On High and Low Sets for the Boolean Hierarchy

Steffen Reith Klaus W. Wagner
Lehrstuhl für Theoretische Informatik
Universität Würzburg
[streit,wagner]@informatik.uni-wuerzburg.de

January 15, 1998

Abstract

The polynomial-time hierarchy (PH) is central for many considerations of complexity theory. We call a set $A \in NP$ *high* (*low*, resp.) for the class Σ_k^p of the polynomial-time hierarchy if Σ_k^p relativized to the oracle A yields Σ_{k+1}^p (Σ_k^p, resp). These concept of high and low sets originates from recursion theory ([Coo74] and [Soa74]) and was translated into the terms of the polynomial-time hierarchy by Schöning ([Sch85]).

Another important hierarchy in complexity theory is the boolean hierarchy (BH) ([WW85]), which is located between the classes NP and Θ_2^p of the polynomial-time hierarchy. In this paper we introduce a concept of highness and lowness for the boolean hierarchy. Informally a set $A \in NP$ is *high* (*low*, resp.) for the class $NP(k)$ of the boolean hierarchy, if certain boolean combinations of A with NP sets yield $NP(k+1)$ ($co\text{-}NP(k)$, resp.).

Using the technique from [CK96], we can show, that every low set for $NP(k)$ is low for Σ_3^p. Thus we obtain a refinement of the first three classes of the Low-hierarchy, described by Schöning in [Sch85].

Keywords: Computational complexity, high/low sets, boolean hierarchy, polynomial-time hierarchy, hard/easy, advice, collapse

1 Introduction

The concept of high/low hierarchies was originally studied in a recursion theoretic context ([Coo74] and [Soa74]). At that time the question arose how to measure the content of information of an oracle, used by a Turing machine (TM). The original definition of those high sets said, that an oracle is called *High* iff it improves the power of a Turing machine, using that oracle, so that these TM can accept languages that belongs to the next level of the arithmetic hierarchy. Now it's self evident to define low sets. A oracle is called *Low* iff it doesn't increase the power of a Turing machine to a higher level of the arithmetic hierarchy.

Many ideas and concepts were translated from recursion theory into the terms of complexity theory. So Schöning ([Sch83],[Sch85]) defined high/low sets for the polynomial-time hierarchy. He defined $High_k^p =_{\text{def}} \{A \in NP | \Sigma_{k+1}^p \subseteq (\Sigma_k^p)^A\}$ and $Low_k^p =_{\text{def}} \{A \in NP | (\Sigma_k^p)^A \subseteq \Sigma_{k+1}^p\}$.

It is possible to show many similar properties of high/low sets for the polynomial-time hierarchy and

the arithmetic hierarchy. This shows that it was important and useful to investigate these sets. The inclusion structure of $high/low$ sets for the arithmetic hierarchy is the same for $High^p_k$ (Low^p_k) sets for the polynomial-time hierarchy, since $High^p_k \subseteq High^p_{k+1}$ and $Low^p_k \subseteq Low^p_{k+1}$ for any $k \geq 0$. Moreover we know if $PH \neq \Sigma^p_k$, than $Low^p_k \cap High^p_k = \emptyset$ and if $PH = \Sigma^p_k$, than $Low^p_k = High^p_k = NP$. Some special classes of $High^p_k/Low^p_k$ sets are known: $Low^p_0 = P$, $Low^p_1 = NP \cap co\text{-}NP$ and $High^p_0 = \{A|A \text{ is } \leq^{\mathrm{p}}_T\text{-complete for } NP\}$. For more details see [Sch85].
Another well known hierarchy is the boolean hierarchy ([WW85], [CGH+88] and [CGH+89]), containing many practical problems. It is widely known, that the boolean hierarchy is embedded into the polynomial-time hierarchy. Moreover Kadin showed in [Kad88] that the polynomial-time hierarchy collapses if the boolean hierarchy collapses. More precisely he showed, that if $NP(k) = co\text{-}NP(k)$, than $PH \subseteq \Theta^p_3$. Later Chang and Kadin could show a improvement of this result. If the boolean hierarchy collapses to its k^{th} level the boolean hierarchy over Σ^p_2 also collapses to its k^{th} level ([CK96]).
It is quite familiar, that the boolean hierarchy and the query hierarchy are intertwined ([KSW87]). So all properties of the boolean hierarchy have consequences for polynomial-time calculations with restricted access to a NP-oracle.
Because of the strong connections between polynomial-time hierarchy and boolean hierarchy and the connections to polynomial-time calculations with restricted access to a NP oracle, we define high/low sets for the boolean hierarchy. It's interesting that the properties of high/low sets for the boolean and the polynomial-time hierarchy are very similar. Using the technique described in [CK96] we can also show, that any low set for the boolean hierarchy is located in Low^p_3. We improve this result again and show that these low sets are also low for the boolean hierarchy over Σ^p_2.

2 Preliminaries

With P^B (NP^B) we denote the class of all languages accepted by a deterministic polynomial time (nondeterministic polynomial time, resp.) oracle machine using the oracle B. If the queries to an oracle depend on the answer to previous queries, so we will call them *adaptive* queries. The queries are said to be made in *parallel*, if a list of all queries is calculated before the machine asks the oracle. The class of all languages accepted by deterministic polynomial time (nondeterministic polynomial time) oracle machines making only parallel queries to an oracle B will be denoted by $P^B_{\|}$ ($NP^B_{\|}$). For a class $\mathcal{C}$ of languages we define $P^{\mathcal{C}} =_{\text{def}} \bigcup_{B\in\mathcal{C}} P^B$, $NP^{\mathcal{C}} =_{\text{def}} \bigcup_{B\in\mathcal{C}} NP^B$ and $P^{\mathcal{C}}_{\|} =_{\text{def}} \bigcup_{B\in\mathcal{C}} P^B_{\|}$.
For a function $r : \mathbb{N} \mapsto \mathbb{N}$ we denote by $P^B[r]$ the class of languages accepted by some deterministic polynomial time oracle machine asking the oracle only $r(n)$ times for an input of length n. Similarly let $P^B_{\|}[r]$ be the class of languages accepted by deterministic polynomial oracle machines making only $r(n)$ parallel queries to B.
For classes $\mathcal{C}$ and $\mathcal{C}'$ of languages we define $co\text{-}\mathcal{C} =_{\text{def}} \{\overline{A}|A \in \mathcal{C}\}$, $\mathcal{C} \wedge \mathcal{C}' =_{\text{def}} \{A \cap B|A \in \mathcal{C}, B \in \mathcal{C}'\}$ and $\mathcal{C} \vee \mathcal{C}' =_{\text{def}} \{A \cup B|A \in \mathcal{C}, B \in \mathcal{C}'\}$
The *polynomial-time hierarchy* consists of the classes Σ^p_k, Π^p_k, Δ^p_k and Θ^p_k by defining inductively $\Sigma^p_0 =_{\text{def}} \Pi^p_0 =_{\text{def}} \Delta^p_0 =_{\text{def}} \Theta^p_0 =_{\text{def}} P$ and $\Sigma^p_{k+1} =_{\text{def}} NP^{\Sigma^p_k}$, $\Pi^p_{k+1} =_{\text{def}} co\text{-}\Sigma^p_{k+1}$, $\Delta^p_{k+1} =_{\text{def}} P^{\Sigma^p_k}$, $\Theta^p_{k+1} =_{\text{def}} P^{\Sigma^p_k}[O(\log n)]$. Obviously $\Sigma^p_k \cup \Pi^p_k \subseteq \Theta^p_{k+1} \subseteq \Delta^p_{k+1} \subseteq \Sigma^p_{k+1} \cap \Pi^p_{k+1}$ for all $k \geq 0$. Finally let $PH =_{\text{def}} \bigcup_{k\geq 0} \Sigma^p_k$.
The *boolean hierarchy* over a complexity class $\mathcal{C}$ consists of the classes $\mathcal{C}(i)$ and $co\text{-}\mathcal{C}(i)$ for $i = 1, 2, \ldots$ which are defined inductively by $\mathcal{C}(1) =_{\text{def}} \mathcal{C}$ and $\mathcal{C}(2i) =_{\text{def}} \mathcal{C}(2i-1) \wedge co\text{-}\mathcal{C}$ as well as $\mathcal{C}(2i+1) =_{\text{def}} \mathcal{C}(2i) \vee \mathcal{C}$ for $i \geq 1$. Furthermore let $BH(\mathcal{C}) =_{\text{def}} \bigcup_{i\geq 0} \mathcal{C}(i)$.

With $\mathcal{C} = NP$ we obtain the well known boolean hierarchy over NP (see [KSW87, CGH+88]). For convenience we set $NP(0) =_{\text{def}} co\text{-}NP(0) =_{\text{def}} P$ and $BH =_{\text{def}} BH(NP)$. In [CK96] the class $BH^{\Sigma_2^p}$ is denoted by BH_3.

3 A Definition of high and low sets for BH

In [Sch85] $High_k^p$ and Low_k^p sets for the polynomial-time hierarchy were defined as follows:

Definition 1 ([Sch85])

$$High_k^p =_{\text{def}} \{A \in NP | \Sigma_{k+1}^p \subseteq (\Sigma_k^p)^A\}$$

and

$$Low_k^p =_{\text{def}} \{A \in NP | (\Sigma_k^p)^A \subseteq \Sigma_k^p\}$$

Theorem 2 ([Sch85])

1. $Low_0^p = P$
2. $Low_1^p = NP \cap co\text{-}NP$
3. $High_0^p = \{A | A \leq_T^{\text{p}}$ *-complete for NP*$\}$

Theorem 3 ([Sch85]) *Let* $k \geq 0$*:*

1. $Low_k^p \subseteq Low_{k+1}^p$
2. $High_k^p \subseteq High_{k+1}^p$
3. *If* $PH \neq \Sigma_k^p$ *than* $High_k^p \cap Low_k^p = \emptyset$
4. *If* $PH = \Sigma_k^p$ *than* $High_k^p = Low_k^p = NP$

Now we are ready to introduce our high and low sets for the boolean hierarchy. For this purpose we define the sets $NP_A(k)$.

Definition 4 *Let* $A \in NP$. *For* $k \geq 1$ *we define the sets* $NP_A(k)$ *as follows:*

$$\begin{array}{lll} NP_A(1) & =_{\text{def}} & \mathcal{R}_m^p(A) \cup P \\ NP_A(2i) & =_{\text{def}} & NP_A(2i-1) \wedge co\text{-}NP \quad (i \geq 1) \\ NP_A(2i+1) & =_{\text{def}} & NP_A(2i) \vee NP \quad\quad\quad (i \geq 1) \end{array}$$

The following properties of $NP_A(k)$ sets can be proven by induction.

Proposition 5 *For* $k \geq 1$ *the inclusion* $NP_A(k) \subseteq NP_A(k+1)$ *holds.*

The proposition below shows the relations between the classes $NP(k)$, $co\text{-}NP(k)$ and $NP_A(k)$.

Proposition 6 *For each* $k \geq 0$ *and any set* $A \in NP$*:*

$$co\text{-}NP(k) \subseteq NP_A(k+1) \subseteq NP(k+1).$$

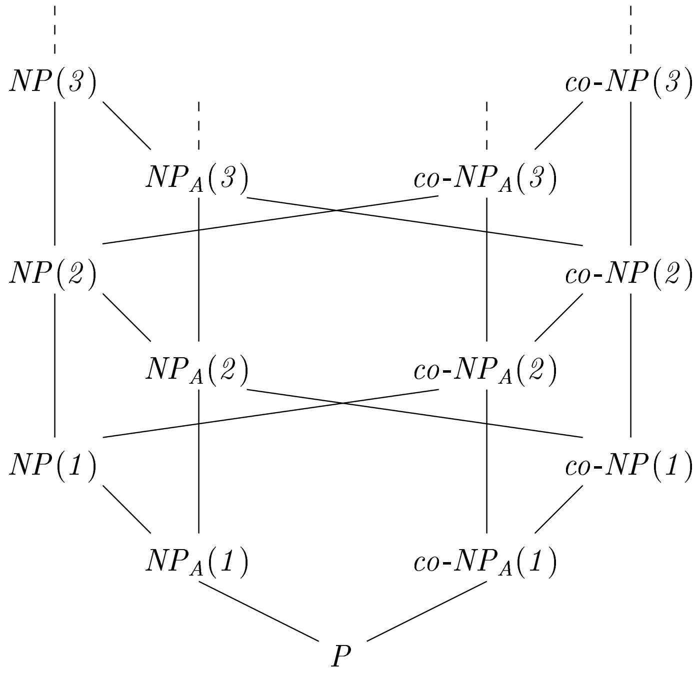

Figure 1: Inclusion structure of $NP(k)$, $co\text{-}NP(k)$, $NP_A(k)$ and $co\text{-}NP_A(k)$

Proposition 7 *If $A \leq_m^{\mathrm{p}} B$ and $k \geq 1$ then $NP_A(k) \subseteq NP_B(k)$.*

Figure 1 shows the inclusion structure of $NP(k)$, $co\text{-}NP(k)$, $NP_A(k)$ and $co\text{-}NP_A(k)$ for $k \geq 1$. Proposition 6 suggests the following definition of high and low sets[1] for the boolean hierarchy:

Definition 8 *Let $k \geq 0$ and $A \in NP$.*

- $A \in high_k^p \Leftrightarrow_{\mathrm{def}} NP_A(k+1) = NP(k+1)$
- $A \in low_k^p \Leftrightarrow_{\mathrm{def}} co\text{-}NP(k) = NP_A(k+1)$
- $high^p =_{\mathrm{def}} \bigcup_{k \geq 0} high_k^p$
- $low^p =_{\mathrm{def}} \bigcup_{k \geq 0} low_k^p$

The following obvious fact shows that the defined highness and lowness properties are compatible with polynomial time m-reducibility.

Proposition 9 *For $k \geq 0$ and $A \leq_m^{\mathrm{p}} B$:*

- *If $A \in high_k^p$ and $B \in NP$ then $B \in high_k^p$.*
- *If $B \in low_k^p$ then $A \in low_k^p$.*

The next lemma demonstrates, that the inclusion structure of $high_k^p$ sets (low_k^p sets, resp.) is similar to that of $High_k^p$ sets (Low_k^p sets, resp.).

[1]Other possible definitions of high and low sets for the boolean hierarchy can be found in [Rei96].

Lemma 10 *For each $k \geq 0$ the inclusions $high_k^p \subseteq high_{k+1}^p$ and $low_k^p \subseteq low_{k+1}^p$ hold.*

Proof: Let $A \in high_k^p$, so $NP(k+1) = NP_A(k+1)$. If k is even, than

$$\begin{aligned} NP_A(k+2) &= NP_A(k+1) \wedge co\text{-}NP \\ &= NP(k+1) \wedge co\text{-}NP \\ &= NP(k+2) \end{aligned}$$

Let $A \in low_k^p$, so $co\text{-}NP(k) = NP_A(k+1)$. If k is even, than

$$\begin{aligned} NP_A(k+2) &= NP_A(k+1) \wedge co\text{-}NP \\ &= co\text{-}NP(k) \wedge co\text{-}NP \\ &= co\text{-}NP(k+1) \end{aligned}$$

The odd cases can be proven similarly. ❑

The next result exhibts the nature of the classes low_0^p, low_1^p and $high_0^p$.

Theorem 11

1. $low_0^p = P$
2. $low_1^p = NP \cap co\text{-}NP$
3. $high_0^p = \{A | A \text{ is } \leq_m^p\text{-complete for } NP\}$

Proof: Follows easily from the definition. We prove statement 2, the other proofs are similar.

$$\begin{aligned} A \in low_1^p &\Leftrightarrow A \in NP \text{ and } NP_A(2) \subseteq co\text{-}NP(1) \\ &\Leftrightarrow A \in NP \text{ and } (\mathcal{R}_m^p(A) \cup P) \wedge co\text{-}NP \subseteq co\text{-}NP \\ &\Leftrightarrow A \in NP \cap co\text{-}NP \end{aligned}$$

❑

Corollary 12

1. $low_0^p = Low_0^p$
2. $low_1^p = Low_1^p$
3. $high_0^p \subseteq High_0^p$

The next theorem shows that it is very unlikely that a set can be simultaneously low^p and $high^p$ (See figure 2 for an illustration of $high_k^p$ and low_k^p classes).

Theorem 13

1. *$high^p \cap low^p \neq \emptyset$ if and only if the boolean hierarchy collapse.*
2. *$high_k^p \cap low_k^p \neq \emptyset$ if and only if $BH = NP(k)$.*

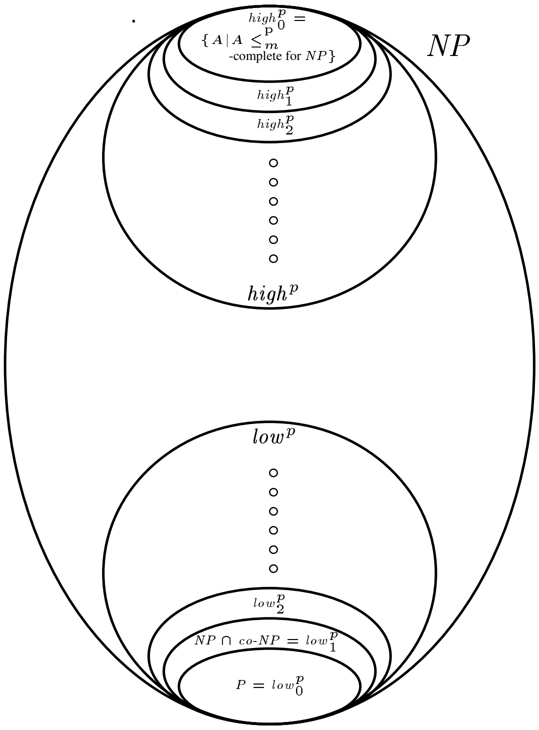

Figure 2: The structure of $high_k^p$ and low_k^p classes.

Proof: First we proof the second statment

$$\begin{array}{rcl} A \in high_k^p \cap low_k^p & \Leftrightarrow & co\text{-}NP(k) = NP_A(k+1) = NP(k+1) \\ & \Leftrightarrow & NP(k) = co\text{-}NP(k) \\ & \Leftrightarrow & NP(k) = BH \end{array}$$

and use it to proof the first one:

$$\begin{array}{rcl} high^p \cap low^p \neq \emptyset & \Leftrightarrow & \text{there exists a } k \geq 0 \text{ s.t. } high_k^p \cap low_k^p \neq \emptyset \\ & \Leftrightarrow & NP(k) = BH \text{ for some } k \geq 0 \\ & \Leftrightarrow & \text{The boolean hierarchy collapses} \end{array}$$

❏

By theorem 11 it is obvious that the classes low_0^p and low_1^p are closed under complement. The next theorem shows the consequences of other low_k^p and $high_k^p$ classes to be closed under complement. The behaviour of low_k^p and $high_k^p$ classes differs from Low_k^p and $High_k^p$ classes, which are obviously closed under complement.

Theorem 14

1. *For* $k \geq 2$, $low_k^p = co\text{-}low_k^p \Leftrightarrow low_k^p = NP \cap co\text{-}NP = low_1^p$
2. *For* $k \geq 0$, $high_k^p = co\text{-}high_k^p \Leftrightarrow NP = co\text{-}NP$.

Proof:

1. Obvious because of $NP \cap co\text{-}NP \subseteq low_k^p \subseteq NP$ for $k \geq 2$.

2. "$\Rightarrow$" We know $\mathbf{SAT} \in high_0^p$ and $high_0^p \subseteq high_k^p \subseteq co\text{-}high_k^p \subseteq co\text{-}NP$. Hence $NP = co\text{-}NP$.

   "$\Leftarrow$" ($k \geq 1$) If $NP = co\text{-}NP$ then $NP(k) = co\text{-}NP(k) = NP$ for all $k \geq 1$. Thus we have $co\text{-}NP(k) = NP_A(k+1) = NP(k+1)$ for all $A \in NP$, i.e. $NP = high_k^p$. Hence $high_k^p = NP = co\text{-}NP = co\text{-}high_k^p$.
   ($k = 0$) If A is in $co\text{-}high_0^p$ then, by theorem 11, $\overline{A}$ is $\leq_m^{\mathrm{p}}$-complete for $NP = co\text{-}NP$. Hence A is $\leq_m^{\mathrm{p}}$-complete for NP, i.e. $A \in high_k^p$.

❑

4 low_k^p sets are low for $BH^{\Sigma_2^p}$

In this section we show, that our low_k^p sets are contained in Low_3^p. To prove this, we use similar *hard* and *easy* arguments as introduced in [CK96].

Definition 15 *The projection function that selects the i^{th} to the j^{th} component of a k tuple will be denoted by:*

$$\langle x_1, \ldots, x_k \rangle_{(i,j)} = \langle x_i, \ldots, x_j \rangle$$

The projection function of the i^{th} component will be denoted by the shortcut:

$$\langle x_1, \ldots, x_k \rangle_{(i)} =_{\mathrm{def}} \langle x_1, \ldots, x_k \rangle_{(i,i)}.$$

Definition 16 *Since we need $\leq_m^{\mathrm{p}}$-complete languages for $NP_A(k)$ we construct them as follows:*

$$\begin{array}{lll} L_{NP_A(1)} & =_{\mathrm{def}} & A \\ L_{NP_A(2i)} & =_{\mathrm{def}} & \{\langle x_1, \ldots, x_{2i} \rangle | \langle x_1, \ldots, x_{2i-1} \rangle \wedge x_{2i} \in \overline{\mathbf{SAT}}\} \\ L_{NP_A(2i+1)} & =_{\mathrm{def}} & \{\langle x_1, \ldots, x_{2i+1} \rangle | \langle x_1, \ldots, x_{2i} \rangle \vee x_{2i+1} \in \mathbf{SAT}\} \end{array}$$

Proposition 17

(i) $L_{NP_A(k)}$ *is $\leq_m^{\mathrm{p}}$-complete for $NP_A(k)$.*

(ii) $L_{NP_{\mathbf{SAT}}(k)}$ *is $\leq_m^{\mathrm{p}}$-complete for $NP(k)$.*

(iii) $\overline{L_{NP_{\mathbf{SAT}}(k)}}$ *is $\leq_m^{\mathrm{p}}$-complete for $co\text{-}NP(k)$.*

Let $L_{NP(k)} =_{\mathrm{def}} L_{NP_{\mathbf{SAT}}(k)}$ and $L_{co\text{-}NP(k)} =_{\mathrm{def}} \overline{L_{NP_{\mathbf{SAT}}(k)}}$.

Proposition 18 *Let $A \in low_k^p$ and $k \geq 1$. With Definition 8 the inclusion $NP_A(k) \subseteq NP_A(k+1) \subseteq co\text{-}NP(k)$ holds. So there exists a function $h \in FP$ with $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via h.*

Definition 19 *Let $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via $h \in FP$. We call $\langle 1^m, x_1, \ldots, x_j \rangle$ a* hard sequence *with respect to h for the length m iff $j = 0$ or* *(i)* $1 \leq j \leq k-1$

(ii) $|x_j| \leq m$

(iii) $x_j \in \overline{\mathbf{SAT}}$

(iv) $\langle 1^m, x_1, \ldots, x_{j-1} \rangle$ *is a hard sequence with respect to* h.

(v) $\forall y_1, \ldots, y_{k-j} \in \Sigma^{\leq m}\ h(\langle y_1, \ldots, y_{k-j}, x_j, \ldots, x_1 \rangle)_{(k-j+1)} \in \overline{\mathbf{SAT}}$.

We will call j order *of the hard sequence* $\langle 1^m, x_1, \ldots, x_j \rangle$*. If there doesn't exist a hard sequence of order* $j' > j$*, we say* $\langle 1^m, x_1, \ldots, x_j \rangle$ *is a* maximal hard sequence *with respect to* h *for the length* m*.*

The next lemma shows, that we can reduce $NP_A(k-j)$ to $co\text{-}NP(k-j)$ by using a hard sequence of order j, for a set $A \in low_k^p$.

Lemma 20 *Let* $A \in low_k^p$ *and therefore* $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ *via* $h \in FP$*. Let* $\langle 1^m, x_1, \ldots, x_j \rangle$ *be a hard sequence with respect to* h *of order* j *and* $0 \leq j \leq k-1$*.*
Then the following equivalence holds for all $y_1, \ldots, y_{k-j} \in \Sigma^{\leq m}$*:*

$$\langle y_1, \ldots, y_{k-j} \rangle \in L_{NP_A(k-j)} \Leftrightarrow h(\langle y_1, \ldots, y_{k-j}, x_j, \ldots, x_1 \rangle)_{(1,k-j)} \in L_{co\text{-}NP(k-j)}$$

Proof: We show this lemma by induction on j.
(IB) Let $j = 0$. So we obtain the following equivalence:

$$\begin{aligned}
\langle y_1, \ldots, y_{k-j} \rangle \in L_{NP_A(k-j)} &\Leftrightarrow \langle y_1, \ldots, y_k \rangle \in L_{NP_A(k)} \\
&\Leftrightarrow h(\langle y_1, \ldots, y_k \rangle) \in L_{co\text{-}NP(k)} \\
&\Leftrightarrow h(\langle y_1, \ldots, y_k \rangle)_{(1,k)} \in L_{co\text{-}NP(k)} \\
&\Leftrightarrow h(\langle y_1, \ldots, y_{k-j} \rangle)_{(1,k-j)} \in L_{co\text{-}NP(k-j)}
\end{aligned}$$

(IS) Now let $j > 0$ and $\langle 1^m, x_1, \ldots, x_{j+1} \rangle$ be a hard sequence with respect to h. Then we know, that $\langle 1^m, x_1, \ldots, x_j \rangle$ is also a hard sequence with respect to h. Using the induction hypothesis we obtain:

$$\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1} \rangle \in L_{NP_A(k-j)} \Leftrightarrow h(\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1}, \ldots, x_1 \rangle)_{(1,k-j)} \in L_{co\text{-}NP(k-j)}$$

$\underline{(k-j)\ \text{even}:}$

$$\left(\begin{array}{ll} & \langle y_1, \ldots, y_{k-(j+1)} \rangle \in L_{NP_A(k-(j+1))} \\ \wedge & x_{j+1} \in \overline{\mathbf{SAT}} \end{array} \right)$$

$$\Leftrightarrow \left(\begin{array}{ll} & h(\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1}, \ldots, x_1 \rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))} \\ \vee & h(\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1}, \ldots, x_1 \rangle)_{(k-j)} \in \mathbf{SAT} \end{array} \right)$$

Since we know, that $x_{j+1} \in \overline{\mathbf{SAT}}$ and $h(\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1}, \ldots, x_1 \rangle)_{(k-j)} \notin \mathbf{SAT}$ (see definition 19) we get:

$$\langle y_1, \ldots, y_{k-(j+1)} \rangle \in L_{NP_A(k-(j+1))} \Leftrightarrow h(\langle y_1, \ldots, y_{k-(j+1)}, x_{j+1}, \ldots, x_1 \rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))}$$

$\underline{(k-j) \text{ odd}:}$

$$\begin{pmatrix} & \langle y_1,\ldots,y_{k-(j+1)}\rangle \in L_{NP_A(k-(j+1))} \\ \vee & x_{j+1} \in \mathbf{SAT} \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} & h(\langle y_1,\ldots,y_{k-(j+1)},x_{j+1},\ldots,x_1\rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))} \\ \wedge & h(\langle y_1,\ldots,y_{k-(j+1)},x_{j+1},\ldots,x_1\rangle)_{(k-j)} \in \overline{\mathbf{SAT}} \end{pmatrix}$$

or

$$\begin{pmatrix} & \langle y_1,\ldots,y_{k-(j+1)}\rangle \notin L_{NP_A(k-(j+1))} \\ \wedge & x_{j+1} \in \overline{\mathbf{SAT}} \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} & h(\langle y_1,\ldots,y_{k-(j+1)},x_{j+1},\ldots,x_1\rangle)_{(1,k-(j+1))} \notin L_{co\text{-}NP(k-(j+1))} \\ \vee & h(\langle y_1,\ldots,y_{k-(j+1)},x_{j+1},\ldots,x_1\rangle)_{(k-j)} \in \mathbf{SAT} \end{pmatrix}$$

As a consequence of both cases:

$$\langle y_1,\ldots,y_{k-(j+1)}\rangle \in L_{NP_A(k-(j+1))} \Leftrightarrow h(\langle y_1,\ldots,y_{k-(j+1)},x_{j+1},\ldots,x_1\rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))}$$

❏

The next two lemmas show, that we can use (maximal) hard sequences to reduce $\overline{A}$ to $\mathbf{SAT}$. In this case the hard sequence is used as an *advice* (See [KL80] and [Wag87] for more information about advices.) by the reduction function.

Lemma 21 *Let $A \in low_k^p$, $h \in FP$ and $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via h. Moreover let $\langle 1^m, x_1, \ldots, x_{k-1}\rangle$ a (maximal) hard sequence with respect to h.*
Then for all $y \in \Sigma^{\leq m}$:

$$y \in \overline{A} \Leftrightarrow h(\langle y, x_{k-1},\ldots,x_1\rangle)_{(1)} \in \mathbf{SAT}$$

Proof: This can be proofed by using lemma 20, because $L_{NP_A(1)} = A$ and $L_{co\text{-}NP(1)} = \overline{\mathbf{SAT}}$. ❏

Lemma 22 *Let $A \in low_k^p$, $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via $h \in FP$, $A \leq_m^{\mathrm{p}} \mathbf{SAT}$ via $g \in FP$ and $|g(x)| \leq r(|x|)$, where r is a polynom. Furthermore let $\langle 1^{r(m)}, x_1, \ldots, x_j\rangle$ be a maximal hard sequence with respect to h with $j < k-1$.*
Then for all $y \in \Sigma^{\leq m}$:

$$y \in \overline{A} \Leftrightarrow \exists y_1,\ldots,y_{k-(j+1)} \in \Sigma^{\leq r(m)} h(\langle y_1,\ldots,y_{k-(j+1)}, g(y), x_j,\ldots,x_1\rangle)_{(k-j)} \in \mathbf{SAT}$$

Proof: "$\Rightarrow$" If $y \in \overline{A}$ we know that $g(y) \in \overline{\mathbf{SAT}}$. Since $\langle 1^{r(m)}, x_1, \ldots, x_j\rangle$ is a maximal hard sequence with respect to h for the length $r(m)$ it follows from definition 19, that $\langle 1^{r(m)}, x_1, \ldots, x_j, g(y)\rangle$ can't be a hard sequence for the length $r(m)$. With $j+1 \leq k-1$, $|g(y)| \leq r(m)$ and $g(y) \in \overline{\mathbf{SAT}}$ we conclude:

$$\exists y_1,\ldots,y_{k-(j+1)} \in \Sigma^{\leq r(m)} h(\langle y_1,\ldots,y_{k-(j+1)}, g(y), x_j,\ldots,x_1\rangle)_{(k-j)} \in \mathbf{SAT}$$

(Condition (v) for hard sequences)

"$\Leftarrow$" Assume that there exists $y_1,\ldots,y_{k-(j+1)} \in \Sigma^{\leq r(m)}$ with $h(\langle y_1,\ldots,y_{k-(j+1)}, g(y), x_j,\ldots,x_1\rangle) \in$

$\mathbf{SAT}$ and a maximal hard sequence $\langle 1^{r(m)}, x_1, \ldots, x_j\rangle$ with respect to h. Because of lemma 20 we know:

$$\langle y_1, \ldots, y_{k-(j+1)}, g(y)\rangle \in L_{NP_A(k-j)} \Leftrightarrow h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(k-j)} \in \mathbf{SAT}$$

Now we have to distinguish two cases:
<u>$(k-j)$ even :</u>

$$\left(\begin{array}{ll} & \langle y_1, \ldots, y_{k-(j+1)}\rangle \in L_{NP_A(k-(j+1))} \\ \wedge & g(y) \in \overline{\mathbf{SAT}} \end{array}\right)$$

$$\Leftrightarrow \left(\begin{array}{ll} & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))} \\ \vee & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(k-j)} \in \mathbf{SAT} \end{array}\right)$$

<u>$(k-j)$ odd :</u>

$$\left(\begin{array}{ll} & \langle y_1, \ldots, y_{k-(j+1)}\rangle \in L_{NP_A(k-(j+1))} \\ \vee & g(y) \in \mathbf{SAT} \end{array}\right)$$

$$\Leftrightarrow \left(\begin{array}{ll} & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(1,k-(j+1))} \in L_{co\text{-}NP(k-(j+1))} \\ \wedge & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(k-j)} \in \overline{\mathbf{SAT}} \end{array}\right)$$

or

$$\left(\begin{array}{ll} & \langle y_1, \ldots, y_{k-(j+1)}\rangle \notin L_{NP_A(k-(j+1))} \\ \wedge & g(y) \in \overline{\mathbf{SAT}} \end{array}\right)$$

$$\Leftrightarrow \left(\begin{array}{ll} & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(1,k-(j+1))} \notin L_{co\text{-}NP(k-(j+1))} \\ \vee & h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(k-j)} \in \mathbf{SAT} \end{array}\right)$$

Since $h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1\rangle)_{(k-j)} \in \mathbf{SAT}$ the right sides of both equivalences are true. So we know, that the left sides must also be true. This shows that $g(y) \in \overline{\mathbf{SAT}}$ and therefore $y \in \overline{A}$. ❏

Definition 23 *Let L_1 (L_2) be a $\leq_m^{\mathrm{p}}$-complete language for NP^A (NP^{NP^A}).*

Lemma 24 *Let $A \in low_k^p$, $A \leq_m^{\mathrm{p}} \mathbf{SAT}$ via $g \in FP$ and r a polynom with $|g(w)| \leq r(|w|)$. Then there exists a NP machine N_1, a polynom p_1, $m \geq r(p_1(n))$ and a maximal hard sequence $\langle 1^m, x_1, \ldots, x_j\rangle$ for the length m with respect to h, where $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via $h \in FP$.*
It holds:

$$\forall w(|w| \leq n \rightarrow (w \in L_1 \Leftrightarrow N_1(w, \langle 1^m, x_1, \ldots, x_j\rangle) \textit{ accepts }))$$

Proof: Let M_1^A be a TM with $L_1 = L(M_1^A)$ and p_1 a polynom, where p_1 is a upper bound for the running time of M_1^A. It is evident, that M_1^A can only ask questions of at most length $p_1(|w|)$ to the oracle A.
Now N_1 can simulate M_1^A as follows:
Input: $(w, \langle 1^m, x_1, \ldots, x_j\rangle)$ with $m \geq r(p_1(|w|)$

1. Simulate M_1^A until it asks a question to its oracle A.
2. If M_1^A asks its oracle "$y \in A$?", then N_1 guess nondeterministically the answer.
3. If N_1 assumes $y \in A$, then N_1 checks this in polynomial time. N_1 rejects if $y \in A$ couldn't be proofed, because N_1 did the wrong assumption. In the other case we continue the simulation of M_1^A with $y \in A$.
4. If N_1 assumes $y \notin A$ then
   (a) If $j = k - 1$ calculate
   $$F = h(\langle y, x_{k-1}, \ldots, x_1 \rangle)_{(1)}$$
   (See lemma 21.)
   (b) If $0 \leq j < k - 1$ guess $y_1, \ldots, y_{k-(j+1)} \in \Sigma^{\leq m}$ and calculate
   $$F = h(\langle y_1, \ldots, y_{k-(j+1)}, g(y), x_j, \ldots, x_1 \rangle)_{(k-j)}$$
   (See lemma 22.)

   Now we test the satisfiability of F in polynomial time. If $F \in \mathbf{SAT}$, then $y \notin A$ was the right guess and we continue the simulation of M_1^A with answer "no".
   If $F \notin \mathbf{SAT}$, then $y \in \overline{A}$ was the wrong guess. In this case N_1 rejects.

Using the lemmas 21 and 22 we see, that $y \in \overline{A} \Leftrightarrow F \in \mathbf{SAT}$. This means, that N_1 finds a satisfiable F in step 4 iff $y \in A$ can't be proofed in step 3.
Moreover the length m is big enough for lemma 21 and 22, because the reduction used in step 4b can at most enlarge y to length $r(p_1(|w|))$. So we obtain

$$w \in L_1 \Leftrightarrow N_1(w, \langle 1^m, x_1, \ldots, x_j \rangle) \; accepts$$

❑

The next lemma shows the existence of a Σ_2^p machine for the language L_2. This machine only needs the order and the length of a maximal hard sequence as an additional input.

Lemma 25 *Let $A \in low_k^p$ and $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$ via $h \in FP$. Then there exists a $NP^{\mathbf{SAT}}$ machine $N_2^{\mathbf{SAT}}$ and a polynom p_2, such that for $m \geq p_2(n)$ and j, where j is the order of a maximal hard sequence with respect to h for the length m, the following holds:*

$$\forall w (|w| \leq n \rightarrow (w \in L_2 \Leftrightarrow N_2^{\mathbf{SAT}}(w, j, 1^m) \; accepts))$$

If j is greater than the order of a maximal hard sequence with respect to h for the length m than:

$$\forall w \in \Sigma^* : N_2^{\mathbf{SAT}}(w, j, 1^m) \; rejects$$

Proof: Let $L_2 = L(M_2^{L_1})$ and s a polynom, giving a upper bound for the running time of $M_2^{L_1}$.
It is evident, that $M_2^{L_1}$ can ask question of at most length $s(n)$. Let $p_2 =_{\mathrm{def}} r(p_1(s(n)))$, where r and p_1 are the polynoms used in lemma 24. The polynom p_2 gives us a upper bound for the length of questions to A, since the enlargement while A is reduced to $\mathbf{SAT}$ is bounded by r.
Definition 19 tells us, that the language of hard sequences is in $co\text{-}NP$. So we know, that the question $\exists \langle 1^m, x_1, \ldots, x_j \rangle$, such that $\langle 1^m, x_1, \ldots, x_j \rangle$ is a maximal hard sequence with $m \geq p_2(n)$, can be answered by a Σ_2^p-machine:

1. Guess a sequence $\langle 1^m, x_1, \ldots, x_j\rangle$.

2. Ask the NP-oracle, if this sequence is hard (co-NP-question).

3. if $\langle 1^m, x_1, \ldots, x_j\rangle$ isn't a hard sequence, then reject.

If j was the maximal order of a hard sequence for the length m, than $\langle 1^m, x_1, \ldots, x_j\rangle$ is a maximal hard sequence.
Now $N_2^{\mathbf{SAT}}$ simulates $M_2^{L_1}$ step by step until $M_2^{L_1}$ asks its oracle "$y \in L_1$?". In this case $N_2^{\mathbf{SAT}}$ asks the $\mathbf{SAT}$ oracle "$(y, \langle 1^m, x_1, \ldots, x_j\rangle) \in L_1$?" (See lemma 24). This means:

$$w \in L_2 \Leftrightarrow N_2^{\mathbf{SAT}}(w, j, 1^m)\ accepts$$

Possibly there are more than one maximal hard sequence for the length m. So we get several computation branches for each maximal hard sequence. All this branches have the same acceptance behavior, because the behavior of $N_2^{\mathbf{SAT}}$ doesn't depend on a certain hard sequence.
If j is greater than the order of a maximal hard sequence, $N_2^{\mathbf{SAT}}$ can't find a maximal hard sequence in step 1. Because of this, any input will be rejected. ❏

Definition 26 *Let $h \in FP$ a reduction function for $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$. We define the set T as follows:*

$$T =_{\mathrm{def}} \{(1^m, j) | \exists x_1, \ldots, x_j \in \Sigma^{\leq m} \textit{ and } \langle 1^m, x_1, \ldots, x_j\rangle \textit{ is a hard sequence with respect to } h\}$$

Lemma 27 *Let $h \in FP$ a reduction function for $L_{NP_A(k)} \leq_m^{\mathrm{p}} L_{co\text{-}NP(k)}$, than $T \in \Sigma_2^p$.*

Proof: The language of hard strings is contained in co-NP (see definition 19). A Σ_2^p-machine guesses $x_1, \ldots, x_j \in \Sigma^{\leq m}$ and checks with its oracle, whether $x_1, \ldots, x_j$ is a hard sequence with respect to h. ❏

Theorem 28 *Let $A \in low_k^p$, then the following inclusion holds:*

$$NP^{NP^A} \subseteq P_{\|}^{\Sigma_2^p}[2k]$$

Proof: If j is the order of a maximal hard sequence, we showed in lemma 25, that there exists a TM $N_2^{\mathbf{SAT}}$ and a polynom p_2, such that for $m \geq p_2(|w|)$:

$$w \in L_2 \Leftrightarrow N_2^{\mathbf{SAT}}(w, j, 1^m)\ accepts$$

holds.
Using that $T \in \Sigma_2^p$, we need k parallel questions to determine the order j of a maximal hard sequence and k additional parallel questions for asking $N_2(w, 0, 1^m), \ldots, N_2(w, k-1, 1^m)$. We accept, if j is the order of a maximal hard sequence and the question $N_2(w, j, 1^m)$ is answered with "yes". In all other cases we reject. ❏

4.1 Some consequences for low_k^p sets

The next corollaries show lowness consequences for low_k^p sets.

Corollary 29 *Let $k \geq 0$ and $A \in low_k^p$, then*

$$P^{NP^{NP^A}} \subseteq P^{NP^{NP}} \textit{ (A is low for } \Delta_3^p\textit{)}$$

Proof: With theorem 28 we obtain:

$$P^{NP^{NP^A}} \subseteq P^{P^{NP^{NP}}[O(1)]} \subseteq P^{NP^{NP}}$$

❑

Corollary 30 *Let $k \geq 0$ and $A \in low_k^p$, then $A \in Low_3^p$.*

Proof: Using corollary 29 and theorem 28 we get:

$$(\Sigma_3^p)^A \subseteq NP^{P^{NP^{NP^A}}} \subseteq NP^{P^{NP^{NP}}} \subseteq \Sigma_3^p$$

❑

Corollary 31 *Let $k \geq 1$, $l \geq 1$ and $A \in low_k^p$, then*

$$P_{||}^{NP^{NP^A}}[l] \subseteq P_{||}^{NP^{NP}}[2kl]$$

Proof: Using theorem 28 we deduce:

$$P_{||}^{NP^{NP^A}}[l] \subseteq P_{||}^{P_{||}^{NP^{NP}}[2k]}[l] \subseteq P_{||}^{NP^{NP}}[2kl]$$

❑

Since the query hierarchy and the boolean hierarchy over Σ_2^p are intertwined (see e.g. [KSW87]), we obtain the following corollary.

Corollary 32 *Let $A \in low_k^p$, $k \geq 1$ and $l \geq 1$, then:*

$$\Sigma_2^p(l)^A \subseteq \Sigma_2^p(2kl + 1)$$

Proof: We can proof this with corollary 31:

$$\Sigma_2^p(l)^A \subseteq P_{||}^{NP^{NP^A}}[l] \subseteq P_{||}^{NP^{NP}}[2kl] \subseteq \Sigma_2^p(2kl + 1)$$

❑

Corollary 33 *If $A \in low_k^p$, then $BH(\Sigma_2^p)^A = BH(\Sigma_2^p)$.*

5 Obtaining better results?

In this section we show that it is very unlikely that we can use Chang and Kadin's ([CK96]) technique to improve our results. In detail we show that the polynomial-time hierarchy collapses to Σ_4^p if we would be able to assimilate one additional lemma.
First we shortly introduce the concept of *advices* originally investigated in [KL80].

Definition 34 ([BDG90]) *A function from* $\mathbb{N}$ *to* Σ^* *will be called* advice function. *Let* $\mathcal{C}$ *be a class of sets, and let* $\mathcal{F}$ *be a class of advice functions. The class* $\mathcal{C}/\mathcal{F}$ *is the class of all sets* B *for which there exists a set* $A \in \mathcal{C}$ *and a function* $f \in \mathcal{F}$ *such that* $B = \{x | \langle x, f(|x|)\rangle \in A\}$.

Definition 35 *With poly we denote the class of polynomial length bounded advice functions.*

$$f \in \textit{poly} \Leftrightarrow_{\text{def}} \textit{There exists a polynom } p \textit{ and for all } n \geq 0,\ |f(n)| \leq p(n)$$

The following lemma can be directly proven by using definition 34.

Lemma 36 *For any class* $\mathcal{C}$ *of languages:*

$$\mathcal{C}/poly/poly = \mathcal{C}/poly$$

Corollary 37 ([Sch85]) $PH/poly = \Sigma_k^p/poly$ *implies* $PH = \Sigma_{k+2}^p$

Theorem 38 *If* $\Pi_2^p \subseteq \Sigma_2^p/poly$ *than the polynomial-time hierarchy collapses to* Σ_4^p.

Proof: Using the lemma 36 we obtain:

$$\begin{array}{rcl} \Pi_2^p \subseteq \Sigma_2^p/poly & \Rightarrow & \Pi_2^p/poly \subseteq \Sigma_2^p/poly/poly \\ & \Leftrightarrow & \Pi_2^p/poly \subseteq \Sigma_2^p/poly \\ & \Leftrightarrow & PH/poly = \Sigma_2^p/poly \end{array}$$

Using corallary 37 we know that the polynomial-time hierarchy collapses to Σ_4^p. ❑

Now we show that is is unlikely that we can't use Chang and Kadin's technique to show better results for our low^p sets.
We would have to proof a similar lemma to lemma 8 in [CK96]. Proof that there exists a nondeterministic polynomial-time oracle machine N such that:

$$w \in \overline{L_2} \Leftrightarrow N^{\mathbf{SAT}}(w, j, 1^m) \text{ accepts}$$

The consequence would be $\Pi_2^p \subseteq \Pi_2^{p\,A} \subseteq \Sigma_2^p/poly$ since any maximal hard sequence can be understood as a polynomially length bounded advice.
Now Theorem 38 shows the collapse of PH to Σ_4^p.

References

[BDG90] J. L. BALCÁZAR, J. DÍAZ, AND J. GABARRÓ. *Structural Complexity I*. Springer-Verlag, first edition, 1990.

[CGH+88] J.-Y. Cai, T. Gundermann, J. Hartmanis, L. A. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232 – 1252, December 1988.

[CGH+89] J.-Y. Cai, T. Gundermann, J. Hartmanis, L. A. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95 – 111, February 1989.

[CK96] R. Chang and J. Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340, April 1996.

[Coo74] S. B. Cooper. Minimal pairs and high recursively enumerable degree. *Journal of Symbolic Logic*, 39:655 – 660, 1974.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263 – 1282, December 1988.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM STOC*, pages 302 – 309. ACM, 1980.

[KSW87] J. Köbler, U. Schöning, and K. W. Wagner. The difference and truth-table hierarchies for NP. *Theoretical Informatics and Applications*, 21(4):419 – 435, 1987.

[Rei96] S. Reith. Low- und High-Mengen in der Booleschen Hierarchie. Master's thesis, University of Würzburg, 1996.

[Sch83] U. Schöning. A low and a high hierarchy within NP. *Journal of Computer and System Sciences*, 27:14 – 28, 1983.

[Sch85] U. Schöning. *Complexity and Structure*, volume 211 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[Soa74] R. I. Soare. Automorphisms of the lattice of recursively enumerable sets. *Bulletin of the American Mathematical Society*, 80:53 – 58, 1974.

[Wag87] K. W. Wagner. Number-of-query hierachies. Technical Report 158, University of Augsburg, 1987.

[WW85] G. Wechsung and K. Wagner. On the boolean closure of NP, manuscript. Extended abstract as: Wechsung, G., On the boolean closure of NP. In *Proc. of the Conference of Foundations on Computation Theory*, volume 199 of *Lecture Notes in Computer Science*, pages 485–493. Springer Verlag, 1985.