Universität Würzburg • Lehrstuhl für Theoretische Informatik

Diplomarbeit

# Low- und High-Mengen in der Booleschen Hierarchie

Steffen Reith

5. Juli 1996

Für meinen Vater

# Inhaltsverzeichnis

# 1 Einleitung

In [Sch85] wurden „Low- und High-Hierarchien“ innerhalb der Polynomialzeithierarchie untersucht. Vereinfacht ausgedrückt wird eine Sprache $A$ als „High“ für eine Klasse der Polynomialzeithierarchie bezeichnet, wenn diese Klasse durch Hinzunahme von $A$ als Orakel mächtiger wird. Formaler ausgedrückt:

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

Offensichtlich ist es dann auch sinnvoll „Low-Mengen“ zu definieren. Eine Sprache $A$ wird deshalb als „Low“ für eine Klasse der Polynomialzeithierarchie bezeichnet, wenn diese Klasse durch Hinzunahme von $A$ als Orakel nicht mächtiger wird. Formaler ausgedrückt:

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

Nun liegt es nahe, eine entsprechende Definition von „High- und Low-Mengen“ auch in der Booleschen Hierarchie zu untersuchen. Deshalb werden in der vorliegenden Arbeit drei verschiedene „Arten“ von solchen Mengen studiert.
Im Grundlagenkapitel werden Formalismus und die ständig verwendeten Begriffe in stark komprimierter Form eingeführt. Daran schließt sich ein umfangreicheres Kapitel über „unabhängig NP-vollständige“ Mengen an, dessen Ergebnisse für die Untersuchung von „$high^1$- und $low^1$-Mengen“ benötigt werden.
In den darauffolgenden Abschnitten werden dann zwei weitere Definitionen von „High- und Low-Mengen“ untersucht.

# 2 Grundlagen

An dieser Stelle sollen die für das Verständnis dieser Arbeit benötigten Grundlagen in stark vereinfachter Form vermittelt werden. Eine umfassende Einführung in die Grundlagen der theoretischen Informatik findet sich in [Wag94].

Ein *Alphabet* ist eine nichtleere endliche Menge. In dieser Arbeit sollen Alphabete immer mit großen griechischen Buchstaben bezeichnet werden. Insbesondere wird das $\Sigma$ immer als Bezeichnung für ein Alphabet verwendet. Ein *Symbol* oder *Buchstabe* ist ein Element eines Alphabets. Mit $|\Sigma|$ wird die Anzahl der Symbole in einem Alphabet bezeichnet.
Ein *Wort* über einem Alphabet ist die Aneinanderreihung von Symbolen aus einem Alphabet. Die Länge eines Wortes $w$ wird mit $|w|$ notiert. Die Menge aller Wörter über einem Alphabet $\Sigma$ wird mit $\Sigma^*$ bezeichnet. Eine Besonderheit hierbei ist das *leere Wort* $\epsilon$, das 0 Symbole enthält. Es ist leicht nachzuprüfen, daß $\Sigma^*$ mit der Verknüpfung "Konkatenation" ein (nichtkommutatives) Monoid bildet.
Eine Menge $A \subseteq \Sigma^*$ wird als *Sprache* bezeichnet. Das Komplement $\overline{A}$ einer Sprache $A$ wird als $\overline{A} =_{def} \{w \in \Sigma^* | w \notin A\} = \Sigma^* \backslash A$ definiert.

Ein *Entscheidungsproblem* ist dann die Frage, ob ein Wort $x$ zur Menge $A$ gehört oder nicht. Formaler wird das durch die sogenannte *charakteristische Funktion* ausgedrückt:

$$c_A(x) =_{def} \begin{cases} 1 & falls \quad x \in A \\ 0 & sonst \end{cases}$$

Untersucht man nun, unter welchen Ressourceneinschränkungen die charakteristische Funktion einer Sprache $A$ berechnet werden kann, so ist es möglich, diese Sprachen in *Komplexitätsklassen* einzuordnen. Dabei bilden alle die Sprachen eine Komplexitätsklasse $\mathcal{C}$ mit der Schrankenfunktion $f$, deren charakteristische Funktionen mit einer Komplexität berechnet werden können, die durch $f$ beschränkt ist.
Die sogenannte *Komplementkomplexitätsklasse* wird dann wie folgt definiert:

$$co\text{-}\mathcal{C} =_{def} \{A \subseteq \Sigma^* | \overline{A} \in \mathcal{C}\}$$

Die wohl wichtigsten Komplexitäten einer Berechnung sind "Speicherplatzbedarf" und "Zeitbedarf".
Der Zeit- bzw. Speicherbedarf einer Berechnung wird dabei in Abhängigkeit der Länge der Eingabe gemessen. Die Klasse $P$ bezeichnet deshalb alle Sprachen

für die ein Polynom $p$ existiert, so daß ihre charakteristische Funktion für jede Eingabe $x$ in der Zeit $p(|x|)$ berechnet werden kann.

Verallgemeinert man deterministische Berechnungsmodelle und läßt zu, daß zu einem Zeitpunkt mehrere Befehle ausgeführt werden dürfen, dann erhält man sogenannte nichtdeterministische Berechnungsmodelle.
Eine Sprache $A$ ist in $NP$ genau dann, wenn ein Polynom $p$ und eine Menge $B \in P$ existieren, so daß

$$x \in A \leftrightarrow \exists y(|y| \leqslant p(|x|) \wedge (x, y) \in B)$$

Interessanterweise hat sich gezeigt, daß die Klassen $P$ bzw. $NP$ sehr robust gegen den Wechsel des Maschinenmodells sind. Deshalb soll hier aufgrund ihrer Einfachkeit die *Turingmaschine* als Modell verwendet werden.

Offensichtlich gilt $P \subseteq NP$. Sei $PSPACE$ die Klasse aller Sprachen, die mit polynomiell beschränktem Speicherplatz entschieden werden können. Dann ist auch die Inklusion $P \subseteq PSPACE$ einfach einzusehen, da ein polynomiell zeitbeschränkter Algorithmus höchstens polynomiell viel Speicher "konsumieren" kann.

Die Frage $P \neq NP$ bzw. $NP = P$ ist eines der großen Probleme der Komplexitätstheorie. Obwohl eine Fülle von Einzelergebnissen vorliegt, konnte dieses Problem nicht gelöst werden. Auch die Ungleichheit der (nichtdeterministischen) Klassen mit logarithmischem Raumbedarf $L$ bzw. $NL$ konnte bisher nicht gezeigt werden.

Ein weiterer wichtiger Begriff ist die *$m$-Reduzierbarkeit*. Eine Menge $A$ heißt genau dann auf eine Menge $B$ $m$-reduzierbar ($A \leqslant_m B$), wenn eine totale berechenbare Funktion $f$ existiert mit $x \in A \leftrightarrow f(x) \in B$. Diese Funktion $f$ wird als *Reduktionsfunktion* bezeichnet. Es ist offensichtlich, daß die Komplexität dieser Funktion einen entscheidenden Einfluß auf den Reduktionsbegriff hat. Aus diesem Grund muß für jede Reduktion die verwendete Funktion genau spezifiziert werden.
Ist die Funktion $f$ in Polynomialzeit berechenbar ($f \in FP$), dann spricht man von einer "many-one $p$-Reduktion" ($A \leqslant_m^p B \Leftrightarrow_{def} \exists f \in FP\, \forall x(x \in A \leftrightarrow f(x) \in B)$). Andere hier benötigte Reduktionsbegriffe werden noch genauer definiert.
Betrachtet man die Menge aller Sprachen, die sich auf Sprachen aus einer bestimmten Komplexitätsklasse reduzieren lassen, so wird dies als *Abschluß* dieser Klasse unter dem verwendeten Reduktionsbegriff bezeichnet. Besonders interessant sind diejenigen Reduktionsbegriffe und Komplexitätsklassen, für die der Abschluß wieder die Komplexitätsklasse selbst ergibt. Man sagt, die Klasse ist unter dieser Reduktion *abgeschlossen*. So sind z.B. $P$ und $NP$ unter many-one $p$-Reduktion abgeschlossen. Genauer:

$$\mathcal{R}_m^p(NP) =_{def} \{A \subseteq \Sigma^* | \exists B \in NP : A \leqslant_m^p B\} = NP$$

und

$$\mathcal{R}_m^p(P) =_{def} \{A \subseteq \Sigma^* | \exists B \in P : A \leqslant_m^p B\} = P$$

Erweitert man das Modell der Turingmaschine um die Möglichkeit, Fragen der Form "$x \in O$" für eine feste Menge $O$ zu stellen, ohne daß dafür der Berechnungsaufwand der charakteristischen Funktion $c_O$ gezählt wird, so nennt man diese Turingmaschine *Orakelturingmaschine* und $O$ ein *Orakel*. Dieser Name soll andeuten, daß die Maschine die Möglichkeit hat Fragen zu stellen, die immer wahrheitsgemäß mit "ja" oder "nein" beantwortet werden. Für den praktisch denkenden Informatiker stellt sich ein Orakel als Unterprogramm mit dem Rückgabewert "boolean" dar, das ohne Benutzung von Ressourcen aufgerufen werden darf.
Um eine Turingmaschine mit Orakel formal beschreiben zu können, wird die von der Maschine verwendete Orakelmenge im Exponenten notiert. D.h., die Turingmaschine $M^A$ kann zu jeder Zeit der Berechnung das Orakel $A$ befragen. Mit $\mathcal{C}^A$ wird die Menge aller Sprachen notiert, deren charakteristische Funktionen mit Orakelturingmaschinen, die das Orakel $A$ verwenden, berechnet werden können, wobei die für die Komplexitätsklasse $\mathcal{C}$ geltende Ressourcenbeschränkung eingehalten wird. Die Menge aller Sprachen aus $\mathcal{C}^A$, wobei das Orakel $A$ aus der Komplexitätsklasse $\mathcal{A}$ stammt, wird mit $\mathcal{C}^{\mathcal{A}}$ bezeichnet.
Die Menge $P^{NP}$ ist also die Menge aller Sprachen, die von in Polynomialzeit arbeitenden Turingmaschinen entschieden werden, wobei sie ohne zusätzlichen Kostenaufwand Fragen an eine Sprache aus $NP$ stellen dürfen. Möchte man die Anzahl der Fragen an das Orakel begrenzen, dann wird dies in eckigen Klammern dahinter notiert. So wird die Menge aller Sprachen, die von polynomialzeitbeschränkten Turingmaschinen mit genau einer Frage an ein Orakel aus $NP$ entschieden werden können, durch $P^{NP}[1]$ notiert. Offensichtlich gilt hier $P^{NP}[1] \subseteq P^{NP}[\log n] \subseteq P^{NP}$.

## 2.1 Die Polynomialzeithierarchie

Die Klassen der Polynomialzeithierarchie sind induktiv wie folgt definiert:

$$\Theta_0^p =_{def} \Delta_0^p =_{def} \Sigma_0^p =_{def} \Pi_0^p =_{def} P$$

und

$$\begin{aligned}
\Theta_{k+1}^p &=_{def} P^{\Sigma_k^p}[O(\log n)] \\
\Delta_{k+1}^p &=_{def} P^{\Sigma_k^p} \\
\Sigma_{k+1}^p &=_{def} NP^{\Sigma_k^p} \\
\Pi_{k+1}^p &=_{def} co\text{-}NP^{\Sigma_k^p}
\end{aligned}$$

Die Klasse $PH$ ist die Vereinigung aller Klassen der Polynomialzeithierarchie:

$$PH =_{def} \bigcup_{i=0}^{\infty} \Sigma_i^p$$

Leider konnte die Echtheit der Polynomialzeithierarchie bis jetzt noch nicht gezeigt werden. Allerdings deuten sehr viele Einzelergebnisse die Echtheit dieser Hierarchie an. Näheres findet sich z.B. in [BDG90a].

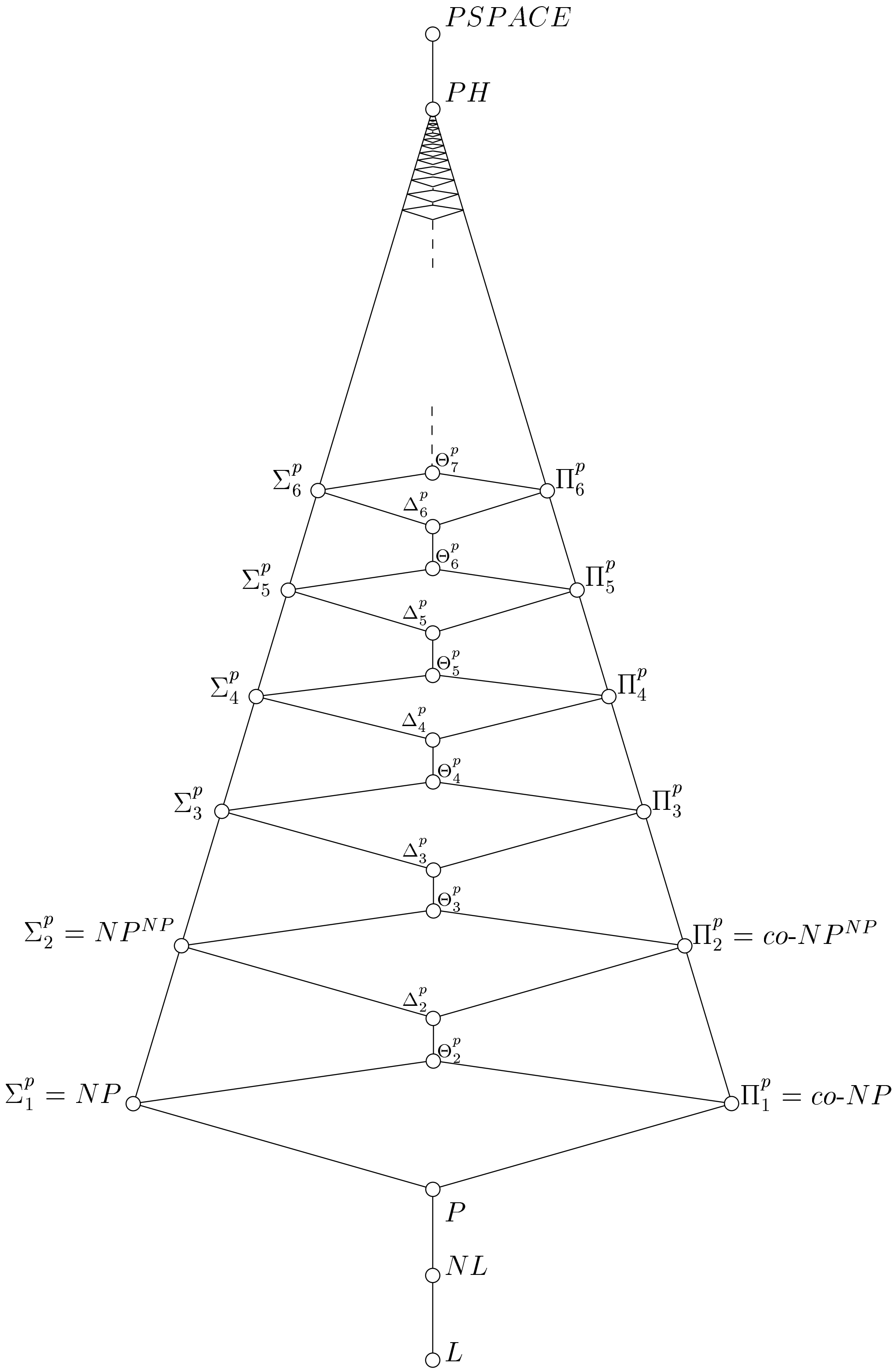


Abbildung 2.1: Die Polynomialzeithierarchie

In Abbildung 2.1 sind die Inklusionsbeziehungen der Klassen der Polynomialzeithierarchie aufgezeichnet.

DEFINITION 2.1.1
*Die „Low- und High-Mengen“ der Polynomialzeithierarchie wurden in [Sch85] wie folgt definiert:*

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

*und*

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

SATZ 2.1.1 ([Sch85])
*Für jedes $k \geqslant 0$ gilt:*

1. *Falls $PH \neq \Sigma_k^p$, dann $High_k^p \cap Low_k^p = \emptyset$*
2. *Falls $PH = \Sigma_k^p$, dann $High_k^p = Low_k^p = NP$*

DEFINITION 2.1.2
*Eine Menge $A$ ist genau dann auf eine Menge $B$* polynomialzeit-turingreduzierbar *($A \leqslant_T^p B$), wenn $A \in P^B$.*

SATZ 2.1.2 ([Sch85])

1. *$Low_0^p = P$*
2. *$Low_1^p = NP \cap co\text{-}NP$*
3. *$High_0^p = \{A \in NP \mid A \text{ ist } \leqslant_T^p \text{-vollständig für} NP\}$*

BEOBACHTUNG 2.1.1 ([SCH85])
*Es sei $k \geqslant 0$, dann gilt*

$$\begin{array}{rcl} Low_k^p & \subseteq & Low_{k+1}^p \\ High_k^p & \subseteq & High_{k+1}^p \end{array}$$

## 2.2 Die Boolesche Hierarchie

Die für die hier vorliegende Arbeit weitaus wichtigere Boolesche Hierarchie soll an dieser Stelle überblickartig präsentiert werden. Um das Verständnis für die Boolesche Hierarchie zu verbessern, sollen auch noch einige grundlegende Sätze und Lemmata vorgestellt werden.

DEFINITION 2.2.1

$$\begin{array}{lcll} \mathcal{C}_1 \wedge \mathcal{C}_2 & =_{def} & \{A \cap B & \mid A \in \mathcal{C}_1, B \in \mathcal{C}_2\} \\ \mathcal{C}_1 \vee \mathcal{C}_2 & =_{def} & \{A \cup B & \mid A \in \mathcal{C}_1, B \in \mathcal{C}_2\} \\ \mathcal{C}_1 \oplus \mathcal{C}_2 & =_{def} & \{A \oplus B & \mid A \in \mathcal{C}_1, B \in \mathcal{C}_2\} \end{array}$$

Die Klassen der Booleschen Hierarchie sind induktiv wie folgt definiert:

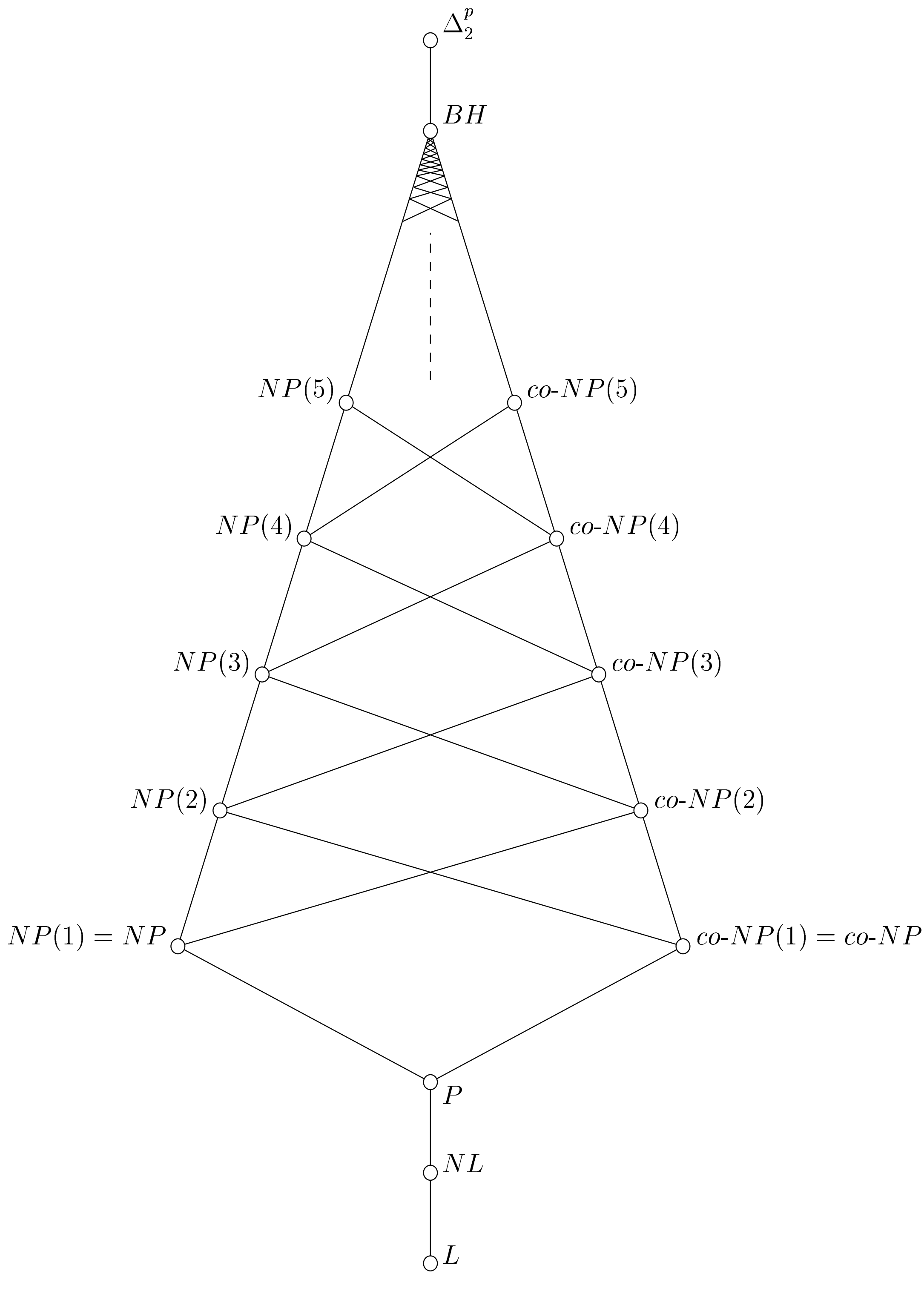


Abbildung 2.2: Die Boolesche Hierarchie

Definition 2.2.2

$$NP(0) =_{def} P$$

*und*

$$\begin{array}{rcll} NP(2i) & =_{def} & NP(2i-1) \wedge co\text{-}NP & (i \geqslant 1) \\ NP(2i+1) & =_{def} & NP(2i) \vee NP & (i \geqslant 0) \end{array}$$

*Die Vereinigung aller Klassen der Booleschen Hierarchie $BH$ ist dann:*

$$BH =_{def} \bigcup_{i=0}^{\infty} NP(i)$$

Diese recht unhandliche Definition der Booleschen Hierarchie kann durch die folgende ersetzt werden:

Definition 2.2.3

$$\begin{array}{rcll} NP(0) & =_{def} & P & \\ NP(1) & =_{def} & NP & \\ NP(i) & =_{def} & NP(i-1) \oplus NP & (i \geqslant 2) \end{array}$$

Die Äquivalenz beider Definitionen wurde in [KSW87] gezeigt.

Lemma 2.2.1 (z.B. in [KSW87])

1. *Die Klassen der Booleschen Hierarchie sind abgeschlossen unter $\leqslant_m^p$-Reduktion, d.h. $\mathcal{R}_m^p(NP(k)) = NP(k)$ und $\mathcal{R}_m^p(co\text{-}NP(k)) = co\text{-}NP(k)$.*

2. *$BH = P^{NP}[O(1)]$.*

Satz 2.2.2 ([CGH+88])
*Für $k \geqslant 1$ gilt:*

$$\begin{array}{rcl} NP(k) = co\text{-}NP(k) & \Leftrightarrow & BH = NP(k) \\ & \Leftrightarrow & NP(k) = NP(k+1). \end{array}$$

Satz 2.2.3 ([Kad88])
*Falls $NP(k) = co\text{-}NP(k)$ für $k \geqslant 1$, dann folgt $PH \subseteq P^{NP^{NP}}[O(\log n)] = \Theta_3^P$.*

# 3 Unabhängig NP-vollständige Mengen

## 3.1 Definition und allgemeine Beobachtungen

DEFINITION 3.1.1
*Das $k$-Tupel $(A_1, A_2, \ldots, A_k)$ heißt $k$-*unabhängig NP-vollständig $\Leftrightarrow_{def}$

- $A_1, A_2, \ldots, A_k \in NP$
- *Für jedes $B_1, B_2, \ldots, B_k \in NP$ gibt es ein $f \in FP$ mit*

$$\begin{array}{ccc} x \in B_1 & \leftrightarrow & f(x) \in A_1 \\ x \in B_2 & \leftrightarrow & f(x) \in A_2 \\ \vdots & & \vdots \\ x \in B_k & \leftrightarrow & f(x) \in A_k \end{array}$$

*Ein 2-unabhängig NP-vollständiges Paar $(A_1, A_2)$ wird kurz unabhängig NP-vollständig bezeichnet.*

Nachdem unabhängig NP-vollständige Mengen eingeführt wurden, muß noch geklärt werden, ob solche Tupel von NP-vollständigen Mengen auch wirklich existieren.
Der nachfolgende Satz zeigt, daß sich solche Tupel aus NP-vollständigen Mengen einfach konstruieren lassen.

SATZ 3.1.1
*Sind $A_1, A_2, \ldots, A_k \subseteq \Sigma^*$ NP-vollständig, so ist*

$$(\underbrace{A_1 \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}}, \underbrace{\Sigma^* \times A_2 \times \ldots \times \Sigma^*}_{k\text{-mal}}, \ldots, \underbrace{\Sigma^* \times \Sigma^* \times \ldots \times A_k}_{k\text{-mal}})$$

*$k$-unabhängig NP-vollständig.*

Beweis:
Seien $B_1, B_2, \ldots, B_k \in NP$, dann existieren $f_1, f_2, \ldots, f_k \in FP$ mit

$$\begin{array}{ccc} x \in B_1 & \leftrightarrow & f_1(x) \in A_1 \\ x \in B_2 & \leftrightarrow & f_2(x) \in A_2 \\ \vdots & & \vdots \\ x \in B_k & \leftrightarrow & f_k(x) \in A_k \end{array}$$

Da $\underbrace{A_1 \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}}, \underbrace{\Sigma^* \times A_2 \times \ldots \times \Sigma^*}_{k\text{-mal}}, \ldots, \underbrace{\Sigma^* \times \Sigma^* \times \ldots \times A_k}_{k\text{-mal}} \in NP$ und mit $f(x) =_{def} (f_1(x), f_2(x), \ldots, f_k(x))$ folgt $f \in FP$ sowie

$$\begin{array}{rcl} x \in B_1 & \leftrightarrow & f(x) \in \underbrace{A_1 \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \\ x \in B_2 & \leftrightarrow & f(x) \in \underbrace{\Sigma^* \times A_2 \times \ldots \times \Sigma^*}_{k\text{-mal}} \\ \vdots & & \vdots \\ x \in B_k & \leftrightarrow & f(x) \in \underbrace{\Sigma^* \times \Sigma^* \times \ldots \times A_k}_{k\text{-mal}} \end{array}$$

Damit ist $(\underbrace{A_1 \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}}, \underbrace{\Sigma^* \times A_2 \times \ldots \times \Sigma^*}_{k\text{-mal}}, \ldots, \underbrace{\Sigma^* \times \Sigma^* \times \ldots \times A_k}_{k\text{-mal}})$ $k$-unabhängig NP-vollständig. ✔

Die nächsten Definitionen und Lemmata beschäftigen sich mit dem Begriff des $p$-Zylinders. Der ursprüngliche Zylinderbegriff stammt aus der Rekursionstheorie und wurde später auch in der Komplexitätstheorie definiert. Eine kurze Einführung zu diesem Thema findet sich z.B. in [BDG90b].

Definition 3.1.2
*Für eine Menge $A \subseteq \Sigma^*$ wird $A \times \Sigma^*$ als* Zylindrifikation *von $A$ bezeichnet.*

Definition 3.1.3
*Zwei Mengen $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$ sind* p-isomorph *(kurz: $A \equiv^p B$) $\Leftrightarrow_{def}$*

- *Es existiert eine Bijektion $f : \Sigma^* \mapsto \Gamma^*$*
- $f, f^{-1} \in FP$
- *$x \in A \leftrightarrow f(x) \in B$ ($A$ läßt sich mittels $f$ auf $B$ reduzieren).*

Definition 3.1.4
*Eine Menge $A \subseteq \Sigma^*$ ist ein* p-Zylinder *$\Leftrightarrow_{def}$ $A \equiv^p A \times \Sigma^*$.*

Lemma 3.1.2
*Ist $A \in NP$ und $f^{-1} \in FP$ eine bijektive Funktion, dann ist auch $f(A) \in NP$.*

Beweis:
Da $f, f^{-1}$ bijektiv und $f^{-1} \in FP$ folgt: $f(A) \leqslant_m^p A$ via $f^{-1}$. Aus der Tatsache $\mathcal{R}_m^p(NP) = NP$ ergibt sich dann $f(A) \in NP$. ✔

Lemma 3.1.3
*Ist $A \subseteq \Sigma^*$ und $k \geqslant 2$ eine natürliche Zahl, dann ist $A$ ein $p$-Zylinder gdw. $\underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \equiv^p A$.*

Beweis:
Zuerst soll die etwas schwierigere Richtung dieses Lemmas gezeigt werden. Die Rückrichtung folgt dann fast direkt aus Definition 3.1.4.

$\Rightarrow$ Die Kodierung eines beliebigen Strings aus $\Sigma^*$ in eine $|\Sigma|$-adisch dargestellte natürliche Zahl ist eine bijektive polynomialzeitberechenbare Funktion. Aus diesem Grund ist $\underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \equiv^p \underbrace{A \times \mathbb{N} \times \ldots \times \mathbb{N}}_{k\text{-mal}}$.

In [Rog88] (§5.3, Seite 63) wird gezeigt, daß $\mathbb{N} \times \mathbb{N} \equiv^p \mathbb{N}$, denn die dort angegebene bijektive Kodierung ist polynomialzeitberechenbar. Durch $(k-2)$-faches Anwenden folgt dann auch, daß $\underbrace{A \times \mathbb{N} \times \ldots \times \mathbb{N}}_{k\text{-mal}} \equiv^p A \times \mathbb{N}$ und da $A$ nach Voraussetzung ein $p$-Zylinder ist ergibt sich:

$$\begin{aligned} \underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} &\equiv^p \underbrace{A \times \mathbb{N} \times \ldots \times \mathbb{N}}_{k\text{-mal}} \\ &\equiv^p A \times \mathbb{N} \\ &\equiv^p A \times \Sigma^* \\ &\equiv^p A \end{aligned}$$

$\Leftarrow$ Da $\underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \equiv^p A$ und $\underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \equiv^p A \times \Sigma^*$ schon oben gezeigt wurde, ist $A$ ein $p$-Zylinder.

✔

Nachdem nun der Begriff des $p$-Zylinders eingeführt ist, soll er dazu verwendet werden, eine Besonderheit im Zusammenhang mit Tupeln von unabhängig NP-vollständigen-Mengen zu zeigen. So ist es möglich, zu jedem $p$-Zylinder NP-vollständige-Mengen zu finden, die zusammen mit diesem $p$-Zylinder ein Tupel unabhängig NP-vollständiger Mengen bilden.

Satz 3.1.4
*Für jeden NP-vollständigen $p$-Zylinder $A$ existieren Mengen $B_1, \ldots, B_{k-1}$, so daß $(A, B_1, \ldots, B_{k-1})$ $k$-unabhängig NP-vollständig ist.*

Beweis:
Da $A \subseteq \Sigma^*$ ein $p$-Zylinder ist, existieren mit Lemma 3.1.3 $f, f^{-1} \in FP$ mit $A \leqslant_m^p \underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}}$ via $f^{-1}$ und $\underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \leqslant_m^p A$ via $f$, wobei $f$ und $f^{-1}$ bijektiv sind.
Seien $B'_1, B'_2, \ldots, B'_{k-1}$ NP-vollständige Mengen und

$$\begin{aligned} \hat{A} &=_{def} \underbrace{A \times \Sigma^* \times \ldots \times \Sigma^*}_{k\text{-mal}} \\ \hat{B}_1 &=_{def} \underbrace{\Sigma^* \times B'_1 \times \ldots \times \Sigma^*}_{k\text{-mal}} \\ &\vdots \qquad\qquad \vdots \\ \hat{B}_{k-1} &=_{def} \underbrace{\Sigma^* \times \Sigma^* \times \ldots \times B'_{k-1}}_{k\text{-mal}} \end{aligned}$$

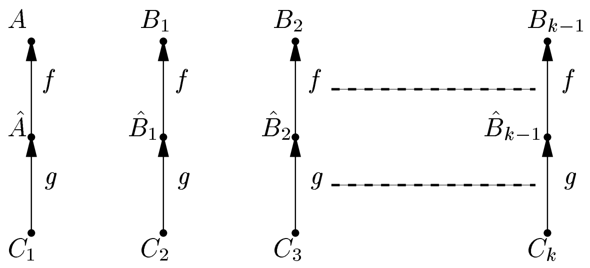


Abbildung 3.1: Reduktionen und Mengen aus Satz 3.1.4

Dann ist das $k$-Tupel $(\hat{A}, \hat{B}_1, \ldots, \hat{B}_{k-1})$ nach Satz 3.1.1 $k$-unabhängig NP-vollständig. Für $C_1, C_2, \ldots, C_k \in NP$ existiert deshalb eine Funktion $g \in FP$, so daß

$$\begin{array}{lllll} x \in C_1 & \leftrightarrow & g(x) \in \hat{A} & \leftrightarrow & f(g(x)) \in A \\ x \in C_2 & \leftrightarrow & g(x) \in \hat{B}_1 & \leftrightarrow & f(g(x)) \in f(\hat{B}_1) \\ \vdots & & \vdots & & \vdots \\ x \in C_k & \leftrightarrow & g(x) \in \hat{B}_{k-1} & \leftrightarrow & f(g(x)) \in f(\hat{B}_{k-1}) \end{array}$$

Sind die Mengen $B_i$ durch $B_1 =_{def} f(\hat{B}_1), \ldots, B_{k-1} =_{def} f(\hat{B}_{k-1})$ definiert[1], dann folgt $\hat{B}_1 \leqslant_m^p B_1, \ldots, \hat{B}_{k-1} \leqslant_m^p B_{k-1}$. Mit Lemma 3.1.2 ergibt sich auch $B_1, \ldots, B_{k-1} \in NP$ und damit ist $(A, B_1, \ldots, B_{k-1})$ $k$-unabhängig NP-vollständig mit der Reduktionsfunktion $f \circ g \in FP$. ✔

Bemerkung 3.1.1
*In [BDG90b] findet sich eine Bemerkung zur* Berman-Hartmanis-Vermutung. *Diese Vermutung sagt aus, daß alle NP-vollständigen Mengen p-isomorph zueinander sind. Weiterhin wird dort erwähnt, daß bisher keine NP-vollständige Menge gefunden wurde, die nicht p-isomorph zum p-Zylinder* **SAT** *ist. Da jede zu* **SAT** *p-isomorphe Menge selbst wieder ein p-Zylinder ist*[2]*, kann mit Satz 3.1.4 gefolgert werden, daß für fast jede bisher bekannte NP-vollständige Menge $A$ Mengen $B_i$ existieren, so daß das $k$-Tupel $(A, B_1, \ldots, B_{k-1})$ $k$-unabhängig NP-vollständig ist. Sollte sich darüberhinaus die* Berman-Hartmanis-Vermutung *als richtig erweisen, kann die Aussage auch auf alle NP-vollständigen Mengen ausgedehnt werden.*
*Allerdings gibt es auch Hinweise, daß die Berman-Hartmanis-Vermutung falsch ist. In [JY85] wurden NP-vollständige Mengen konstruiert, die vermutlich nicht zu* **SAT** *isomorph sind.*

Definition 3.1.5
*Eine Funktion $f$ ist* verlängernd, *falls $|f(x)| > |x|$ für alle $x \in \Sigma^*$ gilt.*

Definition 3.1.6
*Sei $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$, dann ist $A \leqslant_{1,li}^p B \Leftrightarrow_{def}$*

[1] Ein graphischer Überblick über die hier behandelten Mengen findet sich in Abbildung 3.1.
[2] Ein Beweis findet sich in [BDG90b], Lemma 5.1.

- $A \leqslant_m^p B$ *via* $f$
- $f$ *ist eine injektive Funktion*
- $f$ *ist verlängernd*

SATZ 3.1.5 ([BDG90b])
*Folgende Aussagen sind äquivalent:*

- $A$ *ist ein* $p$*-Zylinder.*
- *Falls* $B \leqslant_m^p A$ *gilt auch* $B \leqslant_{1,li}^p A$.

KOROLLAR 3.1.1
*Folgende Aussagen sind äquivalent*

- *Die Menge* $A$ *ist ein NP-vollständiger* $p$*-Zylinder.*
- *Die Menge* $A$ *ist* $\leqslant_{1,li}^p$*-vollständig für NP.*

Beweis:
Mit Hilfe von Satz 3.1.5 können beide Richtungen dieser Aussage leicht gezeigt werden.

⇒ Sei $A$ ein NP-vollständiger $p$-Zylinder und $B \in NP$, dann gilt $B \leqslant_m^p A$. Mit Satz 3.1.5 ergibt sich sogar $B \leqslant_{1,li}^p A$ und damit ist $A$ $\leqslant_{1,li}^p$-vollständig für NP.

⇐ Sei $A$ $\leqslant_{1,li}^p$-vollständig für NP, dann folgt direkt aus Satz 3.1.5, daß $A$ auch ein $p$-Zylinder ist.

✔

Da $p$-Zylinder auch $\leqslant_{1,li}^p$ vollständig für NP sind, wird nun Satz 3.1.4 auf $\leqslant_{1,li}^p$-vollständige Mengen erweitert.

KOROLLAR 3.1.2
*Für jede* $\leqslant_{1,li}^p$*-NP-vollständige Menge* $A$ *existieren* $B_1, B_2, \ldots B_{k-1} \in NP$, *so daß* $(A, B_1, B_2, \ldots B_{k-1})$ $k$*-unabhängig NP-vollständig ist.*

Beweis:
Diese Aussage ist eine direkte Folgerung aus Korollar 3.1.1 und Satz 3.1.4 ✔

## 3.2 „Natürliche“ unabhängig NP-vollständige Mengen

Bisher wurden nur künstlich konstruierte Tupel von NP-Mengen betrachtet. Nun soll aber auch noch gezeigt werden, daß Paare von wohlbekannten NP-vollständigen Mengen, die aus der Graphentheorie stammen, auch unabhängig NP-vollständig sind, um diesem Begriff ein wenig mehr mit „Leben“ zu erfüllen.

### 3.2.1 Hamilton-Teilkreise und Independent Sets

Definition 3.2.1

*Ein Hamilton-Teilkreis (Hamilton subcircuit) der Länge $k$ in einem (gerichteten) Graph $G = (V, E)$ ist eine Folge von Knoten $v_1, v_2, \ldots, v_k \in V$ mit $(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k), (v_k, v_1) \in E$.*

Definition 3.2.2

**HSC**

*Gegeben: Ein gerichteter Graph $G = (V, E)$ und ein $k \leqslant |V|$*

*Frage: Besitzt $G$ einen Hamilton-Teilkreis mit einer Länge von mindestens $k$?*

Lemma 3.2.1

**HSC** *ist NP-vollständig.*

Beweis:

**HSC** ist in NP, da eine geeignete NTM nichtdeterministisch alle Teilmengen $V' \subseteq V$ mit $|V'| \geqslant k$ und eine Anordnung der Knoten $v_1, \ldots, v_k$ mit $v_i \in V'$ raten und dann in Polynomialzeit überprüfen kann, ob diese Anordnung ein Hamilton-Teilkreis ist.

Sei $G = (V, E)$ ein gerichteter Graph und $f$ die polynomialzeitberechenbare Funktion mit $f(G) =_{def} (G, |V|)$, dann folgt $x \in \mathbf{HC} \leftrightarrow f(x) \in \mathbf{HSC}$ und damit $\mathbf{HC} \leqslant_m^p \mathbf{HSC}$. ✔

Definition 3.2.3

$$
\begin{array}{lll}
hc(G) & =_{def} & \textit{Länge eines längsten Hamilton-Teilkreises in } G \\
is(G) & =_{def} & \textit{Größe eines größten Independent Set in } G
\end{array}
$$

Lemma 3.2.2

*Es existiert ein Polynomialzeitalgorithmus, der zu jedem gerichteten Graphen $G$ und jedem $l \geqslant 0$ einen Graphen $G'$ konstruiert mit $is(G') = is(G) + l$ und $hc(G') = hc(G)$.*

Beweis:

Sei $G = (V, E)$, dann kann $G' = (V', E')$ wie folgt konstruiert werden:

$$
\begin{aligned}
V' &= V \cup \{a_1, \ldots, a_l\} \\
E' &= E
\end{aligned}
$$

wobei $V \cap \{a_1, \ldots, a_l\} = \emptyset$.

Durch diese Konstruktion kann sich jeder Independent Set höchstens um die Knoten $\{a_1, \ldots, a_l\}$ vergrößern, da kein anderer Knoten in $G'$ mit einem Knoten in $\{a_1, \ldots, a_l\}$ verbunden ist. Auch eventuell vorhandene Hamilton-Teilkreise werden durch diese Konstruktion nicht verlängert. ✔

Lemma 3.2.3

*Es existiert ein Polynomialzeitalgorithmus, der zu jedem gerichteten Graphen $G$ und $l \geqslant 3$ einen gerichteten Graphen $G'$ konstruiert mit $hc(G') = hc(G) + l$ und $is(G') = is(G) + 1$.*

Beweis:
Es sei $G = (V, E)$ ein beliebiger gerichteter Graph und $K_l = (V_l, E_l)$ ein vollständiger gerichteter Graph mit $|V_l| = l$, $V_l = \{v_1, \ldots, v_l\}$ und $V_l \cap V = \emptyset$. Dann ergibt sich $G'$ durch:

$$\begin{aligned} V' &= V \cup V_l \\ E' &= E \cup E_l \cup \bigcup_{(u,v)\in E} \{(u, v_1), (v_l, v)\} \end{aligned}$$

Da der vollständige Graph $K_l$ einen Hamilton-Teilkreis der Länge $l$ enthält, kann ein in $G$ enthaltener Hamilton-Teilkreis um $l$ verlängert werden (und nicht mehr). Gleichzeitig kann jeder Independent Set um einen Knoten aus $K_l$ vergrößert werden (und nicht mehr), weil je zwei beliebige Knoten in $K_l$ mit einer Kante verbunden sind. ✔

Satz 3.2.4
*Das Paar* (**INDEPENDENT-SET**, **HSC**) *ist unabhängig NP-vollständig.*

Beweis:
Sei $C, D \in NP$. Damit gibt es $g, h \in FP$

$$\begin{array}{lll} C \leqslant_m^p \textbf{INDEPENDENT-SET} & & \text{via g} \\ D \leqslant_m^p \textbf{HSC} & & \text{via h} \end{array}$$

Sei

$$\begin{array}{lll} (G_{\mathbf{IS}}(x), k_{\mathbf{IS}}(x)) =_{def} g(x) & \mathit{mit} & G_{\mathbf{IS}}(x) = (V_{\mathbf{IS}}(x), E_{\mathbf{IS}}(x)) \\ (G_{\mathbf{HSC}}(x), k_{\mathbf{HSC}}(x)) =_{def} h(x) & \mathit{mit} & G_{\mathbf{HSC}}(x) = (V_{\mathbf{HSC}}(x), E_{\mathbf{HSC}}(x)) \end{array}$$

und damit

$$\begin{aligned} x \in C &\leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \\ x \in D &\leftrightarrow hc(G_{\mathbf{HSC}}) \geqslant k_{\mathbf{HSC}} \end{aligned}$$

Offensichtlich gilt $hc(G_{\mathbf{IS}}) \leqslant |V_{\mathbf{IS}}|$ und $is(G_{\mathbf{HSC}}) \leqslant |V_{\mathbf{HSC}}|$ . Sei $k =_{def} k_{\mathbf{HSC}} + k_{\mathbf{IS}} + |V_{\mathbf{HSC}}| + |V_{\mathbf{IS}}| + 1$.
Mit Lemma 3.2.2 kann ein Graph $G'_{\mathbf{IS}} = (V'_{\mathbf{IS}}, E'_{\mathbf{IS}})$ konstruiert werden, so daß

$$\begin{aligned} is(G'_{\mathbf{IS}}) &= is(G_{\mathbf{IS}}) + (k_{\mathbf{HSC}} + |V_{\mathbf{HSC}}| + |V_{\mathbf{IS}}| + 1) \\ hc(G'_{\mathbf{IS}}) &= hc(G_{\mathbf{IS}}) \end{aligned}$$

Dann folgt $x \in C \leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \leftrightarrow is(G'_{\mathbf{IS}}) \geqslant k$.
Mit Hilfe von Lemma 3.2.3 kann ein Graph $G'_{\mathbf{HSC}} = (V'_{\mathbf{HSC}}, E'_{\mathbf{HSC}})$ konstruiert werden mit

$$\begin{aligned} hc(G'_{\mathbf{HSC}}) &= hc(G_{\mathbf{HSC}}) + (k_{\mathbf{IS}} + |V_{\mathbf{HSC}}| + |V_{\mathbf{IS}}| + 1) \\ is(G'_{\mathbf{HSC}}) &= is(G_{\mathbf{HSC}}) + 1 \end{aligned}$$

Damit ergibt sich $x \in D \leftrightarrow hc(G_{\mathbf{HSC}}) \geqslant k_{\mathbf{HSC}} \leftrightarrow hc(G'_{\mathbf{HSC}}) \geqslant k$.
Sei $G^* = (V^*, E^*)$ mit

$$\begin{aligned} V^* &= V'_{\mathbf{HSC}} \cup V'_{\mathbf{IS}} \\ E^* &= E'_{\mathbf{HSC}} \cup E'_{\mathbf{IS}} \cup (V'_{\mathbf{IS}} \times V'_{\mathbf{HSC}}) \end{aligned}$$

Dieser Graph enthält die Graphen $G'_{\mathbf{IS}}$ bzw. $G'_{\mathbf{HSC}}$ als Untergraphen, wobei alle Knoten aus $V'_{\mathbf{IS}}$ mit allen Knoten aus $V'_{\mathbf{HSC}}$ durch eine gerichtete Kante verbunden sind. Aus diesem Grund muß $is(G^*) = \max(is(G'_{\mathbf{IS}}), is(G'_{\mathbf{HSC}}))$ gelten, und wegen $is(G'_{\mathbf{IS}}) \geqslant |V_{\mathbf{HSC}}| + 1 \geqslant is(G_{\mathbf{HSC}}) + 1 = is(G'_{\mathbf{HSC}})$ ergibt sich $is(G^*) = is(G'_{\mathbf{IS}})$.
Da bei der Konstruktion des Graphen $G^*$ aus $G'_{\mathbf{IS}}$ und $G'_{\mathbf{HSC}}$ nur Kanten von Knoten aus $V'_{\mathbf{IS}}$ zu Knoten aus $V'_{\mathbf{HSC}}$ eingebaut werden, gilt $hc(G^*) = \max(hc(G'_{\mathbf{HSC}}), hc(G'_{\mathbf{IS}}))$. Mit $hc(G'_{\mathbf{HSC}}) \geqslant |V_{\mathbf{IS}}| \geqslant hc(G_{\mathbf{IS}}) = hc(G'_{\mathbf{IS}})$ folgt $hc(G^*) = hc(G'_{\mathbf{HSC}})$.
Es ist leicht einzusehen, daß die angegebene Konstruktion eine polynomialzeitberechenbare Funktion $f$ beschreibt und

$$\begin{aligned} x \in C &\leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \\ &\leftrightarrow is(G'_{\mathbf{IS}}) \geqslant k \\ &\leftrightarrow is(G^*) \geqslant k \end{aligned}$$

sowie

$$\begin{aligned} x \in D &\leftrightarrow hc(G_{\mathbf{HSC}}) \geqslant k_{\mathbf{HSC}} \\ &\leftrightarrow hc(G'_{\mathbf{HSC}}) \geqslant k \\ &\leftrightarrow hc(G^*) \geqslant k \end{aligned}$$

Damit ist das Paar (**INDEPENDENT-SET**, **HSC**) unabhängig NP-vollständig. ✔

### 3.2.2 Cliquen und Independent Sets

Definition 3.2.4
*$cl(G) =_{def}$ Größe einer größten Clique in $G$*

Lemma 3.2.5
*Es existiert ein Polynomialzeitalgorithmus, der aus jedem gerichteten Graphen $G$ und $l \geqslant 0$ einen gerichteten Graphen $G'$ konstruiert mit $is(G') = is(G) + l$ und $cl(G') = cl(G)$.*

Beweis:
Konstruktion des Graphen $G'$ analog zu Lemma 3.2.2.
Dabei kann jeder Independent Set um maximal $l$ Knoten vergrößert werden, da aber kein Knoten aus $V$ mit einem der hinzugefügten Knoten verbunden ist, vergrößern sich die Cliquen aus $G$ nicht. ✔

Lemma 3.2.6
*Es existiert ein Polynomialzeitalgorithmus, der zu jedem gerichteten Graphen $G$ und $l \geqslant 0$ einen gerichteten Graphen $G'$ konstruiert mit $cl(G') = cl(G) + l$ und $is(G') = is(G)$.*

Beweis:
Sei $G = (V, E)$ und $K_l = (V_l, E_l)$ ein vollständiger und gerichteter Graph mit $|V_l| = l$, $V_l = \{v_1, \ldots, v_l\}$ und $V \cap V_l = \emptyset$. Dann ergibt sich $G'$ durch:

$$\begin{aligned} V' &= V \cup V_l \\ E' &= E \cup E_l \cup (V \times V_l) \cup (V_l \times V) \end{aligned}$$

Da der vollständige Graph $K_l$ auch eine $l$-Clique ist und jeder Knoten des Graphen $G$ mit jedem Knoten in $K_l$ in beiden Richtungen verbunden ist, ergibt sich $cl(G') = cl(G) + l$. Weiterhin ist es offensichtlich, daß sich kein Independent Set vergrößern kann. ✔

Satz 3.2.7
*Das Paar* (**INDEPENDENT-SET**, **CLIQUE**) *ist unabhängig NP-vollständig.*

Beweis:
Sei $C, D \in NP$. Damit gibt es $g, h \in FP$

$$\begin{array}{lll} C \leqslant_m^p \textbf{INDEPENDENT-SET} & \text{via g} \\ D \leqslant_m^p \textbf{CLIQUE} & \text{via h} \end{array}$$

und

$$\begin{array}{lll} (G_{\mathbf{IS}}(x), k_{\mathbf{IS}}(x)) =_{def} g(x) & mit & G_{\mathbf{IS}}(x) = (V_{\mathbf{IS}}(x), E_{\mathbf{IS}}(x)) \\ (G_{\mathbf{CL}}(x), k_{\mathbf{CL}}(x)) =_{def} h(x) & mit & G_{\mathbf{CL}}(x) = (V_{\mathbf{CL}}(x), E_{\mathbf{CL}}(x)) \end{array}$$

damit folgt

$$\begin{aligned} x \in C &\leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \\ x \in D &\leftrightarrow cl(G_{\mathbf{CL}}) \geqslant k_{\mathbf{CL}} \end{aligned}$$

Es ist leicht einzusehen, daß $cl(G_{\mathbf{IS}}) \leqslant |V_{\mathbf{IS}}|$ und $is(G_{\mathbf{CL}}) \leqslant |V_{\mathbf{CL}}|$. Sei $k =_{def} k_{\mathbf{CL}} + k_{\mathbf{IS}} + |V_{\mathbf{CL}}| + |V_{\mathbf{IS}}|$.
Mit Lemma 3.2.5 kann ein Graph $G'_{\mathbf{IS}} = (V'_{\mathbf{IS}}, E'_{\mathbf{IS}})$ gewonnen werden mit

$$\begin{aligned} is(G'_{\mathbf{IS}}) &= is(G_{\mathbf{IS}}) + (k_{\mathbf{CL}} + |V_{\mathbf{CL}}| + |V_{\mathbf{IS}}|) \\ cl(G'_{\mathbf{IS}}) &= cl(G_{\mathbf{IS}}) \end{aligned}$$

Daraus folgt $x \in C \leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \leftrightarrow cl(G'_{\mathbf{IS}}) \geqslant k$.
Mit Hilfe von Lemma 3.2.6 ist es möglich, einen Graphen $G'_{\mathbf{CL}} = (V'_{\mathbf{CL}}, E'_{\mathbf{CL}})$ zu konstruieren, so daß

$$\begin{aligned} cl(G'_{\mathbf{CL}}) &= cl(G_{\mathbf{CL}}) + (k_{\mathbf{IS}} + |V_{\mathbf{CL}}| + |V_{\mathbf{IS}}|) \\ is(G'_{\mathbf{CL}}) &= is(G_{\mathbf{CL}}) \end{aligned}$$

Daraus folgt $x \in D \leftrightarrow cl(G_{\mathbf{CL}}) \geqslant k_{\mathbf{CL}} \leftrightarrow cl(G'_{\mathbf{CL}}) \geqslant k$.
Sei $G^* = (V^*, E^*)$ mit

$$\begin{aligned} V^* &= V'_{\mathbf{CL}} \cup V'_{\mathbf{IS}} \\ E^* &= E'_{\mathbf{CL}} \cup E'_{\mathbf{IS}} \cup (V'_{\mathbf{IS}} \times V'_{\mathbf{CL}}) \end{aligned}$$

Dieser Graph enthält die Graphen $G'_{\mathbf{IS}}$ bzw. $G'_{\mathbf{CL}}$ als Untergraphen, wobei alle Knoten aus $V'_{\mathbf{IS}}$ mit allen Knoten aus $V'_{\mathbf{CL}}$ durch eine gerichtete Kante verbunden sind. Aus diesem Grund muß $is(G^*) = \max(is(G'_{\mathbf{CL}}), is(G'_{\mathbf{IS}}))$ gelten und wegen $is(G'_{\mathbf{IS}}) \geqslant |V_{\mathbf{CL}}| \geqslant is(G_{\mathbf{CL}}) = is(G'_{\mathbf{CL}})$ ergibt sich $is(G^*) = is(G'_{\mathbf{IS}})$.
Bei der Konstruktion des Graphen $G^*$ werden nur Kanten von Knoten aus $V'_{\mathbf{IS}}$ zu Knoten aus $V'_{\mathbf{CL}}$ eingebaut und damit ist $cl(G^*) = \max(cl(G'_{\mathbf{IS}}), cl(G'_{\mathbf{CL}}))$.
Mit $cl(G'_{\mathbf{CL}}) \geqslant |V_{\mathbf{IS}}| \geqslant cl(G_{\mathbf{IS}}) = cl(G'_{\mathbf{IS}})$ folgt daraus $cl(G^*) = cl(G'_{\mathbf{CL}})$.
Die angegebene Konstruktion beschreibt eine Funktion $f \in FP$ und

$$\begin{aligned} x \in C &\leftrightarrow is(G_{\mathbf{IS}}) \geqslant k_{\mathbf{IS}} \\ &\leftrightarrow is(G'_{\mathbf{IS}}) \geqslant k \\ &\leftrightarrow is(G^*) \geqslant k \end{aligned}$$

sowie

$$\begin{aligned} x \in D &\leftrightarrow cl(G_{\mathbf{CL}}) \geqslant k_{\mathbf{CL}} \\ &\leftrightarrow cl(G'_{\mathbf{CL}}) \geqslant k \\ &\leftrightarrow cl(G^*) \geqslant k \end{aligned}$$

Damit ist das Paar (**INDEPENDENT-SET**, **CLIQUE**) unabhängig NP-vollständig. ✔

# 4 Eine erste Definition

## 4.1 $high^1$-Mengen in der Booleschen Hierarchie

Nun sind alle Begriffe bekannt, die benötigt werden um die erste Definition von „High- und Low“-Mengen zu untersuchen.

Definition 4.1.1
*Es sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $high_k^1 \Leftrightarrow_{def}$*

- $A \in NP$
- $\mathcal{R}_m^p(NP(k) \oplus A) = NP(k+1)$

Lemma 4.1.1
*Sind $A_1, A_2, \ldots, A_k, B_1, B_2, \ldots, B_k \subseteq \Sigma^*$ beliebige Sprachen und $f : \Sigma^* \mapsto \Sigma^*$ eine Reduktionsfunktion mit*

$$\begin{array}{rcl} x \in B_1 & \leftrightarrow & f(x) \in A_1 \\ x \in B_2 & \leftrightarrow & f(x) \in A_2 \\ \vdots & & \vdots \\ x \in B_k & \leftrightarrow & f(x) \in A_k \end{array}$$

*dann gilt auch*

$$x \in B_1 \oplus B_2 \oplus \ldots \oplus B_k \quad \leftrightarrow \quad f(x) \in A_1 \oplus A_2 \oplus \ldots \oplus A_k$$

Beweis:
Da $c_{B_i}(x) = c_{A_i}(f(x))$ und $c_{D \oplus E} = c_D(x) \oplus c_E(x)$ (Die Verknüpfung „$\oplus$“ rechts ist die Addition modulo 2.) gilt, folgt:

$$\begin{array}{rcl} c_{B_1 \oplus \ldots \oplus B_k}(x) & = & c_{B_1}(x) \oplus \ldots \oplus c_{B_k}(x) \\ & = & c_{A_1}(f(x)) \oplus \ldots \oplus c_{A_k}(f(x)) \\ & = & c_{A_1 \oplus \ldots \oplus A_k}(f(x)) \end{array}$$

✔

In den folgenden Zeilen wird gezeigt, daß der Begriff „unabhängig NP-vollständig“ und die Zugehörigkeit zu $high_k^1$ sehr eng miteinander verbunden sind.

Satz 4.1.2
*Ist das $(k+1)$-Tupel $(A, A_1, A_2, \ldots A_k)$ $(k+1)$-unabhängig NP-vollständig, dann gilt $A \in high_k^1$.*

Beweis:
Seien $B, B_1, B_2, \ldots B_k \in NP$. Dann folgt mit Definition 3.1.1 und Lemma 4.1.1

$$B \oplus B_1 \oplus B_2 \oplus \ldots \oplus B_k \leqslant_m^p A \oplus A_1 \oplus A_2 \oplus \ldots \oplus A_k$$

Damit ergibt sich $NP(k+1) \subseteq \mathcal{R}_m^p(A \oplus A_1 \oplus A_2 \oplus \ldots \oplus A_k)$, und da $NP(k+1)$ nach Lemma 2.2.1 unter $\leqslant_m^p$-Reduktion abgeschlossen ist, folgt sogar $NP(k+1) = \mathcal{R}_m^p(A \oplus A_1 \oplus A_2 \oplus \ldots \oplus A_k)$. Damit ist $A \in high_k^1$. ✔

Bemerkung 4.1.1
*Falls $(A_1, A_2, \ldots A_k)$ $k$-unabhängig NP-vollständig ist, ist auch jedes permutierte Tupel offensichtlich wieder $k$-unabhängig NP-vollständig. Damit ist dann auch jede der Mengen $A_i \in high_k^1$.*

Korollar 4.1.1
*Ist $A \subseteq \Sigma^*$ ein NP-vollständiger $p$-Zylinder, dann folgt $A \in high_k^1$.*

Beweis:
Mit Satz 3.1.4 existieren Mengen $B_1, B_2, \ldots, B_k$, so daß $(A, B_1, B_2, \ldots, B_k)$ $k+1$-unabhängig NP-vollständig ist. Mit Satz 4.1.2 folgt dann die Aussage. ✔

Korollar 4.1.2
*Falls $A$ $\leqslant_{1,li}^p$-vollständig für $NP$ ist, dann folgt $A \in high_k^1$.*

Beweis:
Offensichtlich mit den Korollaren 3.1.1 bzw. 4.1.1. ✔

Nachdem bewiesen wurde, daß $p$-Zylinder bzw. $\leqslant_{1,li}^p$-vollständige Mengen in $high_k^1$ enthalten sind, wäre es noch interessant zu wissen, ob von einem Paar unabhängig NP-vollständiger Mengen mindestens eine wieder ein $p$-Zylinder ist. Leider konnte dies nicht gezeigt werden.

## 4.2 $low^1$-Mengen in der Booleschen Hierarchie

Definition 4.2.1
*Es sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $low_k^1 \Leftrightarrow_{def}$*

- $A \in NP$
- $\mathcal{R}_m^p(NP(k) \oplus A) = NP(k)$

Mit Hilfe von Lemma 4.2.1, Satz 4.2.2 und Korollar 4.2.1 kann gezeigt werden, daß jede endliche Menge, wie es der Intuition entspricht, in $low_k^1$ enthalten ist.

Lemma 4.2.1
*Falls für ein $C \in NP$ und jedes $A \in NP$ gilt, daß $A \cap C \in co\text{-}NP$, dann folgt $NP \oplus C = NP$.*

Beweis:

- $\subseteq$ Sei $A \in NP$, dann ist $A \oplus C = (A \cup C) \cap \overline{(A \cap C)} \in NP$, da $NP$ unter Vereinigung und Schnitt abgeschlossen ist und $\overline{(A \cap C)} \in NP$.
- $\supseteq$ Sei $A \in NP$ und sei $A' =_{def} A \oplus C$. Wie schon im ersten Teil dieses Beweises gezeigt wurde, gilt dann auch $A' \in NP$. Damit ergibt sich $A' \oplus C \in NP \oplus C$ und da $A' \oplus C = A \oplus C \oplus C = A$, gilt auch $A \in NP \oplus C$.

Damit ist $NP \oplus C = NP$ gezeigt. ✔

Satz 4.2.2
*Falls für ein $C \in NP$ und jedes $A \in NP$ gilt, daß $A \cap C \in co\text{-}NP$, dann folgt $C \in low_k^1$.*

Beweis:
Mit Lemma 4.2.1 folgt:

$$\begin{aligned} NP(k) \oplus C &= \underbrace{NP \oplus NP \oplus \ldots \oplus NP}_{k\text{-mal}} \oplus C \\ &= \underbrace{NP \oplus NP \oplus \ldots \oplus (NP}_{k\text{-mal}} \oplus C) \\ &= \underbrace{NP \oplus NP \oplus \ldots \oplus NP}_{k\text{-mal}} \end{aligned}$$

Hiermit ergibt sich mit Lemma 2.2.1 $\mathcal{R}_m^p(NP(k) \oplus C) = \mathcal{R}_m^p(NP(k)) = NP(k)$ und dann folgt $C \in low_k^1$. ✔

Korollar 4.2.1
*Jede endliche Menge $C$ ist in $low_k^1$ enthalten.*

Beweis:
Da $C$ endlich ist, muß auch für jede Menge $A \in NP$ der Schnitt $A \cap C$ endlich sein. Damit ist $A \cap C \in P$ und deshalb $A \cap C \in co\text{-}NP$. Mit Satz 4.2.2 folgt die Aussage. ✔

Definition 4.2.2
*Eine Menge $A \subseteq \Sigma^*$ wird als* co-endlich *bezeichnet, falls $\overline{A}$ endlich ist.*

Da die co-endlichen Mengen ebenso einfach zu entscheiden sind wie die endlichen, würde es der Intuition entsprechen, wenn die co-endlichen Mengen ebenfalls in $low_k^1$ enthalten sind.
In den folgenden Zeilen wird aber gezeigt, daß dies sehr unwahrscheinlich ist, da sonst die Boolesche Hierarchie und damit auch die Polynomialzeithierarchie kollabieren würde, wovon man beim heutigen Kenntnisstand nicht ausgeht.

Lemma 4.2.3
*Ist $C \subseteq \Sigma^*$ co-endlich, dann gilt $NP \oplus C = co\text{-}NP$.*

Beweis:

- $\subseteq$ Sei $B \in NP$. Damit ist $B \oplus C = (\overline{B} \cap C) \cup (B \cap \overline{C})$. Da $C$ co-endlich ist, muß $B \cap \overline{C}$ endlich sein und damit folgt $B \cap \overline{C} \in co\text{-}NP$.
  Weil $C$ nach Voraussetzung co-endlich ist, gilt $C \in P$ und damit auch $C \in co\text{-}NP$. Mit der Tatsache, daß $co\text{-}NP$ unter Schnitt abgeschlossen ist und $\overline{B} \in co\text{-}NP$, folgt $\overline{B} \cap C \in co\text{-}NP$. Da $co\text{-}NP$ aber auch unter Vereinigung abgeschlossen ist, ergibt sich $B \oplus C \in co\text{-}NP$.

- $\supseteq$ Ist $B \in co\text{-}NP$, dann folgt $\overline{B} \in NP$, und da $\overline{C}$ endlich ist und damit jeder Schnitt mit einer $NP$-Menge endlich ist, folgt mit Lemma 4.2.1 $\overline{B} \oplus \overline{C} \in NP$. Deshalb ergibt sich $(\overline{B} \oplus \overline{C}) \oplus C = \overline{B} \oplus (\overline{C} \oplus C) = \overline{B} \oplus \Sigma^* = B$, d.h. $B \in NP \oplus C$.

✔

Satz 4.2.4
*Gibt es eine co-endliche Menge $C$ und $C \in low_k^1$, dann ist $NP(k) = BH$.*

Beweis:
Da $C$ co-endlich ist, folgt mit Lemma 4.2.3:

$$\mathcal{R}_m^p(NP(k) \oplus C) = \mathcal{R}_m^p(NP(k-1) \oplus co\text{-}NP) = \mathcal{R}_m^p(co\text{-}NP(k)) = co\text{-}NP(k)$$

Da $C \in low_k^1$ vorausgesetzt wurde, ergibt sich $\mathcal{R}_m^p(NP(k) \oplus C)) = NP(k) = co\text{-}NP(k)$. Mit Satz 2.2.2 ergibt sich dann die Aussage. ✔

Das nun folgende Resultat entspricht nun wieder mehr der Anschauung, da gezeigt werden kann, daß die in $high_k^1$ enthaltenen $p$-Zylinder bzw. $\leqslant_{1,li}^p$-NP-vollständigen Mengen mit hoher Wahrscheinlichkeit (die Boolesche Hierarchie würde kollabieren) nicht in $low_k^1$ enthalten sind.

Satz 4.2.5
*Falls $high_k^1 \cap low_k^1 \neq \emptyset$, dann folgt $BH = NP(k)$.*

Beweis:
Mit $A \in high_k^1 \cap low_k^1$ gilt $NP(k) = \mathcal{R}_m^p(NP(k) \oplus A) = NP(k+1)$. Mit Satz 2.2.2 folgt die Aussage. ✔

Korollar 4.2.2
*Sei $A$ ein NP-vollständiger $p$-Zylinder und $A \in low_k^1$, dann gilt $BH = NP(k)$.*

Beweis:
Nach Korollar 4.1.1 gilt $A \in high_k^1$ und mit Satz 4.2.5 ergibt sich der Rest der Aussage. ✔

Korollar 4.2.3
*Sei $A$ ein NP-vollständiger $p$-Zylinder und $A \in low_k^1$, dann gilt $PH = \Theta_3^P$.*

Beweis:
Mit den Sätzen 2.2.2 und 2.2.3 sowie Korollar 4.2.2 folgt die Aussage. ✔

# 5 Eine alternative Definition

## 5.1 Grundlagen und Beobachtungen

Wegen $NP(k+1) = NP(k) \oplus NP$ gilt, daß mit „$\oplus NP$“ der Sprung zur nächsten Stufe der Booleschen Hierarchie geleistet wird. Die Komplexität einer Menge $A \in NP$ soll deshalb daran gemessen werden, ob dies auch mit „$\oplus \mathcal{R}_m^p(A)$“ möglich ist.

Satz 5.1.1

*1.* $A = \emptyset \Rightarrow NP(k) \oplus \mathcal{R}_m^p(A) = NP(k)$

*2.* $A = \Sigma^* \Rightarrow NP(k) \oplus \mathcal{R}_m^p(A) = co\text{-}NP(k)$

*3.* $A \in NP \backslash \{\emptyset, \Sigma^*\} \Rightarrow NP(k) \oplus P \subseteq NP(k) \oplus \mathcal{R}_m^p(A) \subseteq NP(k+1)$

*4.* *$A$ ist NP-vollständig* $\Rightarrow NP(k) \oplus \mathcal{R}_m^p(A) = NP(k+1)$

Beweis:

1. Da $\mathcal{R}_m^p(\emptyset) = \emptyset$ ist, folgt $NP(k) \oplus \mathcal{R}_m^p(\emptyset) = NP(k) \oplus \emptyset = NP(k)$.
2. Mit $\mathcal{R}_m^p(\Sigma^*) = \Sigma^*$ ergibt sich $NP(k) \oplus \mathcal{R}_m^p(\Sigma^*) = co\text{-}NP(k)$.
3. Da $A \in NP \backslash \{\emptyset, \Sigma^*\}$ gilt $P \subseteq \mathcal{R}_m^p(A)$. Aufgrund der Abgeschlossenheit der Klasse $NP$ „many-one $p$-Reduktion“ gilt auch $P \subseteq \mathcal{R}_m^p(A) \subseteq NP$. Hiermit ergibt sich dann die obige Inklusionskette.
4. Da $A$ NP-vollständig ist, gilt $\mathcal{R}_m^p(A) = NP$. Damit ergibt sich $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k) \oplus NP = NP(k+1)$.

✔

## 5.2 $high^2$-Mengen in der Booleschen Hierarchie

Die nun folgende Definition erweist sich wesentlich einfacher handhabbar. Es können einige Zusammenhänge ohne großen Aufwand gezeigt werden, die sich bei der ersten Definition als schwierig erwiesen haben.

Definition 5.2.1

*Es sei* $k \geqslant 1$. *Eine Menge* $A \subseteq \Sigma^*$ *ist in* $high_k^2 \Leftrightarrow_{def}$

- $A \in NP$
- $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k+1)$

Der folgende Satz zeigt, daß alle NP-vollständigen Mengen in $high_k^2$ enthalten sind. Die ist vergleichbar mit dem Ergebnis über High-Mengen in der Polynomialzeithierarchie (siehe Satz 2.1.2(3) und Beobachtung 2.1.1).

Satz 5.2.1
*Für jede NP-vollständige Menge $A$ gilt $A \in high_k^2$.*

Beweis:
Offensichtlich mit Satz 5.1.1(4). ✔

Bei dieser Definition von „Low- und High"-Mengen kann leicht gezeigt werden, daß die $high^2$-Mengen ineinander enthalten sind. Auch diese Eigenschaft haben sie mit den in [Sch85] beschriebenen High-Mengen der Polynomialzeithierarchie gemeinsam.

Satz 5.2.2
*Für jedes $k \geqslant 1$ gilt $high_k^2 \subseteq high_{k+1}^2$.*

Beweis:
Da für $A \in high_k^2$ nach Definition 5.2.1 $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k+1)$ gilt, ergibt sich:

$$\begin{aligned} NP(k+1) \oplus \mathcal{R}_m^p(A) &= NP \oplus NP(k) \oplus \mathcal{R}_m^p(A) \\ &= NP \oplus NP(k+1) \\ &= NP(k+2) \end{aligned}$$

Damit folgt $A \in high_{k+1}^2$. ✔

Satz 5.2.3
*Für jede Menge $A \in high_k^2$ und jedes $B \in NP$ mit $A \leqslant_m^p B$ gilt $B \in high_k^2$.*

Beweis:
Es ergibt sich $NP(k) \oplus \mathcal{R}_m^p(B) = NP(k+1)$, da

$\subseteq$ Weil $B \in NP$ vorausgesetzt wurde folgt $\mathcal{R}_m^p(B) \subseteq NP$ . Damit ergibt sich $NP(k) \oplus \mathcal{R}_m^p(B) \subseteq NP(k+1)$.

$\supseteq$ Mit $A \leqslant_m^p B$ folgt $\mathcal{R}_m^p(A) \subseteq \mathcal{R}_m^p(B)$. Deshalb folgt

$$NP(k) \oplus \mathcal{R}_m^p(A) \subseteq NP(k) \oplus \mathcal{R}_m^p(B)$$

Mit Definition 5.2.1 ergibt sich

$$NP(k+1) \subseteq NP(k) \oplus \mathcal{R}_m^p(B)$$

Damit gilt $B \in high_k^2$. ✔

## 5.3 $mid^2$-Mengen in der Booleschen Hierarchie

An dieser Stelle sollen noch sogenannte „Mid-Mengen“ untersucht werden. Dies sind Mengen, die vermutlich weder in $high^2$ noch in $low^2$ enthalten sind, sondern im komplexitätstheoretischen Sinn dazwischen angesiedelt sind.

DEFINITION 5.3.1
*Es sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $mid_k^2 \Leftrightarrow_{def}$*

- $A \in NP$
- $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k) \oplus P$

SATZ 5.3.1
*Für jedes natürliche $k \geqslant 1$ gilt $mid_k^2 \subseteq mid_{k+1}^2$.*

Beweis:
Da nach Definition 5.3.1 $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k) \oplus P$ gilt, ergibt sich:

$$\begin{aligned} NP(k+1) \oplus \mathcal{R}_m^p(A) &= NP \oplus NP(k) \oplus \mathcal{R}_m^p(A) \\ &= NP \oplus NP(k) \oplus P \\ &= NP(k+1) \oplus P \end{aligned}$$

Damit folgt $A \in mid_{k+1}^2$. ✔

SATZ 5.3.2
*Sei $A \in mid_k^2$ und $B \equiv^p A$, dann folgt $B \in mid_k^2$.*

Beweis:
Da $A \equiv^p B$ folgt $\mathcal{R}_m^p(A) = \mathcal{R}_m^p(B)$ und damit ergibt sich:

$$\begin{aligned} NP(k) \oplus \mathcal{R}_m^p(B) &= NP(k) \oplus \mathcal{R}_m^p(A) \\ &= NP(k) \oplus P \end{aligned}$$

Damit gilt $B \in mid_k^2$. ✔

SATZ 5.3.3
*Sei $A \in P \backslash \{\Sigma^*, \emptyset\}$, dann gilt $A \in mid_k^2$.*

Beweis:
Da $A \in P \backslash \{\Sigma^*, \emptyset\}$ ist, ergibt sich $\mathcal{R}_m^p(A) = P$. Damit folgt $NP \oplus \mathcal{R}_m^p(A) = NP \oplus P$, d.h $A \in mid_1^2$. Mit Satz 5.3.1 folgt die Aussage. ✔

## 5.4 $low^2$-Mengen in der Booleschen Hierarchie

In diesem Abschnitt sollen $low^2$-Mengen untersucht werden. Dabei zeigt es sich, daß die $low^2$-Klassen ineinander enthalten sind (Siehe Satz 5.4.1).

DEFINITION 5.4.1
*Es sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $low_k^2 \Leftrightarrow_{def}$*

- $A \in NP$
- $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k)$

Satz 5.4.1
*Für jedes natürliche $k \geqslant 1$ gilt $low_k^2 \subseteq low_{k+1}^2$.*

Beweis:
Da nach Definition 5.4.1 $NP(k) \oplus \mathcal{R}_m^p(A) = NP(k)$ gilt, ergibt sich:

$$\begin{aligned} NP(k+1) \oplus \mathcal{R}_m^p(A) &= NP \oplus NP(k) \oplus \mathcal{R}_m^p(A) \\ &= NP \oplus NP(k) \\ &= NP(k+1) \end{aligned}$$

Damit folgt $A \in low_{k+1}^2$. ✔

Satz 5.4.2
*Sei $A \in low_k^2$, $B \in NP \backslash \{\Sigma^*\}$ und $B \leqslant_m^p A$, dann folgt $B \in low_k^2$.*

Beweis:
Es ergibt sich $NP(k) \oplus \mathcal{R}_m^p(B) = NP(k)$, da:

- $\subseteq$ Da $B \leqslant_m^p A$ gilt, folgt auch $\mathcal{R}_m^p(B) \subseteq \mathcal{R}_m^p(A)$ und damit $NP(k) \oplus \mathcal{R}_m^p(B) \subseteq NP(k) \oplus \mathcal{R}_m^p(A) = NP(k)$.
- $\supseteq$ Weil $B \in NP$ und damit $\mathcal{R}_m^p(B) \subseteq NP$, ergibt sich $NP(k) \subseteq NP(k) \oplus \mathcal{R}_m^p(B)$.

Damit gilt $B \in low_k^2$. ✔

Das folgende Lemma zeigt, daß es zumindest die leere Menge in jeder $low_k^2$-Menge enthalten ist.

Lemma 5.4.3
*Es gilt $\emptyset \in low_k^2$.*

Beweis:
Offensichtlich ist $\mathcal{R}_m^p(\emptyset) = \emptyset$. Dann folgt:

$$\begin{aligned} NP \oplus \mathcal{R}_m^p(\emptyset) &= NP \oplus \emptyset \\ &= NP \end{aligned}$$

D.h. $\emptyset \in low_1^2$ und mit Satz 5.4.1 folgt die Aussage. ✔

Lemma 5.4.4 kann auch als Indiz dafür gewertet werden, daß die im Abschnitt 5.3 definierten $mid^2$-Mengen nicht mit den $low^2$-Mengen oder $high^2$-Mengen zusammenfällt.

Lemma 5.4.4
*Falls $NP(k) \oplus P = NP(k)$, dann gilt $NP(k) = co\text{-}NP(k)$.*

Beweis:
Da $\Sigma^* \in P$ und $NP(k) \oplus \Sigma^* = co\text{-}NP(k)$ gilt, ergibt sich:

$$co\text{-}NP(k) = NP(k) \oplus \Sigma^* \subseteq NP(k) \oplus P = NP(k)$$

Da dann aber auch $NP(k) \subseteq co\text{-}NP(k)$ gilt, folgt die Aussage. ✔

Satz 5.4.5 und Korollar 5.4.2 zeigen, daß die Definition von $low^2$ unglücklich gewählt wurde, da mit hoher Wahrscheinlichkeit nur die leere Menge in $low_k^2$ enthalten ist.

Satz 5.4.5
*Für $A \neq \emptyset$ gilt: $A \in low_k^2 \Leftrightarrow NP(k) = co\text{-}NP(k)$*

Beweis:

- $\Rightarrow$ Da $A \neq \emptyset$ folgt $\Sigma^* \in \mathcal{R}_m^p(A)$. Deshalb ergibt sich:

  $$co\text{-}NP(k) = NP(k) \oplus \Sigma^* \subseteq NP(k) \oplus \mathcal{R}_m^p(A) = NP(k)$$

  Es gilt auch $NP(k) = co\text{-}co\text{-}NP(k) \subseteq co\text{-}NP(k)$ und damit ist$NP(k) = co\text{-}NP(k)$.

- $\Leftarrow$ Sei $A \in NP \backslash \{\emptyset\}$ mit Lemma 2.2.2 folgt dann:

  $$\begin{array}{lclcl} NP(k) & \subseteq & NP(k) \oplus \mathcal{R}_m^p(A) & \subseteq & NP(k+1) \\ NP(k) & \subseteq & NP(k) \oplus \mathcal{R}_m^p(A) & \subseteq & NP(k) \end{array}$$

  Damit gilt $A \in low_k^2$.

✔

Korollar 5.4.1

$$low_k^2 = \begin{cases} \{\emptyset\}, & falls\ NP(k) \neq co\text{-}NP(k) \\ NP & sonst \end{cases}$$

Korollar 5.4.2
*Falls die Polynomialzeithierarchie echt ist, gilt für alle $k \geqslant 1$:*

$$low_k^2 = \{\emptyset\}$$

Beweis:
Direkte Folgerung aus den Sätzen 2.2.3, 5.4.5 und Lemma 5.4.3. ✔

# 6 Eine dritte Definition

## 6.1 $high^3$-Mengen in der Booleschen Hierarchie

Die dritte hier untersuchte Definition von „High- und Low-Mengen“ ist stärker an die erste Definition der Booleschen Hierarchie (siehe Definition 2.2.2) angelehnt. Wie bei der zweiten Definition lassen sich die Inklusionen der „High- bzw. Low-Mengen“ mit relativ wenig Aufwand zeigen (siehe Sätze 6.1.3 und 6.2.2).

DEFINITION 6.1.1
*Seien die Mengen $NP_A(k)$ wie folgt induktiv definiert:*

$$\begin{array}{lll}
NP_A(1) & =_{def} & \mathcal{R}_m^p(A) \\
NP_A(2) & =_{def} & NP_A(1) \wedge co\text{-}NP \\
NP_A(3) & =_{def} & NP_A(2) \vee NP \\
\vdots & & \vdots \\
NP_A(2i) & =_{def} & NP_A(2i-1) \wedge co\text{-}NP \quad (i \geqslant 1) \\
NP_A(2i+1) & =_{def} & NP_A(2i) \vee NP \quad (i \geqslant 0)
\end{array}$$

LEMMA 6.1.1
*Es sei $k \geqslant 1$. Für jede Sprache $A \in NP \backslash \{\emptyset\}$ gilt dann:*

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

Beweis:
Diese Aussage kann durch Induktion recht leicht gezeigt werden.
(IA):

$$co\text{-}NP \subseteq \mathcal{R}_m^p(A) \wedge co\text{-}NP \subseteq NP \wedge co\text{-}NP = NP(2)$$

(IS):

$$\begin{array}{rcl}
co\text{-}NP(2i+1) & = & co\text{-}NP(2i) \wedge co\text{-}NP \\
& \subseteq & NP_A(2i+1) \wedge co\text{-}NP = NP_A(2i+2) \\
& \subseteq & NP(2i+1) \wedge co\text{-}NP = NP(2i+2) \\
\\
co\text{-}NP(2i+2) & = & co\text{-}NP(2i+1) \vee NP \\
& \subseteq & NP_A(2i+2) \vee NP = NP_A(2i+3) \\
& \subseteq & NP(2i+2) \vee NP = NP(2i+3)
\end{array}$$

✔

Mit Lemma 6.1.1 werden die nachfolgenden Definitionen von $high^3$- und $low^3$-Mengen etwas verständlicher.

DEFINITION 6.1.2
*Sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $high_k^3 \Leftrightarrow_{def}$*

- $A \in NP$
- $NP_A(k+1) = NP(k+1)$

SATZ 6.1.2
*Für jede NP-vollständige Menge $A$ gilt $A \in high_k^3$.*

Beweis:
Wir zeigen $NP_A(k) = NP(k)$ durch vollständige Induktion über $k \geqslant 1$.
(IA): Da $A$ NP-vollständig ist, folgt $\mathcal{R}_m^p(A) = NP$. Hiermit folgt $NP_A(1) = NP$.
(IS):

$$\begin{array}{rclcl} NP_A(2i+1) & = & NP_A(2i) \vee NP & = & NP(2i) \vee NP \\ NP_A(2i+2) & = & NP_A(2i+1) \wedge co\text{-}NP & = & NP(2i+1) \wedge co\text{-}NP \end{array}$$

Damit folgt $NP_A(k+1) = NP(k+1)$, d.h. $A \in high_k^3$. ✔

SATZ 6.1.3
*Für jedes $k \geqslant 1$ gilt $high_k^3 \subseteq high_{k+1}^3$.*

Beweis:
Sei $A \in high_k^3$, dann folgt:

$2 \mid k$: $NP_A(k+1) = NP(k+1)$ und mit Definition 2.2.3 ergibt sich:

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

$2 \nmid k$: $NP_A(k+1) = NP(k+1)$ und damit:

$$\begin{array}{rcl} NP_A(k+2) & = & NP_A(k+1) \vee NP \\ & = & NP(k+1) \vee NP \\ & = & NP(k+2) \end{array}$$

Damit ist $A \in high_{k+1}^3$. ✔

LEMMA 6.1.4
*Aus $A \leqslant_m^p B$ folgt $NP_A(k) \subseteq NP_B(k)$ für alle $k \geqslant 1$.*

Beweis:
Dieses Aussage kann durch eine Induktion über $k$ gezeigt werden:
(IA):

$$NP_A(1) = \mathcal{R}^p_m(A) \subseteq \mathcal{R}^p_m(B) = NP_B(1)$$

(IS):

$$\begin{array}{rcl} NP_A(2k+1) & = & NP_A(2k) \vee NP \\ & \subseteq & NP_B(2k) \vee NP \\ & = & NP_B(2k+1) \\ & & \\ NP_A(2k+2) & = & NP_A(2k+1) \wedge co\text{-}NP \\ & \subseteq & NP_B(2k+1) \wedge co\text{-}NP \\ & = & NP_B(2k+2) \end{array}$$

✔

SATZ 6.1.5
*Für jede Menge $A \in high^3_k$ und jedes $B \in NP$ mit $A \leqslant^p_m B$ folgt $B \in high^3_k$.*

Beweis:
Mit den Lemmata 6.1.1 und 6.1.4 folgt:

$$NP(k+1) = NP_A(k+1) \subseteq NP_B(k+1) \subseteq NP(k+1)$$

Damit folgt $B \in high^3_k$. ✔

## 6.2 $low^3$-Mengen in der Booleschen Hierarchie

DEFINITION 6.2.1
*Es sei $k \geqslant 1$. Eine Menge $A \subseteq \Sigma^*$ ist in $low^3_k \Leftrightarrow_{def}$*

- $A \in NP$
- $NP_A(k+1) = co\text{-}NP(k)$

Das nächste Resultat ist von Interesse, weil es zeigt, daß bei dieser Definition von Low-Mengen $low^3_1$ mit der in [Sch85] beschriebenen Menge $Low^p_1$ übereinstimmt.
Alle anderen Resultate lassen sich gut mit denen der zweiten Definition vergleichen.

SATZ 6.2.1
$NP \cap co\text{-}NP = low^3_1$.

Beweis:

$\subseteq$ Sei $A \in NP \cap co\text{-}NP$. Da $co\text{-}NP$ unter $\leqslant^p_m$-Reduktion abgeschlossen ist, gilt $\mathcal{R}^p_m(A) \subseteq co\text{-}NP$. Damit folgt:

$$\begin{array}{rcl} NP_A(2) & = & \mathcal{R}^p_m(A) \wedge co\text{-}NP \\ & = & co\text{-}NP \\ & = & co\text{-}NP(1) \end{array}$$

D.h. $A \in low^3_1$.

$\supseteq$ Sei $A \in low_k^3$, d.h. $NP_A(2) = \mathcal{R}_m^p(A) \wedge co\text{-}NP = co\text{-}NP$. Da $A \in \mathcal{R}_m^p(A)$ und $\Sigma^* \in co\text{-}NP$ ergibt sich $A = A \cap \Sigma^* \in co\text{-}NP$. Nach Definition 6.2.1 ist $A \in NP$ und deswegen gilt $A \in NP \cap co\text{-}NP$.

Damit gilt $low_1^3 = NP \cap co\text{-}NP$. ✔

KOROLLAR 6.2.1
*$Low_1^p = low_1^3$.*

Beweis:
Siehe Satz 2.1.2. ✔

SATZ 6.2.2
*Für jedes natürliche $k \geqslant 1$ gilt $low_k^3 \subseteq low_{k+1}^3$.*

Beweis:
Sei $A \in low_k^3$, dann gilt $NP_A(k+1) = co\text{-}NP(k)$. Mit Definition 6.1.1 ergibt sich:

$2 \mid k$:

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

$2 \nmid k$:

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

Damit gilt $NP_A(k+2) = co\text{-}NP(k+1)$, d.h. $A \in low_{k+1}^3$. ✔

SATZ 6.2.3
*Für jede Menge $A \in low_k^3$ und jedes $B \in NP \backslash \{\emptyset\}$ mit $B \leqslant_m^p A$, folgt $B \in low_k^3$.*

Beweis:
Mit den Lemmata 6.1.1 und 6.1.4 folgt:

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

✔

# 7 Zusammenfassung

In Abschnitt 3 wurden unabhängig NP-vollständige-Mengen untersucht. Dabei konnte gezeigt werden, daß solche Tupel von NP-vollständigen Mengen existieren und sich sogar „natürliche“ Beispiele mit anschaulichen NP-vollständigen Mengen finden lassen.
Dabei bleiben noch einige sehr interessante Fragen. Sind zwei verschiedene $p$-Zylinder immer unabhängig NP-vollständig? Ist von einem Paar unabhängig NP-vollständiger Mengen mindestens eine ein $p$-Zylinder?

Der darauffolgende Abschnitt verwendet die Erkenntnisse über unabhängig NP-vollständige Mengen um zu zeigen, daß alle NP-vollständigen $p$-Zylinder in $high_k^1$ enthalten sind. Leider konnte dies nicht auf alle NP-vollständigen Mengen verallgemeinert werden.
Bei der Untersuchung von $high^1$- und $low^1$-Mengen zeigte es sich, daß die Beantwortung der Fragen

$$high_k^1 \overset{?}{\subseteq} high_{k+1}^1$$

und

$$low_k^1 \overset{?}{\subseteq} low_{k+1}^1$$

schwierig ist.

Für die zweite Definition konnten die Inklusionsbeziehungen der $high^2$- und $low^2$-Klassen vergleichsweise einfach gezeigt werden. Dabei wurden auch $mid^2$-Mengen untersucht. Es zeigt sich, daß die Definition von $low^2$-Mengen etwas unglücklich ist, da in diesem Sinne mit großer Wahrscheinlichkeit nur die leere Menge „low“ ist. Die intuitiv genauso schwierigen endlichen Mengen sind aller Wahrscheinlichkeit schon nicht mehr in $low_k^2$ enthalten.
Weiterhin konnte, vergleichbar zu den in [Sch85] definierten „High-Mengen“, gezeigt werden, daß alle NP-vollständigen Mengen in $high_k^2$ enthalten sind.

Die Ergebnisse der dritten Definition sind denen über die zweite Definition sehr ähnlich. Allerdings kann man davon ausgehen, daß diese Definition andere „High- bzw. Low“ Mengen beschreibt. Im Gegensatz zur kompakten zweiten Definition bei der die symmetrische Mengendifferenz verwendet wird, erscheint die dritte Definition etwas unhandlich, da die geradzahligen und ungeradzahligen Stufen der Booleschen Hierarchie unterschieden werden müssen.

# Erklärung zur Diplomarbeit

Hiermit erkläre ich, daß ich diese Arbeit eigenständig verfaßt und keine anderen Hilfsmittel als die angegebenen verwendet habe.

Würzburg den, __________ ______________________

# Literaturverzeichnis


[BDG90a] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, first edition, 1990.

[BDG90b] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II.* 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.

[JY85] D. Joseph and P. Young. Some remarks on witness functions for nonpolynomial and noncomplete sets in NP. *Theoretical Computer Science*, 39:225 – 237, 1985.

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

[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.

[Rog88] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* The MIT Press, second edition, 1988.

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

[Wag94] K. W. Wagner. *Einführung in die Theoretische Informatik.* Springer Verlag, 1994.