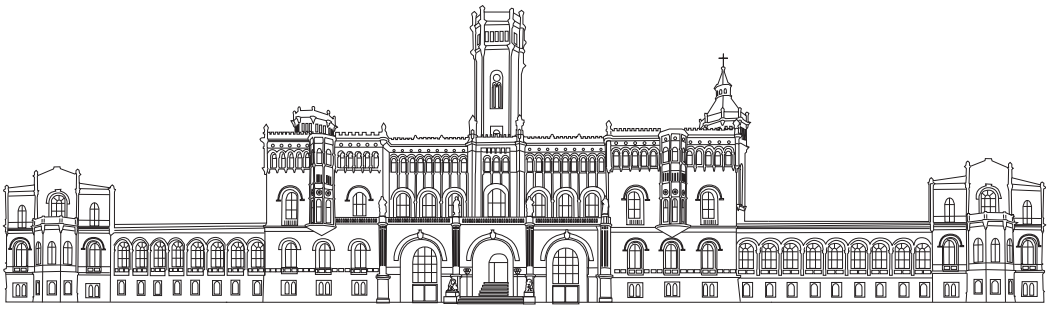


Leibniz Universität Hannover

Fakultät für Elektrotechnik und Informatik

Institut für Theoretische Informatik

Komplexitätstheorie über algebraischen Körpern

Masterarbeit

Vivian Holzapfel

3223460

Erstprüfer	:	Prof. Dr. Heribert Vollmer
Zweitprüfer	:	PD Dr. Arne Meier
Betreuerin	:	M.Sc. Sabrina Alexandra Gaube

Erklärung der Selbstständigkeit

Hiermit versichere ich, die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 18. April 2021

Vivian Holzapfel

Kurzfassung

Diese Arbeit gibt eine Einführung in die Komplexitätstheorie über algebraischen Körpern. Dazu werden die von Valiant definierten algebraischen Komplexitätsklassen VP und VNP vorgestellt sowie eine neue Art Reduktion, unter welcher diese abgeschlossen sind, erläutert. Anhand zweier Beispiele werden VNP-Vollständigkeitsbeweise geführt. Als Gegenstück zur P-NP-Hypothese der klassischen Komplexitätstheorie wird die Valianthypothese präsentiert und die Schwierigkeit ihre Lösung zu finden auf das bekannte mathematische Problem des Zusammenhangs der Permanente und Determinante zurückgeführt. Abschließend wird die algebraische mit der klassischen Komplexitätstheorie verglichen und die Auswirkungen der möglichen Antworten der Valianthypothese auf die Beziehung zwischen P und NP dargestellt.

Abstract

This thesis gives an introduction into algebraic complexity theory. For this purpose the algebraic complexity classes VP and VNP defined by Valiant are presented as well as a new type of reduction under which these classes are closed. Using two examples VNP-completeness proves are done. As counterpart to the P-NP-hypothesis from the classic complexity theory Valiant's hypothesis is introduced and the difficulty to find its solution is traced back to the known mathematical problem of the relation between the permanent and determinant. Lastly algebraic complexity theory is compared to the classical and effects of the possible answers to Valiant's hypothesis on the relation between P and NP are outlined.

Inhaltsverzeichnis

1 Grundlagen der Algebra **1**

2 Komplexitätstheorie **4**

2.1 Klassische Komplexitätstheorie 4

2.2 Algebraische Komplexitätstheorie 6

2.2.1 Straight-Line-Programme 6

2.2.2 Arithmetische Schaltkreise 8

3 Valiants algebraisches Modell **11**

3.1 Die Komplexitätsklassen VP und VNP 11

3.1.1 Die Komplexitätsklasse VP 11

3.1.2 Die Komplexitätsklasse VNP 13

3.1.3 Die Permanente 14

3.1.4 Valiants Kriterium 16

3.2 Valiants VNP-Vollständigkeitsbegriff 18

3.3 VNP-vollständige Probleme 20

3.3.1 Matchings 21

3.3.2 Hamilton'sche Kreise 24

3.4 Valianthypothese 28

3.4.1 VP-Vollständigkeit 29

3.4.2 Erweiterte Valianthypothese 31

3.4.3 Permanente versus Determinante 34

4 Vergleich der klassischen und algebraischen Komplexitätstheorie **37**

4.1 Modell von Cook 37

4.2 Vergleich der Modelle von Cook und Valiant 38

4.2.1 Vergleich der Reduktion und Projektion 40

4.3 Parallele Komplexität 41

4.4 Valiant- versus Cookhypothese 44

4.4.1 Beziehungen der Valiantklassen zur klassischen Komplexitätstheorie 44

4.4.2 Zusammenhang zur Klasse NP 47
4.4.3 Beweiseideen der Beziehungen zur klassischen Komplexitätstheorie 49
4.4.4 Auswirkungen der Valianthypothese auf die klassische Komplexitätstheorie 52
4.4.5 Vergleich zur Cookhypothese 53

5 Ausblick **55**

Abbildungsverzeichnis

2.1 Untere und Obere Schranken 5
2.2 Straight-Line-Programm (SLP) als Multigraph 8
2.3 Arithmetischer Schaltkreis für $f = X^3 + 2X^2$ 10

3.1 Perfektes Matching 15
3.2 Partielles Matching in $K_{2,2}$ 21
3.3 Partielles Matching in G 23
3.4 Hamilton'sche Pfade über XOR-Kopplung 24
3.5 XOR-Kopplung und Symbolbild [Bür13] 25
3.6 Umwandlung kreuzender XOR-Kopplungen in planaren Graphen [Bür13] . 25
3.7 Hilfsgraph C und Symbolbild [Bür13] 25
3.8 Erzeugung der Graphen V_e [Bür13] 26
3.9 Kreisförmige Anordnung der V_e und Hilfgraphen C 26
3.10 Graph G_n [Bür13] 27
3.11 Fall π kein partielles Matching 27
3.12 Klassendiagramm der Algebraischen Komplexitätstheorie (erweiterte Valianthypothese gilt) 35
3.13 Klassendiagramm der Algebraischen Komplexitätstheorie (erweiterte Valianthypothese gilt nicht) 35

4.1 Verhältnis von P und NP 39
4.2 Beweis HP $\leq_m^{\mathsf{P}}$ HC 41
4.3 Auswirkung auf die Polynomialzeithierarchie 53

Abkürzungsverzeichnis

BP Boole'scher Anteil

BSS Blum-Shub-Smale-Modell

CSP Constraint Satisfaction Problem

DET Determinante

$\mathcal{DI}$ Dimer Überdeckungen

GCT Geometrische Komplexitätstheorie

GF erzeugende Funktion

HAMCIRC Hamilton'sche Kreise

HAMPATH Hamilton'sche Pfade

$\mathcal{MD}$ Monomer-Dimer Überdeckungen

PER Permanente

PH Polynomialzeithierarchie

PM perfektes Matching

RAM Random Access Machine

SAT Erfüllbarkeitsproblem aussagenlogischer Formeln

SLP Straight-Line-Programm

VC Vertex Cover

XOR-Kopplung Exklusiv-Oder Kopplung

1 Grundlagen der Algebra

Dieses Kapitel richtet sich nach *Lineare Algebra* [Bos14] und *Algebra* [Bos20] von Bosch und gibt eine Übersicht über die benötigten mathematischen Grundlagen aus der Algebra. Der Begriff des Körpers ist zentraler Bestandteil dieser Arbeit. Um ihn zu definieren werden zuvor die dafür grundlegenden mathematischen Strukturen erläutert. Desweiteren werden mit den multivariaten Polynomringen die untersuchten Elemente der algebraischen Komplexitätstheorie eingeführt.

Definition 1.1. Ein Monoid ist eine Menge S mit einer zweistelligen Verknüpfung

$$\cdot\colon S \times S \to S,\ (s,t) \mapsto s \cdot t$$

und einem neutralen Element $e \in S$ für das $e \cdot s = s \cdot e = s$ gilt. Ein Monoid ist kommutativ, falls $s \cdot t = t \cdot s\ \forall s,t \in S$.

Definition 1.2. Eine Gruppe ist eine Menge G mit einer zweistelligen Verknüpfung $G \times G \to G,\ (g,h) \mapsto g \cdot h = gh$, einem neutralen Element $e \in G$ sowie einer einstelligen Inversion $G \to G,\ g \mapsto g^{-1}$, sodass folgende Bedingungen erfüllt sind:

1. Für alle $g,h,k \in G$ gilt $(gh)k = g(hk)$. (Assoziativität)
2. Für alle $g \in G$ gilt $g \cdot e = e \cdot g = g$. (neutrales Element)
3. Für alle $g \in G$ gilt $gg^{-1} = g^{-1}g = e$. (inverses Element)

Eine Gruppe ist somit ein Monoid, welcher ein Inverses besitzt. Sie heißt abelsch oder kommutativ, wenn zusätzlich das Kommutativgesetz gilt, d. h. $gh = hg$ für alle $g,h \in G$.

Die Definitionen der Gruppe und des Monoiden werden nun zur Definition des Rings herangezogen, auf dessen Basis im Folgenden die Polynomringe eingeführt werden.

Definition 1.3. Ein Ring ist eine Menge R mit zwei zweistelligen Verknüpfungen „$+$“ und „$\cdot$“, dem Element 1 sowie einem additiven Inversen, sodass folgende Bedingungen erfüllt sind:

1. $(R, +, 0)$ ist eine abelsche Gruppe.

2. $(R, \cdot, 1)$ ist ein Monoid.

3. Beide Distributivgesetze gelten.

Ein Ring ist kommutativ, wenn das Monoid $(R, \cdot, 1)$ kommutativ ist.

Für die dritte Anforderung sind zwingend beide Distributivgesetze notwendig, da die Kommutativität nicht zwangsläufig gegeben ist. Ist das Monoid und somit auch der Ring kommutativ, so gelten entweder beide Distributivgesetze oder keins. Es muss daher auch nur eines geprüft werden.

Eine Verallgemeinerung der Ringe sind sogenannte Halbringe. Diese unterscheiden sich darin, dass für die Verknüpfung der Addition nur noch eine Halbgruppe gefordert ist. In einer Halbgruppe muss die Verknüpfung nicht invertierbar sein und es muss kein neutrales Element existieren. Ein Monoid ist eine Halbgruppe mit neutralem Element.

Definition 1.4. Ein Halbring ist eine Menge R mit zwei zweistelligen Verknüpfungen „$+$“ und „$\cdot$“, dem Element 1 sowie einem additiven Inversen, sodass folgende Bedingungen erfüllt sind:

1. $(R, +, 0)$ ist eine Halbgruppe.

2. $(R, \cdot, 1)$ ist ein Monoid.

3. Beide Distributivgesetze gelten.

Halbringe werden in Abschnitt 2.2.2 zur Definition eines Berechnungsmodells der algebraischen Komplexitätstheorie benötigt. Außerdem lassen sich Ringe zu sogenannten Polynomringen, d. h. allen Polynomen über einem Ring R erweitern.

Definition 1.5. Sei R ein kommutativer Ring mit 1, dann ist

$$R[X] \coloneqq \{(a_i)_{i \in \mathbb{N}} \in R^{\mathbb{N}} \mid a_i = 0 \text{ für alle bis auf endlich viele } i\}$$

der Polynomring in der Unbestimmten X über R. $R[X]$ ist kommutativ. Die Elemente von $R[X]$ heißen Polynome.

Basierend auf dieser Definition werden auch Polynome über mehreren Unbestimmten über einem Ring R definiert.

Definition 1.6. Ein multivariater Polynomring in n Unbestimmten $X_1, \ldots, X_n$ über R ist iterativ definiert als

$$R[X_1, \ldots, X_n] = (\ldots((R[X_1])[X_2])\ldots)[X_n],$$

wobei $R[X_1]$ ein Polynomring über der Variable X_1 ist.

Die beschriebenen Berechnungen geschehen im Folgenden immer über einer bestimmten algebraischen Strukturen, sogenannten Körpern. Diese lassen sich mit den vorangegangen Definitionen charakterisieren.

Definition 1.7. Ein Körper ist eine Menge k mit zwei zweistelligen Verknüpfungen, Addition „$+$“ und Multiplikation „$\cdot$“, sodass folgende Bedingungen gelten:

1. $(k, +, 0)$ ist eine abelsche Gruppe.
2. $(k \setminus \{0\}, \cdot, 1)$ ist eine abelsche Gruppe.
3. Das Distributivgesetz gilt.

Die Eigenschaften von Körpern lassen sich insbesondere über ihre Charakteristik definieren und unterscheiden. Dies wird besonders in Abschnitt 4.4.1, welches die Auswirkungen auf die klassische Komplexitätstheorie behandelt, nützlich.

Definition 1.8. Die Charakteristik eines Körpers k ($\operatorname{char} k$) ist definiert als die Primzahl $p > 0$, sodass $p \cdot 1 = 0$. Existiert kein solches p, so ist ist $\operatorname{char} k = 0$.

2 Komplexitätstheorie

Dieses Kapitel gibt eine Übersicht über das Forschungsgebiet der Komplexitätstheorie. Dabei wird zuerst auf die klassische Komplexitätstheorie eingegangen und nachfolgend die titelgebende algebraische Komplexitätstheorie eingeführt sowie ihre Berechnungsmodelle beschrieben.

2.1 Klassische Komplexitätstheorie

Das Gebiet der Komplexitätstheorie befasst sich mit der Bestimmung benötigter Berechnungsressourcen zur Lösung eines Problems. Das Ziel ist, gegeben ein solches Problem, eine zugehörige optimale algorithmische Lösung zu finden, sofern es eine solche gibt, sowie die Optimalität dieser Lösung zu beweisen. Die Kosten der Berechnung einer solchen optimalen Lösung sind über die Anzahl an Schritten, die der zugehörige Algorithmus durchführt, definiert. Sie werden als Komplexität des betrachteten Problems bezeichnet. Die Schwierigkeit liegt darin, die tatsächliche Optimalität einer Lösung zu beweisen. Diese setzt sich aus einer unteren und einer oberen Schranke benötigter Kosten zusammen. Wobei die untere Schranke festlegt, wie viele Kosten mindestens, und die obere wie viele maximal anfallen. Fallen diese beiden Schranken für ein Problem zusammen, so sind die optimalen Kosten ermittelt. Gerade die unteren Schranken sind jedoch schwer zu bestimmen und so stellt es eher eine Seltenheit dar, dass der Fall genau bestimmter Kosten eintritt [BCS13].

Zur besseren Einordnung und Vergleich der Komplexität verschiedener Probleme werden feste Komplexitätsklassen verwendet. Eine Komplexitätsklasse $\mathcal{K}$ ist eine Menge von Funktionen, die mit einer bestimmten Menge gegebener Ressourcen berechnet werden können. Beispielsweise entspricht die Klasse P denjenigen Boole'schen Funktionen deren Berechnung in deterministischer Polynomialzeit erfolgen kann [AB09]. Die Intuition der Schranken lässt sich nun auf die Komplexitätsklassen übertragen. Kann für ein be-

liebiges Problem A gezeigt werden, dass $A \in \mathcal{K}$, so ist dies eine obere Schranke. Eine untere Schranke wird darüber bestimmt dass sich das Problem A nicht in einer darunter liegenden Klasse befindet, d. h. $A \notin \mathcal{K}'$ für $\mathcal{K}' \subsetneq \mathcal{K}$. Dieses Verhalten ist in Abb. 2.1 dargestellt.

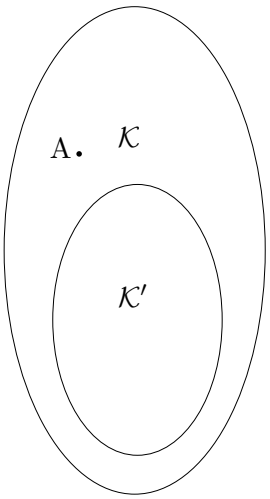


Abbildung 2.1: Untere und Obere Schranken

Verdeutlicht wird dies anhand des Beispiels eines Sortieralgorithmus, welcher n Zahlen in aufsteigender Reihenfolge anordnet. Jedes vergleichsbasierte Sortierverfahren benötigt im schlechtesten Fall $\Omega(n \log n)$ Vergleichsoperationen [OW17]. Dies ist die untere Schranke. Mit dem Algorithmus Heapsort lässt sich als obere Schranke $\mathcal{O}(n \log n)$ zeigen [OW17].

In manchen Fällen ist es jedoch nicht möglich, eine konstruktive algorithmische Lösung zu finden. In diesem Fall können Algorithmen nur über Reduktionen in eine gewisse Klasse eingeordnet werden.

Die klassische Komplexitätstheorie arbeitet auf einem endlichen Binäralphabet und nutzt die Turingmaschine als grundlegendes Berechnungsmodell. Die untersuchten Probleme sind Entscheidungsprobleme, welche durch Spachen formalisiert werden. Ein Beispiel ist die Sprache

$$L_1 = \{w \in \{0,1\}^* \mid w \text{ endet auf } 000\},$$

welche aus all den Wörtern über dem Binäralphabet besteht die auf 000 enden.

2.2 Algebraische Komplexitätstheorie

Die algebraische Komplexitätstheorie schränkt das weite Feld der Komplexitätstheorie ein, indem sie sich nur mit Problemen befasst, welche mit algebraischen Algorithmen gelöst werden können [BCS13]. Algebraisch bedeutet so viel wie das Rechnen mit polynomiellen Gleichungen endlichen Grades [Bos20]. Durch die Beschränkung auf algebraische Algorithmen wird der Suchraum aller möglichen Algorithmen zur Lösung eines Problems reduziert. Diese Algorithmen verwenden ausschließlich die vier grundlegenden arithmetischen Operationen $+, -, *, /$ über Körpern, welche jeweils mit unendlicher Präzision in einem Berechnungsschritt ausgeführt werden können. Im Kontext der algebraischen Komplexitätstheorie werden nun außerdem multivariate Polynome (s. Def. 1.6) evaluiert. Diese stellen das Äquivalent zu den Entscheidungsproblemen der klassischen Komplexitätstheorie dar.

Eine praktische Anwendung liegt in der Computeralgebra. Dieses Gebiet beschäftigt sich mit dem symbolischen Rechnen zum Lösen mathematisch formulierter Probleme durch symbolische Algorithmen, beispielsweise dem Rechnen mit beliebig langen Zahlen, Symbolen, Unbestimmten und Polynomen oder dem exakten Lösen polynomieller Gleichungssysteme. Im Gegensatz zum numerischen Ansatz, welcher zu große Zahlen approximiert und somit nur ebenfalls approximierte Ergebnisse liefert, werden alle Objekte durch Symbole exakt dargestellt und nicht durch ihren tatsächlichen Wert ersetzt [GK03].

Das Berechnungsmodell der algebraischen Komplexitätstheorie sind die Straight-Line-Programme (SLP) und die arithmetischen Schaltkreise, welche genau dieses Verhalten simulieren.

Eine Besonderheit der in dieser Arbeit betrachteteten algebraischen Komplexitätstheorie ist es, dass die Optimalität der Komplexität der betrachteten Probleme häufiger gezeigt werden kann [Gat88].

2.2.1 Straight-Line-Programme

Straight-Line-Programme (SLP) sind, basierend auf den vier arithmetischen Operationen über einem festen Körper k, in der Lage, Lösungen für algebraische Probleme über Polynomringen zu berechnen. Die Berechnungen werden dabei schrittweise und ohne Verzweigungen durchgeführt, wobei jede Operation Kosten von nur einem Schritt hat. Die Gesamtkosten setzen sich aus der Summe aller ausgeführten Operationen zusammen.

Definition 2.1. (1) (Syntax) Ein SLP Γ über einem Körper k und mit Eingabelänge m ist eine Folge von Instruktionen $(\Gamma_1, \ldots, \Gamma_r)$, wobei $\Gamma_p = (\omega_\rho;\ i_\rho,\ j_\rho)$, mit Operationssymbolen $\omega_\rho \in \{+,\ -,\ *\}$ sowie ganzzahligen Adressen $-n < i_\rho,\ j_\rho < \rho$. Die Größe bzw. Länge des SLP ist r.

(2) (Semantik) Ein SLP Γ berechnet für jede Eingabesequenz von Polynomen $a_1,\ \ldots,\ a_m$ eine eindeutige Lösungssequenz $(b_{-n+1}, \ldots, b_r)$, wobei

$$b_\rho = \begin{cases} a_{m+\rho} & \text{für } \rho \leq 0 \\ b_{i_\rho} \omega_\rho b_{j_\rho} & \text{für } \rho > 0. \end{cases}$$

Ein SLP Γ berechnet eine Menge von Polynomen F aus der Eingabe, gdw. $F \subseteq \{b_{-m+1}, \ldots,\ b_r\}$.

(3) Ein SLP Γ definiert einen gerichteten azyklischen Multigraphen, mit den Knoten $\{\rho \in \mathbb{Z} \mid -n < \rho < r\}$ und den Kanten $(i_\rho, \rho),\ (j_\rho, \rho)$. Die Anzahl der Kanten auf dem längsten gerichteten Pfad im Graph wird als Tiefe von Γ bezeichnet.

Die Menge F entspricht hierbei den Sprachen L der klassischen Komplexitätstheorie. Verdeutlichen lässt sich dieses Konzept anhand eines Beispiels. Gegeben sei als Eingabe das Polynom $f = X^3 + 2X^2$ über dem Körper $\mathbb{R}$. Ein mögliches SLP Γ_f könnte wie folgt aussehen:

Beispiel 2.2.

$$\Gamma_1 = (X), \qquad \Gamma_2 = (*;\ 1,\ 1), \qquad \Gamma_3 = (*;\ 2,\ 1), \qquad \Gamma_4 = (2),$$
$$\Gamma_5 = (*;\ 4,\ 2), \qquad \Gamma_6 = (+;\ 5,\ 3).$$

Wie in Punkt (3) aus Definition 2.1 definiert lässt sich auch dieses SLP als gerichteter azyklischer Multigraph darstellen. Die doppelte Kante zwischen den Knoten 1 und 2 ergibt sich daraus, dass die beiden Adressen der Instruktion Γ_2 identisch sind.

Damit berechnet dieses SLP die Menge $\{X,\ X^2,\ X^3,\ 2,\ 2X^2,\ X^3+2X^2,\ X^3+2X^2\}$. Enthalten darin ist das anfangs gewählte Polynom $f = X^3+2X^2$, d. h. nach Definition 2.1 berechnet das SLP Γ_f das Polynom f.

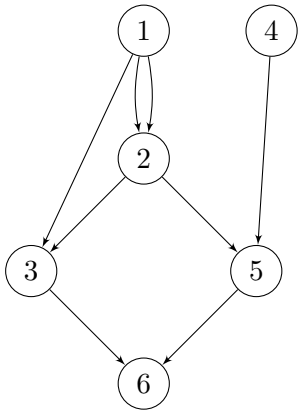


Abbildung 2.2: SLP als Multigraph

2.2.2 Arithmetische Schaltkreise

Ein weiteres Berechnungsmodell ähnlich zu den Straight-Line-Programmen sind die arithmetischen Schaltkreise, eine Abwandlung der Boole'schen Schaltkreise. Anstelle von Boole'schen Operationen führen diese arithmetische Operationen aus und können somit Funktionen berechnen. Dieser Abschnitt richtet sich nach [Vol99].

Definition 2.3. Sei $\mathcal{R} = (R, +, \cdot, 0, 1)$ ein Halbring und $\mathcal{B}$ eine endliche Menge an Funktionen bzw. Familien von Funktionen über R. Mit $(\mathcal{R}, \mathcal{B})$ wird die arithmetische Basis eines Schaltkreises bezeichnet.

Ein arithmetischer Schaltkreis ist über einer arithmetischen Basis $(\mathcal{R}, \mathcal{B})$ definiert. In der algebraischen Komplexitätstheorie handelt es sich bei $\mathcal{R}$ um einen Polynomring, welcher insbesondere ein Ring ist [Bos20]. Die Basis besteht aus der zweistelligen Addition und Multiplikation.

Definition 2.4. Ein arithmetischer Schaltkreis ist ein Tupel $C = (V, E, \alpha, \beta, \omega)$ mit (V, E) einem endlichen gerichteten azyklischen Graphen sowie Funktionen $\alpha\colon E \to \mathbb{N}$, welche die Eingangsreihenfolge festlegt, $\beta\colon V \to \mathcal{B} \cup \{X_1, \ldots, X_n\}$, welche die inneren Gatter markiert und $\omega\colon V \to \{Y_1, \ldots, Y_m\} \cup \{*\}$, welche gleiches für die Ausgangsgatter macht. Es müssen die folgenden Bedingungen gelten:

1. Für alle $v \in V$ mit Eingangsgrad 0 ist $\beta(v) \in \{X_1, \ldots, X_n\}$ oder eine Konstante aus $\mathcal{B}$.

2. Für alle $v \in V$ mit Eingangsgrad $k > 0$ ist $\beta(v)$ eine k-stellige Funktion oder eine Familie aus $\mathcal{B}$.

3. Für alle $i,\ 1 \leq i \leq n$ existiert maximal ein Knoten $v \in V$, sodass $\beta(v) = X_i$. D. h. jede Variable ist eindeutig.

4. Für alle $i,\ 1 \leq i \leq m$ existiert genau ein Knoten $v \in V$, sodass $\omega(v) = Y_i$. D. h. die Ausgabe ist eindeutig.

Für alle $v \in V$ ist $\mathrm{val}_v \colon \mathcal{R}^n \to \mathcal{R}$ für $a_1, \ldots, a_n \in \mathcal{R}$ die Funktion, welche die Knoten auswertet. Es gilt dabei:

1. Besitzt $v \in V$ Eingangsgrad 0 und $\beta(v) = X_i$, für ein $i,\ 1 \leq i \leq n$, dann ist $\mathrm{val}_v(a_1, \ldots, a_n) \coloneqq a_i$. Ist $\beta(v) = a$ für eine konstante Funktion $a \in \mathcal{B}$, so ist $\mathrm{val}_v(a_1, \ldots, a_n) \coloneqq a$.

2. Ist der Eingangsgrad von $v \in V$ gleich $k > 0$, mit Vorgängerknoten $v_1, \ldots, v_k$, welche durch $\alpha(v_1) < \cdots < \alpha(v_k)$ sortiert sind und $\beta(v) = f \in \mathcal{B}$ eine k-stellige Funktion über $\mathcal{R}$, so ist

$$\mathrm{val}_v(a_1, \ldots, a_n) \coloneqq f\left(\mathrm{val}_{v_1}(a_1, \ldots, a_n), \ldots, \mathrm{val}_{v_k}(a_1, \ldots, a_n)\right).$$

   Ist $\beta(v) = f = (f_n)_{n \in \mathbb{N}}$ eine Familie Funktionen über $\mathcal{R}$, so ist

$$\mathrm{val}_v(a_1, \ldots, a_n) \coloneqq f_k\left(\mathrm{val}_{v_1}(a_1, \ldots, a_n), \ldots, \mathrm{val}_{v_k}(a_1, \ldots, a_n)\right).$$

Die von C berechnete Funktion ist $f_C \colon \mathcal{R}^n \to \mathcal{R}^m$ für alle $a_1, \ldots, a_n \in \mathcal{R}$ mit

$$f_c(a_1, \ldots, a_n) \coloneqq (\mathrm{val}_{v_1}(a_1, \ldots, a_n), \ldots, \mathrm{val}_{v_m}(a_1, \ldots, a_n)),$$

wobei $v_i \in V$ für $1 \leq i \leq m$ eindeutige Gatter mit $\omega(v_i) = Y_i$ sind.

Definition 2.5. Eine Familie arithmetischer Schaltkreise ist $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ mit C_n einem arithmetischen Schaltkreis. Sei f_n die von C_n berechnete Funktion, dann berechnet $\mathcal{C}$ die Familie von Funktionen $f_{\mathcal{C}} \colon \mathcal{R}^* \to \mathcal{R}^*$ mit $f_{\mathcal{C}} = (f_n)_{n \in \mathbb{N}}$.

Die Komplexität arithmetischer Schaltkreise wird, wie auch bei den Boole'schen, in ihrer Größe und Tiefe gemessen.

Definition 2.6. Die Größe eines arithmetischen Schaltkreises SIZE ist die Anzahl aller Gatter, welche keine Eingabegatter sind. Die Tiefe DEPTH ist die Länge des längsten gerichteten Pfades im zum Schaltkreis gehörenden Graphen.

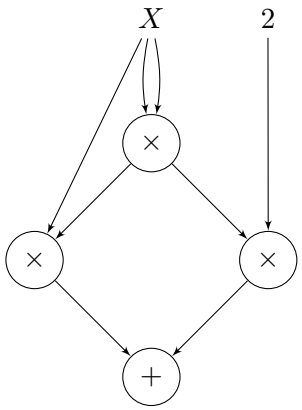


Abbildung 2.3: Arithmetischer Schaltkreis für $f = X^3 + 2X^2$

Jede arithmetische Formel, welche von einem arithmetischen Schaltkreis berechnet werden kann, kann auch durch ein SLP berechnet werden, welches den Schaltkreis simuliert. Dabei stellen die Instruktionen die Gatter in der topologischen Reihenfolge des Schaltkreises dar. Verdeutlicht wird dies in Abb. 2.3 am Beispiel des Polynoms aus Beispiel 2.2. Der resultierende Schaltkreis ähnelt stark der Darstellung des SLPs als Multigraphen. Nach Ben'Or und Cleve [BC92] sind SLP äquivalente Berechnungsmodelle zu parallelen arithmetischen Schaltkreisen. Beide eignen sich je nach Anwendungsbereich als Modell für die algebraische Komplexitätstheorie. Im parallelen Fall wird eher auf die arithmetischen Schaltkreise zurückgegriffen.

3 Valiants algebraisches Modell

In diesem Kapitel werden die Komplexitätsklassen VP und VNP eingeführt, mit der Permanente ein Beispiel für eine Funktion in VNP gegeben sowie über Valiants Kriterium eine Möglichkeit der Überprüfung der Zugehörigkeit zu VNP vorgestellt. Desweiteren wird Valiants Konzept der VNP-Vollständigkeit beschrieben sowie die VNP-Vollständigkeit in zwei Beispiele gezeigt. Abschließend wird die Valianthypothese untersucht und mögliche Lösungsansätze veranschaulicht. Im Zuge dessen wird zusätzlich auf VP-vollständige Probleme eingegangen.

3.1 Die Komplexitätsklassen VP und VNP

Zur Formalisierung der Komplexität algebraischer Probleme werden die Komplexitätsklassen VP und VNP definiert. Diese wurden in Valiant [Val79a] 1979 erstmals beschrieben und 1986 von Strassen [Str86] nach Valiant benannt. Die Definitionen dieses Abschnitts folgen Bürgisser [Bür13].

3.1.1 Die Komplexitätsklasse VP

Sei in diesem Kapitel immer k ein Körper, beispielsweise $\mathbb{R}$, $\mathbb{C}$ oder aber auch ein endlicher Körper, wie $\mathbb{F}_2$, und $k[\underline{X}] \coloneqq k[X_1, ..., X_n]$ der zugehörige Polynomring in n Unbestimmten.

Definition 3.1. Die Komplexität $L(F)$ einer Menge von Polynome $F \subseteq k[X_1, X_2, \ldots, X_n]$ ist die Größe des minimalen SLPs, welches F berechnet.

Die Komplexität des minimalen SLPs, welches beispielsweise X^2 berechnet, wäre 2, eine Instruktion für die Zuweisung der Variable X und eine für die Multiplikation.

Bevor eine Komplexitätsklasse definiert werden kann, werden noch einige grundlegende Definitionen benötigt.

Definition 3.2. Eine Funktion $t\colon \mathbb{N} \to \mathbb{N}$ heißt (polynomiell) p-beschränkt, gdw. es ein $c > 0$ gibt, sodass $t(n) \leq n^c + c \; \forall n \in \mathbb{N}$. Gilt auch die untere Schranke $n^{\frac{1}{c}} - c \leq t(n)$ für alle $n \in \mathbb{N}$, so ist t von oben und unten p-beschränkt.

Die Funktion $f(n) = n^2$ ist z. B. sowohl von oben als auch von unten p-beschränkt, da für $c \geq 3$ die Bedingung $n^{\frac{1}{3}} - 3 \leq n^2 \leq n^3 + 3$ für alle $n \in \mathbb{N}$ gilt.

Diese Definition lässt sich nun auf die Eingaben, Folgen multivariater Polynome, der hier untersuchten algebraischen Probleme übertragen.

Definition 3.3. Eine Folge $f = (f_n)_{n\geq 1}$ multivariater Polynome über k heißt p-Familie, gdw. der Grad von f_n sowie die Anzahl der Variablen von f_n in n p-beschränkt ist.

So ist die Folge $g = (g_n)_{n\geq 1}$ mit $g_n = X_1^n$ eine p-Familie, da alle g_n nur eine Variable enthält und deren Grad n ist. Genauso handelt es sich bei $h = (h_n)_{n\geq 1}$ mit $h_n = \sum_{i=1}^{n} X_i^{3n}$ um eine p-Familie. Die Variablenanzahl ist über die bis n laufende Summe polynomiell beschränkt, der Grad ist dies mit $3n$ für alle Variablen ebenfalls. Im Gegensatz dazu wäre ein Folge $l = (l_n)_{n\geq 1}$ mit $l_n = X_1^{2^n}$ keine p-Familie, da der Grad 2^n exponentiell in n ist.

Definition 3.4. Eine p-Familie $f = (f_n)_{n\geq 1}$ ist p-berechenbar, gdw. die Komplexität $L(f_n)$ in n p-beschränkt ist.

Eine p-Familie ist demnach genau dann auch p-berechenbar, wenn das zugehörige SLP höchstens polynomiell viele Instruktionen benötigt, die Berechnung also in polynomieller Zeit ausführen kann. Darauf basierend kann nun die Komplexitätsklasse VP definiert werden.

Definition 3.5. Die Komplexitätsklasse $\mathsf{VP} = \mathsf{VP}_k$ besteht aus allen p-berechenbaren p-Familien über k.

Einige Beispiele für p-berechenbare Probleme werden in Beispiel 3.6 genannt.

Beispiel 3.6. Die SLP für die p-Familien

$$\mathsf{SUM} \coloneqq (\mathsf{SUM}_n)_{n\geq 1} \qquad \mathsf{SUM}_n \coloneqq X_1 + \cdots + X_n$$
$$\mathsf{PROD} \coloneqq (\mathsf{PROD}_n)_{n\geq 1} \qquad \mathsf{PROD}_n \coloneqq X_1 \cdot \cdots \cdot X_n$$
$$\mathsf{POWSUM} \coloneqq (\mathsf{POWSUM}_n)_{n\geq 1} \qquad \mathsf{POWSUM}_n \coloneqq \sum_{i=1}^{n} X_i^n$$

sind in ihrer Komplexität polynomiell in n beschränkt. Es werden jeweils die grundlegenden Instruktionen der SLPs in einfacher Weise verwendet.

Ein weiteres, nachfolgend wichtiges, Beispiel für ein p-berechenbares Problem dieser Klasse ist DET_n, die Berechnung der Determinante einer $n \times n$ Matrix mit unabhängigen veränderlichen Einträgen.

Definition 3.7. Sei A eine $n \times n$ Matrix, dann ist

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} A_{i,\pi(n)}$$

die Determinante von A, wobei es sich bei S_n um die symmetrischen Gruppe, d. h. allen Permutationen über $\{1, 2, \ldots, n\}$ handelt. Das Signum (sgn) ist das Vorzeichen von π. Es ist definiert als $(-1)^{|\mathrm{inv}(\pi)|}$, wobei inv($\pi$) der Anzahl aller Paare (i, j) mit $\pi(i) > \pi(j)$ für $1 \leq i < j \leq n$ entspricht. Es gilt $\operatorname{sgn}(\pi) = 1$, wenn π gerade, sonst $\operatorname{sgn}(\pi) = -1$.

Mit dem Gauß-Algorithmus kann die Determinante in $\mathcal{O}(n^3)$ berechnet werden. Damit ist die Familie $\mathsf{DET} = (\mathsf{DET}_n)_{n \geq 1}$ p-berechenbar und Teil der Klasse VP.

3.1.2 Die Komplexitätsklasse VNP

Das algebraische Pendant zur Klasse NP umfasst zusätzlich zu VP diejenigen Funktionen, welche p-definierbar, jedoch nicht zwangsläufig p-berechenbar sind.

Definition 3.8. Eine p-Familie $f = (f_n)_{n \geq 1}$ ist p-definierbar, gdw. eine p-berechenbare Familie $g = (g_n),\ g_n \in k[X_1, \ldots, X_{u(n)}]$ existiert, sodass für alle $n \in \mathbb{N}$

$$f_n\left(X_1, \ldots, X_{v(n)}\right) = \sum_{e \in \{0,1\}^{u(n)-v(n)}} g_n\left(X_1, \ldots, X_{v(n)}, e_{v(n)+1}, \ldots, e_{u(n)}\right)$$

gilt.

Die Funktion f_n ist somit von den ersten $v(n)$ Variablen der Familie g_n abhängig. Die übrigen $u(n) - v(n)$ Variablen werden durch Konstanten e_i ersetzt. Die Summe wird über die möglichen Belegungen dieser Konstanten e_i gebildet. Wichtig ist hierbei anzumerken, dass es sich bei f_n im Allgemeinen um eine Summation exponentiell vieler Werte der p-berechenbaren Funktion g_n handelt. Daher kann die Komplexität von f_n exponentiell in n sein.

Definition 3.9. Die Komplexitätsklasse $\mathsf{VNP} = \mathsf{VNP}_k$ besteht aus allen p-definierbaren Familien über k.

Im trivialen Fall gilt für die p-Definierbarkeit $f_n = g_n\ \forall n \in \mathbb{N}$. Daraus folgt eindeutig, dass VP in VNP enthalten ist. Jede p-berechenbare Familie ist auch p-definierbar.

Ein Beispiel für ein Problem der Klasse VNP ist die Permanente (PER) einer $n \times n$ Matrix.

3.1.3 Die Permanente

Definition 3.10. Sei $A \in k^{n\times n}$ eine Matrix, dann ist

$$\operatorname{per}(A) = \sum_{\pi\in S_n} \prod_{i=1}^{n} A_{i,\pi(n)}$$

die Permanente von A.

Die Permanente erinnert in ihrer Definition an die Determinante, unterscheidet sich jedoch durch das fehlende Vorzeichen der Permutation. Im Folgenden sei PER_n die Permanente einer $n \times n$ Matrix mit unabhängigen Unbestimmten sowie $\mathsf{PER} = (\mathsf{PER}_n)_{n\geq 1}$ die p-Familie der Permanenten.

Satz 3.11. *Die p-Familie* $\mathsf{PER} = (\mathsf{PER}_n)_{n\geq 1}$ *ist p-definierbar.*

Beweis. Seien $X = (X_{ij}) \in \{X_{ij}\}^{n\times n}_{1\leq i,j\leq n}$ und $Y = (Y_{lm}) \in \{Y_{lm}\}^{n\times n}_{1\leq l,m\leq n}$ Matrizen über Unbestimmten sowie

$$g_n(X,Y) \coloneqq \alpha_n(Y) \cdot \beta_n(Y) \cdot \mu_n(X,Y)$$

ein Polynom mit

$$\alpha_n(Y) \coloneqq \prod_{i,j,l,m} (1 - Y_{ij}Y_{lm}) \qquad \forall\ 1 \leq i,j,l,m \leq n, \text{ sodass } i = l \iff j \neq m$$

$$\beta_n(Y) \coloneqq \prod_{i=1}^{n}\sum_{j=1}^{n} Y_{ij} \qquad\qquad \mu_n(X,Y) \coloneqq \prod_{i=1}^{n}\sum_{j=1}^{n} X_{ij}Y_{ij}$$

$$\gamma_n(Y) \coloneqq \alpha_n \cdot \beta_n.$$

Für alle $e \in \{0,1\}^{n\times n}$ ist $\gamma(e)$ genau dann ungleich null, wenn es sich bei e um eine Permutationsmatrix handelt. Die Anforderung an eine Permutationsmatrix, jede Spalte und jede Zeile enthält genau eine Eins, wird über die Funktionen α_n, bzw. β_n erzwungen. Enthält jede Spalte und jede Zeile von e höchstens eine Eins ist dies äquivalent dazu, dass $\alpha_n(e) \neq 0$. Gäbe es mehr als eine Eins, so wäre ein Faktor des Produktes null. Unter dieser Bedingung kann für $\beta_n(Y)$ gefolgert werden, dass die Funktion genau dann ungleich null ist, wenn jede Zeile von e mindestens eine Eins enthält. Da in β_n die Zeilensumme gebildet wird, würde eine Nullzeile einen Nullfaktor verursachen.

Damit lässt sich $\mu_n(X,Y)$ nun zu $\mu_n(X,e) = \prod_{i=1}^{n} X_{i\pi(j)}$ vereinfachen, wobei es sich bei π um die zu e zugehörige Permutation handelt.

Die Permutation kann damit als $\mathsf{PER}_n = \sum_{e \in \{0,1\}^{n \times n}} g_n(X, e)$ geschrieben werden und entspricht damit der geforderten Form in der Definition der p-Definierbarkeit 3.8. Die Familie PER ist damit p-definierbar. □

Zur Einordnung der Komplexität von PER im klassischen Fall lässt sich die Äquivalenz zum Problem der Bestimmung der Anzahl perfekter Matchings in einem bipartiten Graphen heranziehen.

Definition 3.12. Sei $G = (V, U, E)$ ein bipartiter Graph. Eine Kantenmenge $M \subseteq E$ ist ein Matching bzgl. G, wenn keine Kanten in M den gleichen Endknoten besitzen. M ist ein perfektes Matching, wenn alle Knoten überdeckt werden [OW17].

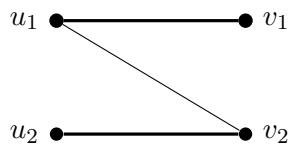


Abbildung 3.1: Perfektes Matching

Zu jeder $n \times n$ Matrix $A = (a_{ij})$ lässt sich der zugehörige bipartite Graph $G_A = (U, V, E)$ definieren, welcher als Knotenmengen die Zeilen bzw. Spaltennummern der Matrix besitzt. Die Kantenmenge E definiert sich über die folgende Beziehung

$$u_i v_i \in E \iff A_{ij} \geq 1.$$

Das Produkt aus der Definition der Permanente, $\prod_{i=1}^{n} A_{i,\pi(n)}$, ist genau dann ungleich null, wenn alle Matrixeinträge $A_{i,\pi(n)}$ ungleich null sind. Dies entspricht der Existenz einer Kante in G_A. Die Summe über alle Permutationen gleicht damit dem Gewicht aller perfekter Matchings in G_A, bzw. im Fall einer Matrix über $\{0, 1\}$ der Anzahl dieser. Die zum Graphen aus Abb. 3.1 gehörige Matrix ist

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Die Gleichung der Permanente ist in diesem Fall

$$\begin{array}{rcccccccc} \mathrm{per}(A) & = & a_{11}a_{12} & + & a_{11}a_{22} & + & a_{21}a_{12} & + & a_{21}a_{22} \\ & = & 1 * 1 & + & 1 * 1 & + & 0 * 1 & + & 0 * 1 \\ & = & 2, & & & & & & \end{array}$$

was auch der Anzahl perfekter Matchings des Graphen entspricht.

Valiant definierte zur Beschreibung der Komplexität die Klasse #P als Klasse der Zählprobleme. Diese sind jeweils zugehörig zu einem Entscheidungsproblem und geben die Anzahl von dessen Lösungen an [Val79b]. Formal lässt sich #P auch über die Anzahl an Zertifikaten y, welche vom Verifizierer V (s. Def. polynomieller Überprüfbarkeit 4.2) des zugehörigen Entscheidungsproblems akzeptiert werden, definieren.

Definition 3.13.

$$\#\mathsf{P} \coloneqq \left\{ f\colon \Sigma^* \mapsto \mathbb{N} \;\middle|\; \begin{array}{l} \Sigma \text{ ist ein Alphabet und es existiert eine} \\ \text{Polynomialzeit NTM } V\text{, so dass für alle } x \\ f(x) = \#\{y \mid (x,y) \in V\} \text{ gilt.} \end{array} \right\}$$

Im Falle der perfekten Matchings liegt das Entscheidungsproblem PM in P [DK11] und das Zählproblem #PM in #P. Valiants Theorem besagt, dass die Permanente für den Sonderfall einer Matrix über $\{0,1\}$ #P-vollständig ist. Im Allgemeinen ist sie mindestens #P-schwer, da 0,1-Matrizen trivialerweise auf beliebige Matrizen reduziert werden können [AB09].

3.1.4 Valiants Kriterium

Für den Beweis der Zugehörigkeit zur Klasse VNP kann anstatt die p-Definierbarkeit einer Funktion direkt zu zeigen auch ein anderes Kriterium verwendet werden. Dieses verwendet eine Funktion welche die Koeffizienten einer Familie von Polynomen erzeugt und beweist durch die Zugehörigkeit dieser Funktion zur Zählklasse #P die p-Definierbarkeit der Familie.

Behauptung 3.14 (Valiants Kriterium). *Angenommen $\phi\colon \{0,1\}^* \to \mathbb{N}$ sei eine Funktion in #P/poly, dann ist die Familie $(f_n)_{n\geq 1}$ von Polynomen*

$$f_n = \sum_{e\in\{0,1\}^n} \phi(e) X_1^{e_1} \dots X_n^{e_n}$$

p-definierbar.

Der Beweis wird für $\phi(e) \in \#\mathsf{P}$ geführt, lässt sich aber auch auf #P/poly übertragen.

Beweis. Angenommen die Funktion ϕ sei in #P. Nach Definition dieser Klasse würde $\phi(e) = \#\{y \mid (e,y) \in V \text{ mit Verfizierer } V\}$ gelten. Nach dem Beweis für die NP-Vollständigkeit von $3-\mathsf{SAT}$ gilt, dass sich die Funktion einer Turingmaschine durch eine Formel Φ_n in 3KNF (konjunktive Normalform mit maximal drei Literalen pro Klausel)

ausdrücken lässt, deren Anzahl erfüllender Belegungen den akzeptierenden Zuständen entspricht. Dies gilt für alle $n \in \mathbb{N}$ [Sip96]. Dabei werden höchstens polynomiell viele neue Variablen $Y_1, \ldots, Y_{m(n)}$ zur Kodierung der Lösung y zu den bestehenden $E_1, \ldots, E_n$, welche die Eingabe e kodieren, hinzugefügt, sodass Φ_n mit $m(n) = n^{\mathcal{O}(1)}$ eine polynomielle Anzahl Klauseln enthält. Für alle $n \in \mathbb{N}$ kann $\phi(e)$ somit auch über

$$\phi(e) = \#\left\{y \in \{0,1\}^{m(n)} \mid \Phi_n(e, y) \text{ ist wahr}\right\}$$

definiert werden. Zur Prüfung der Erfüllbarkeit der gesamten Formel $\Phi_n(e, y)$ wird zuerst die Erfüllbarkeit der einzelnen Klauseln K betrachtet. Diese lässt sich mithilfe eines Polynoms g_k beschreiben. Nach der Definition der 3KNF bestehen die Eingaben aus maximal drei der Variablen aus E_i, Y_i. Der Grad ist ebenfalls durch drei beschränkt, für den Fall dass es sich um dreimal die gleiche Variable handelt.

$$\forall x \in \{0,1\}^3 \colon g_k(x) = \begin{cases} 1, & \text{wenn } K(x) \text{ wahr} \\ 0, & \text{sonst.} \end{cases} \tag{3.1}$$

Sei $K_1 = u \vee v \vee w$ eine beispielhafte Klausel, dann würde die zugehörige Formel g_k würde wie folgt aussehen.

$$\begin{aligned} g_k \coloneqq\ & uvw + uv(1-w) + u(1-v)w + (1-u)vw + (1-u)(1-v)w \\ & + (1-u)v(1-w) + u(1-v)(1-w) \end{aligned}$$

Durch die erfüllenden Belegungen der Variablen u, v und w wird immer genau einer der Summanden 1, während alle anderen 0 ergeben. Die einzige nicht erfüllende Belegung, welche alle Variablen auf 0 setzt, sorgt dafür, dass auch die einzelnen Summanden jeweils 0 sind.

Die Erfüllbarkeit der gesamten Formel lässt sich nach der Korrespondenz der logischen Konjunktion und der Multiplikation über das Produkt p_n der Formeln g_k über alle Klauseln aus Φ_n bestimmen. Die Familie $(p_n)_{n \geq 1}$ ist p-berechenbar. Nach Definition der Formel Φ_n besteht sie aus höchstens polynomiell vielen Variablen, welche einen maximal polynomiellen Grad von $3n^{\mathcal{O}(1)}$ besitzen. Mit der Äquivalenz der Anzahl akzeptierender Zustände und erfüllender Belegungen von Φ_n lässt sich ϕ nun für alle $e \in \{0,1\}^n$ als

$$\phi(e) \coloneqq \sum_{y \in \{0,1\}^{m(n)}} p_n(e, y) \tag{3.2}$$

schreiben. Dies gilt, da p_n genau dann 1 ist, wenn alle Klauseln und somit die gesamte Formel erfüllt sind. Somit tragen nur die erfüllenden Belegungen zur Summe bei.

Um für f_n die in der p-Definierbarkeit (s. Def. 3.8) geforderte Form zu zeigen, werden eine weitere Formel sowie die zusätzlichen Variablen X_i aus f_n benötigt.

$$h_n(X, E, Y) \coloneqq p_n(E, Y) \prod_{i=1}^{n} (E_i X_i + 1 - E_i)$$

Über p_n, die Erfüllbarkeit von Φ_n, wird ähnlich wie in f_n gesteuert welche Variablen in der Formel auftreten. Ist die Formel für eine Belegung nicht erfüllt, so ist $h_n = 0$. Ist $E_i = 1$ entspricht dies einem Faktor von X_i, andernfalls verhindert die Addition von $1 - E_i$ dass das gesamte Produkt 0 wird. Da sowohl p_n als auch das Produkt p-berechenbar sind, gilt dies auch für die zu h_n gehörige Familie $(h_n)_{n\geq 1}$. Die Formel f_n lässt sich nun mithilfe von h_n schreiben:

$$f_n = \sum_{e\in\{0,1\}^n} \phi(e) X_1^{e_1} \cdots X_n^{e_n} = \sum_{e\in\{0,1\}^n} \sum_{y\in\{0,1\}^{m(n)}} h_n(X, e, y).$$

Dabei wird $\phi(e)$ durch Gleichung 3.2 ersetzt. Somit ist f_n p-definierbar, wenn $\phi(e)$ Element von #P ist. □

Lassen sich also die Koeffizienten einer Familie von Polynomen über eine Funktion in #P/poly erzeugen, so ist die Familie in VNP. Dieses Kriterium ist ähnlich wie die Polynomielle Überprüfbarkeit (s. Def. 4.2) der Klasse NP ein „Rezept“ um die Zugehörigkeit zu VNP zu zeigen.

Beispiel 3.15. Sei $f = (f_n)_{n\geq 1}$ eine Familie von Polynomen mit $f_n = 3X_1X_2 + 3X_3$. Alle Koeffizienten sind konstant 3, eine Funktion, welche sie erzeugt, wäre demnach $\phi(e) = 3$. Diese Funktion kann effizient, d. h. in #P/poly berechnet werden und die Familie f liegt demnach in VNP.

3.2 Valiants VNP-Vollständigkeitsbegriff

Als algebraisches Pendant zur NP-Vollständigkeit definiert auch Valiant einen Vollständigkeitsbegriff. Statt der klassischen Reduktion bezieht sich diese Vollständigkeit auf sogenannte Projektionen zwischen Polynomen. Die Definitionen richten sich nach Bürgisser [Bür13].

Definition 3.16. Ein Polynom f ist eine Projektion von einem Polynom g, in Zeichen $f \leq g$ gdw. $f(X_1, \ldots, X_n) = g(a_1, \ldots, a_m)$ mit $a_i \in k \cup \{X_1, \ldots, X_n\}$ gilt. D. h. f kann aus g durch Vertauschen von Veränderlichen bzw. Ersetzen von Unbestimmten durch Konstanten erzeugt werden.

Diese Projektionen lassen sich auf ganze Familien von Polynomen erweitern. Ähnlich wie auch bei der klassischen $\leq_m^{\mathsf{P}}$-Reduktion wird eine polynomielle Beschränktheit gefordert.

Definition 3.17. Eine p-Familie $f = (f_n)_{n \geq 1}$ ist eine p-Projektion von einer p-Familie $g = (g_m)_{m \geq 1}$, $f \leq_p g$ gdw. eine von oben und unten p-beschränkte Funktion $t \colon \mathbb{N} \to \mathbb{N}$ existiert, sodass

$$\exists n_0 \forall n \geq n_0 : f_n \leq g_{t(n)}$$

gilt.

Definition 3.18. Eine p-Familie g ist VNP-vollständig gdw. g p-definierbar ist und $f \leq_p g$ für alle $f \in \mathsf{VNP}$ gilt.

Die Vollständigkeit lässt sich auch im algebraischen Fall über die Zugehörigkeit zur Klasse sowie die generische Projektion aller Elemente der Klasse zeigen.

Bemerkung 3.19. [BCS13]

(1) Die p-Projektion ist transitiv.

(2) Die Klassen VP und VNP sind unter p-Projektion abgeschlossen.

Eines der wichtigsten Resultate in diesem Zusammenhang ist die erste p-Familie, deren VNP-Vollständigkeit gezeigt werden konnte.

Satz 3.20 (Valiant). *Die p-Familie der Permanente PER ist VNP-vollständig über Körpern mit Charakteristik ungleich zwei [Val79a]. Dies gilt insbesondere für $\mathbb{R}$ und $\mathbb{C}$.*

Ein vollständiger Beweis des Satzes findet sich in Bürgisser [Bür04]. Mithilfe dieses Resultats kann die VNP-Vollständigkeit vieler anderer Familien gezeigt werden.

3.3 VNP-vollständige Probleme

Dieser Abschnitt gibt eine Übersicht über eine Auswahl VNP-vollständiger Probleme nach Bürgisser [Bür13]. Speziell wird hier auf graphbasierte Probleme eingegangen. Im Folgenden sei $G = (V, E)$ ein gewichteter Graph und ε eine Grapheigenschaft. Betrachtet wird hier die Komplexität der sogenannten erzeugenden Funktion (GF). Diese Funktion ist polynomiell in den Kantengewichten und gibt für ungewichtete Graphen die erwartete Anzahl spannender Subgraphen von G mit der Eigenschaft ε aus. In gewichteten Graphen entspricht sie dem Gewicht aller dieser Subgraphen.

Definition 3.21. Die erzeugende Funktion eines Graphen $G = (V, E)$ für eine Grapheigenschaft ε ist definiert als

$$\mathrm{GF}(G, \varepsilon) \coloneqq \sum_{\substack{E' \subseteq E \\ (V,E') \in \varepsilon}} \prod_{e \in E'} w(e).$$

Dabei ist $w : E \to I \coloneqq k \cup X$ die Gewichtsfunktion, mit einem Körper k und Variablen X über k.

Es genügt die nachfolgenden Überlegungen für K_n, den vollständigen Graphen mit n Knoten, bzw. $K_{n,n}$, den vollständigen bipartiten Graphen mit zwei Partitionen aus je n Knoten, aufzustellen. Dies ist möglich, da die erzeugende Funktion $\mathrm{GF}(G, \varepsilon)$ für jeden gewichteten Graphen G mit n Knoten eine Projektion von $\mathrm{GF}(K_n, \varepsilon)$ ist. Gleiches gilt für $\mathrm{GF}(K_{n,n}, \varepsilon)$ in bipartiten Graphen.

Die Zugehörigkeit zu VNP kann sowohl direkt über die p-Definierbarkeit als auch über eine aus dem Valiantkriterium (s. Beh. 3.14) folgende Behauptung gezeigt werden.

Behauptung 3.22. *Angenommen, das Problem zu Prüfen ob ein gegebener Graph die Grapheigenschaft ε besitzt, liegt in* P/poly*, so ist die Folge erzeugender Funktionen $GF(K_n, \varepsilon)$ über jedem Körper p-definierbar und damit in* VNP*.*

Die Koeffizienten des multivariaten Polynoms der erzeugenden Funktion werden über diejenigen Teilgraphen bestimmt, welche die geforderte Grapheigenschaft ε aufweisen. Kann dies in P/poly geprüft werden, so folgt, dass die Koeffizienten in #P/poly berechenbar sind. Das Valiantkriterium ist somit anwendbar und zeigt die Zugehörigkeit zu VNP.

3.3.1 Matchings

Wie bereits bekannt, entspricht die Anzahl perfekter Matchings in bipartiten Graphen der Permanente PER_n. Dies lässt sich nun auch auf die erzeugende Funktion übertragen, so gilt $\mathrm{GF}(K_{n,n},\ \mathcal{DI}) = \mathsf{PER}_n$. Die Grapheigenschaft $\mathcal{DI}$ bedeutet, dass alle verbundenen Komponenten exakt zwei Knoten besitzen. In vollständigen bipartiten Graphen entspricht dies der Definition eines perfekten Matchings. Die erzeugende Funktion gibt somit für ungewichtete Graphen die Anzahl perfekter Matchings aus. Da PER_n nach Satz 3.20 VNP-vollständig ist, gilt dies folglich auch für $\mathrm{GF}(K_{n,n},\ \mathcal{DI})$.

Um die VNP-Vollständigkeit weiterer Probleme zu zeigen, eignet sich eine Verallgemeinerung der Permanente besonders gut.

Definition 3.23. Die partielle Permante ist definiert als

$$\mathsf{PER}^{\star}_n \coloneqq \mathrm{GF}(K_{n,n},\ \mathcal{MD}) = \sum_{\pi} \prod_{i \in \operatorname{def} \pi} X_{i\pi(i)}$$

mit der Grapheigenschaft $\mathcal{MD}$, alle verbundenen Komponenten besitzen maximal zwei Knoten sowie π als injektive partielle Abbildung $\underline{n} \to \underline{n}$ ($\underline{n} = \{1, \ldots, n\}$). Das leere Produkt ist nach Definition eins. Die zugehörige p-Familie ist $\mathsf{PER}^{\star}$.

Dieses Problem entspricht dem der Anzahl partieller Matchings eines bipartiten Graphen, sprich Matchings, die nicht alle Knoten überdecken und bei denen nicht gematchte Knoten existieren. Ein solches Matching ist in Abb. 3.2 dargestellt.

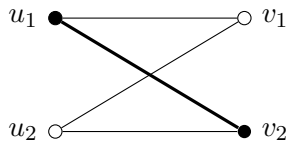


Abbildung 3.2: Partielles Matching in $K_{2,2}$

Dass die Matchings nicht perfekt, sondern nur partiell sind, wird insbesondere über die injektiven partiellen Permutationen π erreicht. Dies wird am Beispiel 3.24 deutlich.

Beispiel 3.24. Die Anzahl parteiller Matchings im $K_{2,2}$ wird wie folgt berechnet:

$$\begin{aligned}
\mathrm{GF}(K_{2,2},\ \mathcal{MD}) &= \textstyle\sum_{\pi} \prod_{i \in \{1,\ldots,n\}} X_{i\pi(i)} \\
&= X_{11}X_{22} + X_{12}X_{21} + X_{11} + X_{22} + X_{12} + X_{21} \\
&= 1 * 1 + 1 * 1 + 1 + 1 + 1 + 1 \\
&= 6.
\end{aligned}$$

Dabei werden auch diejenigen Matchings gezählt, welche, wie z. B. X_{11} nur aus einer Kante bestehen, da durch die partielle Permutation der Fall $i = 2$ nicht betrachtet wird.

Satz 3.25 (Jerrum). *Die p-Familie partieller Permanenten* $\mathsf{PER}^\star$ *ist* VNP*-vollständig.*

Der Beweis wird über die Projektion vom bereits als VNP-vollständig bekannten Problem PER geführt.

Beweis. Die Zugehörigkeit zu VNP kann alternativ zur Behauptung 3.22 auch auf analoge Weise wie für das bereits bekannte Problem PER aus Def. 3.11 gezeigt werden, jedoch muss in diesem Fall keine perfekte Permutationsmatrix gefordert werden. Der Beweis der VNP-Vollständigkeit wird in drei Schritten geführt, welche in Abb. 3.3 am Bsp. des $K_{2,2}$ visualisiert sind.
Schritt 1: Seien $\{u_1, \ldots, u_n\}$ und $\{v_1, \ldots, v_n\}$ die Knotenpartitionen in $K_{n,n}$ sowie $X_{i,j}$ das Gewicht einer Kante $\{u_i, v_i\}$. Angenommen, π wäre ein partielles Matching von $K_{n,n}$ mit freien Knoten u_i, v_j, wobei $i \in I(\pi)$ und $j \in J(\pi)$. Dabei bezeichnen $I(\pi)$ und $J(\pi)$ die Mengen freier Knoten des Matchings π.
Schritt 2: Zusätzlich werden für $i \in \underline{n}$ Knoten u'_i, v'_i sowie die Kanten $\{u'_i, v_i\}$ und $\{u_i, v'_i\}$, jeweils mit Gewicht -1 hinzugefügt. Notiere diesen gewichteten bipartiten Graphen mit G. Die Behauptung ist, dass die Anzahl der perfekten Matchings in vollständigen bipartiten Graphen der Anzahl partieller Matchings im modifizierten Graphen G entspricht, formal $\mathrm{GF}(K_{n,n}, \mathcal{DI}) = \mathrm{GF}(G, \mathcal{MD})$. Daraus folgt, dass $\mathsf{PER}_n \leq \mathsf{PER}^\star_{2n}$. Die partielle Permanente lässt sich mit Hilfe der Permanente berechnen. Nach Definition 3.17 der p-Projektion gilt demnach auch $\mathsf{PER} \leq_p \mathsf{PER}^\star$.
Schritt 3: Für $i \in M$ mit $M \subseteq I(\pi)$ sowie $j \in N$ mit $N \subseteq J(\pi)$ werden nun Kanten $\{u_i, v'_i\}$ bzw. $\{u'_j, v_j\}$ zum partiellen Matching π hinzugefügt. Das neue Matching π' hat damit ein Gewicht von $(-1)^{|M|+|N|} w(\pi)$. Die Anzahl der partiellen Matchings in G lässt sich also über Wahlmöglichkeiten von π sowie M und N definieren.

$$\mathrm{GF}(G, \mathcal{MD}) = \sum_{\pi} w(\pi) \sum_{M \subseteq I(\pi)} (-1)^{|M|} \sum_{N \subseteq J(\pi)} (-1)^{|N|}$$

Existieren in G partielle Matchings, so wird über das Hinzufügen der Kanten mit negativem Gewicht die Summe $\mathrm{GF}(G, \mathcal{MD})$ verringert. Ist eine der Mengen $I(\pi)$ oder $J(\pi)$ nicht leer, dann ist die Summe über die Teilmengen Null. Dies wird in Bsp. 3.26 anhand des $K_{2,2}$ verdeutlicht. Somit tragen nur perfekte Matchings in $K_{n,n}$, für welche nach Definition die Mengen $I(\pi)$ und $J(\pi)$ leer sind, zur Summe bei und die Behauptung gilt. Über die Transitivität der p-Projektion folgt, dass $\mathsf{PER}^\star$ VNP-vollständig ist. □

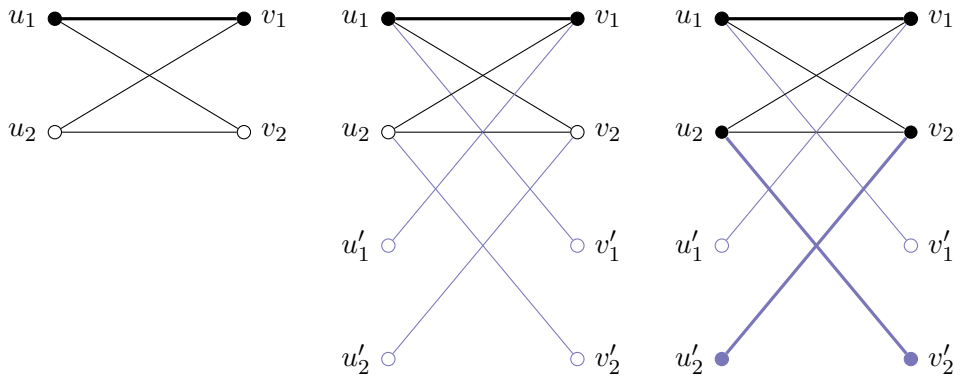


Abbildung 3.3: Partielles Matching in G

Beispiel 3.26. Beispielhaft wird das Matching π aus Abb. 3.3 betrachtet. Die Mengen freier Knoten sind $I(\pi) = \{u_2\}$ und $J(\pi) = \{v_2\}$. Als Teilmengen können also $M = \{u_2\}$ bzw. $N = \{v_2\}$ oder jeweils die leere Menge gewählt werden. Die Summe berechnet sich demnach zu

$$\sum_{M \subseteq I(\pi)} (-1)^{|M|} = (-1)^1 + (-1)^0 = 0.$$

Analoges gilt auch für N. War π also ein partielles Matching, so ist

$$w(\pi) \sum_{M \subseteq I(\pi)} (-1)^{|M|} \sum_{N \subseteq J(\pi)} (-1)^{|N|} = 0$$

und trägt nicht zur Gesamtsumme über π bei. Handelt es sich bei π hingegen um ein perfektes Matching, so sind die Mengen $I(\pi)$ und $J(\pi)$ leer, M und N können folglich auch nur leere Mengen sein. Es gilt für die Summe

$$\sum_{M \subseteq I(\pi)} (-1)^{|M|} = (-1)^0 = 1.$$

Analog für N, letztendlich also

$$w(\pi) \sum_{M \subseteq I(\pi)} (-1)^{|M|} \sum_{N \subseteq J(\pi)} (-1)^{|N|} = w(\pi).$$

3.3.2 Hamilton'sche Kreise

Ein weiteres VNP-vollständiges Problem ist das der Polynome Hamilton'scher Kreise. Ein Hamilton'scher Kreis ist dabei ein einfacher Pfad eines Graphen der jeden Knoten genau einmal beinhaltet und bei dem Start- und Endknoten identisch sind. Die Zugehörigkeit zu VNP ist über Behauptung 3.22 gegeben.

Satz 3.27. *Es existiert eine p-Folge planarer, kubischer, gewichteter Graphen G_n, sodass $(GF(G_n, \{cycles\}))_{n\geq 1}$ VNP-vollständig ist.*

Die erzeugende Funktion $\mathrm{GF}(G, \{cycles\})$ entspricht, aufgrund ihrer Eigenschaft spannende Subgraphen zu zählen, dem Gewicht aller Hamilton'scher Kreise in einem ungerichteten Graphen G.

Der Beweis der VNP-Vollständigkeit dieses Polynoms basiert auf dem Beweis der NP-Vollständigkeit des zugehörigen Entscheidungsproblems von Garey, Johnson und Tarjan [GJT76]. Über die Reduktion von 3-SAT konnte gezeigt werden, dass das Hamilton'sche Kreisproblem auch bei Einschränkung auf planare kubische Graphen NP-vollständig ist. Im algebraischen Fall kann der Beweis dadurch vereinfacht werden, dass nun gewissermaßen vom Zählproblem für 2-SAT reduziert wird, welches Ähnlichkeiten zur partiellen Permanente aufweist. Das Entscheidungsproblem 2-SAT liegt im Gegensatz zu 3-SAT in P, das Zählproblem #2-SAT ist #P-vollständig [Val79c]. Insbesondere bedeutet dies, zu zeigen, dass $(\mathrm{GF}(G_n, \{cycles\})_{n\geq 1}$ eine p-Projektion der partiellen Permanente PER* ist.

Als Grundlage werden zwei Konstruktionen aus dem Beweis von Garey, Johnson und Tarjan [GJT76] benötigt. Die Exklusiv-Oder Kopplung (XOR-Kopplung) zweier Kanten erzwingt, dass genau eine der gekoppelten Kanten $e = \{u, v\}$ und $e' = \{u', v'\}$ auf einem Hamilton'schen Pfad gewählt werden muss. In Abb. 3.4 ist dieses Verhalten verdeutlicht. Symbolisch wird eine XOR-Kopplung, wie in Abb. 3.5 zu sehen, über eine Kante mit einem Plus dargestellt.

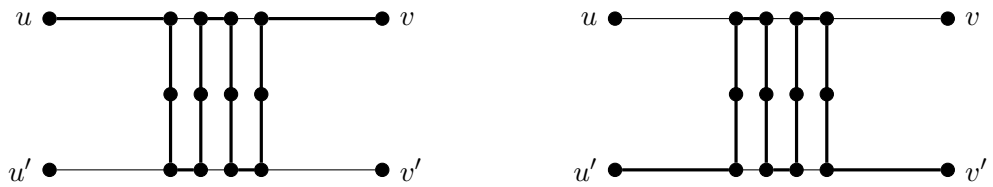


Abbildung 3.4: Hamilton'sche Pfade über XOR-Kopplung

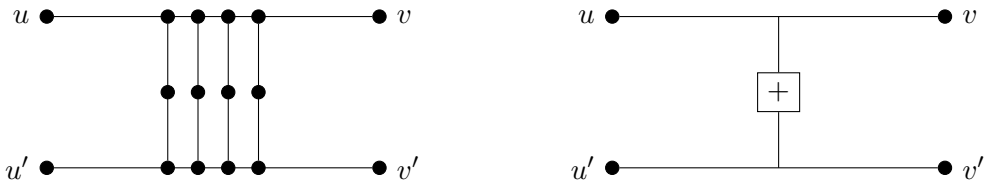


Abbildung 3.5: XOR-Kopplung und Symbolbild [Bür13]

Es ist anzumerken, dass sich kreuzende XOR-Kopplungen die Planarität des Graphen nicht zerstören. Kreuzungen können durch eine Umwandlung des Graphen beseitigt werden. Dieses Vorgehen ist in Abb. 3.6 visualisiert. Die vertikale XOR-Kopplung wird durch ihre vier Kanten dargellt, denen jeweils eine Doppelkante hinzugefügt wird, um darüber die horizontale XOR-Kopplung nachzubilden.

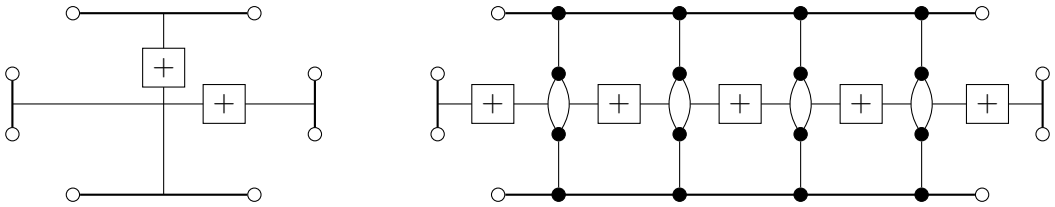


Abbildung 3.6: Umwandlung kreuzender XOR-Kopplungen in planaren Graphen [Bür13]

Die zweite benötigte Konstruktion ist ein Hilfgraph C, siehe Abb. 3.7, welcher bestimmte Eigenschaften bezüglich seiner Hamilton'schen Pfade besitzt. Alle Kanten, mit Ausnahme der mit $\frac{1}{2}$ gekennzeichneten, besitzen Gewicht 1. Es gibt in C genau zwei Hamilton'sche Pfade ohne die Kanten e und f. Diese verwenden jeweils die Kante mit Gewicht $\frac{1}{2}$ und haben daher auch ein Gesamtgewicht von 1. Es existiert je ein Pfad mit Kante e und ohne Kante f bzw. mit f und ohne e von Gewicht 1. Ein Pfad mit sowohl e als auch f ist nicht möglich. Die Summe aller Hamilton'schen Pfade in C ist somit 0, wenn e und f genutzt werden und 1 sonst.

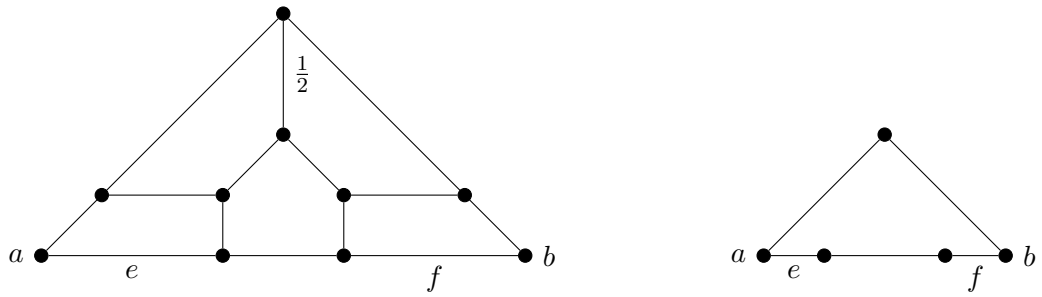


Abbildung 3.7: Hilfsgraph C und Symbolbild [Bür13]

Beweis. Ausgehend vom $K_{n,n}$ wird ein Graph G_n konstruiert. Für jede Kante $e = \{i, j\}$ des $K_{n,n}$ wird ein Graph V_e mit zwei Kanten zu G_n hinzugefügt, s. Abb. 3.8. Diese Kanten werden im Folgenden mit oberer, bzw. unterer Kante bezeichnet. Die obere Kante erhält ein Gewicht von 1, die untere Kante behält das Gewicht X_{ij} der ursprünglichen Kante des $K_{n,n}$.

Abbildung 3.8: Erzeugung der Graphen V_e [Bür13]

Für jedes Kantenpaar $\{e, f\}$, welches sich einen Knoten teilt, wird außerdem ein Hilfsgraph $C_{\{e,f\}}$, s. Abb. 3.7, hinzugefügt, wobei $\{e, f\}$ mit den gekennzeichneten Kanten des Hilfgraphen korrespondieren. Es werden $2n\binom{n}{2}$ solcher Graphen hinzugefügt. Alle $C_{\{e,f\}}$ und V_e werden an ihren Verbindungsknoten zu einem Kreis verbunden, siehe Abb. 3.9.

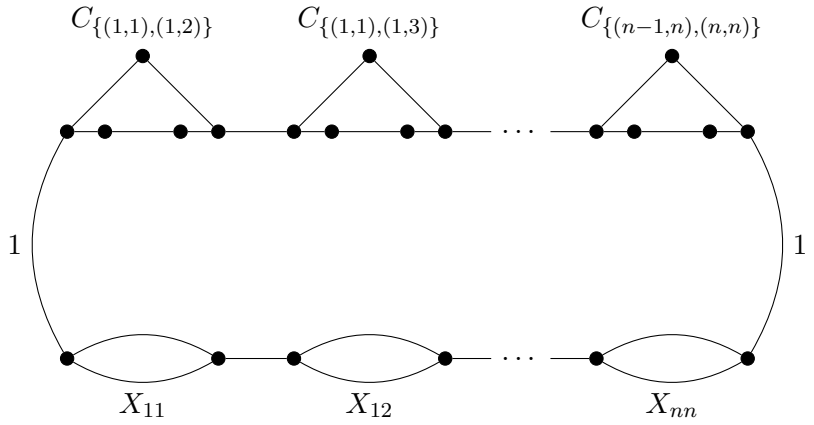


Abbildung 3.9: Kreisförmige Anordnung der V_e und Hilfgraphen C

Zuletzt werden für alle Hilfsgraphen XOR-Kopplungen zwischen der Kante e und der oberen Kante von V_e sowie f und der oberen Kante von V_f hinzugefügt. Kreuzende XOR-Kopplungen werden durch die Struktur in Abb. 3.6 ersetzt, was die Planarität erhält. Der endgültige Graph G_n ist in Abb. 3.10 dargestellt. Der Graph ist weiterhin kubisch und die Anzahl der Knoten ist in n p-beschränkt.

Es ist nun zu zeigen, dass das Gewicht aller Hamilton'schen Kreise in G_n dem der partiellen Matchings in $K_{n,n}$ entspricht, d. h. $\mathsf{PER}* = \mathrm{GF}(G_n, \{cycles\})$.

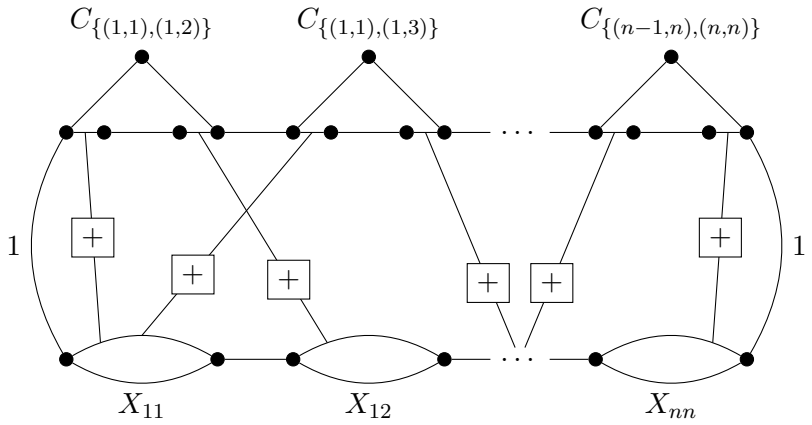


Abbildung 3.10: Graph G_n [Bür13]

Angenommen, σ sei ein Hamilton'scher Kreis in G_n sowie $\pi = \pi(\sigma)$ diejenigen Kanten im $K_{n,n}$, von deren Gegenstück in G_n σ die untere Kante wählt. In diesem Fall handelt es sich bei π um ein partielles Matching.

Der Beweis der Behauptung wird über Widerspruch geführt. Angenommen, π wäre kein partielles Matching und es existierten Kanten $e, f \in \pi$ in $K_{n,n}$ welche einen gemeinsamen Knoten hätten. Dann müssten nach Definition von π beide im Hamilton'schen Kreis σ von G_n gewählt werden, was wiederum bedeutet, dass die oberen Kanten der Graphen V_e und V_f nicht zu σ gehören können. Nach Konstruktion von G_n müsste die XOR-Kopplung zwischen den oberen Kanten und den Hilfsgraphen C sowohl e als auch f wählen, siehe Abb. 3.11. Dies ist jedoch nicht möglich und somit wäre σ kein Hamilton'scher Kreis.

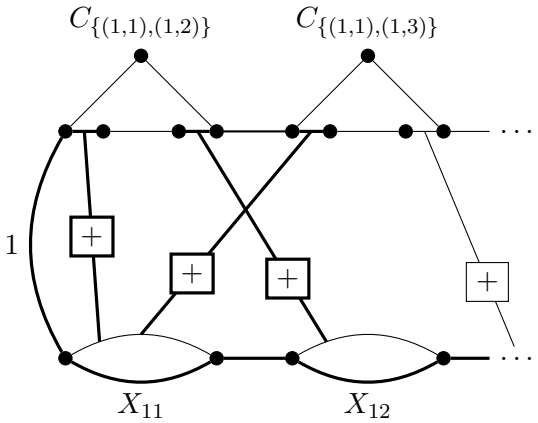


Abbildung 3.11: Fall π kein partielles Matching

Andererseits, angenommen, π sei ein partielles Matching in $K_{n,n}$ und σ mit $\pi(\sigma) = \pi$ der zugehörige Hamilton'sche Kreis. Durch die XOR-Kopplungen wird festgelegt, welche Kanten von e und f in den Hilfsgraphen gewählt werden. Dies können entweder eine oder keine, jedoch nie beide sein. Die Summe der Gewichte alle Hamilton'schen Kreise entspricht somit $\prod_{e \in \pi} X_e$. Insgesamt lässt sich zeigen, dass

$$\mathrm{GF}(G_n, \{cycles\}) = \sum_{\sigma} w(\sigma) = \sum_{\pi} \sum_{\pi(\sigma)=\pi} w(\sigma) = \sum_{\pi} \prod_{e \in \pi} X_e = \mathsf{PER}^*$$

gilt. Damit ist $\mathrm{GF}(G_n, \{cycles\})$ eine p-Projektion von PER^*. Aufgrund der VNP-Schwere von PER^* 3.25 und der Transitivität der p-Projektion 3.19 ist auch $\mathrm{GF}(G_n, \{cycles\})$ VNP-schwer. □

Korollar 3.28. *Die Folgen der Polynome Hamilton'scher Kreise in ungerichteten und gerichteten Graphen,* $(\mathsf{UHC}_n)_{n \geq 1}$ *sowie* $(\mathsf{HC}_n)_{n \geq 1}$ *sind* VNP*-vollständig.*

Beweis. Der ungerichtete Fall folgt direkt aus Satz 3.27. Im gerichteten Fall wird jede Kante $\{i, j\}$ des zuvor ungerichteten Graphen durch gerichtete Kanten (i, j) und (j, i) ersetzt, sodass ein Hamilton'scher Kreis ungeachtet der Richtung möglich ist. □

Durch die Reduktion von #2-SAT vereinfacht sich der Beweis im Gegensatz zum klassischen Fall. Da dort die Reduktion vom NP-schweren Problem 3-SAT, dessen Formeln drei Literale pro Klausel enthalten, geführt werden muss, wird eine Graphkonstruktion benötigt, welche ein ODER mit drei Eingaben simuliert [GJT76]. Der Aufbau der Formeln von #2-SAT mit zwei Literalen pro Klausel erlaubt es hier, den direkten Bezug zum Problem der partiellen Matchings im bipartiten Graphen herzustellen.

3.4 Valianthypothese

Nachdem nun die Klassen VP und VNP definiert wurden sowie ein Vollständigkeitsbegriff bzgl. der p-Projektion eingeführt wurde, bleibt zu klären, in welchem Verhältnis die beiden Klassen zueinander stehen. Valiant [Val79a] stellte im Zuge der erstmaligen Definition der Klassen auch die Vermutung auf, dass diese ungleich sind. Strassen [Str86] prägte für diese Vermutung 1986 den Begriff der Valianthypothese.

Valianthypothese. $\mathsf{VP} \neq \mathsf{VNP}$

Da VP und VNP als algebraische Versionen der klassischen Klassen P und NP verstanden werden können, von welchen die Gleichheit unbekannt ist, könnte eine Lösung im algebraischen Fall Aufschlüsse über dieses klassische Problem geben. Daher ist es interessant diese zu untersuchen. Der Zusammenhang zwischen der Valianthypothese und der Cookhypothese $\mathsf{P} \neq \mathsf{NP}$ wird in Abschnitt 4.4.5 diskutiert.

VP ist die Klasse der p-berechenbaren Funktionen, VNP die der p-definierbaren, die Valianthypothese ist also genau dann wahr, wenn es VNP-vollständige Familien gibt, welche nicht p-berechenbar sind. Da die Permanente als bekanntestes Problem VNP-vollständig ist, lässt sich dies auch auf die Frage übertragen, ob die Permanente p-berechenbar ist [Gat88]. Eine Antwort darauf würde insbesondere zeigen, ob zu ihrer Berechnung schnellere Algorithmen als die bisher bekannten existieren können.

3.4.1 VP-Vollständigkeit

Zur Differenzierung von VP und VNP bietet es sich an, die jeweils vollständigen Probleme der Klassen zu vergleichen. Während für VNP viele solcher Probleme bekannt sind, wie das Polynom für Hamilton'sche Kreise welches in Abschnitt 3.3 vorgestellt wurde, ist dies für VP nicht der Fall. Die VP-Vollständigkeit ist analog zur VNP-Vollständigkeit definiert.

Definition 3.29. Eine p-Familie g ist VP-vollständig gdw. g p-berechenbar ist und $f \leq_p g$ für alle $f \in \mathsf{VP}$ gilt [Bür13].

Dieser Umstand ist Grund zur Überlegung, ob die Klasse VP tatsächlich die geeignetste ist, um effizient lösbare Polynome zu beschreiben. Für eine Teilklasse von VP hingegegen konnten Malod und Portier [MP08] die Vollständigkeit vieler Probleme der linearen Algebra zeigen. Dazu gehören beispielsweise die Potenz sowie die Spur, also die Summe aller Diagonalelemente, einer Matrix. Zu VP zugehörige Probleme lassen sich finden, wie die in Abschnitt 3.1.1 genannte Summe bzw. das Produkt in n Unbestimmten, die VP-Schwere dieser Probleme ist jedoch unbekannt. Auch fehlte lange eine natürliche Definition der Klasse VP, welche nicht auf einer Form arithmetischer Schaltkreise oder äquivalenten Berechnungsmodellen wie den Straight-Line-Programmen basiert. Mengel [Men11] konnte schließlich nachweisen, dass sich VP über Constraint Satisfaction Probleme (CSPs) eingeschränkter Struktur charakterisieren lässt. Bei CSPs handelt es sich um Erfüllbarkeitsprobleme, an welche zusätzliche Bedingungen in Form sogenannter „Constraints“ gestellt werden. Diese sind von der Form $R(x_1, \ldots, x_n)$, wobei R eine n-stellige logische Relation der Variablen $x_1, \ldots, x_n$ ist. Eine Constraint Sprache Γ ist eine endliche Menge dieser Art Relationen, eine Γ-Formel eine Konjunktion von Relationen aus

Γ. Eine Γ-Formel ist genau dann durch eine Belegung θ erfüllt, wenn alle Constraints gleichzeitig durch θ erfüllt werden [CV08]. Der Dichotomiesatz von Schäfer besagt, dass CSP NP-vollständig ist, solange die Constraints nicht aus einer Menge Constraints bestimmter Eigenschaften stammen. Ansonsten liegt CSP in P [Sch78]. Im algebraischen Fall sind die aus CSPs folgenden Probleme im allgemeinen VNP-vollständig. Wird jedoch die Struktur der Constraints eingeschränkt und das Problem im mächtigeren, nicht Boole'schen Kontext, betrachtet, so charakterisiert es VP [Men11].

Ein Beispiel für ein VP-vollständiges Problem zeigten Durand et al. [Dur+14]. Ein Homomorphismus zweier Graphen $G = (V_G, E_G)$ und $H = (V_H, E_H)$ ist eine Abbildung der Knotenmenge V_G nach V_H, welche die Struktur der Kanten erhält. Zur Darstellung als Polynom werden die Knoten von H mit der Variablenmenge X und die Kanten mit der Variablenmenge Y bezeichnet. Dabei gilt

$$X = \{X_u \mid u \in V_H\} \qquad \text{und} \qquad Y = \{Y_{uv} \mid (u, v) \in E_H\}.$$

Für jeden Homomorphismus $\phi\colon G \to H$ kann das zugehörige Monom folgenderweise aufgestellt werden. Die Funktion $\alpha\colon V_G \to \mathbb{N}$ markiert dabei die Knoten.

$$\mathrm{mon}(\phi) \triangleq \left(\prod_{u \in V_G} X_{\phi(u)}^{\alpha(u)}\right)\left(\prod_{(u,v)\in E_G} Y_{\phi(u),\phi(v)}\right)$$

Das vollständige Polynom f ergibt sich aus der Summe der Monome aller möglichen Homomorphismen von G zu H. Diese Menge wird mit $\mathcal{H}$ bezeichnet.

$$f_{G,H,\alpha,\mathcal{H}}(X, Y) = \sum_{\phi \in \mathcal{H}} mon(\phi)$$

Für Graphen bestimmter Form ist dieses Polynom f unter p-Projektion VP-vollständig. Der Beweis basiert auf der Isomorphie zwischen den gewählten Graphen. Mahajan und Saurabh [MS18] konnten dieses Ergebnis auf Homomorphismen zwischen beschränkten Graphen G und vollständigen Graphen H verallgemeinern.

3.4.2 Erweiterte Valianthypothese

Trotz der Fortschritte in der Charakterisierung von VP ist es weiterhin unbekannt, ob die Determinante VP-vollständig ist. Ein Resultat, das insbesondere im Hinblick auf ihre Ähnlichkeit zur Permanente von Interesse ist. Jedoch kann gezeigt werden, dass sie unter strengeren Anforderungen an die p-Projektion in der resultierenden Klasse vollständig ist. Dieser Abschnitt basiert auf Bürgisser [Bür13].

Definition 3.30. Eine Funktion $t\colon \mathbb{N} \to \mathbb{N}$ heißt quasi-polynomiell bzw. *qp*-beschränkt, gdw. es ein $c > 0$ gibt, sodass $t(n) \leq n^{\mathcal{O}((\log n)^c)}$.

Im Gegensatz zur p-Beschränktheit aus Definition 3.2 wird hier nicht nur ein Polynom als Schranke gefordert. Äquivalent zur p-Berechenbarkeit folgt hieraus auch die *qp*-Berechenbarkeit.

Definition 3.31. Eine p-Familie $f = (f_n)_{n\in\mathbb{N}}$ ist *qp*-berechenbar, gdw. die Komplexität $L(f_n)$ in n *qp*-beschränkt ist.

Über dieser Art Familien lässt sich ebenfalls eine Klasse definieren, über welche ein weiterer Ansatz zur Lösung der Valianthypothese geschaffen wird.

Definition 3.32. Die Komplexitätsklasse VQP besteht aus allen *qp*-berechenbaren Familien über k.

Für das in VP liegende Problem der Determinante kann unter einem anderen Projektionsbegriff Vollständigkeit in VQP gezeigt werden, etwas, was in VP bisher eine offene Frage ist.

Definition 3.33. Eine p-Familie $f = (f_n)_{n\geq 1}$ ist eine *qp*-Projektion von einer p-Familie $g = (g_m)_{m\geq 1}$, d. h. $f \leq_{qp} g$ gdw. eine *qp*-beschränkte Funktion $t\colon \mathbb{N} \to \mathbb{N}$ existiert, sodass

$$\exists n_0 \forall n \geq n_0 \colon f_n \leq g_{t(n)}$$

gilt.

Satz 3.34. *Die p-Familie der Determinante* $\mathsf{DET} = (\mathsf{DET}_n)_{n\geq 1}$ *ist* VQP*-vollständig unter qp-Projektion.*

Beweis. Siehe Bürgisser [Bür13]. □

Die Klasse VP ist eine echte Teilmenge von VQP. Außerdem gilt, dass VQP keine Teilmenge von VNP ist. Dies ist besonders hervorzuheben, da die Frage, ob die Echtheit der Teilmenge auch für VP und VNP gilt, Aussage der ungelösten Valianthypothese ist. Diese Frage lässt sich mit den Ergebnissen dieses Kapitels zu der Frage erweitern, ob die Klasse VNP in VQP enthalten ist.

Korollar 3.35. *Die folgenden Aussagen sind äquivalent.*

1. $\mathsf{VNP} \not\subseteq \mathsf{VQP}$
2. $\mathsf{PER} \notin \mathsf{VQP}$
3. PER *ist keine qp-Projektion von* DET ($\mathsf{PER} \not\leq_{qp} \mathsf{DET}$)

Die erste dieser Aussagen wird als erweiterte Valianthypothese bezeichnet. Nachfolgend werden die Beziehungen von VQP zu VP und VNP untersucht.

Definition 3.36. VQP^i für $i \in \mathbb{N}$ ist die Klasse derjenigen p-Familien $(f_n)_{n\geq 1}$, für die gilt, dass ihre Komplexität durch $L(f_n) \leq n^{\mathcal{O}\left((\log n)^i\right)}$ beschränkt ist. Desweiteren ist $\mathsf{VQP} = \bigcup_i \mathsf{VQP}^i$.

Außerdem ist VQP^i Teilmenge von VQP^{i+1}, da für alle $i \in \mathbb{N}$ für die Komplexität $L(f_n) \leq n^{\mathcal{O}\left((\log n)^i\right)} \leq n^{\mathcal{O}\left((\log n)^{i+1}\right)}$ gilt.

Satz 3.37. *Die Klasse* VP *ist eine Teilmenge der Klasse* VQP*. Es gilt* $\mathsf{VP} \subseteq \mathsf{VQP}$*.*

Beweis. Nach Definition 3.36 enthält die Klasse VQP^0 diejenigen p-Familien f_n^0, für die $L(f_n^0) \leq n^{\mathcal{O}\left((\log n)^0\right)} = n^{\mathcal{O}(1)}$ gilt. Sie entspricht damit der Klasse VP der p-beschränkten Funktionen. Da außerdem VQP^0 eine Teilmenge von VQP ist, zeigt dies für die Beziehung von VP zu VQP, dass $\mathsf{VP} = \mathsf{VQP}^0 \subseteq \mathsf{VQP}$ ist. □

Satz 3.38. *Die Teilmengenbeziehung in der Hierarchie von* VQP *ist echt. Es gilt für alle* $i \in \mathbb{N}$*, dass* $\mathsf{VQP}^i \subsetneq \mathsf{VQP}^{i+1}$ *ist.*

Beweis. Sei $f^i = \left(f_n^i\right)_{n\geq 1}$ für $i \in \mathbb{N}$ eine Familie von Funktionen mit

$$f_n^i = \sum_{\mu} 2^{2^{j_n(\mu)}} X_1^{\mu_1} \cdots X_{m(n)}^{\mu_{m(n)}},$$

wobei

$$j_n(\mu) \coloneqq \sum_{j=1}^{m(n)} \mu_j n^{j-1} \qquad \text{und} \qquad m(n) \coloneqq m_i(n) \coloneqq \left\lceil (\log n)^i \right\rceil,$$

sowie die Summe über μ aus $\{0, 1, \ldots n-1\}^{m(n)}$ ist. Weiterhin lässt sich durch Induktion über m zeigen, dass für jedes Polynom $f \in k[X_1, \ldots, X_m]$, für welches der Grad jeder Variable in f durch n beschränkt ist, $L(f) \leq 2n^m - 2$ gilt. Für obiges Polynom bedeutet dies, dass $L\left(f^i\right) \leq 2n^{m(n)} - 2 = 2n^{\lceil (\log n)^i \rceil} \in n^{\mathcal{O}\left((\log n)^i\right)}$ ist, d. h. die Familie f^i liegt in VQP^i. Um VQP^i und VQP^{i-1} zu separieren, muss nun nachgewiesen werden, dass f^i nicht in VQP^{i-1} liegt. Die Komplexität $L\left(f^i\right)$ muss also strikt größer als $n^{\mathcal{O}\left((\log n)^{i-1}\right)}$ sein. Zur Lösung wird ein weiteres univariates Polynom g_n herangezogen, dessen untere Schranke für die Komplexität bekannt ist.

$$g_n \coloneqq \sum_j 2^{2^j T^j} \quad 0 \leq j < n^{m(n)} \tag{3.3}$$

Außerdem lässt sich g_n mithilfe von f_n über $g_n = f_n\left(T^{n^0}, T^{n^1}, \ldots T^{n^{m(n)-1}}\right)$ darstellen. Die Komplexität, die Potenz zu berechnen, liegt bei $L(T^n) \leq 2 \log n$. Daraus folgt auch für die Komplexität $L(g_n) \leq L(f_n) + 2m(n) \log n$, der zusätzliche Term stammt aus der Berechnung der $m(n)$ Potenzen der Variablen T. Mit

$$L(g_n) \geq c \sqrt{\frac{n^{m(n)}}{m(n) \log n}} \geq c \sqrt{\frac{n^{(\log n)^i}}{(\log n)^i \log n}} \tag{3.4}$$

folgt für

$$\begin{aligned} L(g_n) \leq L(f_n) + 2m(n) \log n \iff L(f_n) &\geq L(g_n) - 2m(n) \log n \\ &\geq c \sqrt{\frac{n^{(\log n)^i}}{(\log n)^i \log n}} - 2m(n) \log n \\ &\geq n^{\frac{1}{2}(\log n)^i} (\log n)^{-\frac{1}{2}(i+1)} - 2(\log n)^{i+1} \\ &\geq n^{\mathcal{O}(\log n)^i} \\ &> n^{\mathcal{O}(\log n)^{i-1}} \end{aligned}$$

Die anderen Terme sind gegenüber $n^{\mathcal{O}(\log n)^{i-1}}$ vernachlässigbar. Insgesamt bedeutet dies, dass $f^i \in \mathsf{VQP}^i$, aber auch $f^i \notin \mathsf{VQP}^{i-1}$. VQP^i ist also eine echte Teilmenge von VQP^{i+1}. □

Korollar 3.39. *Die Teilmengenbeziehung von* VP *und* VQP *ist echt. Es gilt* $\mathsf{VP} \subsetneq \mathsf{VQP}$.

Beweis. Nach Satz 3.38 existieren auch für $i = 0$ Polynome in VQP, welche in VQP^1, aber nicht in $\mathsf{VQP}^0 = \mathsf{VP}$ liegen, d. h. $\mathsf{VP} \subsetneq \mathsf{VQP}$. □

Im Gegensatz zur Teilmengenbeziehung von VP und VNP, welche direkt aus der Definition von VNP folgte, lässt sich zeigen, dass dies für VQP und VNP nicht der Fall ist. Es existieren p-Familien, welche zwar qp-berechenbar, aber nicht p-definierbar sind.

Satz 3.40. *Sei $(f_n)_{n\geq 1}$ eine p-Familie über $\mathbb{Q}$ sowie $t\colon \mathbb{N} \to \mathbb{N}$ mit $n^{\log n} \leq t(n) \leq 2^n$. Sind alle 2^{2^j} mit $0 \leq j < t(n)$ für alle n Koeffizienten von f_n, so ist $(f_n)_{n\geq 1}$ nicht p-definierbar.*

Beweis. Siehe Bürgisser, Clausen und Shokrollahi [BCS13]. □

Korollar 3.41. *Die Klasse* VQP *ist keine Teilmenge von* VNP.

Beweis. Sei $f^1 = \left(f_n^1\right)_{n\geq 1}$ die p-Familie aus dem Beweis zu Satz 3.38 mit $i = 1$. Nach Definition liegt f^1 in VQP^1 und damit in VQP. Für $t(n) = n^{\lceil \log n \rceil}$ besitzt f^1 die in Satz 3.40 geforderten Eigenschaften, jeder Wert 2^{2^j} für $0 \leq j < t(n) = n^{\lceil \log n \rceil}$ ist nach Definition Koeffizient von f^1. Damit ist f^1 nicht p-definierbar. Es existieren also p-Familien welche in VQP, aber nicht in VNP liegen, VQP ist dementsprechend keine Teilmenge von VNP. □

Ob die umgekehrte Annahme, dass auch VNP keine Teilmenge von VQP ist, gilt, ist Bestandteil der erweiterten Valianthypothese 3.35. Abhängig davon, ob die erweiterte Valianthypothese gilt, ergeben sich somit die Klassendiagramme in Abb. 3.12 bzw. 3.13.

Gilt die erweiterte Valianthypothese, so auch die einfache, da VP eine Teilmenge von VQP ist. Deutlicher wird dies, wenn die Hypothesen als $\mathsf{VNP} \setminus \mathsf{VQP} \neq \emptyset$ und $\mathsf{VNP} \setminus \mathsf{VP} \neq \emptyset$ geschrieben werden. Anderenfalls, wenn $\mathsf{VNP} \setminus \mathsf{VQP} = \emptyset$ ist, d. h. die erweiterte Valianthypothese nicht gilt, lässt sich keine Aussage treffen, ob auch $\mathsf{VNP} \setminus \mathsf{VP} = \emptyset$ ist.

3.4.3 Permanente versus Determinante

Durch die Einführung einer neuen Klasse, in welcher die Determinante vollständig ist, lässt sich das Problem der Valianthypothese auch über ein lange bekanntes Problem der Mathematik definieren. Ist es möglich, die Permanente durch Substitution mithilfe der Determinante auszudrücken? Die Frage wurde zuerst 1913 von Pólya und Szegő [PS98] gestellt und ist seither unbeantwortet. Die Auswirkungen auf die Komplexitätstheorie macht sie für die Theoretische Informatik interessant. Die Permanente wurde bereits in Abschnitt 3.1.3 eingeführt und ist nach Satz 3.20 VNP-vollständig über Körpern mit Charakteristik ungleich zwei. Das Problem des Zusammenhangs zwischen Determinante und Permanente definiert sich wie folgt: Sei A eine $n \times n$ und B eine $m \times m$ Matrix aus

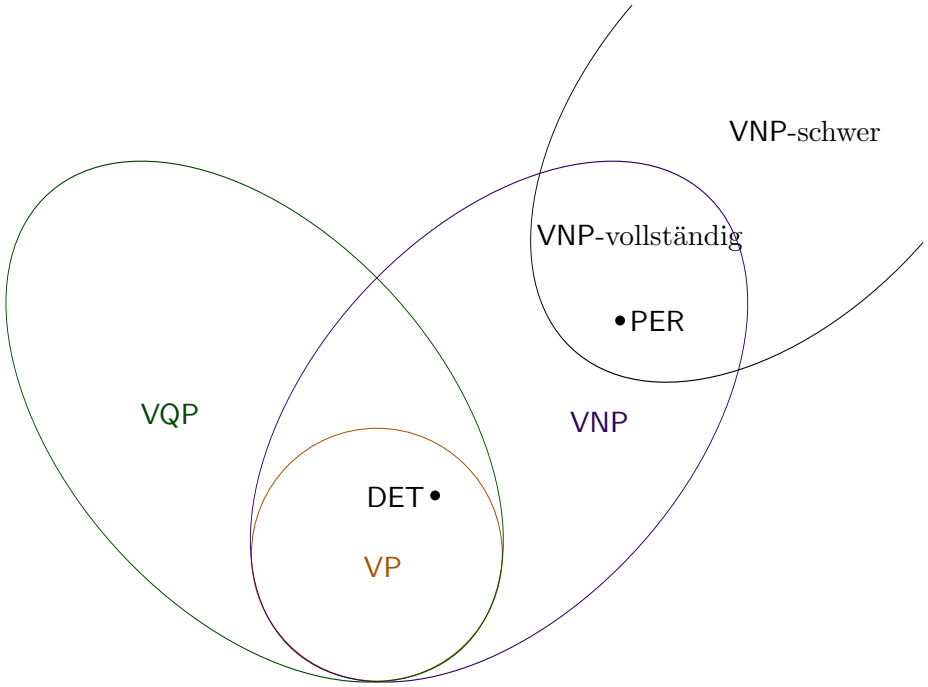


Abbildung 3.12: Klassendiagramm der Algebraischen Komplexitätstheorie (erweiterte Valianthypothese gilt)

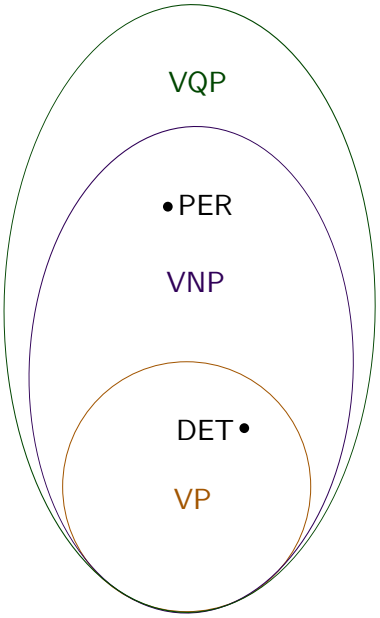


Abbildung 3.13: Klassendiagramm der Algebraischen Komplexitätstheorie (erweiterte Valianthypothese gilt nicht)

Konstanten und Unbestimmten über einem Körper der Charakteristik ungleich zwei. Gesucht ist das kleinste m, sodass die Determinante von B der Permanente von A entspricht, d. h. $\det(B) = \mathrm{per}(A)$. Die Vermutung ist, dass dieses m polynomiell größer als n ist, d. h. $m > poly(n)$ [Agr06].

Andererseits konnte Gathen [Gat87] zeigen, dass PER_n für $m < \sqrt{2n}$ keine Projektion von DET_m ist. Die Bestimmung der Schranken für m ist weiterhin Thema der Forschung. Für Körper der Charakteristik 2 entspricht die Determinante der Permanente. Der Grund dafür ist, dass die Basis des Signums aus der Formel zur Berechnung der Determinante $(-1) \equiv 1 \mod 2$ ist. Somit fällt der einzige Unterschied zur Berechnung der Permanente weg. Jedoch ist in diesem Fall der VNP-Vollständigkeitsbeweis für PER nicht anwendbar, da dieser auf eine Division durch 2 zurückgreift. Es wird angenommen dass die Permanente über solchen Körpern nicht VNP-vollständig ist.

4 Vergleich der klassischen und algebraischen Komplexitätstheorie

Von besonderem Interesse ist die Beziehung zwischen der klassischen und der algebraischen Komplexitätstheorie in Bezug auf das ungelöste Millennium-Problem, ob die Klasse P ungleich der Klasse NP ist. Im Folgenden werden die beiden Modelle nach Cook und Valiant verglichen sowie die Auswirkungen der Valianthypothese (s. 3.4) auf die klassische Komplexitätstheorie beschrieben.

4.1 Modell von Cook

Zuerst wird das klassische Modell von Cook nach Sipser [Sip96] in Erinnerung gerufen. Hierzu werden die beiden Klassen P und NP definiert sowie die Reduktion und der zugehörige Vollständigkeitsbegriff vorgestellt. Das zugrunde liegende Berechnungsmodell ist die Turingmaschine.

Die Komplexitätsklassen $\mathsf{TIME}(t)$ bzw. $\mathsf{NTIME}(t)$, mit $t\colon \mathbb{N} \to \mathbb{N}$, bestehen aus allen Sprachen A, für die es eine (nichtdeterministische) Mehrband-Turingmaschine gibt, welche A entscheidet und in Zeit $\mathcal{O}(t)$ arbeitet.

Definition 4.1. Die Klasse P enthält die Sprachen, die in deterministischer polynomieller Zeit entscheidbar sind. In Zeichen

$$\mathsf{P} = \bigcup_k \mathsf{TIME}(n^k).$$

Für Probleme in P kann ein Algorithmus angegeben werden, welcher das Problem auf einer deterministischen Turingmaschine in Polynomialzeit entscheidet.

Die Klasse NP hingegen wird über die Polynomielle Überprüfbarkeit definiert.

Definition 4.2. Eine Sprache A heißt polynomiell überprüfbar, wenn es einen Verifikationsalgorithmus V mit polynomieller Laufzeit in $|x|$ gibt, sodass für alle Eingaben x

$$x \in A \iff V \text{ akzeptiert bei Eingabe } \langle x, e\rangle \text{ für ein } e$$

gilt.

Definition 4.3. Die Klasse NP enthält die Sprachen die polynomiell überprüfbar sind.

$$\mathsf{NP} = \bigcup_k \mathsf{NTIME}(n^k)$$

Demnach enthält also P diejenigen Probleme, die effizient berechenbar sind, und NP die effizient überprüfbaren Probleme.

Definition 4.4. Seien $A \subseteq \Sigma^*,\ B \subseteq \Delta^*$ Sprachen. A heißt auf B in Polynomialzeit m-reduzierbar $A \leq_m^{\mathsf{P}} B$, gdw. eine in Polynomialzeit berechenbare Funktion $f\colon \Sigma^* \to \Delta^*$ existiert, sodass für alle $x \in \Sigma^*$ $x \in A \iff f(x) \in B$ gilt.

Bemerkung 4.5.

(1) $\leq_m^{\mathsf{P}}$ ist transitiv.

(2) Die Klassen P und NP sind unter $\leq_m^{\mathsf{P}}$ abgeschlossen.

Definition 4.6. Eine Sprache A ist NP-vollständig, gdw. $A \in \mathsf{NP}$ und A NP-schwer ist. D. h. für alle Sprachen $B \in \mathsf{NP}$ gilt $B \leq_m^{\mathsf{P}} A$.

Die Frage, ob die Klassen P und NP identisch sind, ist ein offenes Problem. Es wird von verschiedenen Quellen jedoch angenommen, dass dies nicht der Fall ist [Aar16].

Cookhypothese. $\mathsf{P} \neq \mathsf{NP}$

Abhängig von der Antwort auf diese Frage gibt sich ein unterschiedliches Bild des Verhältnisses der beiden Klassen zueinander, wie in Abb. 4.1 dargestellt.

4.2 Vergleich der Modelle von Cook und Valiant

Die Definitionen der Klassen P (s. Def. 4.1), respektive VP (s. Def. 3.5) unterscheiden sich hauptsächlich durch das zugrunde liegende Berechnungsmodell, bedingt durch die unterschiedliche Art der Eingabe. In beiden Fällen wird jedoch die Existenz eines Polynomialzeitalgorithmus gefordert, welcher das gegebene Problem (Entscheidungsproblem bzw. multivariates Polynom) löst/berechnet.

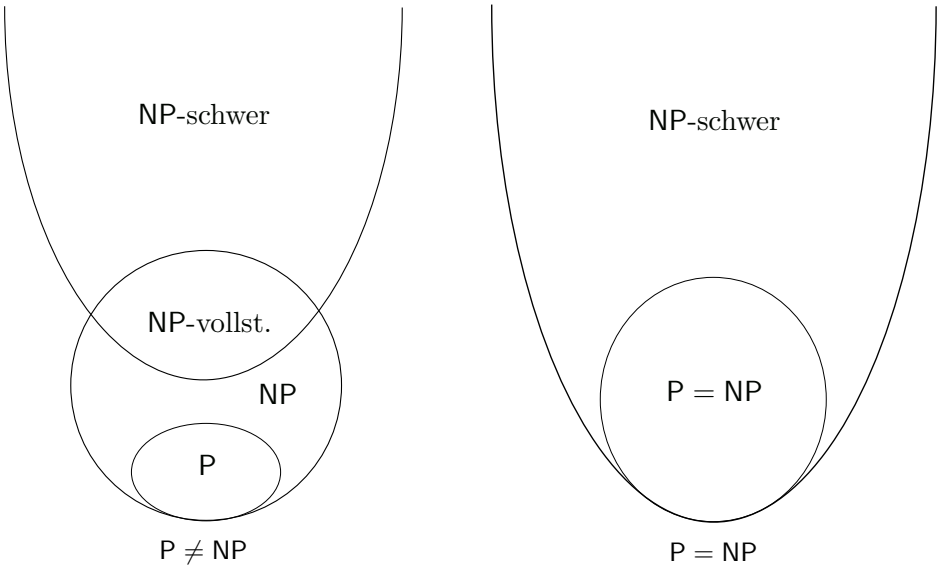


Abbildung 4.1: Verhältnis von P und NP

Für NP und VNP lassen sich die Parallelen jedoch nicht so direkt ziehen. Deutlicher wird der Zusammenhang, wenn die Probleme in NP über ihre charakteristische Funktion definiert werden. So liegt eine Sprache $A \subseteq \Sigma^*$ genau dann in NP, wenn ihre charakteristische Funktion für alle $x \in \Sigma^n$ durch

$$\chi_A(x) = \bigvee_{e \in \Sigma^{t(n)}} \chi_V(x, e) \tag{4.1}$$

berechnet wird. Dabei bezeichnet χ_V die charakteristische Funktion des Verifizierers, welche in Polynomialzeit berechenbar ist sowie e jeweils ein mögliches Zertifikat. Die Funktion $t\colon \mathbb{N} \to \mathbb{N}$ ist p-beschränkt [BCS13]. Damit $A \in$ NP gilt, muss es demnach nur ein solches Zertifikat e geben, welches vom Verifizierer akzeptiert wird. Wird nun die Disjunktion durch eine Summation sowie die charakteristischen Funktionen durch p-Familien ersetzt, so ist das Resultat die p-Definierbarkeit aus Definition 3.8. Die Funktion $t(n)$ entspricht der Differenz der Eingabelänge des Verifizierers und der Eingabelänge der Funktionen der p-definierbaren p-Familie [BCS13].

Aufgrund der Summation über alle möglichen Belegungen von $e_1, \ldots, e_{u(n)}$ ähnelt VNP der Klasse #P (s. Definition 3.13). Die Definition von VNP erinnert an die Anzahl aller möglichen Zertifikate eines Entscheidungsproblems, über welche #P definiert ist. Auch die hier als VNP-vollständig gezeigten Probleme sind ausschließlich Zählprobleme, wie

die Anzahl perfekter bzw. partieller Matchings sowie die erzeugende Funktion für Hamilton'sche Kreise. Dabei hervorzuheben ist das Problem perfekter Matchings, welches in P liegt, dessen zugehörige Zählproblem $\#\mathsf{P}$-vollständig ist. Die p-Familie der äquivalenten Permanente $\mathsf{PER} = (\mathsf{PER}_n)_{n\geq 1}$ hingegen ist VNP-vollständig [BCS13].

Ist jedoch eine effektive Möglichkeit bekannt, die Anzahl von Lösungen eines Entscheidungsproblems zu bestimmen, so lässt sich dies auf das Lösen des Entscheidungsproblems an sich übertragen [BCS13]. Zusammenfassend lässt sich sagen, dass VNP über Boole'schen Berechnungen ungefähr NP entspricht, über arithmetischen hingegen $\#\mathsf{P}$. Grund dafür ist, dass im Boole'schen Kontext über $\{0, 1\}$ nicht gezählt wird, im arithmetischen Fall jedoch schon [MR13]. Darin begründet liegt auch die später in Abschnitt 4.4.5 angesprochene Schwierigkeit, die Bereiche der klassischen und algebraischen Komplexitätstheorie in Verbindung zu setzen. Grundlegend ähneln sich die Gebiete sehr. Mithilfe eines Reduktions- bzw. Projektionsbegriffes werden die schwersten Probleme unter den polynomiell überprüfbaren bzw. den p-definierbaren ermittelt. Die Klassen P und NP sind unter der $\leq_m^{\mathsf{P}}$-Reduktion, die Klassen VP und VNP unter p-Projektion abgeschlossen. Für alle Klassen existieren jeweils vollständige Probleme. Für P sind jedoch deutlich mehr Probleme als vollständig bekannt als für VP, für welche es lange offen war, ob natürliche VP-vollständige Probleme existieren. Ein Problem ist P-vollständig, wenn es in P liegt und sich alle anderen Probleme in P in logarithmischem Platzbedarf darauf reduzieren lassen. Zu den P-vollständigen Problemen gehört unter anderen die Lineare Programmierung, bei welcher eine lineare Zielfunktion gegeben einer Menge linearer (Un-)Gleichungen optimiert wird [GHR91].

4.2.1 Vergleich der Reduktion und Projektion

Wie in Abschnitt 3.3 gezeigt, existieren VNP-vollständige Probleme, welche auf NP-vollständigen Entscheidungsproblemen basieren. Es liegt nahe, nach dem Zusammenhang zwischen der p-Projektion und der $\leq_m^{\mathsf{P}}$-Reduktion zu suchen. Da VNP jedoch eine algebraische Klasse ist, die Ausgaben der zugehörigen Probleme also Funktionswerten und keinen Wahrheitswerten entsprechen, fehlt der allgemeinen Definition der $\leq_m^{\mathsf{P}}$-Reduktion eine wichtige Eigenschaft.

Definition 4.7. Seien $A \subseteq \Sigma^*,\ B \subseteq \Delta*$ Sprachen sowie R eine Reduktion. A heißt auf B anzahlerhaltend reduzierbar, wenn

$$|\{x \mid x \in A\}| = |\{R(x) \mid R(x) \in B\}|$$

gilt [Gol08].

Das bedeutet, dass sich die Anzahl der Lösungen durch die Reduktion nicht verändert. Veranschaulichen lässt sich dies an einem einfachen Beispiel.

Beispiel 4.8.
Betrachtet werden die Entscheidungsprobleme HAMPATH und HAMCIRC auf gerichteten Graphen.

$$\mathsf{HAMPATH} = \{\langle G,\ s,\ t\rangle \mid G = (v, E) \text{ besitzt Hamilton'schen Pfad von } s \text{ nach } t.\}$$
$$\mathsf{HAMCIRC} = \{\langle G\rangle \mid G = (V, E) \text{ besitzt Hamilton'schen Kreis.}\}$$

HAMPATH ist $\leq_m^{\mathsf{P}}$-reduzierbar auf HAMCIRC via einer Reduktionsfunktion f, welche einen neuen Knoten u, gerichtete Kanten $e_1 = (t, u)$ und $e_2 = (u, s)$ zur Kantenmenge E des Graphen G hinzufügt. Der neue Graph wird mit G' bezeichnet, siehe Abb. 4.2. Die Anzahl der Hamilton'schen Kreise in G' entspricht derer der Hamilton'schen Pfade in G, da jeder Pfad durch das Hinzufügen der Kanten zum Kreis wird. Der Knoten u erzwingt, dass diese Kanten auf dem Kreis gewählt werden müssen.

Abbildung 4.2: Beweis HP $\leq_m^{\mathsf{P}}$ HC

Mit Hilfe der anzahlerhaltenden Reduktionen kann die #P-Vollständigkeit des zu einem Entscheidungsproblem aus NP gehörigem Zählproblem gezeigt werden [Gol08].

Die anzahlerhaltenden Reduktionen NP-vollständiger Probleme können in manchen Fällen als Grundlage für p-Projektionen dienen, um die VNP-Vollständigkeit der erzeugenden Funktion des Problems zu zeigen [Bür13]. Ein Beispiel dafür ist der in Abschnitt 3.3.2 aufgeführte Beweis für die erzeugende Funktion Hamilton'scher Kreise.

4.3 Parallele Komplexität

Das Berechnungsmodell der algebraischen Komplexitätstheorie, die Straight-Line-Programme, wurde in Abschnitt 2.2.1 als Hintereinanderausführung mehrerer Instruktionen definiert. Intuitiv stellt sich die Frage, ob eine Parallelisierung dieser Programme generell möglich ist und welche Zeitersparnis sich in der Ausführung des resultierenden

parallelisierten SLPs ergeben würde. In der klassischen Komplexitätstheorie existiert die Klasse NC^i, mit $i \in \mathbb{N}$, welche diejenigen Sprachen enthält, die von einem Boole'schen Schaltkreis in polynomieller Größe und polylogarithmischer Tiefe entschieden werden können [Vol99]. Die Existenz eines solchen Schaltkreises kann als Äquivalenz zur Existenz eines parallelen Maschinenmodell mit $n^{\mathcal{O}(1)}$ Prozessoren verstanden werden, auf welchem ein Problem in Zeit von $(\log n)^{\mathcal{O}(1)}$ lösbar ist. Die Gatter des Schaltkreises entsprechen hierbei indirekt den Prozessoren. Liegt eine Sprache in P, so besitzt sie einen effizienten Algorithmus. Entsprechend dessen liegt eine Sprache genau dann in NC, wenn zu ihrer Berechnung ein effizienter **paralleler** Algorithmus existiert. Es gilt jeweils $\mathsf{NC}^i \subseteq \mathsf{NC}^{i+1}$, die Echtheit dieser Teilmengenbeziehung ist jedoch nur für $\mathsf{NC}^0 \subsetneq \mathsf{NC}^1$ bewiesen. Desweiteren ist unklar, ob die Klassen der effizient lösbaren Sprachen und effizient parallel lösbaren Sprachen übereinstimmen, d. h. ob $\mathsf{P} = \mathsf{NC}$. Es wird allgemein jedoch angenommen, dass dies nicht der Fall ist [AB09]. In der algebraischen Komplexitätstheorie sei nun die zu NC äquivalente Klasse VNC wie folgt definiert.

Definition 4.9. Die Klasse VNC^i enthält alle p-Familien $f = (f_n)_{n\geq 1}$ über einem Körper k, für welche eine Folge von Straight-Line-Programmen (Γ_n) existiert, sodass Γ_n f_n berechnet und Γ_n Größe einer in n p-beschränkten Funktion und Tiefe von $\mathcal{O}\left((\log n)^i\right)$ besitzt.

Aufgrund der Äquivalenz von Straight-Line-Programmen und arithmetischen Schaltkreisen [BC92] lässt sich VNC^i analog zur klassischen Klasse NC^i auch über ebendiese definieren.

Definition 4.10. Die Klasse VNC^i enthält alle p-Familien $f = (f_n)_{n\geq 1}$ über einem Körper k, für welche eine arithmetische Schaltkreisfamilie $\mathcal{C}$ mit

$$\mathsf{SIZE\text{-}DEPTH}\left(n^{\mathcal{O}(1)}, (\log n)^i\right)$$

existiert, welche f berechnet.

Auch in diesem Fall gilt $\mathsf{VNC}^1 \subseteq \mathsf{VNC}^2 \subseteq \cdots \subseteq \mathsf{VP}$. Bemerkenswerterweise lässt sich jedoch zeigen, dass diese Hierarchie kollabiert.

Satz 4.11. *Ist f ein Polynom vom Grad n, für welches ein arithmetischer Schaltkreis von Größe s existiert, so existiert ein arithmetischer Schaltkreis von Größe $(sn)^{\mathcal{O}(1)}$ und Tiefe $\mathcal{O}(\log n \log s)$.*

Beweis. Siehe Valiant et al. [Val+83]. □

Insbesondere bedeutet dies, dass für Polynome mit arithmetischen Schaltkreise von Größe $n^{\mathcal{O}(1)}$ arithmetische Schaltkreise existieren, welche eine Tiefe von

$$\begin{aligned}\mathcal{O}(\log n \log s) &= \mathcal{O}\left(\log n \log\left(n^{\mathcal{O}(1)}\right)\right) \\ &= \mathcal{O}\left(\log n \log n \; \mathcal{O}(1)\right) \\ &= \mathcal{O}\left((\log n)^2\right)\end{aligned}$$

besitzen.

Satz 4.12. *Es gilt* $\mathsf{VP} = \mathsf{VNC}^2$ *über jedem Körper.*

Beweis. Siehe Bürgisser [Bür13]. □

Für jede p-Familie in VP existiert somit ein SLP, welches parallelisiert eine maximale Tiefe von $\mathcal{O}\left((\log n)^2\right)$ aufweist, bzw. ein Schaltkreis mit Tiefe $\mathcal{O}\left((\log n)^2\right)$. Verdeutlicht wird dies am Beispiel von POWSUM aus Beispiel 3.6.

Beispiel 4.13. Die Familie wurde als

$$\mathsf{POWSUM} \coloneqq (\mathsf{POWSUM}_n)_{n\geq 1} \qquad \text{mit} \qquad \mathsf{POWSUM}_n \coloneqq \sum_{i=1}^{n} X_i^n$$

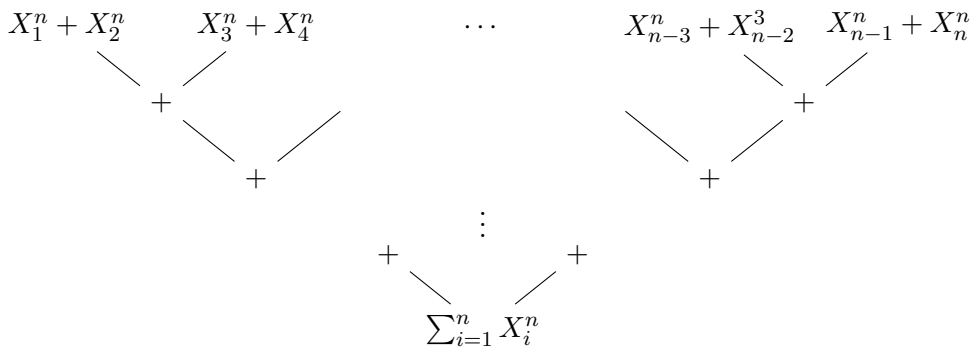
definiert. Ein mögliches paralleles SLP würde zuerst für alle n Variablen die n-te Potenz berechnen. Dies kann für alle X_i parallel geschehen und benötigt jeweils $\mathcal{O}(\log n)$ Instruktionen. In einem Schaltkreis würde die gleiche Anzahl Multiplikationsgatter benötigt.

$$\begin{aligned}\Gamma_1 &= X_i & X_i \\ \Gamma_2 &= (*, 1, 1) & X_i^2 \\ \Gamma_3 &= (*, 2, 2) & X_i^4 \\ &\vdots & \vdots\end{aligned}$$

Nachfolgend kann die Summation der Variablen paarweise erfolgen, was ebenfalls mit $\mathcal{O}(\log n)$ Instruktionen bzw. Gattern möglich ist. Als Gesamttiefe ergibt sich

$$\mathcal{O}(\log n) + \mathcal{O}(\log n) = \mathcal{O}(\log n) \leq \mathcal{O}\left((\log n)^2\right)$$

Die p-Familie aus VP liegt also auch in VNC^2.



Ein Beispiel für vollständige Probleme der algebraischen Schaltkreisklassen ist das iterierte Produkt von n vielen 3×3 Matrizen, welches unter p-Projektion VNC^1-vollständig ist [BC92].

Die Erklärung für die unterschiedlichen Resultate der beiden Modelle liegt im zugrunde liegenden Berechnungsmodell. Für Straight-Line-Programme bzw. arithmetische Schaltkreise kann der Satz 4.12 durch die von der Algebra gegebenen Struktur bewiesen werden. Für das parallele Maschinenmodell des klassischen Falls fehlt diese.

4.4 Valiant- versus Cookhypothese

Es lässt sich zeigen, dass sämtliche Aussagen über die Valianthypothese abhängig vom zugrunde liegenden Körper k und insbesondere dessen Charakteristik sind. Körper mit Charakteristik 0 sind z. B. $\mathbb{N}$, $\mathbb{R}$ oder $\mathbb{C}$, über welchen auch in der klassischen Komplexitätstheorie gerechnet wird. Wohingegen Körper mit einer Charakteristik ungleich 0 z. B. eher Restklassenkörpern der zugehörigen Primzahl entsprechen. Nach Definition der Valiantklassen gilt $\mathsf{VNP} = \mathsf{VNP}_k$ sowie $\mathsf{VP} = \mathsf{VP}_k$. Valiant [Val92] zeigte bereits, dass über $\mathbb{F}_2$ aus der nichtuniformen Variante der Cookhypothese, d. h. $\mathsf{P/poly} \neq \mathsf{NP/poly}$, folgen würde, dass $\mathsf{VP} \neq \mathsf{VNP}$ gilt. Über dem Körper der komplexen Zahlen $\mathbb{C}$ würde in diesem Fall auch die generelle Cookhypothese $\mathsf{P} \neq \mathsf{NP}$ verifiziert. Im Folgenden wird aufgeführt, welche Folgen eine Gleichheit der Klassen VP und VNP hätte und der Vergleich zu den Folgen der Gleichheit der klassischen Klassen P und NP gezogen. Die Resultate basieren auf Bürgisser [Bür13].

4.4.1 Beziehungen der Valiantklassen zur klassischen Komplexitätstheorie

Damit die algebraischen Komplexitätsklassen besser mit den klassischen vergleichbar sind, werden diese im Folgenden einheitlich über sogenannte Stringfunktionen definiert.

Definition 4.14. Sei $(\varphi_n)_{n\geq 1}$ eine Folge Boole'scher Funktionen mit

$$\varphi_n \colon \{0,1\}^n \to \{0,1\}^{m(n)},\ x \mapsto (\varphi_{n,1}(x), \ldots, \varphi_{n,m(n)}(x)),$$

dann wird mit

$$\varphi \colon \{0,1\}^* \to \{0,1\}^*,\ x \in \{0,1\}^n \mapsto \varphi_n(x)$$

die zugehörige Stringfunktion bezeichnet, welche auf die einzelnen Boole'schen Funktionen abbildet.

Die Auswirkungen der Valianthypothese auf die klassische Komplexitätstheorie werden deutlich, wenn sich auf einen Teil der Valiantklassen, den sogenannten Boole'schen Anteil BP beschränkt wird.

Definition 4.15. (1) Sei $(f_n)_{n\geq 1}$ eine p-Familie mit $f_n \in k[X_1, \ldots, X_n]$ über einem Körper k, dann wird eine Stringfunktion $(\varphi_n)_{n\geq 1}$ als Boole'scher Anteil (BP) von $(f_n)_{n\geq 1}$ bezeichnet, wenn gilt:

- $\operatorname{char} k = 0 :\ m(n) = n^{\mathcal{O}(1)}$

$$\forall x \in \{0,1\}^n :\ f_n(x) = \sum_{i=1}^{m(n)} \varphi_{n,i}(x) 2^{i-1}$$

- $\operatorname{char} k = p > 0 :\ m(n) = m \coloneqq \lfloor \log p \rfloor + 1$

$$\forall x \in \{0,1\}^n :\ f_n(x) = \sum_{i=1}^{m(n)} \varphi_{n,i}(x) 2^{i-1} \mod p$$

(2) Die Menge aller Boole'schen Anteile p-berechenbarer Familien über k ist $\mathrm{BP}(\mathsf{VP}_k)$, die p-definierbarer Familien $\mathrm{BP}(\mathsf{VNP}_k)$.

Die Boole'schen Anteile entsprechen somit der Binärdarstellung der Funktionswerte der zugehörigen p-Familie. Sind die Boole'schen Anteile bekannt, dienen sie als eine Art Hilfsfunktion zur Bestimmung der Funktionswerte. Diese Eigenschaft erinnert an die Advice-Klassen aus der klassischen Komplexitätstheorie [Gol08].

Definition 4.16. Eine Funktion $\alpha \colon \mathbb{N} \to \{0,1\}^*$ mit p-beschränkter Länge $|\alpha(n)|$ heißt polynomieller Advice. Sei $\mathcal{K}$ eine Komplexitätsklasse aus Stringfunktionen, die zugehörige nichtuniforme Klasse $\mathcal{K}/\mathsf{poly}$ besteht aus Funktionen $\psi(x) = \varphi(x, \alpha(|x|))$, mit $\varphi \in \mathcal{K}$ und einer polynomiellen Advice-Funktion α.

Es lassen sich nun Annahmen über die Beziehungen von $\mathrm{BP}(\mathsf{VP}_k)$ und $\mathrm{BP}(\mathsf{VNP}_k)$ zu den nichtuniformen Varianten der angeführten Komplexitätsklassen aufstellen. Dabei sind diese abhängig vom zugrunde liegenden Körper k und basieren auf der Gültigkeit der verallgemeinerten Riemannhypothese. Die verallgemeinerte Riemannhypothese ist eine Generalisierung der Riemannhypothese, welche Aussagen über die Nullstellen der Riemann'schen Zetafunktion macht. Diese stehen mit der genauen Verteilung der Primzahlen in Verbindung [Koc00].

Über einem Körper mit einer Charakteristik von 0, gilt

Satz 4.17 (A)**.** *Unter der verallgemeinerten Riemannhypothese gilt*

$$\begin{array}{rcccl} \mathsf{FNC}^1/\mathsf{poly} & \subseteq & \mathrm{BP}(\mathsf{VP}_k) & \subseteq & \mathsf{FNC}^3/\mathsf{poly} \\ \#\mathsf{P}/\mathsf{poly} & \subseteq & \mathrm{BP}(\mathsf{VNP}_k) & \subseteq & \mathsf{FP}^{\#\mathsf{P}}/\mathsf{poly}. \end{array}$$

Bei einem endlichen Körper mit Charakteristik $k = p > 0$, mit p einer Primzahl, folgt hingegen:

Satz 4.18 (B)**.**

$$\begin{array}{rcccl} \mathsf{FNC}^1/\mathsf{poly} & \subseteq & \mathrm{BP}(\mathsf{VP}_k) & \subseteq & \mathsf{FNC}^2/\mathsf{poly} \\ \#_p\mathsf{P}/\mathsf{poly} & = & \mathrm{BP}(\mathsf{VNP}_k). & & \end{array}$$

Im Folgenden werden die auftretenden Komplexitätsklassen erläutert, die Beweisideen der Sätze 4.17 und 4.18 werden in Abschnitt 4.4.3 gebracht. Auf Basis dieser Annahmen wird nun untersucht, welche Folgen die Gleichheit von VP_k und VNP_k hätte.

Es handelt sich bei den in den Annahmen auftretenden klassischen Klassen um nichtuniforme Funktionsklassen, unter anderem die Klasse FNC^i, zugehörig zu NC^i sowie FP, zugehörig zu P.

Definition 4.19. Die Klasse FNC^i beinhaltet Stringfunktionen, welche sich von einem Boole'schen Schaltkreis mit polynomieller Größe und Tiefe von $\mathcal{O}\left((\log n)^i\right)$ berechnen lassen [Vol99].

$$\mathsf{FNC}^i \coloneqq \mathsf{FSIZE\text{-}DEPTH}\left(n^{\mathcal{O}(1)}, (\log n)^i\right)$$

Definition 4.20. Die Klasse FP besteht aus denjenigen Stringfunktionen, welche in polynomieller Zeit von einer Turingmaschine berechnet werden können.

Es gilt $\mathsf{FNC}^1 \subseteq \mathsf{FNC}^2 \subseteq \cdots \subseteq \mathsf{FP} \subseteq \#\mathsf{P}$. Die Inklusion der Schaltkreisklassen folgt direkt aus der ansteigenden Tiefe, ein Schaltkreis der Tiefe $\mathcal{O}\left(\log n\right)^i)$ kann einen von Tiefe $\mathcal{O}\left(\log n\right)^{i-1})$ simulieren. Die Schaltkreisklassen sind Teilmenge der Funktionsklasse FP, da sie über ihre Tiefe eine zusätzliche Anforderung an die in polynomieller Zeit berechenbaren Funktionen der Klasse FP stellen. FP wiederum ist in $\#\mathsf{P}$ enthalten, da zu einer Eingabe x in Polynomialzeit der Funktionswert $f(x)$ berechnet und dies als Anzahl von Zertifikaten gewählt werden kann.

4.4.2 Zusammenhang zur Klasse NP

Die bekannte Klasse NP lässt sich alternativ zu Definition 4.3 auch über $\#\mathsf{P}$ definieren. Dabei entspricht NP der Menge aller Sprachen $\{x \in \{0,1\}^* \mid \phi(x) > 0\}$ mit einer Funktion $\phi \in \#\mathsf{P}$. Nach Definition von $\#\mathsf{P}$ handelt es sich bei ϕ um eine Funktion, welche die Anzahl an Lösungen ausgibt. Die neue Definition stellt NP als die Klasse aller Sprachen dar, für welche es mindestens ein polynomiell überprüfbares Zertifikat gibt.

In Annahme (B) aus Satz 4.18 wird $\mathrm{BP}(\mathsf{VNP}_k)$ in Beziehung mit der Klasse $\#_p\mathsf{P}/\mathsf{poly}$ gesetzt.

Definition 4.21. Die Klasse $\#_p\mathsf{P}/\mathsf{poly}$, mit einer Primzahl p, besteht aus allen Funktionen

$$\psi\colon \{0,1\}^* \to \mathbb{F}_p,\ x \mapsto \phi(x) \mod p,$$

wobei ϕ eine Funktion aus $\#\mathsf{P}$ ist.

Ähnlich wie im Zusammenhang zwischen $\#\mathsf{P}$ und NP existiert eine zu $\#_p\mathsf{P}$ zugehörige Sprachklasse $\mathsf{Mod}_p\mathsf{NP}$.

Definition 4.22. Die Klasse $\mathsf{Mod}_p\mathsf{NP}$, mit einer Primzahl p, ist die Menge an Sprachen

$$\{x \in \{0,1\}^* \mid \phi(x) \equiv 1 \mod p\},$$

wobei ϕ eine Funktion aus $\#\mathsf{P}$ ist.

Für den Beweis des Zusammenhangs mit den algebraischen Klassen sind die nichtuniformen Varianten der beiden Klassen NP und $\mathsf{Mod}_p\mathsf{NP}$ und ihre Beziehung zueinander von Interesse.

Satz 4.23. *Sei p eine Primzahl, dann gilt $\mathsf{NP}/\mathsf{poly} \subseteq \mathsf{Mod}_p\mathsf{NP}/\mathsf{poly}$.*

Der Beweis basiert auf dem von Valiant und Vazirani [VV85] und nutzt dabei Adlemans Trick [Adl78]. Als Grundlage dient die nichtuniforme Polynomialzeitreduktion.

Definition 4.24. Eine Sprache A ist nichtuniform in Polynomialzeit reduzierbar auf eine Sprache B, wenn eine Stringfunktion $\rho\colon \{0,1\}^* \to \{0,1\}^* \in \mathsf{FP/poly}$ existiert, sodass $A = \rho^{-1}(B)$ gilt.

Ist eine Komplexitätsklasse $\mathcal{K}$ unter Polynomialzeitreduktion abgeschlossen, so gilt dies übertragend auch für die zugehörige nichtuniforme Klasse $\mathcal{K}/\mathsf{poly}$ und die nichtuniforme Polynomialzeitreduktion.
Da daher die Klasse $\mathsf{Mod}_p\mathsf{NP/poly}$ unter nichtuniformer Polynomialzeitreduktion abgeschlossen ist, reicht es für den Beweis, eine solche Reduktion eines NP-vollständigen Problems auf ein Problem in $\mathsf{Mod}_p\mathsf{NP}$ zu zeigen. Das NP-vollständige Erfüllbarkeitsproblem aussagenlogischer Formeln SAT bietet sich in diesem Kontext an. Dabei sei $\#\phi$ die Anzahl erfüllender Belegungen für eine aussagenlogische Formel ϕ. Das analoge Problem $\mathrm{Mod}_p\mathrm{SAT}$ ist so definiert, dass $\#\phi \equiv 1 \mod p$ gilt. $\mathrm{Mod}_p\mathrm{SAT}$ liegt nach Definition in $\mathsf{Mod}_p\mathsf{NP}$. D. h. es ist zu zeigen, dass eine in nichtuniformer Polynomialzeit berechenbare Funktion existiert, welche eine KNF ϕ auf eine KNF χ abbildet, sodass $\#\phi > 0 \iff \#\chi \equiv 1 \mod p$ gilt.

Beweis. Valiant und Vazirani [VV85] zeigten einen Zusammenhang zwischen NP und randomisierten Problemen, insbesondere für SAT, über eine randomisierte Polynomialzeitreduktion. Bei diesen Reduktionen gilt die Rückrichtung mit einer minimalen Wahrscheinlichkeit. Sei ϕ eine KNF in n Variablen und w ein zufälliger Bitstring der Länge n. Nach Valiant und Vazirani [VV85] kann diesen in polynomieller Zeit eine KNF Φ zugewiesen werden, sodass

$$\begin{aligned} \#\phi = 0 &\implies \#\Phi = 0, \\ \#\phi > 0 &\implies \mathrm{Prob}[\#\Phi \neq 1] \leq 1 - \frac{1}{4n} \end{aligned} \tag{4.2}$$

gilt. Nun wird dieser Ansatz auf mehrere zufällige Bitstrings erweitert. Seien q eine ungerade Zahl und $w_1,\ w_2, \ldots, w_q$ q zufällige Bitstring jeweils der Länge n. Die KNF Φ_i sei wie oben die zu ϕ und w_i zugewiesene Formel. Außerdem kann aus den Φ_i und einer Primzahl p in Polynomialzeit eine weitere KNF χ berechnet werden, sodass

$$\#\chi = 1 + \prod_{j=1}^{q} (p - 1 + \#\Phi_j)$$

gilt. Damit folgt, dass sich die Implikationen aus Gleichung 4.2 zu

$$\begin{aligned} &\#\phi = 0 \implies \#\chi = 0, \\ &\#\phi > 0 \implies \text{Prob}[\#\chi \not\equiv 1 \mod p] \leq \left(1 - \frac{1}{4n}\right)^q \end{aligned} \tag{4.3}$$

umformen lassen. Die erste Implikation zeigt bereits die Rückrichtung der Reduktion. Um auch die Hinrichtung zu zeigen, muss die Wahrscheinlichkeit der zweiten Implikation möglichst klein werden. Nach Adleman [Adl78] lassen sich die zufälligen Bitstrings durch ein Orakel ersetzen, welches festgelegte Strings als Advice liefert. Diese haben eine Größe polynomiell in der Eingabelänge. Sei nun N_a die Anzahl an KNFs der Größe $a \geq n$. Es gilt $\log N_a = a^{\mathcal{O}(1)}$. Wird $q = a^{\mathcal{O}(1)}$ groß genug gewählt so ergibt sich desweiteren

$$N_a \left(1 - (4n)^{-1}\right)^q \leq N_a e^{\frac{-q}{4n}} < 1,$$

was wiederum zeigt, dass für alle a zufällige Bitstrings $w_1,\ w_2, \ldots, w_q$ existieren, sodass für alle KNFs der Größe a

$$\#\phi > 0 \implies \#\chi \equiv 1 \mod p$$

gilt. Die Bitstrings $w_1, \ldots, w_n$ sind dabei der Advice für KNFs der Größe a. Zusammen mit den Implikationen aus Gleichung 4.3 ist

$$\#\phi > 0 \iff \#\chi \equiv 1 \mod p$$

gezeigt. □

4.4.3 Beweiseideen der Beziehungen zur klassischen Komplexitätstheorie

Im Folgenden werden die Beweisideen der Hauptannahmen der Beziehungen von $\text{BP}(\mathsf{VP}_k)$ und $\text{BP}(\mathsf{VNP}_k)$ zu den klassischen Komplexitätsklassen aufgeführt. Die vollständigen Beweise finden sich in Bürgisser [Bür13]. Begonnen wird mit der Annahme (A) aus Satz 4.17

$$\begin{array}{ccccc} \mathsf{FNC}^1/\mathsf{poly} & \subseteq & \text{BP}(\mathsf{VP}_k) & \subseteq & \mathsf{FNC}^3/\mathsf{poly} \\ \#\mathsf{P}/\mathsf{poly} & \subseteq & \text{BP}(\mathsf{VNP}_k) & \subseteq & \mathsf{FP}^{\#\mathsf{P}}/\mathsf{poly}. \end{array}$$

Beweis Satz 4.17 *(A).*

(A1) $\mathsf{FNC}^1/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VP}_k)$

Straight-Line-Programme können Boole'sche Schaltkreise in einer durch die Komplexität der Schaltkreise begrenzten Komplexität simulieren. Sei $(C_n)_{n\in\mathbb{N}}$ eine Familie von Schaltkreisen in $\mathsf{FNC}^1/\mathsf{poly}$ sowie φ_n die von C_n berechnete Funktion. Das SLP Γ_n simuliert den Schaltkreis in $\mathsf{SIZE}(\Gamma_n) \leq 3 \cdot \mathsf{FSIZE}(C_n)$ und $\mathsf{DEPTH}(\Gamma_n) \leq 2 \cdot \mathsf{FDEPTH}(C_n)$. Dabei lassen sich die Boole'schen Funktionen wie folgt als Polynome des SLPs darstellen.

$$
\begin{aligned}
&\forall x, y \in \{0,1\}: \\
&\qquad x \wedge y = x \cdot y \\
&\qquad \bar{x} = 1 - x \\
&\qquad x \vee y = \overline{\bar{x} \wedge \bar{y}} \text{ (nach de Morgan)} \\
&\qquad\quad = 1 - ((1-x)(1-y)) \\
&\qquad\quad = 1 - (1 - x - y + xy) \\
&\qquad\quad = 1 - 1 + x + y - xy \\
&\qquad\quad = x + y - x \cdot y
\end{aligned}
$$

Für alle Polynome des SLPs und alle $x \in \{0,1\}$ gilt somit $g_{n,i} = \varphi_{n,i}$. Die φ_n bzw. g_n sind somit Boole'sche Anteile einer Funktion f_n, mit $f_n \coloneqq \sum_{i=1}^{m(n)} g_{n,i} 2^{i-1}$. Die Familie $(f_n)_{n\geq 1}$ ist p-berechenbar, da sie von maximal polynomiellen Grad ist. Es gilt $\deg g_n \leq 2^{\mathsf{DEPTH}(\Gamma_n)} \leq 2^{2\log n} \leq n^{\mathcal{O}(1)}$.

(A2) $\#\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$

Grundlage für den Beweis von (A2) ist das Valiant Kriterium aus Behauptung 3.14, welches einen Zusammenhang zwischen den Zählklassen und VNP herstellt. Zunächst wird gezeigt, dass $\#\mathsf{P} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$ und das Resultat dann auf $\#\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$ erweitert.

Angenommen, (φ_n) sei eine Stringfunktion in $\#\mathsf{P}$ und stellt den Boole'schen Anteil einer Funktion $\phi(x) \coloneqq \sum_i \varphi_{n,i} 2^{i-1}$ dar. Nach Definition liegt $\phi(x)$ in $\#\mathsf{P}$. Mit dem Valiant Kriterium folgt, dass es eine p-definierbare Funktion $g_n(x) = \phi(x)$ gibt. Damit ist (φ_n) Boole'scher Anteil der Funktion $(g_n)_{n\geq 1}$ aus VNP_k.

Nun sei angenommen, $(\psi_n) \in \#\mathsf{P}/\mathsf{poly}$. Nach Definition der Klasse gibt es eine Funktion $(\varphi_n) \in \#\mathsf{P}$ und Advice α, sodass $\psi_n = \varphi_{t(n)}(x, \alpha)$ mit $t(n) = |(x,\alpha)|$. Zur Funktion φ gibt es, wie bereits gezeigt, eine Funktion $(g_n) \in \mathsf{VNP}_k$, sodass $g_n(x) = \sum_i \varphi_{n,i} 2^{i-1}$. Um darüber den Bezug zur Funktion (ψ_n) herzustellen, wird

eine Funktion $h_n(X) = g_{t(n)}(X, \alpha(n))$ eingeführt. Diese ist eine p-Projektion von $(g_n)_{n\geq 1}$ und somit ebenfalls in VNP_k. Es gilt nun

$$h_n(x) = g_{t(n)}(x, \alpha(n)) = \sum_i \varphi_{t(n),i}(x, \alpha(n)) 2^{i-1} = \sum_i \psi_{n,i} 2^{i-1}.$$

Damit handelt es sich bei (ψ_n) um einen Boole'schen Anteil (h_n), (ψ_n) liegt also in $\mathrm{BP}(\mathsf{VNP}_k)$ und damit ist $\#\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$.

(A3) $\mathrm{BP}(\mathsf{VP}_k) \subseteq \mathsf{FNC}^2/\mathsf{poly}$
Sei $(f_n)_{n\geq 1}$ eine Funktion in VP, welche Boole'sche Anteile besitzt. Es existiert ein Straight-Line-Programm Γ_n mit Größe $n^{\mathcal{O}(1)}$ und Tiefe $\mathcal{O}\left((\log n)^2\right)$, welches $(f_n)_{n\geq 1}$ berechnet. Das SLP Γ_n lässt sich durch einen Boole'schen Schaltkreis in $\mathsf{FNC}^3/\mathsf{poly}$ simulieren.

(A4) $\mathrm{BP}(\mathsf{VNP}_k) \subseteq \mathsf{FP}^{\#\mathsf{P}}/\mathsf{poly}$
Ähnlich wie in (A3) sei $(f_n)_{n\geq 1}$ eine Funktion in VNP_k, welche Boole'sche Anteile besitzt. Nach Definition von VNP_k ist $(f_n)_{n\geq 1}$ über eine p-berechenbare Funktion g_n p-definierbar. Im dem Fall in dem diese Funktion Boole'sche Anteile besitzt, gilt $(f_n) \in \mathsf{FNC}^3/\mathsf{poly} \subseteq \mathsf{FP}/\mathsf{poly} \subseteq \#\mathsf{P}/\mathsf{poly}$. Andernfalls kann über Simulation der Straight-Line-Programme durch Schaltkreise gezeigt werden, dass die Boole'schen Anteile von $(f_n)_{n\geq 1}$ in $\mathsf{FP}^{\#\mathsf{P}}/\mathsf{poly}$ berechnet werden können.

□

Die Beweise der Resultate aus (A) können für den Fall über Körpern einer Charakteristik ungleich 0 angepasst werden. Auf diese Weise kann die Annahme (B) aus Satz 4.18

$$\begin{aligned} \mathsf{FNC}^1/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VP}_k) \subseteq \mathsf{FNC}^2/\mathsf{poly} \\ \#_p\mathsf{P}/\mathsf{poly} = \mathrm{BP}(\mathsf{VNP}_k) \end{aligned}$$

gezeigt werden.

Beweis Satz 4.18 *(B).* Die Ergebnisse aus (A1) und (A2) sind direkt auf (B) übertragbar, es gilt auch im Falle von $\mathrm{char}\, k = p > 0$ $\mathsf{FNC}^1/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VP}_k)$. Das Resultat (A2) ändert sich zu $\#_p\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$ aufgrund der durch die Charakteristik p bestimmten Struktur des Körpers k wodurch Modulo p gerechnet wird. Auf ähnliche Weise wie in (A3) folgt nun $\mathrm{BP}(\mathsf{VP}_k) \subseteq \mathsf{FNC}^2/\mathsf{poly}$. Da der Körper k endlich, ist kann ein Straight-Line-Programm direkt von einem Schaltkreis in $\mathsf{FSIZE\text{-}DEPTH}\left(n^{\mathcal{O}(1)}, \mathcal{O}\left((\log n)^2\right)\right)$ si-

muliert werden. Im Gegensatz zu (A4) kann hier nun jedoch gezeigt werden, dass sowohl $\#_p\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k)$ als auch $\mathrm{BP}(\mathsf{VNP}_k) \subseteq \#_p\mathsf{P}/\mathsf{poly}$ und damit dann die Gleichheit $\mathrm{BP}(\mathsf{VNP}_k) = \#_p\mathsf{P}/\mathsf{poly}$ gilt. Es wird angenommen, dass auch $\mathrm{BP}(\mathsf{VNP}_k) = \#\mathsf{P}/\mathsf{poly}$ gilt. □

4.4.4 Auswirkungen der Valianthypothese auf die klassische Komplexitätstheorie

Wäre die Valianthypothese falsch, d. h. es würde $\mathsf{VP}_k = \mathsf{VNP}_k$ gelten, so lassen sich zwei Fälle unterscheiden. Ist $\operatorname{char} k = 0$, so folgt aus Annahme (A) Satz 4.17, dass

$$\#\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathsf{VNP}_k) = \mathrm{BP}(\mathsf{VP}_k) \subseteq \mathsf{FNC}^3/\mathsf{poly} \subseteq \mathsf{FP}/\mathsf{poly} \subseteq \#\mathsf{P}/\mathsf{poly} \tag{4.4}$$

gelten muss. Bekannt ist unter anderem nach Johnson [Joh90], dass $\mathsf{P} = \mathsf{NP} \implies \mathsf{P} = \mathsf{PH}$ gilt. Ein ähnliches Resultat existiert auch für die nichtuniformen Klassen,

$$\mathsf{P}/\mathsf{poly} = \mathsf{NP}/\mathsf{poly} \implies \mathsf{P}/\mathsf{poly} = \mathsf{PH}.$$

Karp und Lipton [KL80] zeigten, dass wenn $\mathsf{P}/\mathsf{poly} = \mathsf{NP}/\mathsf{poly}$ gilt, die Polynomialzeithierarchie (PH) auf die zweite Stufe kollabiert. Da aus Gleichung 4.4 die Gleichheit aller Klassen der Polynomialzeithierarchie, insbesondere auch $\mathsf{P}/\mathsf{poly} = \mathsf{NP}/\mathsf{poly}$ folgt, würde dieser Kollaps bei Gleichheit von VP_k und VNP_k eintreten.

Im Falle einer Charakteristik von $p > 0$ würde nach Annahme (B) Satz 4.18

$$\mathsf{FNC}^1/\mathsf{poly} \subseteq \ \mathrm{BP}(\mathsf{VP}_k) = \ \mathrm{BP}(\mathsf{VNP}_k) = \#_p\mathsf{P}/\mathsf{poly} \subseteq \mathsf{FNC}^2/\mathsf{poly}$$

gelten. Vereinfacht bedeutet dies

$$\#_p\mathsf{P}/\mathsf{poly} \subseteq \mathsf{FNC}^2/\mathsf{poly} \subseteq \mathsf{FP}/\mathsf{poly}. \tag{4.5}$$

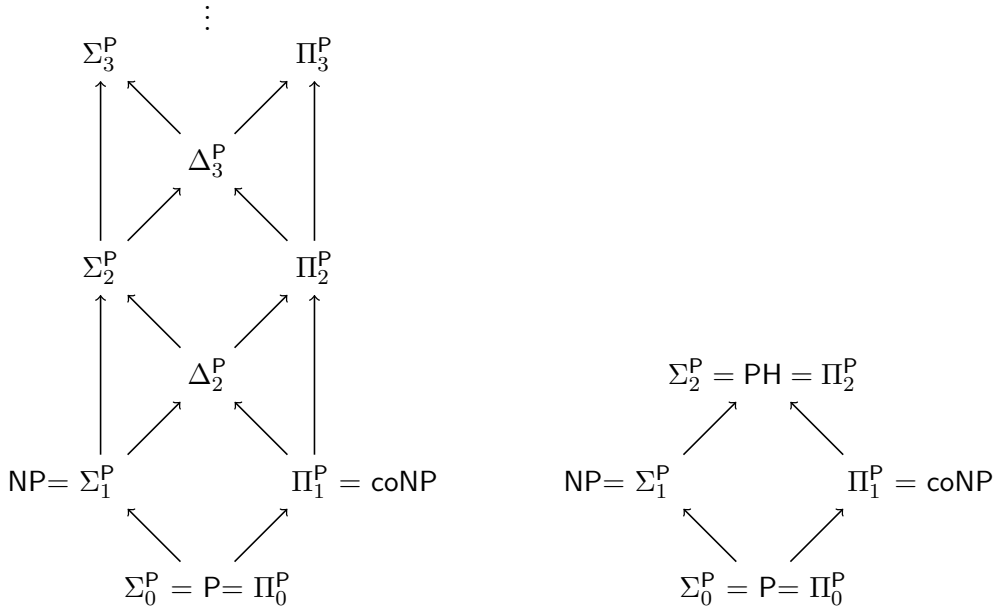
Werden nun die zu diesen Zählklassen zugehörige Sprachklassen betrachtet, so folgt

$$\mathsf{Mod}_p\mathsf{NP}/\mathsf{poly} \subseteq \mathsf{NC}^2/\mathsf{poly} \subseteq \mathsf{P}/\mathsf{poly} \subseteq \mathsf{NP}/\mathsf{poly}. \tag{4.6}$$

Der Satz 4.23 sagt jedoch aus, dass $\mathsf{NP}/\mathsf{poly} \subseteq \mathsf{Mod}_p\mathsf{NP}/\mathsf{poly}$ ist und damit folgt die Gleichheit der Klassen in Gleichung 4.6. Mit gleicher Argumentation wie bei Körpern mit Charakteristik 0 würde somit die Polynomialzeithierarchie auf die zweite Stufe kollabieren.

4.4.5 Vergleich zur Cookhypothese

Gilt die Valianthypothese nicht, d. h. es ist $\mathsf{VP} = \mathsf{VNP}$, so folgt, dass $\mathsf{P/poly} = \mathsf{NP/poly}$ ist. Im Vergleich zur Cookhypothese folgt aus $\mathsf{P/poly} = \mathsf{NP/poly}$, dass die Polynomialzeithierarchie „nur“ auf die zweite Stufe kollabiert, wie in Abb. 4.3b dargestellt. Wäre hingegen $\mathsf{P} = \mathsf{NP}$, so würde sie komplett kollabieren, d. h. $\mathsf{P} = \mathsf{NP} = \mathsf{PH}$.



(a) Polynomialzeithierarchie

(b) Polynomialzeithierarchie kollabiert auf die zweite Stufe

Abbildung 4.3: Auswirkung auf die Polynomialzeithierarchie

Aus Gleichheit der Valiantklassen VP und VNP lässt sich demnach nach aktuellem Forschungsstand keine direkte Aussage über die Gültigkeit der uniformen Cookhypothese ziehen, da unbekannt ist, ob aus Gleichheit von P/poly und NP/poly auch die Gleichheit von P und NP folgt. Bisher ist nur über $\mathbb{C}$ die umgekehrte Implikation $\mathsf{P/poly} \neq \mathsf{NP/poly} \implies \mathsf{P} \neq \mathsf{NP}$ bekannt. Die Schwierigkeit des Vergleichs der klassischen und der Valiantklassen liegt nach diesem Ansatz auch darin, einen Zusammenhang zwischen der nichtuniformen und uniformen Komplexität herzustellen. Es wird angenommen, dass die Vermutung $\mathsf{NP} \subseteq \mathsf{P}$ äquivalent zu $\mathsf{NP} \subseteq \mathsf{P/poly}$ ist, welche aber ebenfalls bisher nicht bewiesen werden konnte. Ein weiteres Problem ist es, gefundene Ergebnisse auf beliebige Körpern zu verallgemeinern, da diese Ergebnisse, wie in Ab-

schnitt 4.4.1 gezeigt, stark vom zugrunde liegenden Körper abhängig sind. Resultate lassen sich nicht allein auf Basis der gemeinsamen Körperaxiome aufstellen. Es wird jedoch auch im algebraischen Fall angenommen dass $\mathsf{VP} \neq \mathsf{VNP}$ ist, da ein Kollaps der Polynomialzeithierarchie als unwahrscheinlich gilt [Bür13; MP08].

5 Ausblick

Viele Herangehensweisen und Konzepte, die in der klassischen Komplexitätstheorie entwickelt wurden, lassen sich auf die algebraische Komplexitätstheorie übertragen. Dazu gehören z. B. die Intermediate-Probleme sowie die parametrisierte Komplexität.

Ähnlich zu NP-Intermediate-Problemen der klassischen Komplexitätstheorie kann gezeigt werden, dass es, wenn die Valianthypothese gilt, p-Familien in VNP gibt, welche nicht in VP liegen und nicht VNP-vollständig sind. Die Vollständigkeit ist dabei über sogenannten c-Reduktionen definiert, ein Reduktionsbegriff ähnlich zur Turingreduktion. Diese Probleme werden analog VNP-Intermediate genannt. Die Besonderheit liegt in der algebraischen Variante darin, dass eine spezifische p-Familie als VNP-Intermediate bewiesen werden konnte. Im Gegensatz zu bekannten NP-Intermediate Problemen des Cook- oder BSS-Modells ist diese Familie natürlicher und nicht nur auf Basis des benötigten Beweises definiert [MS18].

Die parametrisierte Komplexitätstheorie befasst sich damit, die Komplexität von Problemen abhängig von der Wahl ihrer Parameter zu bestimmen. Dabei ist insbesondere interessant, von welchen Parametern die Laufzeit abhängig ist. Eine neue Klasse FPT beinhaltet die parametrisierbaren (fixed parameter tractable) Probleme, welche sich in einer Laufzeit von $f(k)p(n)$ von einem Algorithmus lösen lassen. Dabei bezeichnet k den gewählten Parameter und n die Eingabelänge, f ist eine berechenbare Funktion, sowie p ein beliebiges Polynom. Über parametrisierte Zählprobleme und die resultierende Klasse #FPT sowie den Zusammenhang vieler (Boole'scher) Zählprobleme mit Auswertungsproblemen von Polynomen lassen sich auch parametrisierte Varianten der Valiantklassen definieren. Ein Beispiel für ein parametrisiertes algebraisches Problem ist die erzeugende Funktion für Vertex Cover (VC) der Größe k

$$\mathrm{GF}(\mathsf{VC}) = \sum_{\substack{C \subseteq \{1,\dots,n\},\\ |C|=k}} \prod_{i \in C} X_i.$$

Interessant ist außerdem der Vergleich nicht nur mit dem klassischen, sondern auch einem weiteren Modell der Komplexitätstheorie, dem BSS-Modell. Das BSS-Modell, benannt nach Blum, Shub und Smale [BSS88], welche es 1988 einführten, befasst sich mit Berechnungen über den reellen Zahlen. Ziel ist es, über die Kombination algebraischer Berechnungsmodelle und Uniformität einen einzigen Algorithmus zur Lösung eines Problems in allen Dimensionen anzugeben. Das Berechnungsmodell ist die BSS-Maschine, eine Random Access Machine (RAM), welche reelle Zahlen speichern und rationale Funktionen in einem Schritt berechnen kann. Es stellt damit wie die SLP einen symbolischen Ansatz des Rechnens dar, da Zahlen jeweils in einem Register gespeichert werden. Über einem Körper k werden analog zur klassischen Komplexitätstheorie die Klassen NP_k und P_k definiert, welche diejenigen Entscheidungsprobleme enthalten, die mit Hilfe einer BSS-Maschine über k ungeachtet der Dimension der Eingabe in (nichtdeterministisch) polynomiell beschränkter Zeit entschieden werden können. Ebenfalls analog definiert ist die BSS-Hypothese, dass auch $\mathsf{P}_k \neq \mathsf{NP}_k$ gilt [Bür13]. Es ist unter anderem nach Cucker et al. [Cuc+95] bekannt, dass aus der nichtuniformen Cookhypothese folgt, dass im BSS-Modell $\mathsf{P} \neq \mathsf{NP}$ über $\mathbb{C}$ gilt. Ob dieses Resultat auch auf $\mathbb{R}$ übertragbar ist ist unbekannt. Genauso fehlt eine Relation zwischen dem algebraischen und dem BSS-Modell. Es wird jedoch unter anderem von Bürgisser in [Bür04] vermutet, dass aus $\mathsf{VP} \neq \mathsf{VNP}$ über $\mathbb{C}$ auch $\mathsf{P} \neq \mathsf{NP}$ über $\mathbb{C}$ im BSS-Modell folgt. Die Schwierigkeit liegt hier, wie auch im Zusammenhang zwischen dem algebraischen und klassischen Modell, darin, eine Beziehung der Komplexität zwischen algebraischen Berechnungs- und Entscheidungsproblemen herzustellen.

Aktueller Stand der Forschung ist eine Erweiterung der algebraischen Komplexitätstheorie, die Geometrische Komplexitätstheorie (GCT), welche versucht, mit Hilfe algebraischer Geometrie und Darstellungstheorie untere Schranken zu finden. Dabei werden unter anderem die algebraischen Klassen untersucht. Eines der Ziele ist es über diesen Ansatz eine Antwort auf die P-NP-Frage zu finden [Mul11].

Literatur

[Aar16] Scott Aaronson. „P $\stackrel{?}{=}$ NP“. In: *Open problems in mathematics*. Springer, 2016, S. 1–122. DOI: 10.1007/978-3-319-32162-2.

[AB09] Sanjeev Arora und Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.

[Adl78] Leonard Adleman. „Two theorems on random polynomial time“. In: *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*. IEEE. 1978, S. 75–83. DOI: 10.1109/SFCS.1978.37.

[Agr06] Manindra Agrawal. „Determinant versus permanent“. In: European Mathematical Society, 2006. DOI: 10.4171/022-3/48.

[BC92] Michael Ben-Or und Richard Cleve. „Computing algebraic formulas using a constant number of registers“. In: *SIAM Journal on Computing* 21.1 (1992), S. 54–58. DOI: 10.1137/0221006.

[BCS13] Peter Bürgisser, Michael Clausen und Mohammad A Shokrollahi. *Algebraic complexity theory*. Bd. 315. Springer Science & Business Media, 2013. ISBN: 3-540-60582-7.

[Bos14] Siegfried Bosch. *Lineare Algebra*. Bd. 5. Springer, 2014. ISBN: 978-3-662-62615-3. DOI: 10.1007/978-3-662-62616-0.

[Bos20] Siegfried Bosch. *Algebra*. Bd. 9. Springer, 2020. ISBN: 978-3-662-61648-2. DOI: 10.1007/978-3-662-61649-9.

[BSS88] Lenore Blum, Mike Shub und Steve Smale. „On a Theory of Computation over the Real Numbers; NP Completeness, Recursive Functions and Universal Machines (Extended Abstract)“. In: (1988), S. 387–397. DOI: 10.1109/SFCS.1988.21955.

[Bür04] Peter Bürgisser. „The complexity of factors of multivariate polynomials“. In: *Foundations of Computational Mathematics* 4.4 (2004), S. 369–396. DOI: 10.1007/s10208-020-09477-6.

[Bür13] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*. Bd. 7. Springer Science & Business Media, 2013. ISBN: 978-3-540-66752-0.

[Cuc+95] Felipe Cucker et al. „On real Turing machines that toss coins“. In: *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*. 1995, S. 335–342. DOI: 10.1145/225058.225155.

[CV08] Nadia Creignou und Heribert Vollmer. „Boolean Constraint Satisfaction Problems: When Does Post's Lattice Help?“ In: *Complexity of Constraints*. Springer, 2008, S. 3–37. DOI: 10.1007/978-3-540-92800-3.

[DK11] Ding-Zhu Du und Ker-I Ko. *Theory of computational complexity*. Bd. 58. John Wiley & Sons, 2011. ISBN: 978-1-118-59497-1.

[Dur+14] Arnaud Durand et al. „Homomorphism polynomials complete for VP“. In: *34th International Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2014. DOI: 10.4230/LIPIcs.FSTTCS.2014.493.

[Gat87] Joachim von zur Gathen. „Permanent and determinant“. In: *Linear Algebra and its Applications* 96 (1987), S. 87–100. DOI: 10.1109/SFCS.1986.42.

[Gat88] Joachim von zur Gathen. „Algebraic complexity theory“. In: *Annual review of computer science* 3.1 (1988), S. 317–348. DOI: 10.1146/ANNUREV.CS.03.060188.001533.

[GHR91] Raymond Greenlaw, H James Hoover und Walter L Ruzzo. *A compendium of problems complete for P*. Citeseer, 1991. DOI: 10.7939/R39Z90F7X.

[GJT76] Michael R Garey, David S. Johnson und R Endre Tarjan. „The planar Hamiltonian circuit problem is NP-complete“. In: *SIAM Journal on Computing* 5.4 (1976), S. 704–714. DOI: 10.1137/0205049.

[GK03] Johannes Grabmeier und Erich Kaltofen. *Computer Algebra Handbook: Foundations, Applications, Systems*. Springer Science & Business Media, 2003. ISBN: 978-3-642-62988-4.

[Gol08] Oded Goldreich. „Computational complexity: a conceptual perspective“. In: *ACM Sigact News* 39.3 (2008), S. 35–39. DOI: 10.1017/CBO9780511804106.

[Joh90] David S Johnson. „A catalog of complexity classes“. In: *Algorithms and complexity*. Elsevier, 1990, S. 67–161. DOI: 10.1016/B978-0-444-88071-0.50007-2.

[KL80] Richard M Karp und Richard J Lipton. „Some connections between nonuniform and uniform complexity classes“. In: *Proceedings of the twelfth annual ACM symposium on Theory of computing*. 1980, S. 302–309.

[Koc00] Helmut Koch. *Number Theory: Algebraic numbers and functions*. 24. American Mathematical Soc., 2000. ISBN: 978-0-8218-2054-4.

[Men11] Stefan Mengel. „Characterizing arithmetic circuit classes by constraint satisfaction problems“. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2011, S. 700–711. DOI: 10.1007/978-3-642-22006-7_59.

[MP08] Guillaume Malod und Natacha Portier. „Characterizing Valiant's algebraic complexity classes“. In: *Journal of complexity* 24.1 (2008), S. 16–38. DOI: 10.1016/j.jco.2006.09.006.

[MR13] Meena Mahajan und BV Raghavendra Rao. „Small Space Analogues of Valiant's Classes and the Limitations of Skew Formulas“. In: *computational complexity* 22.1 (2013), S. 1–38. DOI: 10.1007/s00037-011-0024-2.

[MS18] Meena Mahajan und Nitin Saurabh. „Some complete and intermediate polynomials in algebraic complexity theory“. In: *Theory of Computing Systems* 62.3 (2018), S. 622–652. DOI: 10.1007/s00224-016-9740-y.

[Mul11] Ketan D Mulmuley. „On P vs. NP and geometric complexity theory: Dedicated to Sri Ramakrishna“. In: *Journal of the ACM (JACM)* 58.2 (2011), S. 1–26. DOI: 10.1145/1944345.1944346.

[OW17] Thomas Ottmann und Peter Widmayer. *Algorithmen und Datenstrukturen*. Springer, 2017. ISBN: 978-3-8274-2803-5. DOI: 10.1007/978-3-8274-2804-2.

[PS98] George Pólya und Gabor Szegő. *Problems and Theorems in Analysis I*. Springer, 1998. DOI: 10.1007/978-3-642-61983-0.

[Sch78] Thomas J Schaefer. „The complexity of satisfiability problems“. In: *Proceedings of the tenth annual ACM symposium on Theory of computing*. 1978, S. 216–226. DOI: 10.1145/800133.804350.

[Sip96] Michael Sipser. „Introduction to the Theory of Computation“. In: *ACM Sigact News* 27.1 (1996).

[Str86] Volker Strassen. „The work of LG Valiant“. In: *Proc. Int. Congress of Mathematicians, Berkeley CA*. 1986. DOI: 10.1145/1536414.1536415.

[Val+83] Leslie G. Valiant et al. „Fast Parallel Computation of Polynomials Using Few Processors“. In: Bd. 12. 4. 1983, S. 641–644. DOI: 10.1137/0212043.

[Val79a] Leslie G Valiant. „Completeness classes in algebra“. In: *Proceedings of the eleventh annual ACM symposium on Theory of computing*. 1979, S. 249–261. DOI: 10.1145/800135.804419.

[Val79b] Leslie G Valiant. „The complexity of computing the permanent“. In: *Theoretical computer science* 8.2 (1979), S. 189–201. DOI: 10.1016/0304-3975(79)90044-6.

[Val79c] Leslie G Valiant. „The complexity of enumeration and reliability problems“. In: *SIAM Journal on Computing* 8.3 (1979), S. 410–421. DOI: 10.1137/0208032.

[Val92] Leslie G Valiant. „Why is Boolean complexity theory difficult“. In: *Boolean Function Complexity* 169.84-94 (1992), S. 3. DOI: 10.1017/CBO9780511526633.008.

[Vol99] Heribert Vollmer. *Introduction to circuit complexity: a uniform approach*. Springer Science & Business Media, 1999. ISBN: 978-3-540-64310-4. DOI: 10.1007/978-3-662-03927-4.

[VV85] Leslie G Valiant und Vijay V Vazirani. „NP is as easy as detecting unique solutions“. In: *Proceedings of the seventeenth annual ACM symposium on Theory of computing*. 1985, S. 458–463. DOI: 10.1016/0304-3975(86)90135-0.