq_{mT}^P B \leq_{mT}^P C \Rightarrow A \leq_{mT}^P C\).

(i) Let \(A \subseteq \Sigma^*\) be any language. We have to prove that the following statements are equivalent:

(1) \(A \in \text{NP}\);
(2) \((\exists C \in \text{NP})\): \(A \leq_{mT}^P C\);
(3) \((\exists C \subseteq \Sigma^*)\): \(A \leq_{mT}^P C\).

"(2)\(\Rightarrow\)(3)"\: This direction is trivial.

"(1)\(\Rightarrow\)(2)"\: Let \(B \in \text{P}\), and let \(p\) be a polynomial such that \(A = \{x \in \Sigma^* | (\exists y \in \Sigma^{\leq p(|x|)}; \langle x, y \rangle \in B \}. \)

We define the set \(C \in \text{NP}\) as follows:

\[ C := \{ \langle x, y \rangle | x \in \Sigma^*; y \in \Sigma^{\leq p(|x|)}; (\exists y' \in \Sigma^{p(|x|) - |y|}; \langle x, yy' \rangle \in B \}. \]

It is easy to see that there exists a deterministic, polynomially time-bounded oracle machine \(M\) that witnesses the fact \(A \leq_{mT}^P C\).

"(3)\(\Rightarrow\)(1)"\: Let \(C \subseteq \Sigma^*\) be any set, and let \(M\) be a deterministic, polynomially time-bounded oracle machine that witnesses the fact \(A \leq_{mT}^P C\). Without loss of generality we may assume that the machine \(M\) (on any input using any set as its oracle language) queries its oracle for the same string only one time. Let \(T\) be a nondeterministic, polynomially time-bounded Turing machine that, on an input \(x\), behaves like \(M\) on input \(x\) with the following exception: Any oracle query made by \(M\) is replaced by a guess of the answer. It follows that \(A = L(T)\). Hence, \(A \in \text{NP}\). \(\square\)

By the Theorem 4.14 we get that \(\text{P/Pol} \supseteq \text{NP}\) implies \(\text{MCPNP} = \text{NP}\). Note that this result follows also from Theorem 4.6, and the disjunctive self-reducibility of the NP-complete set SAT.
4.3. Dense proof systems

In this section we will characterize probabilistic complexity classes in terms of nonuniform proof systems. Schöning [25] defines probabilistic ("Monte Carlo") classes as follows:

**Definition 4.15.** Let $C$ be a class of languages. BPC denotes the class of all languages $A$ such that for some $B \subseteq C$, some $\delta > 0$, some polynomial $p$, all $n \in \mathbb{N}$ and all inputs $x \in \Sigma^n$

$$\| \{ y \in \Sigma^{-p(n)} | \langle x, y \rangle \in B \iff x \in A \} \| > (1 + \delta)^* 2^p(n).$$

RC denotes the class of all languages $A$ such that for $B \subseteq C$, some $\delta > 0$, some polynomial $p$, all $n \in \mathbb{N}$ and all inputs $x \in \Sigma^n$

1. $x \in A \Rightarrow \| \{ y \in \Sigma^{-p(n)} | \langle x, y \rangle \in B \} \| > \delta^* 2^p(n);
2. $x \notin A \Rightarrow (\forall y \in \Sigma^*) : \langle x, y \rangle \notin B.$

Originally, probabilistic complexity classes were defined by a model of computation similar to that of nondeterministic machines. The difference is that instead of "guessing" a next move, computation steps depend on an ideal random experiment, e.g. tossing a coin. This leads to a different definition of accepting: while nondeterministic machines accept their input iff at least one computation path leads to an accepting state, in probabilistic machines the probability of getting an accepting computation is considered.

BPP contains all languages $A$ for which a probabilistic, polynomially time-bounded machine $M$ exists satisfying the following condition. For some constant $\delta > 0$, $M$ accepts the strings from $A$ and rejects the strings from $A^c$ with probability at least $1/2 + \delta$. Many other probabilistic complexity classes have been defined, e.g. R, PP, ZPP and AM.

The description of "Monte Carlo" classes as given in Definition 4.15 is based on the following observation. The "coin-tossing" experiments which are a part of the computation can be done at the beginning. This allows interpreting their outcome as an additional, random part of the input.

To describe probabilistic classes by nonuniform proof systems, we have to analyze the density of proof sets.

**Definition 4.16.** A proof system $(I, B, p)$ is called dense iff

$$(\exists \delta > 0) (\forall n \in \mathbb{N}) : \| B \cap \{ \langle 0^n, u \rangle | u \in \Sigma^{-p(n)} \} \| > (1 + \delta)^* 2^p(n).$$

Let $C_1$ and $C_2$ be classes of languages. $C_1$-$C_2$-DPSL denotes the class of all languages which have dense $C_1$-$C_2$ proof systems. $C_1$-$C_2$-SDPSL denotes the class of all languages which have $C_1$-$C_2$ proof systems that are both secure and dense.

The statements (2.4) and (2.5) hold for dense proof systems and for proof systems that are both secure and dense in an analogous way. For dense proof systems, (2.3) also holds.
In order to relate dense proof systems to "Monte Carlo" classes we use the idea of positive reducibility [27]. Recall that, for a set $A$, $P_{\text{pos}}(A)$ ($NP_{\text{pos}}(A)$) denotes the class of all languages $B$ for which there is a deterministic (nondeterministic), polynomially time-bounded oracle machine $M$ such that

1. $B = L(M, A)$;
2. $(\forall C, D \in \mathcal{P}_x): C \supseteq D \Rightarrow L(M, C) \supseteq L(M, D)$.

**Lemma 4.17.** (i) For all classes $C$, all languages $A \in \mathcal{BPC}$, and all polynomials $q$, there exists a dense $P_{\text{pos}}(C)$-$P_x$ proof system $(I, B, p)$ for $A$ such that

$$(\forall n \in \mathbb{N}): \| B \cap \{ \langle 0^n, u \rangle \mid u \in \Sigma^p \} \| > (1 - 2^{-q(n)}) \cdot 2^p(n).$$

(ii) For all classes $C$, all languages $A \in \mathcal{RC}$ and all polynomials $q$, there exists a secure and dense $P_{\text{pos}}(C)$-$P_x$ proof system $(I, B, p)$ for $A$ such that

$$(\forall n \in \mathbb{N}): \| B \cap \{ \langle 0^n, u \rangle \mid u \in \Sigma^p \} \| > (1 - 2^{-q(n)}) \cdot 2^p(n).$$

**Proof.** Part (i) follows immediately from the second amplification lemma in [25] that can easily be adapted to our notions. Part (ii) follows by the standard technique of iterating probabilistic algorithms [24].

Using this lemma, we get the following theorem.

**Theorem 4.18.** For all language classes $C$ that are closed under $P_{\text{pos}}$-reductions, $\mathcal{BPC} = \mathcal{C-P}_x-\mathcal{DP}_x$ and $\mathcal{RC} = \mathcal{C-P}_x-\mathcal{SDP}_x$.

In [27] it is shown that the levels of the polynomial-time hierarchy are closed under $P_{\text{pos}}$-reductions. This fact and the preceding theorem show that probabilistic complexity classes can be described in terms of our framework, e.g.

(4.3) $(R =) \mathcal{RP} = P-\mathcal{P}_x-\mathcal{SDP}_x$,

(4.4) $\mathcal{BPP} = P-\mathcal{P}_x-\mathcal{DP}_x$,

(4.5) $(\mathcal{AM} =) \mathcal{BPPNP} = NP-\mathcal{P}_x-\mathcal{DP}_x$,

(4.6) $\mathcal{co-AM} = \mathcal{co-(NP-P}_x-\mathcal{DP}_x) = (\mathcal{co-NP})-\mathcal{P}_x-\mathcal{SDP}_x$.

For a broad discussion of the above classes see [2, 3, 9, 16, 28, 32, 33].

Let $C$ be any language class that is closed under $\leq^p_k$-reductions. From Theorem 4.6 and Theorem 4.18 we get that the class $\mathcal{BPC} \cap \mathcal{DSR}$ is included in $\mathcal{RC}$, e.g. $\mathcal{BP}(\Sigma_x^p \cap \Pi_x^p) \cap \mathcal{DSR} \subseteq \mathcal{R}(\Sigma_x^p \cap \Pi_x^p)$ ($k \geq 0$). By the disjunctive self-reducibility of the NP-complete set SAT, we obtain the following generalization of a result from Ko [13]:

(4.7) $(\forall k \geq 0)$: $\mathcal{BP}(\Sigma_x^p \cap \Pi_x^p) \supseteq \mathcal{NP} = \mathcal{R}(\Sigma_x^p \cap \Pi_x^p) = \mathcal{NP}$.

The complexity of the proof set of dense proof systems can be bounded in the complexity of the interpretation set. This result improves Proposition 2.6 for dense proof systems.
Theorem 4.19. For all language classes $C$ that are closed under $P_{\text{pol}}$-reductions

$C-P_2$-DPSL $\subseteq C-\Sigma^p_2(C)$-PSL.

Proof. The proof is based on the following lemma from [24, p. 74]. Here $\circ$ denotes the bitwise addition modulo 2.

Lemma 4.20. For all $n \in \mathbb{N}$, all polynomials $p$, and all sets $E \subseteq \Sigma^p$,

(i) $\| E \| \leq 2^{p(n)-n} \Rightarrow (\forall u = \langle u_1, u_2, \ldots, u_{p(n)} \rangle, |u_i| = p(n))$

$(\exists v \in \Sigma^p) [ (\forall i \leq p(n): u_i \circ v \notin E) ];$

(ii) $\| \Sigma^p - E \| \leq 2^{p(n)-n} \Rightarrow (\exists u = \langle u_1, u_2, \ldots, u_{p(n)} \rangle, |u_i| = p(n))$

$(\forall v \in \Sigma^p)(\exists i \leq p(n): u_i \circ v \in E)$.}

Note that the predicates \(\lceil \forall i \leq p(n): u_i \circ v \notin E\rceil\) and \(\lceil \exists i \leq p(n): u_i \circ v \in E\rceil\) can be decided relative to the set $E$ in polynomial time. Suppose, for a set $I \in C$, a set $B \subseteq \Sigma^*$, and a polynomial $p$, $(I, B, p)$ is a dense proof system for a given language $A$. Using Lemma 4.17, we may assume without loss of generality that

\[(\forall n \in \mathbb{N}): \| B \cap \{ \langle 0^n, u \rangle \mid u \in \Sigma^p \} \| > (1 - 2^{-n})*2^{p(n)}.\]

We construct a set $B'$ such that $(I, B', p)$ is a $C-\Sigma^p_2(C)$ proof system for the language $A$. Using the relativized version of Proposition 2.4 (compare also with the observation following this proposition), this proves that $A \in C-\Pi^p_2(C)$-PSL:

\[B' = \{ \langle 0^n, u \rangle \mid n \in \mathbb{N}; u \in \Sigma^p \}; (3u' = \langle u_1, u_2, \ldots, u_{p(n)} \rangle, |u_i| = p(n))$

$(\forall v \in \Sigma^p)(\forall x \in \Sigma^p): [ (\exists i \leq p(n): \langle x, u \rangle \notin I) \Rightarrow \langle x, u_i \circ v \rangle \notin I]$.}

From both following claims, and the fact that $(I, B, p)$ is a $C-P_2$ proof system for the set $A$, we get that $(I, B', p)$ is a $C-\Sigma^p_2(C)$ proof system for $A$.


Proof. For any $n \in \mathbb{N}$ we define $E_n := \{ u \in \Sigma^p \mid \langle 0^n, u \rangle \in B \}$. Then, $\| E_n \| > 2^{p(n) - 2^{p(n)-n}}$. Hence, $\| \Sigma^p - E_n \| \leq 2^{p(n)-n}$. Using statement (ii) from the above lemma, we get that

$(\exists u' = \langle u_1, u_2, \ldots, u_{p(n)} \rangle, |u_i| = p(n))(\forall v \in \Sigma^p) [ (\exists i \leq p(n): u_i \circ v \in E_n \}].$

For all advice strings $w, w' \in E_n$, $I_{w^n} \subseteq I_{w'^n}$. By these facts and our construction of the set $B'$ the claim follows. \(\square\)

Claim 4.22. $(\forall n \in \mathbb{N})(\forall \langle 0^n, u \rangle \in B'): A \subseteq_n I_{w^n} - I_{w'^n}$. \

Proof. Suppose, for any \( n \in \mathbb{N} \) and any \( u \in \Sigma^* \), \( \langle 0^n, u \rangle \in B' \). For a contradiction we assume that there is a string \( x \in \Sigma^{\leq n} \) such that \( x \in A \iff \langle x, u \rangle \notin I \). We define

\[
E_x := \{ u \in \Sigma^{\leq p(n)} \mid x \in A \iff \langle x, u \rangle \notin I \}.
\]

Then, \( \| E_x \| \leq 2^{p(n)-n} \). Using the above lemma we get that

\[
\forall u' = \langle u_1, u_2, \ldots, u_{p(n)} \rangle, |u_i| = p(n) \ (\exists v \in \Sigma^{\leq p(n)}): \forall i \leq p(n): u_i \circ v \notin E_x.
\]

Therefore, the string \( x \) implies that \( \langle 0^n, u \rangle \) does not belong to the set \( B' \). This contradiction proves Claim 2. \( \square \)

One instance of this result is of particular interest:

\[
BPP = \text{P}\cdot\text{P}_z\cdot\text{DPSL} \subseteq \text{P}\cdot\Pi^p_1\cdot\text{PSL}.
\]

By Proposition 2.5 the known result \( BPP \subseteq \Sigma^p \cap \Pi^p_1 \) [16, 28] follows. For all \( k \geq 1 \), \( \Sigma^p \cap \Pi^p_1 \) is closed under \( \leq^p_{\text{pos}} \)-reductions, and \( \Pi^p_1(\Sigma^p \cap \Pi^p_1) \) is included in \( \Pi^p_1 \). Therefore, we get the following generalization of this result:

\[
(4.8) \forall k \geq 1 : \text{BPP}(\Sigma^p \cap \Pi^p_1) = (\Sigma^p \cap \Pi^p_1)\cdot\text{P}_z\cdot\text{DPSL} \subseteq \Sigma^p_1 \cap \Pi^p_1 \cdot\text{DPSL} \subseteq \Sigma^p_{k+1} \cap \Pi^p_{k+1}.
\]

We continue with an astonishing connection between dense and secure proof systems.

**Theorem 4.23.** Let \( C \) be a language class closed under \( \leq_{\text{pol}} \)-reductions for which \( \text{co-C} = \text{NP}_{\text{pol}}(\text{co-C}) \). Then, \( C\cdot\text{P}_z\cdot\text{DPSL} \subseteq C\cdot\text{P}_z\cdot\text{SDPSL} \).

**Proof.** For a language \( A \in C\cdot\text{P}_z\cdot\text{DPSL} \), \( A^- \in (\text{co-C})\cdot\text{P}_z\cdot\text{DPSL} \). By the fact that \( \text{co-C} \) is closed under \( \text{NP}_{\text{pol}} \)-reductions, from a theorem in [25, p. 5] it follows the existence of a dense (co-C)-\text{P}_z proof system \( (I, B, p) \) for the set \( A^- \) such that

\[
(\forall n \in \mathbb{N})(\forall u \in \Sigma^{= p(n)}): (A^-)^{\leq n} \subseteq I_u^{\leq n}.
\]

We get easily that \( (I^-, B, p) \) is a secure and dense proof system for \( A \). \( \square \)

It is unknown whether the equation

\[
(\text{P}\cdot\text{P}_z\cdot\text{SDPSL} =) R =? = BPP = (\text{P}\cdot\text{P}_z\cdot\text{DPSL})
\]

holds. From the preceding theorem, and the fact that the classes \( \Sigma^p_k \) \( (k \geq 1) \) are closed under \( \text{NP}_{\text{pol}} \)-reductions [27] it follows that the analogous question for interpretation sets from \( \Pi^p_k \) \( (k \geq 1) \) can be answered positively, i.e.

\[
(4.9) \forall k \geq 1 : \Pi^p_k\cdot\text{P}_z\cdot\text{SDPSL} = \Pi^p_k\cdot\text{P}_z\cdot\text{DPSL}.
\]

Using this equation, the equation (4.6), and Corollary 4.4 we get the recently proved result that \( \text{co-AM} \cap \text{NP} \) is included in the second level of the low hierarchy [23].
In section 3 we have located languages from NP for which there exist proof systems with interpretation sets from co-NP inside the low hierarchy. Using this main theorem, in section 4 we have developed the known lowness results in our framework. Figure 1 summarizes the obtained inclusions among classes restricted to NP.

5. Nonuniform proof systems with interpretation sets from the low hierarchy

We conclude our investigations of nonuniform proof systems by considering systems with low interpretation sets. We will show that for a language \( A \) from NP such a proof system exists only if \( A \) is low itself. First we restrict our investigations to proof sets from the polynomial-time hierarchy. The proof of the following theorem exploits ideas that we have used already in the proof of Theorem 3.3.

**Theorem 5.1.** \( (\forall k \geq 1): L_k^p - \Sigma_k^p - \text{PSL} \cap \text{NP} = L_k^p \).

**Proof.** "\( \geq \)" follows from (2.6).

"\( \subseteq \)" Let \( A \in L_k^p - \Sigma_k^p - \text{PSL} \), and let \( (I, B, p) \) be a \( L_k^p - \Sigma_k^p \) proof system for \( A \). We have to show, \( \Sigma_k^p(A) \subseteq \Sigma_k^p \). Fix \( L \in \Sigma_k^p(A) \). Then there are deterministic, polynomially
time-bounded oracle Turing machine \( M \) and a polynomial \( q \) such that
\[
L = \{ x \in \Sigma^* | (\exists y_1 \in \Sigma^{\leq q(|x|)})(\forall y_2 \in \Sigma^{\leq q(|x|)}) \cdots (Q_k y_k \in \Sigma^{\leq q(|x|)}) : \langle x, y_1, y_2, \ldots, y_k \rangle \in L(M, A) \}.
\]

Let \( M' \) be the deterministic, polynomial-time oracle Turing machine that, on an input \( \langle x, y_1, y_2, \ldots, y_k, u \rangle \), behaves like \( M \) on input \( \langle x, y_1, y_2, \ldots, y_k \rangle \) with the following exception: Any oracle query \( \langle s, u \rangle \in \mathcal{A} \) made by \( M \) is replaced by the query \( \langle s, u \rangle \in \mathcal{I} \). Moreover, let \( r \) be a polynomial such that, for all \( n \in \mathbb{N} \), \( r(n) \) bounds the length of oracle queries of \( M \) on an input of the form \( \langle x, y_1, y_2, \ldots, y_k \rangle \), where \( x \in \Sigma^{\leq n} \) and \( y_i \in \Sigma^{\leq q(n)} \). Then \( L \) can be characterized by \( M' \) as follows:
\[
L = \{ x \in \Sigma^* | (\exists y_1 \in \Sigma^{\leq q(|x|)})(\exists u \in \Sigma^{= r(|x|)}) : \langle x, y_1, y_2, \ldots, y_k, u \rangle \in L(M', I) \}.
\]
By the fact that \( B \in \Sigma^p_k \), we get \( L \in \Sigma^p_k \). Since \( I \) belongs to the \( k \)th level of the low hierarchy, it follows \( L \in \Sigma^p_k \). \( \square \)

In the preceding sections we have bounded the complexity of computing correct advice strings in the complexity of interpreting these advice strings and the complexity of the described language. Using these results, we extend the above theorem to proof sets of unlimited complexity.

**Theorem 5.2.**

(i) (\( \forall k \geq 3 \)) holds: \( L_k^p - P_\Sigma^p \cap NP = L_k^p \),

(ii) (\( \forall k \geq 2 \)) holds: \( L_k^p - P_\Sigma^p \cap NP = L_k^p \),

(iii) (\( \forall k \geq 2 \)) holds: \( L_k^p - P_\Sigma^p \cap NP = L_k^p \).

**Proof.** The inclusions of the right-hand sides in the left-hand sides follow easily from our definitions. Using Proposition 2.6 [for part (i)], Proposition 4.3 [for part (ii)], Theorem 4.19 [for part (iii)], the following observations, and Theorem 5.1, the inclusions of the left sides in the right sides follow.

(i) (\( \forall k \geq 3 \)) holds: \( \Pi_k^p (NP \oplus L_k^p) \subseteq \Pi_k^p (L_k^p) \subseteq \Sigma_k^p (L_k^p) \subseteq \Sigma_k^p \);

(ii) (\( \forall k \geq 2 \)) holds: \( \Pi_k^p (L_k^p) \subseteq \Sigma_k^p (L_k^p) \subseteq \Sigma_k^p \);

(iii) All levels from the low hierarchy are closed under \( P_{\text{pos}}\)-reductions, and (\( \forall k \geq ? \)) holds: \( \Pi_k^p (L_k^p) \subseteq \Sigma_k^p (L_k^p) \subseteq \Sigma_k^p \).

Hence, Theorem 5.2 is proved. \( \square \)

This theorem shows that polynomially length-bounded advice strings do not augment the power of the higher levels of the low hierarchy. Hence, it seems that nonuniform complexity measures cannot be used to extend one level of the low
hierarchy to a higher level of the low hierarchy beginning at the third level. New
techniques will be required to prove lowness results for higher levels.

6. Conclusions

We have used the new concept of nonuniform proof systems as a framework for the
study of various complexity classes. Nonuniform classes can be described directly in
terms of this notion. For the study of probabilistic classes an analysis of the density of
proof sets is necessary. Demanding that interpretation sets make errors only in one
direction, it is possible to describe probabilistic classes with one-sided error, and
classes defined by simultaneous nonuniform and nondeterministic time bounds. This
homogeneous description allows a better understanding of the relationships between
the considered classes, and also between these classes and uniform complexity
measures.

We have obtained the known lowness results from our main theorem and a corre-
spanding inclusion structure. Moreover, we have shown that, for any \( k \geq 3 \), poly-
nomially length-bounded advice strings do not augment the power of the \( k \)th level of
the low hierarchy. This result gives evidence that new techniques will be required to
obtain lowness results concerning higher levels.

There still remain many possibilities for further investigations:

1. It will be interesting to consider other classes of interpretation sets and/or proof
sets and/or length functions as we have done.

2. In Section 4.1 we have seen that nonuniform proof systems for disjunctive
self-reducible languages can always be transformed into secure proof systems. If for
each language \( L \in \text{NP} \) there is a disjunctive self-reducible language \( B \) such that \( B \leq_{m}^{p} L \)
and \( L \leq_{m}^{p} B \), then from this result some interesting consequences follow such as
\( \text{MCPNP} = \text{NP} \cap \text{P/Poly} \) and \( R = \text{NP} \cap \text{BPP} \) (compare with [15, 32]).

3. Are there languages in the low hierarchy which cannot be described by inter-
pretation sets from co-NP and polynomially length-bounded advice strings?

4. Can the result \( (\text{co-NP} \cap \text{NP}) / \text{Poly} \supseteq \text{NP} \Rightarrow \text{PH} = \Sigma_{2}^{P} \) be extended to interpre-
tation sets from co-NP?

References

classes, manuscript, 1986.


