Komplexität von Suchproblemen und Beweissystemen

Anton Ehrmanntraut

20. November 2023

Inhaltsverzeichnis

1 Einleitung

Viele andere Beweise der Komplexitätstheorie funktionieren auch in relativierten Umgebungen, also wenn alle beteiligten TM zu OTM ausgewechselt werden. Das Diagonalargument in einem typischen Beweis von $\mathrm{P} \subsetneq \mathrm{E}$ relativiert beispielsweise, sodass auch $\mathrm{P}^O \subsetneq \mathrm{E}^O$ für jedes beliebige Orakel O gilt. Wir sagen dann auch, dass diese Aussagen bzw. Beweise *relativieren*.

Die typischen Beweistechniken der Komplextätstheorie relativieren. Das macht eigens konstruierte Orakel zu Indizien, dass gewisse Aussagen schwer zu beweisen sind. Beispielsweise konstruieren Baker, Gill und Solovay (1975) ein Orakel A sodass $\mathrm{P}^A \neq \mathrm{NP}^A$. Mit diesem Fakt ist die Aussage „$\mathrm{P} = \mathrm{NP}$“ nicht mit relativierbaren Methoden beweisbar, da sonst ja auch $\mathrm{P}^A = \mathrm{NP}^A$ gelten würde.

Die algorithmische Komplexitätstheorie fokussiert sich primär auf Entscheidungsprobleme, und weniger auf Suchprobleme. Dabei sind Suchprobleme mindestens genauso natürlich wie Entscheidungsprobleme, und in einigen Fällen auch (vor allem für Nicht-Fachleute) intuitiver: wir sind nicht daran interessiert *ob* eine Dreifärbung eines Graphen existiert, sondern sind an der konkreten Färbung interessiert. Diese Fokussierung auf Entscheidungsprobleme ist durchaus fundiert: es lässt sich argumentieren, dass für viele relevante Instanzen das Suchproblem auf das entsprechende Entscheidungsproblem „reduziert“ werden kann, womit aus algorithmischer Perspektive das Suchproblem nicht schwerer ist als das Entscheidungsproblem. Die Konzentration auf Suchproble kommt dann unter anderem auch mit dem Vorteil, dass viele theoretische Konzepte einfacher zu fassen sind und komakter zu formulieren sind; so z.B. der Begriff von *Sprache* und Turing-Maschinen die als „Ausgabe“ nur akzeptieren bzw. ablehnen können. Das hat auch historische Gründe: Entscheidungsprobleme waren der primäre Fokus innerhalb der Berechenbarkeitstheorie einerseits und in der Theorie der formalen Sprachen und Grammatiken andererseits.

SAT	:	es ex. keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge für NP mit p-optimalem Beweissystem
TAUT	:	es ex. keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge für coNP mit p-optimalem Beweissystem
$\mathsf{CON}^{\mathsf{N}}$	:	es ex. keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge für coNP mit optimalem Beweissystem
NP ∩ coNP	:	es ex. keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge für UP
UP	:	es ex. keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge für NP ∩ coNP
DisjNP	:	es ex. kein $\leq_{\mathrm{m}}^{\mathrm{pp}}$-vollständiges disjunktes NP-Paar für DisjNP
DisjCoNP	:	es ex. kein $\leq_{\mathrm{m}}^{\mathrm{pp}}$-vollständiges disjunktes coNP-Paar für DisjCoNP

Vermutung 1.1 (Q, Fenner u. a. 2003)**.** *Für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ eine akzeptierende Berechnung von $N(x)$ ist.*

2 Grundlagen

Dieses Kapitel legt die definitorischen Grundlagen für die folgenden Kapitel fest. In Abschnitt 2.1 legen wir mathematische Notationen für diese Arbeit fest. Abschnitt 2.2 spezifiziert das Maschinenmodell. Abschnitt 2.3 wiederholt einige Standarddefinitionen aus der Komplexitätstheorie. Abschnitt 2.4 setzt das hier verwendete Verständnis von Relativierungen fest. Abschließend geht Abschnitt 2.5 kurz auf Beweissysteme im Sinne von Cook und Reckhow (1979) ein.

2.1 Notation

Sei Σ das standardmäßige Alphabet mit $\Sigma = \{0, 1\}$. Elemente von Σ^* nenenn wir Wörter, sind also endliche Sequenzen von Zeichen aus Σ. Teilmengen von Σ^* nennen wir auch Sprachen. Wir bezeichnen die Länge eines Wortes $w \in \Sigma^*$ mit $|w|$. Das leere Wort bezeichnen wir mit ε. Das i-te Zeichen eines Wortes w für $0 \leq i < |w|$ identifizieren wir mit $w[i]$. Diese Notation erweitern wir auf Sequenzen von Indizes: für $0 \leq i_1, i_2, \ldots, i_k < |w|$ und $\alpha = (i_1, i_2, \ldots, i_k)$ sei $w[\alpha] = w[i_1]w[i_2]\cdots w[i_k]$. Insbesondere ist damit $w[0, 1, 2, \ldots, |w| - 1] = w$. Falls w ein (echter) Präfix von w ist dann schreiben wir $w \sqsubseteq v$ (bzw. $w \sqsubsetneq v$).

Die Menge aller natürlichen (nicht-negativen) Zahlen wird mit $\mathbb{N}$ bezeichnet. Die leere Menge notieren wir wie üblich als $\emptyset$. Die Kardinalität einer Menge A notieren wir wie üblich als $|A|$. Für eine Menge $A \subseteq \Sigma^*$ und $n \in \mathbb{N}$ definieren wir $A^{\leq n} = \{w \in A \mid |w| \leq n\}$. Analog definieren wir $A^{<n}, A^{=n}$, usw. Außerdem bezeichnet $\ell(A) = \sum_{w \in A} |w|$. Für solche Teilmengen A von Σ^* verstehen wir das Komplement $\overline{A}$ als $A - \Sigma^*$.

Relationen und Funktionen

Zweistellige bzw. binäre Relationen $R \subseteq A \times B$ können wir mit den üblichen Eigenschaften beschreiben: die Relation R ist

- *(links-)total* wenn jedes Element aus A mit mindestens einem Element aus B reliert,
- *rechtstotal* bzw. *surjektiv* wenn jedes Element aus B mit mindestens einem Element aus A reliert,
- *linkseindeutig* bzw. *injektiv* wenn jedes Element aus B mit höchstens einem Element aus A reliert,
- *(rechts-)eindeutig* bzw. *funktional* wenn jedes Element aus A mit höchstens einem Element aus B reliert,
- *bijektiv* wenn jedes Element aus A mit genau einem Element aus B reliert und umgekehrt, also genau dann wenn R funktional, surjektiv und injektiv ist.

Binäre Relationen nennen wir eine (partielle) *Funktion* wenn diese Relation funktional ist. Eine Funktion sei also im Folgenden im Allgemeinen nicht total. Sollte (Links-)Totalität explizit gefordert sein, sprechen wir von *totalen Funktionen*. Binäre Relationen über Wörtern aus Σ^*, welche nicht unbedingt Funktionen sind, verstehen wir manchmal auch aus historischen Gründen als (partielle) Multifunktionen, dem Begriff der „*partial multivalued function*“ nachempfunden.

Für eine binäre Relation $R \subseteq \Sigma^* \times \Sigma^*$ schreiben wir $\mathrm{Proj}(R)$ für die Menge $\{x \mid (x, y) \in R\}$. Für ein Wort $x \in \Sigma^*$ schreiben wir *set-*$R(x) = \{y \mid (x, y) \in R\}$ für die Bildmenge von x auf R Manchmal werden wir binäre Relationen auch über die Spezifikation der jeweiligen Bildmengen definieren, also z.B. *set-*$Q(n) = \{0, 1, \ldots, n\}$ schreiben um die Relation $Q = \{(a, b) \mid b \leq a\}$ zu definieren. Falls f eine Funktion bzw. funktional ist, meinen wir mit $f(x)$ wie üblich das *Bildelement* der Funktion f meinen, und nicht die *Bildmenge*.

Für eine Funktion f bezeichnen wir die Urbild- bzw. Bildmenge (domain und range) mit $\mathrm{dom}(f)$ und $\mathrm{ran}(f)$. (Beachte dass $\mathrm{Proj}(f) = \mathrm{dom}(f)$. Wir führen diese Unterscheidung nur wegen den Gewohnheiten dieser zwei Notationen ein.) Ist f eine Funktion, dann bezeichnen wir mit f^{-1} dessen Umkehrrelation. Beobachte dass f^{-1} funktional ist, wenn f injektiv ist. Ist f sogar bijektiv, dann ist die Umkehrfunktion f^{-1} eine totale Funktion.

Eine Funktion $f\colon \Sigma^* \to \Sigma^*$ nennen wir *verlängernd* wenn $|f(x)| \geq |x|$ für alle $x \in \mathrm{dom}(f)$. Die Funktion f nennen wir *polynomiell längenbeschräkt* wenn ein Polynom p existiert sodass $|x| \leq p(|x|)$ für alle $x \in \mathrm{dom}(f)$. Die Funktion f nennen wir *ehrlich* wenn ein Polynom q existiert sodass $q(|f(x)|) \geq |x|$ für alle $x \in \mathrm{dom}(f)$.

Beachte dass Funktionen nur spezielle Relationen sind. Wenn also f eine Funktion ist, meinen wir mit „$f \in \mathrm{P}$“ dass *der Graph* von f in Polynomialzeit entschieden werden kann

(i.e., gegeben Tupel (x, y), gilt $f(x) = y$?). Das ist eine schwächere Aussage als „$f \in \mathrm{FP}$“ die wie in üblicher Interpretation besagen soll, dass aus x das Bild $f(x)$ in Polynomialzeit berechnet werden kann.

Im Folgenden definieren wir noch den Begriff der *Verfeinerung*. Seien F, G zwei Multifunktionen. Wir nennen G eine *Verfeinerung* von F wenn $\mathrm{Proj}(F) = \mathrm{Proj}(G)$ und *set-*$G(x) \subseteq$ *set-*$F(x)$ für alle $x \in \mathrm{Proj}(F)$ (bzw. äquivalent $\in \mathrm{Proj}(G)$). Ist F eine Multifunktion, und $\mathcal{G}$ eine Klasse von Multifunktioen, schreiben wir $F \in_c \mathcal{G}$ wenn $\mathcal{G}$ eine Verfeinerung $G \in \mathcal{G}$ von F enthält. Für zwei Klassen $\mathcal{F}, \mathcal{G}$ von Multifunktionen schreiben wir $\mathcal{F} \subseteq_c \mathcal{G}$ falls für jede Multifunktion $F \in \mathcal{F}$ auch $F \in_c \mathcal{G}$ gilt.

Codierungen, Identifikation von Zahlen und Wörtern

Die endlichen Wörter Σ^* können über ihre quasi-lexikographischen Ordnung $\prec_{\mathrm{lex}}$ linear geordnet werden. Diese ist eindeutig definiert indem wir $0 \preceq_{\mathrm{lex}} 1$ fordern. Unter dieser Definition existiert ein Ordnungsisomorphismus zwischen $(\Sigma^*, \preceq_{\mathrm{lex}})$ und $(\mathbb{N}, <)$, welcher insbesondere eine Bijektion zwischen Σ^* und $\mathbb{N}$ induziert, der sowohl in Polynomialzeit berechenbar als auch invertiertbar ist. (Eine solcher Isomorphismus wird zum Beispiel durch eine dyadische Codierung realisiert.) Durch diese Identifikation können wir Wörter aus Σ^* als Zahlen aush $\mathbb{N}$ behandeln und umgekehrt. Es könenn also auch Notationen, Beziehungen und Operationen für Σ^* auf $\mathbb{N}$ übertragen werden und umgekehrt. Insbesondere können wir dann von einer Länge $|n|$ des Wortes sprechen, welches von $n \in \mathbb{N}$ repräsentiert wird. Insbesondere meint dieser Ausdruck nicht den Betrag von n. Ebenso bezeichnet die Ordnung $\leq$ sowohl die Kleiner-oder-gleich-Ordnung auf den natürlichen Zahlen als auch der quasi-lexikographischen Ordnung $\prec_{\mathrm{lex}}$ auf den endlichen Wörtern. Diese Übereinstimmung ist nach den Eigenschaften des Ordnungsisomorphismus auch kompatibel mit der Identifikation von Wörtern mit Zahlen. Beachte dass der Längenoperator $|\cdot|$ ordnungserhaltend ist: wenn $a \leq b$ (oder eben äquivalent $a \preceq_{\mathrm{lex}} b$) für zwei Wörter $a, b \in \Sigma^*$ dann ist auch $|a| \leq |b|$, bzw. ist das Wort a höchstens so lang wie das Wort b. Mit den Ausdrücken 0^n und 1^n meinen wir immer die zwei Wörter $000\cdots$ und $111\cdots$ aus Σ^n.

Wir definieren mit $\langle\cdots\rangle$ eine Paarungsfunktion von $\bigcup_{i\geq 0}(\Sigma^*)^i \to \Sigma^*$, welche injektiv und in Polynomialzeit sowohl berechenbar als auch invertierbar ist, und die im folgenden Sinne längeneffizient ist: $|\langle u_1, \ldots, u_n\rangle| = 2(|u_1| + \cdots + |u_n| + n)$. Eine solche Paarungsfunktion kann beispielsweise über $\langle u_1, \ldots, u_n\rangle \mapsto f(\#u_1\# \cdots \#u_n)$ realisiert werden, wobei f eine Codierung vom Alphabet $\{0, 1, \#\}$ auf Σ^* mittels $\{0 \mapsto 00, 1 \mapsto 11, \# \mapsto 01\}$ ist. Diese Paarungsfunktion werden wir häufig verwenden, um Tupel an Wörtern zu codieren, z.B. damit eine Turing-Maschine ein Tupel an Wörtern als Eingabe entgegen nehmen kann. Auf die konkrete Angabe dieser Paarungsfunktion wird aber im Folgenden meist verzichtet und sie wird nur implizit mitgedacht. So meinen wir mit dem Tupel (a, b) für $a, b \in \Sigma^*$ je nach Kontext entweder mathematisch präzise das Element aus dem Produkt $\Sigma^* \times \Sigma^*$, oder das Wort $\langle a, b\rangle \in \Sigma^*$. Ebenso verstehen wir je nach Kontext eine binäre Relation $R \subseteq \Sigma^* \times \Sigma^*$ auch als eine Sprache im Sinne einer Teilmenge von Σ^*, die bspw. von einer Turing-Maschine entschieden werden kann. Algorithmen und Turing-Maschinen verarbeiten nicht nur Wörter, sondern auch andere Objekte wie z.B. Graphen oder Turing-Maschinen. Daher werden wir die obige implizit mitgedachte Codierung auch auf andere Objekte ausweiten. Hierbei seien die jeweiligen Codierungen angemessen effizient, in dem Sinne dass die Codierungen kompakt sind und entsprechende Operationen auf den codierten Objekten in Polynomialzeit zulassen. Zum Beispiel lässt sich ein Graph mit Knotenmenge V und Kantenmenge E in polynomieller Länge abh. von $|V|$ und $|E|$ codieren, und auf der entsprechenden Codierung kann z.B. die Nachbarschaft eines ausgezeichneten Knotens ebenso in Polynomialzeit aufgezählt werden.

2.2 Maschinenmodell

Diese Arbeit baut auf dem Berechnungsmodell der Turing-Maschine (TM) auf. Wir betrachten hierbei sowohl die deterministische als auch die nichtdeterministische Variante. In dieser Arbeit haben TM sowohl ausgezeichnete Zustände zum Akzeptieren bzw. Ablehnen, ein Eingabeband, ein Arbeitsband, und ein Ausgabeband. Im Folgenden betrachten wir nur TM die immer terminieren. (Es ist einer TM im Allgemeinen nicht ansehbar, ob diese immer terminiert. Im Verlauf dieser Arbeit werden die TM aber so beschaffen sein, dass diese offensichtlich immer terminieren.)

Wir betrachten zunächst deterministische TM. Sei M eine deterministische TM, und x eine Eingabe. Dann induziert eine Berechnung $M(x)$ einen Rechenweg α, der in einem ausgezeichnetem Zustand q terminiert. Wir sagen dann auch, dass α *der Rechenweg von Berechnung* $M(x)$ ist. Wenn der terminierende Zustand q dieses Rechenwes α ein akzeptierender Zustand ist, dann sagen wir auch dass $M(x)$ *mit Ausgabe* y *akzeptiert* oder kurz $M(x)$ *akzeptiert* wobei y jenes Wort ist, welches auf dem Ausgabeband steht.

Eine solche deterministische TM M setzt nun gleichzeitig zwei unterschiedliche Funktionsweisen um. Einerseits die eines Akzeptors einer Menge, und andererseits die einer Funktion:

- Die von M *entschiede Sprache* ist die Menge $L(M) = \{x \in \Sigma^* \mid M(x) \text{ akzeptiert}\}$.
- Die von M *berechnete Funktion* ist die Funktion $f_M \colon \Sigma^* \to \Sigma^*$ mit

$$f_M(x) = \begin{cases} y & \text{wenn } M(x) \text{ mit Ausgabe } y \text{ akzeptiert,} \\ \bot & \text{sonst.} \end{cases}$$

Wenn wir M im Kontext der zweiten Funktionsweise verstehen, dann sprechen wir auch von einem Turing-Transduktor. Wir kürzen dann auch „die von M berechnete Funktion“ durch „die Funktion M“ ab und verstehen den Turing-Transduktor M als genuine Funktion, und schreiben dann z.B. $M(x) = y$ anstelle $f_M(x) = y$.

Diese zwei Arten von Funktionsweisen einer TM erweitern wir nun auf nichtdeterministische TM. Sei N eine nichtdeterministische TM, und x eine Eingabe. Dann induziert analog eine Berechnung $N(x)$ nicht nur eine, sondern ggf. mehrere terminierende Rechenwege, die wir ebenso die Rechenwege von Berechnung $N(x)$ nennen. Terminiert ein solcher Rechenweg von $N(x)$ in einem akzeptierenden Zustand, nennen wir diesen Rechenweg auch einen *akzeptierenden Rechenweg*. Ähnlich wie im deterministischen Fall sagen wir dass $N(x)$ *auf Rechenweg α (mit Ausgabe y) akzeptiert* wenn α ein akzeptierender Rechenweg von $N(x)$ ist (und y auf dem Eingabeband steht). Beachte dass die Angabe eines Rechenwegs zwingend notwendig ist, da zu einer Berechnung $N(x)$ ja mehrere Rechenwege mit je unterschiedlichen Akzeptierverhalten und Ausgaben existieren. Im Sinne eines existentiellen Akzeptierverhaltens sagen wir dass $N(x)$ *akzeptiert* wenn *mindestens* ein akzeptierender Rechenweg α auf $N(x)$ existiert.

Analog ergeben sich nun wieder zwei Funktionsweisen, einerseits als Akzeptor, andererseits als Multifunktion:

- Die von N *entschiede Sprache* ist die Menge

$$L(N) = \{x \in \Sigma^* \mid N(x) \text{ akzeptiert } \} = \{x \in \Sigma^* \mid \text{ ex. akz. Rechenweg auf } N(x)\}.$$

- Die von N *berechnete Mulitfunktion* ist die Multifunktion $f_N \subseteq \Sigma^* \times \Sigma^*$ mit

$$f_N(x) = \{y \mid N(x) \text{ akz. auf einem Rechenweg mit Ausgabe } y\}$$

Die berechnete Mulitfunktion kann in anderen Worten auch so verstanden werden, dass x den Ausgaben von $N(x)$ zugeordnet wird, wobei jeder akzeptierender Rechenweg eine Ausgabe macht, nämlich das Wort was auf dem Ausgabeband steht. Wie im deterministischen Fall können wir von nichtdeterministischen Turing-Transduktoren sprechen, wenn wir die zweite Funktionsweise betonen wollen. Ebenso können wir wieder abkürzend von „der Multifunktion N“ sprechen.

In sowohl dem deterministischen und nichtdeterministischen Fall können wir Berechnungen eine *Laufzeit* zuordnen: für eine TM M sei

$$\text{time}_M(x) = \max\{\text{Anz. Rechenschritte in } \alpha \mid \alpha \text{ ist ein Rechenweg von } M(x)\}.$$

Ist $\text{time}_M(x)$ durch ein Polynom in Abhängigkeit von $|x|$ beschränkt, und M eine deterministische (bzw. nichtdeterministische) TM, sagen wir auch dass M eine *deterministische (bzw. nichtdeterministische) Polyomialzeit-Turing-Maschine* (PTM bzw. NPTM) ist.

2.3 Komplexitätsklassen

Auf Basis der Turing-Maschinen als Berechnungsmodell können die üblichen Komplexitätsklassen der Entscheidungsprobleme bzw. Sprachen $\mathrm{P}, \mathrm{NP}, \mathrm{coNP}$ usw. definiert werden:

$$\begin{aligned}
\mathrm{P} &= \{L \subseteq \Sigma^* \mid \text{ex. PTM } M \text{ die } L \text{ entscheidet}\} \\
\mathrm{NP} &= \{L \subseteq \Sigma^* \mid \text{ex. NPTM } M \text{ die } L \text{ entscheidet}\} \\
\mathrm{UP} &= \{L \subseteq \Sigma^* \mid \text{ex. NPTM } M \text{ die } L \text{ entscheidet,} \\
&\qquad\qquad \text{und } M(x) \text{ akz. auf höchstens einem eindeutigen Rechenweg}\} \\
\mathrm{coNP} &= \{L \subseteq \Sigma^* \mid \overline{L} \in \mathrm{NP}\}
\end{aligned}$$

Die Einfach- und Doppelt-Exponentialzeitklassen definieren wir wie folgt:

$$\begin{aligned}
\mathrm{E} &= \{L \subseteq \Sigma^* \mid \text{ex. TM } M \text{ die } L \text{ entscheidet, und ex. } c > 0 \text{ mit } \text{time}_M(x) \leq 2^{c|x|} \text{ für alle } x\} \\
\mathrm{EE} &= \{L \subseteq \Sigma^* \mid \text{ex. TM } M \text{ die } L \text{ entscheidet, und ex. } c > 0 \text{ mit } \text{time}_M(x) \leq 2^{2^{c|x|}} \text{ für alle } x\}
\end{aligned}$$

Die nichtdeterministischen Varianten NE, NEE und Komplementklassen coNE, coNEE sind analog definiert.

Die Funktioneklassen FP, NPMV, NPSV ist analog definiert (Selman 1994):

$$\begin{aligned}
\mathrm{FP} &= \{f : \Sigma^* \to \Sigma^* \mid f \text{ ist eine Funktion und ex. PTM-Transduktor } M \text{ der } f \text{ berechnet}\} \\
\mathrm{NPSV} &= \{f : \Sigma^* \to \Sigma^* \mid f \text{ ist eine Funktion und ex. NPTM-Transduktor } M \text{ der } f \text{ berechnet}\} \\
\mathrm{NPMV} &= \{f \subseteq \Sigma^* \times \Sigma^* \mid f \text{ ist eine Multifunktion und ex. NPTM-Transduktor } M \text{ der } f \text{ berechnet}\}
\end{aligned}$$

Wir defineren NPMV_t als die Teilmenge von NPMV der Multifunktionen, die linkstotal sind. Analog NPSV_t. Ist für eine Funktion $f \in \mathrm{FP}$ auch $f^{-1} \in_\mathrm{c} \mathrm{FP}$, also eine (funktionale) Verfeinerung g von f^{-1} in FP, dann sagen wir auch, dass f *p-invertierbar* ist. Beachte, dass die Ehrlichkeit von f eine notwendige Bedingung für die p-Invertierbarkeit von f ist.

Grollmann und Selman (1988) erarbeiten in ihrer Untersuchung zu Public-Key-Kryptosystemen den Begriff von *disjunkten NP-Paaren* heraus.

Definition 2.1 (DisjNP, DisjCoNP)**.** Zwei Mengen $A, B \in \Sigma^*$ bilden ein *disjunktes NP-Paar* (A, B) falls $A, B \in \mathrm{NP}$ und $A \cap B = \emptyset$. Die Klasse aller disjunkten NP-Paare schreiben wir mit DisjNP.

Analog können wir die Klasse DisjCoNP aller disjunkten coNP-Paare definieren. $\lhd$

Intuitiv mit dieser Definition verknüpft ist das folgende Promise-Problem: gegeben eine Instanz $x \in A \cup B$, entscheide ob $x \in A$ oder $x \in B$. Das Versprechen bwz. Promise ist hierbei, dass x sicher in A oder B enthalten ist; ein entsprechender Entscheidungsalgorithmus kann sich beliebig verhalten für Eingaben $x' \notin A \cup B$.

Entspreched diesem Promise-Problem ergibt sich formal folgende Definition von „Lösbarkeit“: Wir nennen ein disjunktes NP-Paar (A, B) *P-separierbar* wenn ein Separator $S \in \mathrm{P}$ existiert sodass $A \subseteq P$ und $B \subseteq \overline{P}$.

Reduktionen

Wie üblich können wir mittels Reduktionen die Sprachen der Komplexitätsklassen ordnen. Seien A, B zwei Sprachen:

- $A \leq^\mathrm{p}_\mathrm{T} B$ wenn $A \in \mathrm{FP}^B$ (Turing- bzw. Cook-Reduzierbarkeit).
- $A \leq^\mathrm{p}_\mathrm{m} B$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (Many-one- bzw. Karp-Reduzierbarkeit).
- $A \leq^\mathrm{p}_1 B$ wenn eine injektive Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (One-one-Reduzierbarkeit).
- $A \leq^\mathrm{p}_{1,\mathrm{i}} B$ wenn eine injektive und p-invertierbare Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$.

Für die Funktionenklassen hat sich folgender sehr starke Begriff von Many-one-Reduzierbarkeit herausgebildet (Köbler und Messner 2000; Beyersdorff, Köbler und Messner 2009; Pudlák 2017). Seien g, h zwei Multifunktionen:

- $g \leq^\mathrm{p}_\mathrm{m} h$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit $\textit{set-}g(x) = \textit{set-}h(f(x))$.

Auf den Paaren aus DisjNP und DisjCoNP hat sich folgender Begriff von Reduzierbarkeit herausgebildet.[1] Seien $(A, B), (C, D)$ zwei disjunkte NP-Paare (bzw. zwei disjunkte coNP-Paare):

- $(A, B) \leq^\mathrm{pp}_\mathrm{m} (C, D)$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit $f(A) \subseteq B$ und $f(B) \subseteq C$.

1. Vgl. insb. Glaßer, Selman, Sengupta und Zhang (2004) und Glaßer, Selman und Sengupta (2005) für eine ausführlichen Vergleich und Diskussion Reduktions- und Vollständigkeitsbegriffen. Insgesamt zeigen die Arbeiten, dass dieser schwache Begriff von Reduktion geeignet gewählt ist, denn er ist insbesondere äquivalent zu alternativen stärker wirkenden Reduktionsbegriffen ist.

Jede dieser Ordnungsrelationen ist eine Quasiordnung, i.e. reflexiv und transitiv. Beachte, dass (auf Mengen) $\leq^\mathrm{p}_{1,\mathrm{i}}$ feiner als $\leq^\mathrm{p}_1$ ist, und diese feiner als $\leq^\mathrm{p}_\mathrm{m}$, und diese feiner als $\leq^\mathrm{p}_\mathrm{T}$ ist. Beachte auch, dass P und NP auf $\leq^\mathrm{p}_\mathrm{m}$ (und $\leq^\mathrm{p}_\mathrm{T}$) nach unten abgeschlossen sind. Ebenso ist FP auf $\leq^\mathrm{p}_\mathrm{m}$ nach unten abgeschlossen und die p-separierbaren Paare auf $\leq^\mathrm{pp}_\mathrm{m}$ nach unten abgechlossen:

$$\begin{aligned}
&A \leq^\mathrm{p}_\mathrm{m} B \text{ und } B \in \mathrm{NP} && \implies A \in \mathrm{NP} \\
&A \leq^\mathrm{p}_\mathrm{m} B \text{ und } B \in \mathrm{P} && \implies A \in \mathrm{P} \\
&g \leq^\mathrm{p}_\mathrm{m} h \text{ und } h \in_\mathrm{c} \mathrm{FP} && \implies g \in_\mathrm{c} \mathrm{FP} \\
&(A, B) \leq^\mathrm{pp}_\mathrm{m} (C, D) \text{ und } (C, D) \text{ ist p-sep.} && \implies (A, B) \text{ ist p-sep.}
\end{aligned}$$

Sei $\mathcal{C}$ eine Komplexitätsklasse und $\preceq$ eine der obigen Reduktionsordnungen. Wie üblich nennen wir nun eine Sprache A *$\preceq$-hart für $\mathcal{C}$* wenn A eine obere Schranke für $\mathcal{C}$ geordnet über $\preceq$ ist (d.h. $B \preceq A$ für alle $B \in \mathcal{C}$). Wir nennen A *$\preceq$-vollständig für $\mathcal{C}$* wenn $A \in \mathcal{C}$ ein

größtes Element von $\mathcal{C}$ geordnet über $\preceq$ ist (d.h. $B \preceq A$ für alle $B \in \mathcal{C}$ und $A \in \mathcal{C}$). Auf Grundlage der Existenz universeller effizienter Turing-Maschinen können für die Klassen P und NP jeweils eine kanonische $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge angegeben werden. Für NP ist diese

Definition 2.2.

$\texttt{KAN} = \{(N, x, 1^n) \mid N \text{ ist eine NTM und akz. } x \text{ auf einem RW mit} \leq n \text{ vielen Schritten}\}.\lhd$

Lemma 2.3. *Die Menge* KAN *ist* $\leq^{\mathrm{p}}_{1,\mathrm{i}}$*-vollständig.*

Beweis. Sei $A \in$ NP. Wir wollen zeigen dass $A \leq^{\mathrm{p}}_{1,\mathrm{i}}$ KAN. Sei hierfür N eine NPTM welche A entscheidet. Es gibt also auch ein Polynom p welches die Laufzeit von N beschränkt. Definiere nun die Funktion $f(x) = (N, x, 1^{p(|x|)})$. Es gilt nun

$$x \in A \iff N(x) \text{ akz. auf RW mit} \leq p(|x|) \text{ Schritten}$$
$$\iff (N, x, 1^{p(|x|)}) \in \texttt{KAN} \iff f(x) \in \texttt{KAN}.$$

Ferner ist leicht zu sehen, dass $f \in$ FP, dass f injektiv und auch p-invertierbar ist. □

Polyomialzeit-Isomorphie

Auf Mengen erzeugen die obigen Reduktionsordungen je eine kanonische Äquivalenzordung („Duplikatrelation“):

- $A \equiv^{\mathrm{p}}_{\mathrm{m}} B \iff A \leq^{\mathrm{p}}_{\mathrm{m}} B$ und $B \leq^{\mathrm{p}}_{\mathrm{m}} A$.
- $A \equiv^{\mathrm{p}}_{1} B \iff A \leq^{\mathrm{p}}_{1} B$ und $B \leq^{\mathrm{p}}_{1} A$.
- $A \equiv^{\mathrm{p}}_{1,\mathrm{i}} B \iff A \leq^{\mathrm{p}}_{1,i} B$ und $B \leq^{\mathrm{p}}_{1,i} A$.

Wir definieren nun auch noch die *p-Isomorphie* als eine Verfeinerung von $\equiv^{\mathrm{p}}_{1,\mathrm{i}}$:

- $A \equiv^{\mathrm{p}} B \iff$ es existiert eine bijektive und p-invertierbare Funktion $f \in$ FP mit $x \in A \leftrightarrow f(x) \in B$.

Gilt $A \equiv^{\mathrm{p}} B$ dann sagen wir auch dass A und B *p-isomorph* sind. Im Folgenden werden noch die wichtigsten bekannten Aussagen bezüglich p-Isomorphie zusammengefasst:

Hartmanis und Berman (1976) zeigen dass aus $\equiv^{\mathrm{p}}_{1,\mathrm{i}}$-äquivalente Sprachen dann p-isomorph sind, wenn die jeweiligen Reduktionsfunktionen verlängernd sind.

Satz 2.4 (Hartmanis und Berman 1976)**.** *Gilt $A \leq^{\mathrm{p}}_{1,\mathrm{i}} B$ via f und $B \leq^{\mathrm{p}}_{1,\mathrm{i}} A$ via g, und f und g sind verlängernd, dann gilt $A \equiv^{\mathrm{p}} B$.*

Um die Voraussetzungen vom vorigen Satz 2.4 zu vereinfachen, führen sie den Begriff der *paddability* ein.

Definition 2.5. Eine Sprache $A \neq \emptyset$ heißt (Berman–Hartmanis-)*paddable* genau dann wenn eine injektive und p-invertierbare Funktion $g \in FP$ existiert sodass für alle $x, y \in \Sigma^*$ gilt:

$$x \in A \iff g(x, y) \in A.$$

Das heißt, g fügt einen beliebigen String y zur „Probleminstanz“ x hinzu, sodass die Mitgliedschaft zu A unverändert bleibt, und die beiden originalen Strings x und y wieder rekonstruiert werden können. $\lhd$

Es gilt:

Satz 2.6. *(1) Ist A paddable so gibt es für jedes B mit $B \leq^{\mathrm{p}}_{\mathrm{m}} A$ eine injektive p-invertierbare verlängernde Funktion die $B \leq^{\mathrm{p}}_{1,\mathrm{i}}$ realisiert.*

(2) Sind A, B paddable, so folgt aus $A \equiv^{\mathrm{p}}_{\mathrm{m}}$ stets $A \equiv_{\mathrm{p}}$.

Alle bekannten $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen für NP sind zueinander p-isomorph. Berman und Hartmanis vermuteten, dass das für *alle* $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen gilt:

Vermutung 2.7 (IC)**.** *Alle $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen für* NP *sind p-isomorph. In anderen Worten: die $\leq^{\mathrm{p}}_{\mathrm{m}}$-Äquivalenzklasse der vollständigen Mengen ist gleich der $\equiv^{\mathrm{p}}$-Äquivalenzklasse von* SAT*.*

Mit obigem Begriff von Paddability lässt sich die $\equiv^{\mathrm{p}}$-Äquivalenzklasse von SAT folgendermaßen charakterisieren:

Satz 2.8. *Eine Menge $A \in$ NP ist genau dann p-isomorph zu* SAT *wenn $A \leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig und und paddable ist.*

Als Konsequenz ergibt sich hieraus, dass die *bekannten* $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen alle paddable sind. (Das ist die eigentliche empirische Beobachtung von Berman und Hartmanis, auf welcher diese IC vermuteten.)

2.4 Orakel und Relativierungen

Wie in der Einleitung schon angesprochen, ist die Idee hinter Orakel-Berechnungen zu untersuchen, welche Probleme B effizient(er) durch einen Algorithmus gelöst werden können, wenn die Algorithmen eine (fiktive) Möglichkeit haben, ein (ggf. sehr schweres) Problem A ohne Rechenaufwand zu lösen. Der Zugriff auf A kann also wie ein „Nachschlagewerk“ eine „Blackbox-Funktion“ verstanden werden, die auf magische Weise A augenblicklich löst.

Diese Idee wird im Berechnungsmodell der Orakel-Turing-Maschine (OTM) formalisiert. Orakel-Turing-Maschinen sind eine Erweiterung der (deterministischen und nichtdeterministischen) Turing-Maschinen, die zum Eingabe-, Arbeits- und Ausgabeband auch noch ein separates Orakelband haben. Ferner existieren drei ausgezeichnete Zustände $q_?, q_{\text{yes}}, y_{\text{no}}$.

Gegeben ein Orakel $A \subseteq \Sigma^*$ können OTM nun Fragen der Form $x \stackrel{?}{\in} A$ an das Orakel stellen, indem sie ein Wort x auf das Frageband schreiben, und in den Zustand $q_?$ übergeht. Im unmittelbar nächsten deterministischen Schritt der Berechnung wird der Zustand q_{yes} eingenommen falls $x \in A$, sonst den Zustand q_{no}.

Aus dieser Beschreibung wird klar, dass eine Berechnung einer OTM sowohl von der Eingabe x abhängig ist, als auch vom Orakel A, *relativ* zu diesem $M(x)$ rechnet. Wir schreiben dann auch kurz M^A wenn wir die OTM M mit festem Orakel A meinen, und $M^A(x)$ die Berechnung der OTM M auf Eingabe x mit Orakel A. Entsprechend können wir auch die Laufzeit $\text{time}_M^A(x)$ definieren, und von (deterministischen bzw. nichtdeterministischen) Polynomialzeit-Orakel-Turing-Maschinen (POTM, NPOTM) sprechen, wenn die Laufzeit auf allen Eigaben und allen Orkalen polynomiell durch die Eingabelänge beschränkt ist.

Wir können nun die relativierten Komplexitätsklassen $\text{P}^O, \text{NP}^O, \text{FP}^O, \text{NPMV}^O, \ldots$ relativ zu einem gegebenen Orakel O definieren, wobei in der jeweiligen Definition die TM mit OTM ersetzt werden, die Zugriff auf das Orakel O haben. Diese Relativierung überträgt sich auch auf unsere weiteren Definitionen, wie z.B. Reduktion und Vollständigkeit. Wir schreiben z.B. $A \leq_{\text{m}}^{\text{p},O} B$ wenn eine Funktion $f \in \text{FP}^O$ existiert mit $x \in A \leftrightarrow f(x) \in B$. Die kanonische NP-vollständige Menge $\texttt{KAN}$ kann ebenso zu $\texttt{KAN}^O$ relativiert werden. Für *natürliche* Mengen wie $\texttt{SAT}$ usw. werden wir dagegen keine relativierte Variante definieren. In diesem Sinne ist $\texttt{SAT}$ im Allgemeinen *nicht* $\leq_{\text{m}}^{\text{p},O}$-vollständig, in dem Sinne dass ein Orakel O existiert und eine Menge $A \in \text{NP}^O$ sodass $A \not\leq_{\text{m}}^{\text{p},O} \texttt{SAT}$.

In allgemeineren Beweisen, die nicht konkrete natürliche Mengen betreffen, lassen sich üblicherweise alle beteiligten TM mit OTM auswechseln, ohne die Gültigkeit der Aussage zu verändern. Aussagen bzw. Beweise, die in solchen relativierten Umgebungen relativ zu jedem beliebigen Orakel O gelten, nennen wir *relativierende* Aussagen bzw. Beweise. Das Diagonalargument in einem typischen Beweis von $\text{P} \subsetneq \text{E}$ relativiert beispielsweise, sodass auch $\text{P}^O \subsetneq \text{E}^O$ für jedes beliebige Orakel O gilt. Ebenso relativiert die Aussage „$\texttt{KAN} \in \text{NP}$ ist $\leq_{\text{m}}^{\text{p}}$-vollständig“ (zu „$\texttt{KAN}^O \in \text{NP}^O$ ist $\leq_{\text{m}}^{\text{p},O}$-vollständig“).

Im Folgenden soll jede Aussage als relativierbar verstanden werden, es sei denn es wird auf die Nichtrelativierbarkeit hingewiesen, oder von konkreten natürlichen Mengen gesprochen, welche ohnehin nicht relativieren.

2.5 Beweissysteme

Beweissysteme wurden in der Einleitung schon kurz definiert. In diesem Abschnitt wird die präzise Definition von Cook und Reckhow (1979) wiedergegeben.

Definition 2.9. Eine Funktion $f \in \text{FP}$ ist ein *Beweissystem* für L wenn $\text{ran}(f) = L$. Ist $f(w) = x$ schreiben wir auch, dass w ein *f-Beweis* für x ist.

Existiert zudem ein Polynom q sodass für jedes $x \in L$ ein f-Beweis x der Länge $\leq q(|x|)$ existiert, sagen wir dass f *kurzen Beweise* hat. $\lhd$

Hieraus stellt sich die erste Frage, welche Mengen Beweissysteme mit kurzen Beweisen haben.

Ein einfaches Beweissystem für die Menge $\texttt{SAT} \in \text{NP}$ wäre z.B. das *Standardbeweissystem* *std* für $\texttt{SAT}$:

$$std(\varphi, w) = \begin{cases} \varphi & \text{wenn } w \text{ eine erfüllende Belegung für } \varphi \text{ ist,} \\ \bot & \text{sonst.} \end{cases}$$

Dieses Beweissystem hat kurze Beweise.

Cook und Reckhow machen dagegen die Beobachtung im Fall von der Menge $\texttt{TAUT}$ der aussagenlogischen Tautologien die Beobachtung, dass $\texttt{TAUT}$ genau dann kein Beweissystem mit kurzen Beweisen hat, wenn $\text{NP} \neq \text{coNP}$. Diese Einsicht motivierte das sogenannte *Cook–Reckhow-Programm*: man nähert sich der Frage $\text{NP} \neq \text{coNP}$ mittels Untersuchung immer stärerer Beweissysteme. Um nun die relative Stärke unterschiedlicher Beweissysteme zu vergleichen, führen Cook und Reckhow den Begriff der Simulation ein.

Definition 2.10. Seien f, g zwei Beweissysteme für L. Wir sagen dass f das Beweissystem g *simuliert* wenn eine (nicht notwendigerweise effiziente) polynomiell längenbeschränkte Funktion π existiert sodass

$$f(\pi(w)) = g(w).$$

Heißt, für jeden g-Beweis w für x existiert auch ein f-Beweis $\pi(w)$ für das gleiche x, und dieser f-Beweis $\pi(w)$ ist nur polynomiell länger als w.

Ist zusätzlich $\pi \in \mathrm{FP}$, dann ist sagen wir, dass f das Beweissystem g *p-simuliert*. Das ist äquivalent zur Aussage $g \leq_{\mathrm{m}}^{\mathrm{p}} f$. ◁

Wenn f das Beweissystem g p-simuliert, kürzen wir das entsprechend der Beobachtung in der Definition auch kurz mit $g \leq_{\mathrm{m}}^{\mathrm{p}} f$ ab.

Beobachte dass Beweissysteme mit kurzen Beweisen unter Simulation abgeschlossen sind: wenn Beweissystem f das Beweissystem g simuliert, und g kurze Beweise hat, dann hat auch f kurze Beweise. Auf ähnliche Beweise sind die Beweissysteme unter p-Simulation abgeschlossen, die einfach auffindbare Beweise haben: wenn Beweissystem f für L das Beweissystem g p-simuliert, und für jedes $x \in L$ in Polynomialzeit ein g-Beweis w gefunden werden kann, dann kann auch ein f-Beweis in Polynomialzeit gefunden werden.

Allgemein generiert die Relation der (p-)Simulation wieder eine Quasiordnung, die nach der Existenz von größten Elementen untersucht werden kann. Hieraus ergibt sich der Begriff der (p-)Optimalität.

Definition 2.11. Ein Beweissystem f für L ist *(p-)optimal* wenn es jedes Beweissyem g für L (p-)simulieren kann.

Die p-Optimalität von f ist äquivalent zur $\leq_{\mathrm{m}}^{\mathrm{p}}$-Vollständigkeit von f für die Teilmenge der Funktionen aus FP mit Bildmenge L. ◁

Die beiden Definition relativieren natürlicherweise, und wir können so z.B. von p^O-optimalen Beweissystemen sprechen.

Es ist leicht zu sehen, dass jedes Beweissystem mit kurzen Beweisen auch optimal ist. Im Zusammenhang mit dem Cook–Reckhow-Programm weisen Krajíček und Pudlák (1989) darauf hin, dass die Existenz eines optimalen Beweissystems für TAUT wahrscheinlich schwächer ist, als die Existenz eines Beweissystems mit kurzen Beweisen, denn Ersteres folgt schon aus NE = coNE. (Köbler, Messner und Torán, 2003, schwächen die Voraussetzung auf NEE = coNEE ab.)

Für die Mengen aus P bzw. NP existieren p-optimale bzw. optimale Beweissysteme:

Beobachtung 2.12. *(1) Ist $A \in \mathrm{P}$, dann existiert ein p-optimales Beweissystem für A mit kurzen Beweisen.*

(2) Ist $A \in \mathrm{NP}$, dann existiert ein optimales Beweissystem für A mit kurzen Beweisen.

Beweis. 1. Zu (1): Betrachte die Funktion

$$h(x) = \begin{cases} x & \text{wenn } x \in A \\ \bot & \text{sonst} \end{cases}.$$

Diese Funktion ist definitiv ein Beweissystem für A. Sie ist in FP, ist ja der Test „$x \in A$“ in Polynomialzeit möglich. Klar ist auch dass h kurze Beweise hat. Dieses Beweissystem ist p-optimal, denn wenn g ein weiteres Beweissystem für A ist, und wenn w ein g-Beweis für x ist, dann ist $h(g(w)) = h(x) = x$, also $g(w)$ ein h-Beweis für x, wie gewünscht.

2. Zu (2): Kann durch die bekannte Zertifikats-Definition von NP gezeigt werden (was im späteren Teil der Arbeit auch geschieht), zur Vollständigkeit lässt sich an dieser Stelle aber auch ein Beweis über die NPTM-Definition von oben angeben. Sei N eine NPTM, die A mit polynomieller Laufzeit p entscheidet. Definere nun

$$h(x, \alpha) = \begin{cases} x & N(x) \text{ akz. auf Rechenweg } \alpha \\ \bot & \text{sonst} \end{cases}$$

Nachdem $L(N) = A$ ist h definitiv ein Beweissystem für A. Es ist leicht zu sehen dass $h \in \mathrm{FP}$, ist der Test „α ist ein gültiger Rechenweg und akzeptiert“ in Polynomialzeit möglich. Ferner existiert für jedes $x \in A$ auch ein akzeptierender Rechenweg α auf $N(x)$ der Länge $\leq p(|x|)$, womit der Beweis (x, α) auch nur polynomiell länger als x ist. Da Beweissystem h hat also kurze Beweise, und ist damit auch optimal. □

Insbesondere mit dem letzten Punkt können die optimalen Beweissysteme der Mengen aus NP auch als Beweissysteme mit kurzen Beweisen charakterisiert werden:

Beobachtung 2.13. *Sei $A \in \mathrm{NP}$ und f ein Beweissystem für A. Das Beweissystem f ist optimal genau dann wenn jedes $x \in A$ einen f-Beweis der Länge $\leq q(|x|)$ hat, wobei q ein Polynom ist.*

Beweis. Richtung von rechts nach links klar, das gilt schon im Allgemeinen Fall. Für die andere Richtung sei f ein optimales Beweissystem für A. Dann muss f auch das Beweissystem h mit kurzen Beweisen aus voriger Beobachtung simulieren. Für jeden kurzen h-Beweis w für x existiert dann ein höchstens polynomiell längerer f-Beweis $\pi(w)$. □

Die Existenz eines (p-)optimalen Beweissystems für eine Menge L ist eine Eigenschaft, die sich bzgl. $\leq^{\mathrm{p}}_{\mathrm{m}}$ nach unten überträgt:

Lemma 2.14 (Messner 2000, Thm. 3.2)**.** *Hat A ein (p-)optimales Beweissystem und $B \leq^{\mathrm{p}}_{\mathrm{m}} A$, dann hat auch B ein (p-)optimales Beweissystem.*

Beweis. Sei f die Reduktionsfunktion, die $B \leq^{\mathrm{p}}_{\mathrm{m}} A$ realisiert, und sei h ein p-optimales Beweissystem für A. Definere

$$h'(x,w) = \begin{cases} x & \text{wenn } h(w) = f(x), \text{ also } w \text{ ein } h\text{-Beweis für } f(x) \text{ ist,} \\ \bot & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen dass h' ein Beweissystem für B ist.

Wir zeigen nun dass h' auch p-optimal ist. Sei hierfür g' ein Beweissystem für B. Wir definieren nun

$$g(y) = \begin{cases} f(g'(w)) & \text{wenn } y = 1w, \\ h(w) & \text{wenn } y = 0w. \end{cases}$$

Es ist leicht zu sehen dass g ein Beweissystem für A ist. Dann kann h auch das Beweissystem g via $\pi \in \mathrm{FP}$ p-simulieren. Beachte dass $1w$ ein g-Beweis für $f(g'(w))$ ist, und damit $\pi(1w)$ ein h-Beweis für $f(g'(w))$ ist. Damit

$$h'(\underbrace{g'(w), \pi(1w)}_{\pi'(w)}) = g'(w).$$

bzw. kann jeder g'-Beweis w für x in einen h'-Beweis $\pi'(w)$ übersetzt werden.

Der Beweis für die Aussage mit optimalen Beweissystemen läuft ähnlich. □

Beachte insbesondere dass dieser Beweis relativiert. In Kombination mit vollständigen Mengen erhalten wir hieraus folgendes Korollar:

Korollar 2.15. *Folgende Aussagen sind äquivalent:*

(1) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge A für NP*, für welche ein (p-)optimales Beweissystem h existiert. (Das ist die Aussage* $\neg\mathsf{SAT}$*.)*

(2) Für jede Menge $B \in$ NP existiert eine (p-)optimales Beweissystem.

Analoge Äquivalenzen gelten für coNP*.*

Das erklärt auch die Form, in der wir in der Einleitung die Hypothese $\mathsf{SAT}, \mathsf{TAUT}$ gewählt haben. Historisch sagte die Hypthese TAUT aus, dass kein p-optimales aussagenlogische Beweissystem (i.e. für die coNP-vollständige Menge TAUT) existiert. Mit vorigem Korrolar ist klar, dass diese Aussage äquivalent zu unserer hier gewählten Definition von TAUT ist: keine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge A für NP hat ein p-optimales Beweissystem.

Das hat vor allem den Vorteil, dass $\mathsf{SAT}, \mathsf{TAUT}, \mathsf{CON}^{\mathsf{N}}$ auf natürliche Weise relativieren. So relativiert beispielsweise SAT auf Orakel O zur Aussage „kein $\leq^{\mathrm{p},O}_{\mathrm{m}}$-vollständiges $L \in \mathrm{NP}^O$ hat ein p^O-optimales Beweissystem $f \in \mathrm{FP}^O$“. Das entspricht genau der Form von Relativierung, welche als erstes von Dose (2020) vorgeschlagen wurde.

3 Zur Konzeptionalisierung und Ordnung von Suchproblemen

TODO: Einleitung und Übersicht über das Kapitel. Was passiert hier?

3.1 Definition von Suchproblemen

Wir geben hier noch einmal die Definition von Suchproblemen wieder, welche schon in der Einleitung erarbeitet wurde. Als Suchprobleme verstehen wir das algorithmische Problem, gegeben eine Probleminstanz x, eine entsprechende positive Lösungsinstanz y zu berechnen, oder negativ abzulehnen. Hier noch einmal ein Beispiel aus der Einleitung: gegeben eine aussagenlogische Formel φ, berechne entweder eine Belegung y welche φ erfüllt, oder gebe „unerfüllbar“ aus. Die wesentliche Einschränkung, welche wir auch schon in der Einleitung festgelegt haben, ist die Einschränkung auf „NP-Suchprobleme“. Wir meinen damit, dass

- die Lösungen nur polynomiell länger als die Probleminstanzen sind, und
- effizient in Polynomialzeit verifiziert werden kann, ob zu einer gegebenen Probleminstanz x ein beliebiges Wort y tatsächlich eine (positive) Lösung im Sinne des Suchproblems darstellt oder nicht.

(Wir fordern im Übrigen nicht, dass negatives Ablehnen effizient verifiziert werden kann.) Um das Beispiel wieder aufzugreifen: Zum einen haben Formeln φ, welche überhaupt erfüllbar sind, eine erfüllende Belegung in Länge von φ. Zum anderen kann effizient geprüft werden, ob y tatsächlich eine erfüllbare Belegung von φ ist.

Diese Einschränkung wird durch die empirische Einsicht gestützt, dass viele natürliche Suchprobleme, für die momentan kein effizienter Algorithmus bekannt ist, genau in eine solche Einschränkung fallen. Also Suchprobleme, die „verifizierbar“ sind und „kurze Lösungen“ haben. Einige weitere Beispiele werden wir im Folgenden noch betrachten.

Zunächst werden wir die beiden obigen Punkte noch einmal in eine formale Definition gießen:

Definition 3.1 (NP-Relation, FNP)**.** Eine *NP-Relation* ist eine zweistellige Relation $R \subseteq \Sigma^* \times \Sigma^*$, sodass diese

(1) in Polynomialzeit entscheidbar ist, d.h. $R \in \mathrm{P}$, und

(2) p-balaciert ist, d.h. es existiert ein Polynom q, sodass

$$(x, y) \in R \implies |y| \leq q(|x|) \quad \text{für alle } x, y \in \Sigma^*. \tag{3.1}$$

Die Wörter der ersten Komponente nennen wir *Probleminstanzen* oder *Instanzen* oder *Probleme* von R, die Wörter der zweiten Komponente nennen wir die *Zertifikate* (oder manchmal *Lösungen*) von R. Wir sagen dann für $(x, y) \in R$, dass y *ein Zertifikat für* x ist. In diesem Sinne sagt (3.1) aus, dass Zertifikate y für x nicht superpolynomiell länger als x sein dürfen. Das Polynom q nennen wir auch die *Zertifikatsschranke* zu R.

Wir schreiben FNP für die Klasse aller NP-Relationen. ◁

Das oben diskutierte Suchproblem zu einer NP-Relation R kann jetzt wie folgt formal formuliert werden:

> Suchproblem zur Relation R:
>
> **Gegeben:** Instanz x.
>
> **Gesucht:** Zertifikat y mit $(x, y) \in R$ falls ein solches y überhaupt existiert, sonst „keine Lösung“ ausgeben.

Zur Erinnerung:

$$\mathrm{Proj}(R) = \{x \mid \exists y \in \Sigma^*, (x, y) \in R\} \in \mathrm{NP}.$$

Die Menge $\mathrm{Proj}(R)$ ist also die Menge der Probleminstanzen, für welche ein zugehöriges Zertifikat existiert; damit entspricht $\mathrm{Proj}(R)$ derjenigen Menge, die üblicherweise bei algorithmischen Entscheidungsproblemen betrachtet wird. Um die beiden Varianten noch einmal gegenüberzustellen: das entsprechende Entscheidungsproblem einer Relation R lautet

Entscheidungsproblem zur Relation R:

Gegeben: Instanz x.

Gesucht: Akzeptieren falls ein Zertifikat y mit $(x, y) \in R$ existiert, sonst ablehnen.

Das entspricht also dem Entscheiden der Sprache $\text{Proj}(R)$. Damit wird auch klar, dass das entsprechende Entscheidungsproblem bzw. die Sprache $\text{Proj}(R)$ nicht von der konkreten Relation R abhängig ist. Vielmehr: es existieren zur Sprache L ggf. unendlich viele NP-Relationen R mit $\text{Proj}(R) = L$. Für eine Sprache L sagen wir dann auch, dass R *eine NP-Relation für* L ist.

Die Zugehörigkeit des entsprechenden Suchproblems zu NP folgt hierbei unmittelbar aus der Definition von NP-Relationen. (Rate nichtdeterministisch ein Zertifikat und akzeptiere wenn dieses korrekt ist.) Im nächsten Abschnitt wird die Beziehung zwischen Suchproblemen bzw. NP-Relationen einerseits, und Entscheidungsproblemen bzw. Mengen aus NP andererseits, weiter behandelt. Festhalten können wir hier aber schon, dass das Suchproblem offenbar „schwieriger“ ist als das alleinige Entscheidungsproblem.

Im Folgenden werden eine Beispiele von natürlichen NP-Relationen angegeben. Um diese von den aonst üblicherweise verwendeten Labels für Mengen bzw. Suchprobleme abzugrenzen, sind im Verlauf dieser Arbeit *NP-Relationen* zu natürlichen Suchproblemen immer mit einem `r` am Anfang gekennzeichnet.

- $\texttt{rPERFECTMATCHING} = \{(G, M) \mid G \text{ ist ein Graph, } M \text{ ein perfektes Matching auf } G\}$.
- $\texttt{rSAT} = \{(\varphi, w) \mid \varphi \text{ ist eine aussagenlogische Formel, } w \text{ erfüllende Belegung für } \varphi\}$.
- $\texttt{rVC} = \{((G, k), C) \mid G \text{ ist ein Graph, } C \text{ eine Knotenüberdeckung, und } |C| \leq k\}$.
- $\texttt{rHAMCYCLE} = \{(G, P) \mid G \text{ ist ein Graph, } P \text{ ein Zyklus der jeden Knoten genau einmal berührt}\}$.
- $\texttt{rANOTHERHAMCYCLE} = \{((G, P), P') \mid G \text{ ist ein Graph, } P, P' \text{ je ein Zyklus der jeden Knoten genau einmal berührt, } P \neq P'\}$.
- $\texttt{rFACTORIZATION} = \{(n, (p_1, p_2, \ldots, p_k)) \mid n \in \mathbb{N}, n > 1, \text{ alle } p_i \text{ Primzahlen ungleich 2 oder } n, \text{ und } n = p_1 \cdots p_k\}$.
- $\texttt{rFACTOR} = \{(n, p) \mid n \in \mathbb{N} \text{ ist nicht prim, und } p \text{ ist ein nichttrivialer Faktor von } n\}$.
- $\texttt{rSMALLFACTOR} = \{((n, a), p) \mid n \in \mathbb{N} \text{ ist nicht prim, und } p \text{ ist ein nichttrivialer Faktor von } n \text{ und } p \leq a\}$.
- $\texttt{rGI} = \{((G, H), \sigma) \mid G, H \text{ sind Graphen mit gleicher Knotenmenge, und } \sigma \text{ ist ein Graphisomorphismus von } G \text{ nach } H\}$.

Es ist leicht zu sehen, dass jede dieser relationen auch eine NP-Relation ist. Beachte dass die Menge der Primzahlen in Polynomialzeit etnscheidbar ist Agrawal, Kayal und Saxena (2004). Bei jeder der obigen natürlichen Relationen gilt, dass die Projektion auch der üblichen Sprache aus NP entspricht. Wir haben z.B.

$$\text{Proj}(\texttt{rVC}) = \{(G, k) \mid \text{ex. Knotenüberdeckung } C \text{ von Graph } G \text{ mit } |C| \leq k\}.$$

Die Definition Suchproblemen als NP-*Relationen* lässt es zu, Suchprobleme bzw. NP-Relationen als „partielle Multifunktionen“ zu verstehen. Selman (1994) definiert in seiner Taxonomie der Funktionsklassen die Klasse NPMV_g als die Klasse derjenigen Multifunktionen $f \in \text{NPMV}$, für die (der Graph) f in P liegt. Es lässt sich leicht sehen, dass die hier definierte Klasse FNP identisch zu Selman definierten Klasse NPMV_g ist, solange man Multifuktionen mit binäre Relationen identifiziert.

Mit dieser Perspektivierung ist auch einfach zu definieren, was mit „Suchproblem lösen“ gemeint ist. Wir machen hierbei Gebrauch von Verfeinerungen (von Multifunktionen). Wir sagen, dass das Suchproblem zur NP-Relation R *in Polynomialzeit lösbar ist*, wenn $R \in_\text{c} \text{FP}$. Diese Aussage bedeutet ja, dass eine Verfeinerung f von R existiert, und f ist dabei eine (partielle) Funktion. Für eine Eingabeinstanz x wird also entweder $f(x)$ einen Wert y ausgeben für den $y \in \textit{set-}R(x)$ gilt, bzw. in anderen Worten, ein y für das $(x, y) \in R$ und damit ist die Ausgabe y eine Lösung für x. Oder, falls $f(x)$ ablehnt, dann ist $x \notin \text{dom}(f) = \text{Proj}(R)$, heißt „$f(x)$ lehnt ab“ bedeutet dass x keine Lösung hat.

Wir können damit auch schon die obige intuitive Aussage beweisen, dass das Suchproblem „schwieriger“ ist als das entsprechende Entscheidungsproblem, in dem Sinne dass sich das Entscheidungsproblem auf das Suchproblem reduzieren lässt:

Beobachtung 3.2. *Sei R eine NP-Relation. Falls $R \in_\text{c} \text{FP}$, dann gilt* $\text{Proj}(R) \in \text{P}$.

Beweis. Sei $f \in \mathrm{FP}$ die Verfeinerung von R nach Voraussetzung. Teste ob $f(x) \neq \perp$. Falls ja, dann ist $f(x) \in$ *set-*$R(x)$ und damit hat x eine Lösung; akzeptiere. Falls nicht, dann ist $x \notin \mathrm{dom}(f) = \mathrm{Proj}(x)$; lehne ab. □

Der aktuelle Stand zur Lösbarkeit der oben gennanten natürlichen Suchprobleme ist:

- $\mathtt{rPERFECTMATCHING} \in_{\mathrm{c}} \mathrm{FP}$.
- $\mathrm{NP} = \mathrm{P} \iff \mathtt{rSAT} \in_{\mathrm{c}} \mathrm{FP} \iff \mathtt{rVC} \in_{\mathrm{c}} \mathrm{FP} \iff \mathtt{rHAMCYCLE} \in_{\mathrm{c}} \mathrm{FP} \iff \mathtt{rANOTHERHAMCYCLE} \in_{\mathrm{c}} \mathrm{FP}$.
- Unklar ob $\mathtt{rSMALLFACTOR}, \mathtt{rFACTOR}, \mathtt{rFACTORIZATION} \overset{?}{\in}_{\mathrm{c}} \mathrm{FP}$. Wir haben aber $\mathrm{UP} \cap \mathrm{coUP} = \mathrm{P} \implies \mathtt{rSMALLFACTOR} \in_{\mathrm{c}} \mathrm{FP} \iff \mathtt{rFACTOR} \in_{\mathrm{c}} \mathrm{FP} \iff \mathtt{rFACTORIZATION} \in_{\mathrm{c}} \mathrm{FP}$.
- Unklar ob $\mathtt{rGI} \overset{?}{\in}_{\mathrm{c}} \mathrm{FP}$.

Bevor nun im nächsten Abschnitt die Suchprobleme den Entscheidungsproblemen näher gegenübergestellt werden, schließen wir diesen Abscnhitt noch mit einer kurzen Diskussion zu *totalen* Suchchproblemen ab.

Totale NP-Suchprobleme

Die oben formulierte Definition von FNP ist genau diejenige, die von Megiddo und Papadimitriou (1991) als erstes in dieser Form und Bezeichnung definiert wurde. Ihre Motivation war, hierbei insbesondere die *totalen* Suchprobleme in den Blick zu nehmen. Also solche Suchprobleme, bei der jede Proleminstanz immer mindestens ein Zertifikat bzw. Lösung hat. Die Faktorisierung ist beispielsweise ein solches totales Suchproblem, da ja jede natürliche Zahl sich faktorisieren lässt.

Das sind – entsprechend dieser Definition von FNP bzw. Konzeptionalisierung von Suchproblemen – genau jene NP-Relationen welche (links-)total sind: für jedes $x \in \Sigma^*$ existiert ein $y \in \Sigma^*$ mit $(x, y) \in R$. Die Relationen $\mathtt{rFACTORIZATION}$ und $\mathtt{rFACTOR}$ wie oben definiert sind nicht total, nachdem die negativen Instanzen aber besonders „einfach“ sind, können für beide NP-Relationen effektiv äquivalente Relationen angegeben werden, die total sind:

- $\mathtt{rFACTORIZATION}' = \mathtt{rFACTORIZATION} \cup \{(n, \text{„}\textit{ungültig}\text{“}) \mid n \leq 1\}$.
- $\mathtt{rFACTOR}' = \mathtt{rFACTOR} \cup \{(n, \text{„}\textit{ungültig}\text{“}) \mid n \leq 1 \text{ oder } n \text{ ist prim}\}$.

Megiddo und Papadimitriou (1991) fassen diese totalen NP-Relationen zur Klasse TFNP zusammen:

Definition 3.3 (TFNP)**.** Die Klasse TFNP ist die Teilmenge von FNP derjenigen NP-Relationen R, welche linkstotal sind, heißt zu jedem $x \in \Sigma^*$ existiert ein $y \in \Sigma^*$ mit $(x, y) \in R$. ◁

Hierzu gehören die oben genannten Varianten $\mathtt{rFACTORIZATION}'$ und $\mathtt{rFACTOR}'$. Für Megiddo und Papadimitriou befinden sich in TFNP eine Vielzahl von interessanten und schwierigen Suchproblemen, bei denen die Frage der Lösbarkeit in Polynomialzeit noch offen ist. Das betrifft u.a. zahlentheoretische Probleme aus der Kryptographie wie Faktorisierung, diskreter Logarithmus. Beachte dass TFNP nicht identisch ist zur Klasse $\mathrm{NPMV}_{\mathrm{t}}$; es macht sich hier die gleiche Unterscheidung wie bei FNP vs. NPMV auf: Die Klasse TFNP ist eine Teilmenge von $\mathrm{NPMV}_{\mathrm{t}}$ jener totalen Multifunktionen $f \in \mathrm{NPMV}_{\mathrm{t}}$, für die der (der Graph) f in P liegt. Beachte dass TFNP sogar eine echte Teilmengen von $\mathrm{NPMV}_{\mathrm{t}}$ ist, außer $\mathrm{P} = \mathrm{NP}$:

Beobachtung 3.4 (vgl. Fenner u. a. 2003, Prop. 5)**.** *Wenn für alle* $f \in \mathrm{NPMV}_{\mathrm{t}}$ *auch (der Graph)* $f \in \mathrm{P}$ *ist, dann gilt* $\mathrm{P} = \mathrm{NP}$.

Beweis. Betrachte folgenden NPTM-Transduktor N auf Eingabe $\varphi \in \Sigma^*$: zunächst spaltet sich die Berechnung nichtdeterministisch auf. In der ersten Rechnung wird sofort 1 ausgegeben. In der zweiten Rechnung wird eine Belegung w für die aussagenlogische Formel φ geraten, und 2 ausgegeben wenn w die Formel φ erfüllt. Sei f die Multifunktion, welche von N berechnet wird. Damit gilt:

$$\textit{set-}f(x) = \{1, 2\} \text{ falls } x \in \mathtt{SAT}, \text{ und } \{1\} \text{ sonst}$$

und $f \in \mathrm{NPMV}_{\mathrm{t}}$. Nach Annahme ist $f \in \mathrm{P}$. Nun kann aber SAT in Polynomialzeit entschieden werden, denn $\varphi \in \mathtt{SAT}$ genau dann wenn $(\varphi, 2) \in f$.

Die Aussage relativiert, wenn anstelle SAT z.B. das kanonische vollständige Problem gewählt wird. □

Mit der Beschäftigung mit TFNP-Problemen kam es zu einer umfassenden Theoriebildung. So kam z.B. eine verfeinerte Betrachtung durch Unterklassen von TFNP hinzu. Jede dieser Unterklassen verinnerlicht hierbei jeweils das kombinatorische Prinzip, „warum“ ein Suchproblem total ist. Exemplarisch werden hier zwei Unterklassen skizziert:

- Die Unterklasse PLS („polynomial local search“) umfasst die Suchprobleme, welche in die Form eines Suchgraphen polynomiellen Grads gebracht werden können, worauf ein

lokales Optimum gesucht ist. Das zugrunde liegende kominatorische Prinzip zur Totalität wäre „*endliche Suchgraphen haben immer ein lokales Optimum*“ oder allgemeiner „*Jeder endliche gerichtete azyklische Graph hat eine Senke*“.

Ein Beispiel hierfür wäre die Suche nach einem lokal optimalen Schnitt in einem Graphen; hier meint „lokal optimal“ dass kein Flip eines Knotens zu mehr Kantenschnitten führt. Nachdem es nur exponentiell viele Schnitte gibt, muss mindestens einer davon lokal optimal sein.

- Die Unterklasse PPP („polynomial pigeon principle“) umfasst Suchprobleme, welche aufgrund des kombinatorischen Schubfachprinzip total sind.

  Ein Beispiel hierfür ist das Gleiche-Summe-Suchproblem: gegeben n positive ganze Zahlen die sich zu $< 2^n - 1$ aufsummieren, finde zwei unterschiedliche nichtleere Teilmengen dieser Zahlen welche die gleiche Summe haben. (Existiert nach Schubfachprinzip: es existieren $2^n - 1$ viele nichtleere Teilmengen, jede davon mit Summe $< 2^n - 1$, die Summen können also nicht alle unterschiedlich sein.)

Auf die weitere Theorie der TFNP-Probleme wird in dieser Arbeit nicht weiter eingegangen. Wir werden aber in Abschnitt **??** noch Reduktionen auf NP-Relationen definieren; dieser Reduktionsbegriff ist der identische wie auf den TFNP-Problemen Megiddo und Papadimitriou (1991).

3.2 Suchprobleme vs. Entscheidungsprobleme

Wie in der Einleitung schon ausgeführt, konzentriert sich die algorithmische Komplexitätstheorie primär auf die Entscheidungsprobleme und weniger auf die Suchprobleme. Das ist durchaus fundiert: es kommt mit einer Vereinfachung der Konzepte, Definitionen und Theorien, und gleichzeitig lässt sich für viele relevante Instanzen das Suchproblem auf das entsprechende Entscheidungsproblem „reduzieren“. Dieses Argument wird gerne als *search reduces to decision* beschrieben.

In diesem Abschnitt werden detailiert Suchprobleme und Entscheidungsprobleme gegebübergestellt und Forschungsergebnisse hierzu aus der Literatur präsentiert. Zum einen wird die eben genannte Reduzierbarkeit und das *search-reduces-to-decision*-Argument ausgeführt, und zum anderen werden Ergebnisse vorgestellt, die darauf hinweisen dass genau dieses Argument nicht für alle Suchprobleme zutrifft.

Wir wollen zunächst auf die Beziehungen zwischen NP-Relationen und NP-Sprachen hinweisen. Über die Projektionen von NP-Relationen können wir genau die Klasse NP von Entscheidungsproblemen charakterisieren: zu jeder Sprache bzw. Entscheidungsproblem $L \in \mathrm{NP}$ existiert (mind.) eine NP-Relation R mit $L = \mathrm{Proj}(R)$, und zu jeder NP-Relation bzw. NP-Suchproblem R ist $\mathrm{Proj}(R) \in \mathrm{NP}$. Das ist die übliche „Zerifikats-Charakterisierung“ von NP aus den Lehrbüchern.

Beobachtung 3.5 (Zertifikats-Definition von NP)**.** $\mathrm{NP} = \{\mathrm{Proj}(R) \mid R \textit{ ist eine NP-Relation}\}$.

Beweis. Für die Inklusion von links nach rechts starten wir mit einer Sprache L und einer NPTM N die L entscheidet, wobei die Laufzeit durch das Polynom p beschränkt ist. Definiere nun die Relation

$$R_N = \{(x, \alpha) \mid N(x) \text{ akz. mit RW } \alpha,\ \alpha \text{ hat} \leq p(|x|) \text{ viele Schritte}\}$$

Diese Relation ist eine NP-Relation. Der Test ist offenbar in Polynomialzeit möglich, und die Relation ist p-balanciert, ist $|\alpha| \in O(\#\text{ Schritte von } \alpha) \in O(p(|x|))$. Aus Definition geht hervor dass $L(N) = \mathrm{Proj}(R_N)$.

Für die Inklusion von rechts nach links konstruieren wir uns zu einer gegebenen NP-Relation R mit Zertifikatsschranke q eine NPTM N_R die $\mathrm{Proj}(R)$ entscheidet. Die NPTM N_R rät auf Eingabe x zunächst ein $y \in \Sigma^{\leq q(|x|)}$ und akzeptiert genau dann wenn $(x, y) \in R$. Es ist wegen $R \in \mathrm{P}$ klar, dass diese NTM in Polynomialzeit läuft. Wieder geht aus Definition hervor dass $L(N) = \mathrm{Proj}(R_N)$. □

Damit ist im Übrigen die obige Definition von NP-Relationen auch nicht „neu“ sondern schon immer mitgedacht. Die eben formulierte Charakterisierung findet sich in allen üblichen Einführungswerken zur Komplexitätstheorie. Dagegen machen die unterschiedlichen Einführungswerke ihren Zugang manchmal stärker von der Perspektive der Suchprobleme abhängig, und manchmal stärker von der typischen Herangehensweise über Entscheidungsproblemen. Vgl. z.B. Goldreich (2008) wlecher in seinem Lehrbuch die P-vs.-NP-Frage zunächst als die äquivalente Frage der Beziehung zwischen den „*efficiently solvable search problems*“ und den „*search problems with efficiently checkable solutions*“ (letzteres sind genau die NP-Relationen) formuliert. Erst später wird mittels *search-reduces-to-decision*-Argumenten dafür argumentiert, NP-Entscheidungsproble als die zentralen Untersuchungsobjekte der Komplexitätstheorie anzusehen.

Zumindest für die P-NP-Frage ist es irrelevant, ob man sich auf Suchprobleme von NP-Relationen oder auf Entscheidungsproblemen von NP-Mengen bezieht. Jedes NP-Suchproblem ist in Polynomialzeit lösbar genau dann wenn jede Menge in NP in deterministischer Polynomialzeit entscheidbar ist.

Lemma 3.6. $\mathrm{FNP} \subseteq_{\mathrm{c}} \mathrm{FP} \iff \mathrm{P} = \mathrm{NP}$.

Beweis. Die Richtung von links nach rechts ist klar, sind ja Suchprobleme schwieriger als Entscheidungsprobleme (Beobachtung 3.2 mit Beobachtung 3.5.)

Die Richtung von rechts nach links zeigen wir mittels Präfixsuche. Sei R eine beliebige NP-Relation mit Zertifikatsschranke q. Wir zeigen dass $R \in_{\mathrm{c}} \mathrm{FP}$. Betrachte folgende Menge

$$A = \{(x, z) \mid \exists z' \in \Sigma^{\leq q(|x|)}, (x, zz') \in R\}$$

Es ist leicht zu sehen dass $A \in \mathrm{NP}$. Also gilt nach Annahme aus $A \in \mathrm{P}$. Nun kann gegeben eine Instanz x iterativ ein Präfix eines Zertifikats verlängert werden:

```
1  z ← ε
2  solange |z| ≤ q(|x|) tue (Invariante: wenn ein Zertifikat y für x ex., dann z ⊑ y)
3  |  wenn (x, z) ∈ R dann
4  |  |  akzeptiere mit z
5  |  sonst wenn (x, z0) ∈ A dann
6  |  |  z ← z0
7  |  sonst wenn (x, z1) ∈ A dann
8  |  |  z ← z1
9  |  sonst
10 |  |  ablehnen
11 ablehnen
```

Korrektheit klar. Unter Annahme $A \in \mathrm{P}$ ist auch klar, dass diese Funktion von einem PTM-Transduktor berechnet werden kann. Damit $R \in_{\mathrm{c}} \mathrm{FP}$. □

Search reduces to decision

Es ist leicht zu sehen, dass der Suchalgorithmus von obigem Beweis so geändert werden kann, dass anstelle der Entscheidung von A auch Orakelfragen an ein (externes) Orakel A gestellt werden können, d.h. das Suchproblem von R kann *à la Cook* auf das Entscheidungsproblem von A reduziert werden. In anderen Worten, $R \in_{\mathrm{c}} \mathrm{FP}^A$. Das generalisiert sogar, wenn statt A irgendein beliebiges für NP-vollständiges Orkel gewählt wird. Ist also $\mathrm{Proj}(R) \leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig für NP, dann gilt trivialerweise der Spezialfall $R \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(R)}$. Das ist genau das *search-reduces-to-decision*-Argument: ist das Entscheidungsproblem zu R effizient lösbar, dann auch das Suchproblem zu R effizient lösbar.

Korollar 3.7 (*Search reduces to decision* für die NP-Vollständigen)**.** *Sei R eine NP-Relation, für die* $\mathrm{Proj}(R)$ *auch $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig für* NP *ist. Dann gilt $R \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(R)}$.*

Beweis. Wir zeigen die Aussage mit einem Relativierbarkeits-Argument.

Relativ zum Orkakel $\mathrm{Proj}(R)$ gilt $\mathrm{P} = \mathrm{NP}$, ist ja $\mathrm{Proj}(R)$ vollständig für NP. Damit gilt mit vorigem Lemma 3.6 auch $\mathrm{FNP} \subseteq_{\mathrm{c}} \mathrm{FP}$ relativ zu $\mathrm{Proj}(R)$. Da $R \in \mathrm{FNP}$, gilt also auch $R \in_{\mathrm{c}} \mathrm{FP}$ relativ zu $\mathrm{Proj}(R)$. □

Für die NP-Intermediates, also Entscheidungsprobleme aus NP, die weder in P liegen, noch NP-vollständig sind, ist aber unklar, ob immer das Suchproblem auf das Entscheidungsproblem reduziert werden kann.

Wie beim Suchalgoritmus aus obigem Beweis ist aber klar, dass Suchprobleme immer auf eine Präfix- bzw. Bisektion-Entscheidungsvariante reduziert werden können. Im allgemeinen Fall: für jede NP-Relation R mit Laufzeitschranke q gilt

$$R \in_{\mathrm{c}} \mathrm{FP}^{L_R} \text{ wobei } L_R = \{(x, z) \mid \exists z' \in \Sigma^{\leq q(|x|)} \mid (x, zz') \in R\}$$

und

$$R \in_{\mathrm{c}} \mathrm{FP}^{L'_R} \text{ wobei } L'_R = \{(x, z) \mid \exists y \in \Sigma^{\leq q(|x|)} \mid (x, y) \in R, y \leq z\}$$

Konkret ist das zum Beispiel der Fall bei der NP-Relation `rSMALLFACTOR`. Zur Erinnerung, wir haben

$$\mathrm{Proj}(\texttt{rSMALLFACTOR}) = \{(n, a) \mid n \text{ nicht prim, ex. nichttrivialer Faktor } p \text{ von } n \text{ mit } p \leq a\}.$$

Durch Orakelfragen an $\mathrm{Proj}(\texttt{rSMALLFACTOR})$ kann dann mit binärer Suche ein solcher Faktor auch gefunden werden.

Das *search-reduces-to-decision*-Argument hat aber auch Grenzen: Diese Technik scheitert insbesondere, wenn wir wirklich immer die exakte Projektion als Entscheidungsproblem verstehen. Betracht zum Beispiel die NP-Relation zur linearen Teilbarkeit:

$$\texttt{rLINDIV} = \{((a, b), k) \mid a, b, k \in \mathbb{N}, a \cdot k + 1 \text{ teilt } b\}.$$

Wir wissen dass $\mathrm{Proj}(\texttt{rLINDIV}) \notin \mathrm{P}$ außer $\mathrm{NP} = \mathrm{coNP}$ (Adleman und Manders 1977); ob $\mathrm{Proj}(\texttt{rLINDIV})$ NP-vollständig ist, bleibt unklar. Bei dieser NP-Relation wäre nun nicht ersichtlich, wie das Suchproblem auf das Entscheidungsproblem reduziert werden könnte; eine triviale binäre Suche wie oben ist ja nicht möglich.

Für andere Suchprobleme existieren aber nichttriviale Möglichkeiten das Suchproblem auf das (natürliche) Entscheidungsproblem zu reduzieren, auch wenn das Entscheidungsproblem nicht in der Form einer Bisektion/Präfixsuche ist. Hierbei wird die spezifische Struktur des Problems ausgenutzt. Ein Beispiel ist $\mathtt{rSAT}$: Gegeben Formel φ, teste mittels dem Orakel, ob $\varphi[x_1/0] \in \mathtt{SAT}$ oder $\varphi[x_1/1] \in \mathtt{SAT}$. Hier meit $\varphi[x_1/0]$ die Formel sein soll, die entsteht wenn alle Vorkommen von Variable x_1 in φ mit 0 ersetzt werden, $\varphi[x_1/1]$ analog. Sollte jetzt $\varphi[x_1/0] \in \mathtt{SAT}$ stimmen, dann wissen wir dass es eine Belegung für φ existiert die φ erfüllt und gleichzeitig x_1 auf 0 setzt. Wir können dann iterativ auf dem gleichen Weg eine Belegung für die nächste Variable x_2 bestimmen usw. (Der Fall dass $\varphi[x_1/1] \in \mathtt{SAT}$ ist analog.) Es gilt $\mathtt{rSAT} \in \mathrm{FP}^{\mathrm{Proj}(\mathtt{rSAT})}$.

Beachte aber, dass $\mathrm{Proj}(\mathtt{rSAT}) = \mathtt{SAT}$ schon NP-vollständig ist. Damit folgt $\mathtt{rSAT} \in \mathrm{FP}^{\mathtt{SAT}}$ schon aus Korollar 3.7.

Ein nichttriviales Beispiel für ein Suchproblem, deren Projektion (mutmaßlich) nicht NP-vollständig ist, wäre die NP-Relation $\mathtt{rGI}$. Zur Erinnerung: dieses Suchproblem sucht nach einem Graphisomorphismus zwischen zwei gegebenen Graphen. Hier gilt $\mathtt{rGI} \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(\mathtt{rGI})}$: es lässt sich ein Graphisomorphismus zwischen G und H bestimmen, indem mehrmals mittels des Orakels bei (anderen) Paaren von Graphen getestet wird, ob diese isomorph sind (vgl. Goldreich 2008, S. 65, 100). Ob eine nichttriviale Reduktion für $\mathtt{rLINDIV}$ möglich ist, scheint in der Literatur nicht untersucht.

Abschließend wollen wir noch theoretische Resultate präsentieren. Die ersten zwei plausbilisieren, dass wahrscheinlich eine NP-Relation existiert, für die das Suchproblem nicht auf das entsprechende Entscheidungsproblem reduziert werden kann (also wie bei $\mathtt{rLINDIV}$ vermutet).

Satz 3.8 (Impagliazzo und Sudan 1991; Borodin und Demers 1976, Thm. 5)**.** *Angenommen* $\mathrm{E} \neq \mathrm{NE}$ *oder* $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP}$. *Dann existiert eine NP-Relation mit* $\mathrm{Proj}(R) \in \mathrm{NP} - \mathrm{P}$ *für die* $R \notin_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(R)}$ *gilt.*

Unter stärkeren Bedingungen kann sogar zeigen, dass sogar *Mengen* $L \in \mathrm{NP}$ existieren, für die das Suchproblem *jeder* NP-Relation für L nicht auf das Entscheidungsproblem reduziert werden kann. In anderen Worten, unabhängig davon wie das „Zertifikatssystem" für L aussieht, ist keins so einfach dass Zertifikate mit Hilfe eines Orakels für das Suchproblem gefunden werden können.

Satz 3.9 (Bellare und Goldwasser 1994, Thm. 1.1; Impagliazzo und Sudan 1991)**.** *Angenommen* $\mathrm{EE} \neq \mathrm{NEE}$ *oder* $\mathrm{NE} \neq \mathrm{coNE}$. *Dann existiert eine Menge* $L \in \mathrm{NP} - \mathrm{P}$ *sodass* $R \notin_{\mathrm{c}} \mathrm{FP}^{L}$ *für jede NP-Relation* R *für* L*, i.e. für die* $\mathrm{Proj}(R) = L$ *gilt.*

Beachte dass zu keiner dieser Relationen R aus den beiden vorigen Sätzen die Projektion $\mathrm{Proj}(R)$ eine NP-Intermediate ist; $\mathrm{Proj}(R)$ kann nicht $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig sein, denn das wäre ein Widerspruch zu Korollar 3.7.

Folgendes Resultat charakterisiert diejenigen Sprachen $L \in \mathrm{NP}$ die zumindest *eine* NP-Relation R für L haben, sodass das Suchproblem (bzgl. R) auf das Entscheidungsproblem reduzierbar ist.

Definition 3.10. (1) Eine deterministishe OTM heißt *robust für* A falls $L(M^O) = A$ für alle Orakel O.

(2) Eine Menge A heißt *selbsthelfend* falls eine für A robuste OTM M existiert, für die $\mathrm{time}^A_M(x)$ polynomiell in abh. von $|x|$ wächst, i.e. M^A ist eine POTM (zumindest mit dem Orakel A angeschlossen). $\lhd$

Balcázar fass die Intuition hinter dieser Definition wie folgt zusammen: man will die Situation abbilden, dass ein Entscheidungsalgoritmus existiert der, mit genug Zeit, immer zu einem korrekten Ergebnis kommt, aber auch mit einem externen „Helfer" interagieren darf, welcher dem Algorithmus helfen kann, schneller fertig zu rechnen.

Satz 3.11 (Balcázar 1989)**.** *Sei* $A \in NP$*. Folgende Aussagen sind äquivalent:*

(1) A *ist selbsthelfend.*

(2) es existiert eine NP-Relation R *sodass* $\mathrm{Proj}(R) = A$ *und* $R \in_{\mathrm{c}} \mathrm{FP}^{A}$.

Selbstreduzierbarkeit in TFNP

Für *totale* Suchprobleme (i.e. aus TFNP) kann nicht sinnvoll gefragt werden, ob hier das Suchproblem auf das Entscheidungsproblem reduziert werden kann, ist ja für $R \in \mathrm{TFNP}$ das entsprechende Entscheidungsproblem $\mathrm{Proj}(R) = \Sigma^*$ trivial.

Stattdessen können wir uns aber fragen, ob das Suchproblem eines Zertifikats zu x einfacher wird, wenn wir Lösungen zu „kleineren" Instanzen x' gratis abfragen dürfen. Hierzu

können wir folgenden Begriff von Reduzierbarkeit definieren: Betrachte hierbei folgende Variante eine POTM-Transduktors relativ zu $R \in \text{TFNP}$: Dieser Transduktor ist wie ein üblicher PTM-Transduktor, hat zusätzlich aber Zugriff auf ein *funktionales* Orakel, in dem Sinne dass er Orakelfragen der Form „*gib mir ein Zertifikat y für x′*“ stellen kann. Das Orakel antwortet dann mit einem solchen Zerifikat y mit $(x', y) \in R$. Das existiert, ist ja R total.

Es ist klar, dass mit einem solchen Transduktor relativ zu R auch das Suchproblem zu R lösbar ist. (Gegeben x, stelle einfach die Frage „gib mir Zertifikat für x“.) Deshalb nehmen wir folgende Enschränkung vor: der Transduktor darf bei Eingabe x in den Orakelfragen nur nach Zertifikaten für x' fragen, die kürzer sind als x. Falls selbst unter dieser Einschränkung der Fragen das Suchproblem durch eien solchen Transduktor relativ zu R gelöst werden kann, sagen wir, dass R *nach unten selbstreduzierbar ist.*

Zum Verständnis: Wäre die TFNP-Relation $\texttt{rFACTOR}' \in \text{TFNP}$ (i.e., suche einen nichttriviane Faktor für n, oder gebe „prim“ aus) nach unten selbstreduzierbar, dann würde das bedeuten dass ein Faktor von n effizient gefunden werden kann, wenn wir nach Faktoren von Zahlen $\leq n/2$ fragen dürfen. Welche TFNP-Probleme nach unten selbstreduzierbar sind, ist erstaunlich wenig untersucht, und eine Beforschung in dieser präzisen Formulierung wurde wohl erst durch Harsha, Mitropolsky und Rosen (2023) angetreten. Sie zeigen die Selbstreduzierbarkeit nach unten für folgendes TFNP-Problem „Iterate with source“, welches als ein „kanonischer“ Repräsentant für die Unterklasse PLS (zur Erinnerung: *polynomial local search*) gilt. Zur Verständlichkeit geben wir hier nur eine intuitive Formulierung an:

> **Iterate with source:**
>
> Gegeben ist ein Nachfolger-Schaltkreis $S\colon \Sigma^n \to \Sigma^n$ (dieser induziert einen gerichteten Graphen auf den Knoten Σ^n, i.e. $S(v)$ ist einziger Nachfolger von v) polynomieller Größe abh. von n, und ein Startknoten $s \in \Sigma^n$.
>
> Finde einen Knoten $v \in \Sigma^n$ sodass $v < S(v) \not< S(S(v))$ gilt.

Die Selbstreduzierbarkeit macht dabei nur Orakelfragen mit kleineren Schaltkreisen $S'\colon \Sigma^{n-1} \to \Sigma^{n-1}$ (die auch eine kürzere Repräsentation haben) und Startknoten Σ^{n-1}.

Für „natürliche“ TFNP-Probleme ist offen, welche davon nach unten selbstreduzierbar sind. Harsha, Mitropolsky und Rosen fragen explizit danach, ob z.B. die Suche nach einem maximalen Schnitt in der Flip-Umgebung auch nach unten abgeschlossen ist.

Zumindest im Bezug auf die Faktorisierung zeigen Harsha, Mitropolsky und Rosen, dass diese wahrscheinlich nicht nach unten selbstreduzierbar ist.

Satz 3.12 (Harsha, Mitropolsky und Rosen 2023)**.** *Die NP-Relation* $\texttt{rFACTOR}' \in \text{TFNP}$ *ist nicht nach unten selbstreduzierbar, außer* $\texttt{rFACTOR}' \in \texttt{PLS}$.

Es ist offen ob $\texttt{rFACTOR}' \overset{?}{\in} \texttt{PLS}$ und zumindest unplausibel, weil unklar ist wie Faktorisierung als lokales Suchproblem repräsentiert werden kann. Tatsächlich zeigen die Autorinnen sogar die Konsequenz $\texttt{rFACTOR}' \in \text{UEOPL}$, was auch noch $\texttt{rFACTOR}' \in \text{PPAD}$ zur Folge hätte. Auch die Frage ob $\texttt{rFACTOR}' \overset{?}{\in} \text{PPAD}$ ist offen, und wurde breit untersucht. Eine positive Antwort wäre zumindest sehr überraschend (vgl. Harsha, Mitropolsky und Rosen 2023, 67:15; siehe ebd. auch für eine Def. von UEOPL, PPAD). Insgesamt ist die Forschung bezüglich Selbstreduzierbarkeit nach unten für TFNP-Probleme (und allgemeiner für FNP-Probleme) noch sehr klein, und es bedarf auf jeden Fall weiterer Untersuchungen, unter anderem auch im Richtung einer Konzeptionalisierung von „kleinerer Instanz“, die robuster als „kürzerer String“ ist. Vergleiche mit der *disjunktiver Selbstreduzierbarkeit* (Balcázar 1989, vgl. Selman 1988, vgl. vgl. Wechsung 2000, Abschn. 9.5) – einem ähnlichen Begriff der Selbstreduzierbarkeit auf der Ebene der Entscheidungsprobleme – liegen nahe.

3.3 Levin-Reduzierbarkeit

Ähnlich wie auf den üblichen Entscheidungsproblemen können wir auch von Reduzierbarkeiten zwischen verschiedenen Suchproblemen sprechen. In der Literatur hat sich folgender Begriff von Reduzierbarkeit zwischen NP-Relationen herausgebildet (vgl. Papadimitriou 1994, S. 229; Goldreich 2008, S. 61; Arora und Barak 2009, S. 50):

Definition 3.13 (Levin-Reduzierbarkeit)**.** Seien Q, R zwei NP-Relationen. Wir sagen dass Q *sich auf* R *(Polynomialzeit-)Levin-reduzieren lässt*, bzw. $Q \leq_{\text{L}}^{\text{p}} R$ wenn zwei Funktionen $f, g \in \text{FP}$ existieren sodass

(1) $x \in \text{Proj}(Q) \iff f(x) \in \text{Proj}(R)$,

(2) $(f(x), y) \in R \implies (x, g(x, y)) \in Q$.

Punkt (1) sagt also nur aus, dass f eine Many-one-Polynomialzeit-Reduktion zwischen den entsprechenden Entscheidungsproblemen ist. Punkt (2) sagt nun aus, dass wenn y ein Zertifikat für die Instanz $f(x)$ aus R ist, dann lässt sich aus y wieder ein Zertifikat $g(x, y)$ für die originale Instanz x berechnen.

Die Funktion f nennen wir *Reduktionsfunktion*, die Funktion g nennen wir *Translationsfunktion*.

Wir schreiben $Q \leq^{\mathrm{p}}_{\mathrm{L},1} R$ falls f zusätzlich injektiv ist. Wir schreiben $Q \leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}} R$ falls f zusätzlich injektiv und p-invertierbar ist. Klar ist:

$$Q \leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}} R \implies Q \leq^{\mathrm{p}}_{\mathrm{L},1} R \implies Q \leq^{\mathrm{p}}_{\mathrm{L}} R \implies \mathrm{Proj}(Q) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(R).$$

Definiere die Duplikatrelation $\equiv^{\mathrm{p}}_{\mathrm{L}}$ entsprechend. ◁

Die Bezeichnung *Levin*-Reduktion ist hier in Anlehung an bisherige Verwendung gewählt, und bezieht sich darauf, dass in der Etablierung der NP-Vollständigkeit durch Karp (1972), Cook (1971) und Levin (1973) gerade Levin die Suchprobleme in den Blick genommen hat, während Karp und Cook sich auf Entscheidungsprobleme konzentriert haben. Die Formalisierung von NP-Suchproblemen durch NP-Relationen (Definition 3.1) findet sich in Grundzügen schon in Levins Präsentation. Es sei aber darauf hingewiesen, dass sich die hier genannte Definition der Levin-Reduzierbarkeit (Definition 3.13) eine schwächere Form der Reduzierbarkeit ist als die eigentliche von Levin vorgeschlagene. Die hier genannte Definition ist jedoch hinreichend für alle relevanten Eigenschaften, sowie für die Aussagen aus Levins eigener Publikation.

Beachte dass $\leq^{\mathrm{p}}_{\mathrm{L}}$-Reduktionen eine Verstärkung von $\leq^{\mathrm{p}}_{\mathrm{m}}$-Reduktionen auf den jeweiligen Projektionen darstellt:

Beobachtung 3.14. *Wenn* $R \leq^{\mathrm{p}}_{\mathrm{L}} Q$ *dann gilt* $\mathrm{Proj}(R) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(Q)$.

Die Relationen $\leq^{\mathrm{p}}_{\mathrm{L}}$, $\leq^{\mathrm{p}}_{\mathrm{L},1}$ und $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{i}}$ sind reflexiv und transitiv, bilden also eine Quasiordnung. Intuitiv formt die Levin-Reduktion $\leq^{\mathrm{p}}_{\mathrm{L}}$ auf den Suchproblemen das Analog der Many-One-Reduktion $\leq^{\mathrm{p}}_{\mathrm{m}}$ auf den Entscheidungsproblemen.

Genau so wie wir es bei der üblichen $\leq^{\mathrm{p}}_{\mathrm{m}}$-Reduktion auf den Suchproblemen gewohnt sind, ordnet $\leq^{\mathrm{p}}_{\mathrm{L}}$ die Suchprobleme der NP-Relationen nach ihrer „Schwierigkeit“: wenn $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ dann ist Q höchstens so „schwer“ wie R: gegeben einen Lösungsalgorithmus für R lässt sich auch Q effizient lösen, und das sogar mit nur einer Anfrage an den Lösungsalgorithmus. Damit folgt: wenn das Suchproblem zu R effizient gelöst werden kann, dann kann auch das Suchproblem zu Q gelöst werden. Formal ausgedrückt ist FP nach unten abgeschlossen unter der $\leq^{\mathrm{p}}_{\mathrm{L}}$-Ordnung:

Lemma 3.15. *Wenn* $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ *und* $R \in_{\mathrm{c}} \mathrm{FP}$ *dann ist ist* $Q \in_{\mathrm{c}} \mathrm{FP}$.

Beweis. Seien f, g die Reduktions- bzw. Translationsfuktion, welche $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ realisieren, und sei $r \in \mathrm{FP}$ eine Verfeinerung von R. Definiere nun

$$q(x) = \begin{cases} g(x, r(f(x))) & \text{falls } r(f(x)) \neq \bot \\ \bot & \text{sonst.} \end{cases}$$

Offenbar ist $q \in \mathrm{FP}$. Wir zeigen nun dass q eine Verfeinerung von Q ist. Zum einen gilt $\mathrm{dom}(q) = \mathrm{Proj}(Q)$:

$$x \in \mathrm{Proj}(Q) \iff f(x) \in \mathrm{Proj}(R) \iff f(x) \in \mathrm{dom}(r) \iff q(x) \neq \bot \iff x \in \mathrm{dom}(q).$$

Hier folgt die erste Äquivalenz nach Definition 3.13(1). Zum anderen haben wir

$$x \in \mathrm{Proj}(Q) \implies f(x) \in \mathrm{Proj}(R) \implies (f(x), r(f(x))) \in R \implies (x, g(x, r(f(x)))) \in Q \implies q(x) \in \mathit{set}\text{-}Q(x),$$

i.e, $q(x)$ ist ein Zertifikat für x, wie gewünscht. Hier folgt die dritte Implikation nach Definition 3.13(2). □

Genau so wie bei der Many-one-Reduktion auf den Suchproblemen können wir nach größten Elementen auf der $\leq^{\mathrm{p}}_{\mathrm{L}}$-Ordnung fragen.

Definition 3.16. Sei $\mathcal{F}$ eine Klasse von Multifuktionen (z.B. FNP oder TFNP). Wir nennen $R \in \mathcal{F}$ $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* $\mathcal{F}$ wenn R ein größtes Element von $\mathcal{F}$ geordnet über $\leq^{\mathrm{p}}_{\mathrm{L}}$ ist: Für alle $Q \in \mathcal{F}$ gilt $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$.

Die $\leq^{\mathrm{p}}_{\mathrm{L},1}$- und $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{i}}$-Vollständigkeit ist analog definiert. ◁

Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation für FNP. Diese ist im Wesentlichen die natürliche Erweiterung der kanonischen $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Menge $\mathtt{KAN}$:

$$\mathtt{rKAN} = \{((N, x, 1^n), \alpha) \mid \alpha \text{ ist ein akz. Rechenweg auf } N(x) \text{ und } |\alpha| \leq n\}.$$

Beachte dass $\mathrm{Proj}(\mathtt{rKAN}) = \mathtt{KAN}$.

Satz 3.17. *Die kanonische NP-Relation* $\mathtt{rKAN}$ *ist* $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$*-vollständig für* FNP.

Beweis. Sei R eine beliebige NP-Relation mit Zertifikatsschranke r, i.e. $(x,y) \in R \implies |y| \leq r(|x|)$. Sei M die PTM welche R entscheidet, mit Laufzeitschranke p. Sei N eine NPTM welche auf Eingabe x zunächst ein Zertifikat $y, |y| \leq r(|x|)$ rät, und dann testet ob $M(x,y)$ akzeptiert. Die Laufzeit von N ist beschränkt auf $p(|(x,y)|) \in O(p(r(|x|)))$ (hier nutzen wir die effiziente Listencodierung von **??** aus). Sei daher q ein Polynom, welches die Laufzeit von N beschränkt.

Definiere die Reduktionsfunktion $f(x) = (N, x, 1^{q(|x|)})$. Wir zeigen zunächst dass

$$x \in \mathrm{Proj}(R) \iff f(x) \in \mathrm{Proj}(\mathtt{rKAN}).$$

Wenn $x \in \mathrm{Proj}(R)$, dann existiert ein $y, |y| \leq r(|x|)$ sodass $(x,y) \in R$. Dann wird auch $N(x)$ akzeptieren, nämlich auf jenem Pfad welcher y rät. Es existiert also ein Rechenweg α mit $|\alpha| \leq q(|x|)$ sodass $N(x)$ auf α akzptiert. Dann gilt aber auch $(f(x), \alpha) = ((N, x, 1^{q(|x|)}), \alpha) \in \mathtt{rKAN}$. Die Rückrichtung $x \notin \mathrm{Proj}(R) \implies f(x) \notin \mathrm{Proj}(R)$ folgt analog. Es ist klar, dass f injektiv ist, dass f Polynomialzeit-berechenbar und -invertierbar ist.

Es lässt sich außerdem einfach eine Translationsfunktion $g \in \mathrm{FP}$ angeben, die für $g(f(x), \alpha) = y$ aus α das entsprechende geratene Zertifikat y aus α berechenen kann. □

Die Ordnung $\leq^{\mathrm{p}}_{\mathrm{L}}$ und dessen Vollständigkeits-Begriff verhält sich auch sonst wie bei dem Analog $\leq^{\mathrm{p}}_{\mathrm{m}}$ gewohnt.

Lemma 3.18. *Sei R eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation für* FNP*. Es gelten folgende Aussagen:*

(1) $\mathrm{Proj}(R)$ *ist eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge für* NP*.*

(2) Wenn Q eine NP-Relation ist und $R \leq^{\mathrm{p}}_{\mathrm{L}} Q$, dann ist auch Q $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig.

(3) $R \in_{\mathrm{c}} \mathrm{FP} \iff \mathrm{FNP} \subseteq_{\mathrm{c}} \mathrm{FP} \iff \mathrm{NP} = \mathrm{P}$*.*

Es existieren auch natürliche NP-Relationen, die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig für FNP sind. Das Bekannteste ist rSAT. Zur Erinnerung:

$$\mathtt{rSAT} = \{(\varphi, w) \mid \varphi \text{ ist eine aussagenlogische Formel, } w \text{ erfüllende Belegung für } \varphi\}.$$

Die Aussage „rSAT ist $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig“ entspricht dem üblichen Cook–Levin-Satz, und wird hier nur wiederholt:

Satz 3.19 (Satz von Cook und Levin)**.** *Die NP-Relation* rSAT *ist $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständig für* FNP*.*

Skizze. Wir zeigen nur, dass $\mathtt{rCSAT} = \{(C, w) \mid C \text{ ist SAT-Schaltkreis und } C(w) = 1\}$, i.e. Schaltkreiserfüllbarkeit, $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständig ist. Es ist leicht zu sehen dass $\mathtt{rCSAT} \leq^{\mathrm{p}}_{\mathrm{L,1,inv}} \mathtt{rSAT}$ mit den gleichen Argumenten wie $\mathtt{CSAT} \leq^{\mathrm{p}}_{\mathrm{m}} \mathtt{SAT}$.

Ein üblicher Beweis des Satzes von Cook und Levin (welcher auf der Seite der Entscheidungsprobleme operiert) zeigt als erstes, dass eine PTM M und eine „Eingabegröße“ n als Schaltkreis C repräsentiert werden kann, sodass $C_{M,n}(x) = 1 \iff M(x)$ akzeptiert, für alle $x \in \Sigma^n$.

Sei nun R eine beliebige NP-Relation mit Laufzeitschranke q. Ohne Beschränkung können wir in diesem Beweis annehmen, dass alle Zertifikate für x bezüglich R genau die Länge $q(|x|)$ haben. Dann existiert eine PTM M die R entscheidet. Sei $x \in \Sigma^*$ gegeben. Für geeignetes n haben wir

$$\forall y \in \Sigma^{q(|x|)}. \quad C_{M,n}(x,y) = 1 \iff M(x,y) \text{ akz.} \iff (x,y) \in R.$$

Wenn wir jetzt x in $C_{M,n}$ „hart verdrahten“ erhalten wir einen Schaltkreis $C_{M,n,x}$ und es gilt

$$\forall y \in \Sigma^{q(|x|)}. \quad C_{M,n,x}(y) = 1 \iff C_{M,n}(x,y) = 1 \iff (x,y) \in R.$$

Und damit ist $f(x) = C_{M,n,x}$ eine Reduktionsfunktion von R nach rCSAT:

$$x \in \mathrm{Proj}(R) \iff \exists y \in \Sigma^{q(|x|)}.(x,y) \in R \iff \exists y \in \Sigma^{q(|x|)}.C_{M,n,x}(y) = 1$$
$$\iff C_{M,n,x} \in \mathrm{Proj}(\mathtt{rCSAT}) \iff f(x) \in \mathrm{Proj}(\mathtt{rCSAT}).$$

Eine Translationsfunktion g kann auch einfach angegeben werden, wenn $(f(x), y) \in \mathtt{rCSAT}$ dann gilt $C_{M,n,x}(y) = 1$ und damit (modulo einer ggf. notwendigen Umcodierung) $(x,y) \in R$.

Es ist leicht zu sehen dass f injektiv ist, denn wenn die Schaltkreise $f(x), f(x')$ identisch sind, dann muss auch $x = x'$. Genauso ist klar, dass aus $f(x) = C_{M,n,x}$ einfach das „hineincodierte“ x wieder ausgelesen werden kann. □

Die bekannten NP-Relationen R mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständiger Projektion $\mathrm{Proj}(R)$ sind auch Levin-vollständig. Typische Präsentationen von $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigkeit, z.B. $\mathtt{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} \mathtt{VC}$ geben uns nicht nur eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-Reduktionsfunktion f von Instanzen x der einen Menge (i.e. SAT) zu Instanzen $f(x)$ der anderen Menge (i.e. VC), sondern beinhalten meist im Beweis eine (implizit mitgedachte) effiziente Übersetzung von Zertifikaten von x nach $f(x)$ und umgekehrt. Die Reduktionsfunktion f mit der Rückübersetzung der (natürlichen) Zertifikate für $f(x)$ nach Zertifikaten für x reichen dann aus, um eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-Reduktion zu realisieren, und damit Levin-Vollständigkeit zu zeigen.

Nach Goldreich (2008, S. 104) sind die folgenden NP-Relationen jedenfalls definitiv $\leq^{\mathrm{p}}_{\mathrm{L,1,i}}$-vollständig: rSAT, rSETCOVER, rVC, rCLIQUE, r3COLORABILITY. Die von Papadimitriou (1994, S. 193–198) angegebene Reduktion $\mathtt{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} \mathtt{rHAMCYCLE}$ lässt sich leicht zu $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} \mathtt{rHAMCYCLE}$ erweitern, womit rHAMCYCLE auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Ebenso ist rANOTHERHAMCYCLE auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig (1994, S. 232). Damit gilt mit Lemma 3.18(3) auch die in Abschnitt **??** angegebene Äquivalenz NP = P genau dann wenn rSAT, rVC, ...$\in_{\mathrm{c}}$ FP.

NP-Relationen R für welche die Projektion zwar $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig, aber R nicht $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig sind, scheinen nicht bekannt zu sein. Aus dieser empirischen Beobachtung ergibt sich die Frage, ob das auch für *alle* NP-Relationen R gilt:

Frage 3.20. *Wenn* $\mathrm{Proj}(R)$ *eine* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständige Menge für* NP *ist, ist dann auch* R *eine* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständige NP-Relation für* FNP*?*

Aus Präsentationsgründen werden wir die pessimistische negative Antwort auf diese Frage als Vermutung formulieren:

Vermutung 3.21 (Karp-vs-Levin-Vermutung; KvL)**.** *Es existiert eine NP-Relation* R *sodass* $\mathrm{Proj}(R) \leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständig für* NP *ist, aber* R *ist nicht* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* FNP*.*

Obwohl diese Frage erstaunlich auf natürlich scheint, und zwei umfassende Reduktionsbegriffe der Komplexitätstheorie in Beziehung setzen versucht, gibt es erstaunlicherweise kaum Forschung welche sich dieser Hypothese annähert. Ein Beweis von KvL ist jedenfalls mindestens so schwer wie die P-NP-Frage, denn KvL $\Rightarrow$ P $\neq$ NP. In Abschnitt **??** werden wir diese Hypothese und dessen Beziehung zu anderen Hypothesen erarbeiten. Dort werden dann auch Argumente geliefert, die für diese negative Antwort sprechen.

Die oben genannte Frage lässt sich auf natürliche Weise abschwächen, indem man von der konkreten NP-Relation abstrahiert: Wenn L eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge für NP ist, existiert dann zumindest *eine* NP-Relation R_L für L (i.e., $\mathrm{Proj}(R) = L$) sodass R_L $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist? In anderen Worten, existiert ein hinreichend ausdruckstarkes „Zertifikatssystem“ R für L sodass R $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist? Diese abgeschwächte Frage lässt sich positiv beantworten, falls man die Berman–Hartmanis-Vermutung IC annimmt:

Beobachtung 3.22 (Buhrman, Kadin und Thierauf 1998)**.** *Für jede Menge* $L \in$ NP *die p-isomorph zu* SAT *ist, existiert eine NP-Relation* R_L *sodass* $\mathrm{Proj}(R_L) = L$ *und* R_L *auch* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig ist.*

Beweis. Nach Voraussetzung haben wir eine bijektive p-invertierbare Funktion $h \in \mathrm{FP}$ mit $x \in L \iff h(x) \in \mathtt{SAT}$. Definiere nun

$$R_L = \{(x, w) \mid (h(x), w) \in \mathtt{rSAT}\}.$$

Es ist leicht zu sehen, dass R_L eine NP-Relation ist. Es ist auch leicht zu sehen dass $\mathrm{Proj}(R_L) = L$.

Wir zeigen nun, dass R_L auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Sei hierfür Q eine beliebige NP-Relation. Nachdem rSAT ja $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist, existieren Reduktions- und Translationsfunktionen f, g die $Q \leq^{\mathrm{p}}_{\mathrm{L}}$ rSAT realisieren. Definiere nun

$$f'(x) = h^{-1}(f(x)).$$

Insbesondere ist $h^{-1}(\cdot)$ wohldefiniert, ist ja h surjektiv. Damit gilt zum einen für f'

$$x \in \mathrm{Proj}(Q) \iff f(x) \in \mathtt{SAT} \iff h(\underbrace{h^{-1}(f(x))}_{f'(x)}) \in \mathtt{SAT} \iff f'(x) \in \mathrm{Proj}(R_L),$$

und zum anderen gilt

$$(f'(x), w) \in R_L \implies (h(h^{-1}(f(x))), w) \in \mathtt{rSAT} \implies (f(x), w) \in \mathtt{rSAT} \implies (x, g(x, w)) \in Q.$$

Damit erfüllen also f' und g die Voraussetzungen an eine Reduktions- bzw. Translationsfunktion und $Q \leq^{\mathrm{p}}_{\mathrm{L}} R_L$, wie gewünscht. □

(Es ist leicht zu sehen, dass diese Aussage relativiert, wenn anstelle rSAT eine andere beliebige $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Relation R gewählt wird.)

Damit haben (im unrelativierten Fall) insbesondere alle *bekannten* $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen, i.e. zu SAT p-isomorphen Mengen, eine entsprechende $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation. Es muss aber gleichzeitig darauf hingewiesen werden, dass die Zertifikate in den entsprechenden Relationen R_L nicht natürlich sind; die Zertifikate sind nur Belegungen für die Formeln $h(x)$ und haben an sich keinen Bezug bwz. Interpretierbarkeit gegenüber der Instanz x.

Wir schließen mit einer Diskussion zur Vollständigkeit bezüglich TFNP ab. Zum einen ist klar, dass unter dem hier vorgeschlagenen Reduktionsbegriff keine Reduktion von einer nicht-totalem NP-Relation auf eine (totale) TFNP-Relation möglich ist:

$$R \not\leq^{\mathrm{p}}_{\mathrm{L}} Q, \text{ für alle } Q \in \mathrm{TFNP}, R \in \mathrm{FNP}, \mathrm{Proj}(R) \neq \Sigma^*,$$

denn jede potentielle Reduktion würde Definiton 3.13(1) verletzen, i.e. die Many-one-Reduktion auf den jeweiligen Projektionen.

Andererseits kann gefragt werden, ob TFNP-Relationen existieren die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig für TFNP sind. In der Literatur (vgl. Pudlák 2017) wird hierauf eine negative Antwort vermutet:

Vermutung 3.23 (TFNP)**.** *Es existiert keine NP-Relation* $R \in$ TFNP *die* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* TFNP *ist.*

Auch hier ist ein Beweis für diese Vermutung mindestens so schwer wie ein Beweis für NP $\neq$ coNP. Die Beziehung dieser Vermutung mit weiteren Vermutungen betreffend Promise-Probelemen wird in Abschnitt **??** erarbeitet.

3.4 Zur gemeinsamen Struktur von vollständigen Suchproblemen

Feinere Reduktionsbegriffe

Aus den Erfahrungen in den Entdeckungen von NP-vollständigen Mengen bzw. der Entwicklung der hierfür notwendigen Reduktionen wurde intuitiv deutlich, dass die NP-vollständigen Mengen sich viele nicht-offensichtliche Eigenschaften teilen, die über „Equi-Lösbarkeit“ ($A \in \mathrm{P}$ genau dann wenn $B \in \mathrm{P}$ für $\leq^{\mathrm{p}}_{\mathrm{T}}$-vollständige Mengen A, B) hinaus geht. Das beginnt schon bei der sonst vorgezogenen und üblichen Many-one-Reduktion. Hier wird auf die fundamental ähnliche Struktur der $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen A, B hingewiesen: eine Instanz x von A kann als ein „äquivalenter“ Fall $f(x)$ von B repräsentiert werden.

Die intuitive Beobachtung, dass sämtlichen Beweise der $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit zu Beweisen der $\leq_{1,\mathrm{i}}$-Vollständigkeit verstärkt werden können, führte schlussendlich zur (Berman–Hartmanis-)Isomorphievermutung `IC`, in der postuliert wird, dass es im Wesentlichen nur *eine* NP-vollständige Menge gibt, und die verschiedenen Ausprägungen unterschiedlicher NP-vollständiger Mengen nur triviale Umcodierungen des selben Problems sind. Obwohl die Forschung zu NP-Suchproblemen im Vergleich zu den entsprechenden Entscheidungsproblemen im Hintergrund blieb, wurde in der Forschung augrund der oben genannten Erfahrungen und Intuitionen die Beobachtung gemacht, dass viele der entsprechenden Suchprobleme eine inhärente strukturelle Ähnlichkeit untereinander haben (vgl. auch die Diskussion von Hemaspaandra 1998). Im Folgenden werden einige dieser Arbeiten kurz skizziert.

Simon (1975, S. 83) machte beispielsweise die Beobachtung, dass die ihm bekannten Reduktionsfunktionen $f\colon A \to B$ in den Beweisen zur NP-Vollständigkeit so gebaut sind, dass die Instanz x genau k „Lösungen“ bezüglich A hat genau dann wenn $f(x)$ genau k „Lösungen“ bezüglich B hat. „Lösung“ hier in Anführungszeichen weil auf Mengen überhaupt kein Begriff von Lösungen bzw. Zertifikaten existiert; Simon dachte in seinen Überlegungen die zugrunde liegende kombinatorischen (Such-)Probleme zu A und B nur mit.

Auf NP-Relationen lässt sich sein Reduktionsbegriff aber formal präzise formulieren:

Definition 3.24 (Geizige Reduktionen)**.** Seien Q, R NP-Relationen. Wir sagen dass sich Q auf R (in Polynomialzeit) *geizig* (*„parsimonious“*) reduzieren lässt, bzw. $Q \leq^{\mathrm{p}}_{\mathrm{pars}} R$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit

$$|\mathit{set}\text{-}Q(x)| = |\mathit{set}\text{-}R(f(x))|.$$

$\lhd$

Beachte dass geizige Reduktionen immer eine Many-one-Reduktion realisieren: wir haben $x \in \mathrm{Proj}(Q)$ genau dann wenn mind. ein Zertifikat für x bzlg. Q existiert, also genau dann wenn eines für $f(x)$ bzgl. R existiert, bzw. $f(x) \in \mathrm{Proj}(R)$.

Geizige Reduktionen wurden insbesondere in der Komplexitätstheorie des Zählens aufgegriffen (Simon 1975; Valiant 1979). Typische algorithmische Probleme sind z.B. *„wie viele Belegungen w erfüllen die aussagenlogische Formel φ“* oder, kanonischer, *„Auf wievielen Rechenwegen akzeptiert die Berechnung $N(x)$ der NPTM N?“*. Es ist einfach zu sehen, dass sich geizige Reduktionen $\#\mathtt{SAT} \leq^{\mathrm{p}}_{\mathrm{pars}} \#\mathtt{KAN}$ und $\#\mathtt{KAN} \leq^{\mathrm{p}}_{\mathrm{pars}} \#\mathtt{SAT}$ angeben lassen können. Damit kann das Zählproblem zur $\#\mathtt{SAT}$-Instanz x als $\#\mathtt{KAN}$-Instanz $f(x)$ repräsentiert werden und umgekehrt – die beiden Zählprobleme sind relativ zum jeweils anderem gleich schwer (wobei letztere Aussage auch mit schwächeren Reduktionsbegriffen gezeigt werden kann). Auf eine weitere Präsentation der Komplexitätstheorie des Zählens muss hier verzichtet werden.

Beachte, dass geizige Reduktionen nicht mit Levin-Reduktionen vergleichbar sind. Levin-Reduktionen erhalten im Allgemeinen nicht die Anzahl an Zertifikaten, während umgekehrt geizige Reduktionen keine effektive Übersetzung zwischen den Zertifikaten für $f(x)$ auf Zertifikate für x zulassen.

Lynch und Lipton (1978) verstärkt den Reduktionsbegriff der sparsamen Reduktionen und setzt voraus, dass Zertifikate für x in Zertifikate für $f(x)$ effizient umgerechnet werden können. Beachte aber, dass diese „Vorwärtsübersetzung“ nicht hinreichend ist um Levin-Reduzierbarkeit zu zeigen (die ja umgekehrt Zertifikate für $f(x)$ in Zertifikate für x umrechnet).

Fischer, Hemaspaandra und Torenvliet 1995 gingen noch einen Schritt weiter und definieren *zertifikats-isomorphe* Reduktionen („witness-isomorphic reduction“) zwischen zwei NP-Relationen. Hier erhält die Reduktionsfunktion $A \to B$ auf den Instanzen nicht nur die *Anzahl* der Zertifikate, sondern es werden zusätzlich die Zertifikate für $x \in A$ mit den Zertifikateion für $f(x)$ in B in eine effiziente in Polynomialzeit berechenbare Eins-zu-Eins-Korrespondenz gesetzt. Dieses Vorgehen ist intendiert als eine Generalisierung der p-Isomorphie (Hartmanis und Berman 1976) nicht nur auf Instanzen sondern auch auf Zertifikaten.

Definition 3.25 (Zertifikats-Isomorphie; Fischer, Hemaspaandra und Torenvliet 1995)**.** Seien Q, R NP-Relationen. Wir sagen dass sich Q auf R (in Polynomialzeit) *zertifikats-isomorph reduzieren lässt*, bzw. $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$, wenn Funktion $f : \Sigma^* \to \Sigma^*, f \in \mathrm{FP}$ und Funktion $g : \Sigma^* \times \Sigma^* \to \Sigma^* \in \mathrm{FP}$ existieren, die p-invertierbar sind, und

(1) $(x, y) \in Q \implies (f(x), g(x, y)) \in R$ (Vorwärts-Translation von Zertifikaten),

(2) $(f(x), z) \in R \implies \exists y.\, (x, y) \in R \wedge g(x, y) = z$ (g ist quasi "surjektiv"),

(3) Falls $y_1, y_2 \in \textit{set-}Q(x)$ und $y_1 \neq y_2$, dann ist auch $g(x, y_1) \neq g(x, y_2)$ (g ist quasi "injektiv"). ◁

(Der oben skizzierte Reduktionsbegriff von Lynch und Lipton (1978) geht aus der Zertifikats-Isomorphie hervor, wenn „p-invertierbar" in der Definition gestrichen wird.)

Die Punkte (1)–(3) in der Definition der Zertifikats-Isomorphie können alternativ auch folgendermaßen äquivalent formuliert werden:

$$\forall x. \quad |\textit{set-}Q(x)| = |\textit{set-}R(f(x))| \wedge \{g(x, y) \mid y \in \textit{set-}Q(x)\} = \textit{set-}R(f(x)). \tag{3.2}$$

Damit ist auch leicht zu sehen, dass für alle $x \in \mathrm{Proj}(Q)$ die Funktion $g_x(y) = g(x, y)$ eine Bijektion zwischen den Zertifikaten für x und den Zertifikaten für $f(x)$ ist, sowie dass f eine sparsame Reduktion realisiert (und damit auch eine Many-one-Reduktion).

Um die Analogie zur p-Isomorphie abzuschließen: Fischer, Hemaspaandra und Torenvliet definieren einen *witness-isomorphic isomorphism* zwischen zwei NP-Relationen Q, R wenn $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$ via f, g und $R \leq^{\mathrm{p}}_{\mathrm{wi}} Q$ via f^{-1}, g^{-1}. Wieder analog zur Berman–Hartmanis ist für einen *witness-isomorphic isomorphism* ausreichend, wenn $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$ und $R \leq^{\mathrm{p}}_{\mathrm{wi}} Q$ jeweils über verlängernde Funktionen, *à la Schröder–Bernstein*.

Dieser Reduktionsbgeriff stellt eine Verstärkung der Levin-Reduktion und sparsamen Reduktion dar:

Lemma 3.26. *Seien Q, R zwei NP-Relationen. Falls $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$, dann ist $Q \leq^{\mathrm{p}}_{\mathrm{pars}} R$ und $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ und $Q \leq^{\mathrm{p}}_{\mathrm{m}} R$.*

Beweis. Seien $f, g \in \mathrm{FP}$ die zwei Funktionen, welche $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$ realisieren. Wir haben in der vorhergehenden Diskussion über Gleichung 3.2 schon gesehen, dass f eine sparsame (Many-one-)Reduktion darstellt.

Wir müssen für Levin-Reduzierbarkeit nur noch zeigen, dass wir Zertifikate z für $f(x)$ wieder zu Zertifikaten für x rückübersetzen können. Das wegen der p-Invertierbarkeit von g möglich: Mit Definition 3.25(2) gilt

$$(f(x), z) \in R \implies g^{-1}(z) = (x, y) \text{ und } (x, y) \in R.$$

Insbesonderee ist mit Definition 3.25(2) auch $g^{-1}(z) = (x, y)$ definiert, und nachdem $g^{-1} \in \mathrm{FP}$, lässt sich leicht eine entsprechende Translationsfunktion für die Levin-Reduktion angeben. □

Wir werden später zeigen, dass natürliche $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relationen exitiere, welche mutmaßlich nicht $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig sind, und damit auch nicht zertifikats-isomorph zu `rSAT` sein können.

Universelle Relationen

Einen anderen Weg gingen Agrawal und Biswas (1992a). Sie sind konkret daran interessiert, wie die natürlichen NP-Vollständigen Mengen/Relationen konkret strukturiert sind, bzw. welche sie sich teilen. Die Intuition ist am verständlichsten, wenn man sich in Erinnerung ruft, wie übliche Beweise der NP-Vollständigkeit auf Mengen funktionieren. Zum Beispiel beinhaltet ein Beweis der NP-Vollständigkeit von `VC` eine Many-one-Reduktion von `3CNFSAT` auf `VC`. Eine übliche Strategie für diese Reduktion ist es nun, „Gadgets" auf `VC` (hier: Teilgraphen) zu definieren, welche Klausen und Variablen simulieren. Diese werden dann zu einer Instanz $f(\varphi)$ zusammengesetzt, um ganze SAT-Formeln φ zu simulieren; damit kann dann die Reduktion von Formeln auf `VC` realisiert werden.

Agrawal und Biswas formalisieren diese Eigenschaft nun, und nennen NP-Relationen *universell*, wenn diese die Konstruktion eines *building blocks* (ungefähr eine Klausel) zulässt, und die *joinable* (entspricht ungefähr Disjunktion) und die *couplable* (entspricht ungefähr Konjunktion) sind. Damit lässt sich die NP-Vollständigkeit vieler Probleme bzw. Relationen in einer uniformen und allgemeinen Weise aus rein strukturellen Eigenschaften ableiten. Die Eigenschaft *universell* stellt eine Verstärkung der Levin-Vollständigkeit da, und ist eine der stärksten strukturellen Eigenschaften, die Agrawal und Biswas bei allen natürlichen Levin-vollständigen NP-Relationen (modulo einer trivialen Umcodierung der Zertifikate) vermuten. Daher wird deren Arbeit hier auch extensiver ausgeführt.

Eine zentrale Methode von Agrawal und Biswas ist, oft relevante Informationen direkt aus dem „String" der Zertifikate auslesen. Daher beschränken wir auf folgende Teilmenge der NP-Relationen, bei der die Zertifikatsstring möglichst uniform sind:

Definition 3.27 (Strenge NP-Relation)**.** Sei R eine NP-Relation mit Zertifikatsschranke q. Wir nennen R *streng* wenn für R gilt, dass $(x, y) \in R \implies |y| = q(|x|) > 0$. In anderen Worten, jedes Zertifikat y für x ist nicht ε und hat genau die Länge $q(|x|)$. $\lhd$

Dies sollte keine Einschränkung darstellen. zu jeder natürlichen NP-Relation R kann eine strenge NP-Relation R' angegeben werden mit $\mathrm{Proj}(R) = \mathrm{Proj}(R')$ und $R \equiv^{\mathrm{p}}_{\mathrm{L}}$, wobei R' noch einigermaßen natürlich bleibt.

Nun können wir, wie oben schon angedeutet, *joinable*, *couplable* und *building block* definieren.

Definition 3.28 (Universelle Relation; Agrawal und Biswas 1992a)**.** Sei R eine strenge NP-Relation mit Zertifikatsschranke q. Die Relation R ist *universell* wenn alle drei folgenden Eigenschaften erfüllt sind:

(1) Die Relation R hat einen *building block*: es gibt ein Element $\mathit{block} \in \mathrm{Proj}(R)$, sowie paarweise verschiedene $b_1, b_2, b_3 \in \mathbb{N}$ sodass

$$\{y[b_1, b_2, b_3] \mid y \in \mathit{set}\text{-}R(\mathit{block})\} = \Sigma^3 - \{000\}$$

(2) Die Relation R ist *joinable*: es gibt eine Funktion $\mathit{join} \in \mathrm{FP}$ sodass $\mathit{join}(x_1, \dots, x_n) = (z, \alpha)$, wobei $x_1, \dots, x_n, z \in \Sigma^*$, $\sum_{k=1}^{n} q(|x_k|) = |\alpha| \leq q(|z|)$, und wobei $\alpha \in \mathbb{N}^*$ eine Sequenz von *paarweise verschiedenen* Indizes $< q(|z|)$ ist, und

$$\{y'[\alpha] \mid y' \in \mathit{set}\text{-}R(z)\} = \{y_1 \circ y_2 \circ \cdots \circ y_n \mid (\forall k \leq n).\, y_k \in \mathit{set}\text{-}R(x_k)\}.$$

(3) Die Relation R ist *couplable*: es gibt eine Funktion $\mathit{cpl} \in \mathrm{FP}$ sodass

$$\mathit{cpl}(x, (i_1, \dots, i_n), (j_1, \dots, j_n)) = (z, \alpha)$$

wobei $x \in \Sigma^*$, $i_1, \dots, i_n, j_1, \dots, j_n \leq q(|x|) - 1$ und $|\alpha| = q(|x|)$, und wobei wieder $\alpha \in \mathbb{N}^*$ eine Sequenz von *paarweise verschiedenen* Indizes $< q(|z|)$ ist, und

$$\{y'[\alpha] \mid y' \in \mathit{set}\text{-}R(z)\} = \{y \mid y \in \mathit{set}\text{-}R(x) \text{ und } (\forall k \leq n)(y[i_k] \neq y[j_k])\}. \qquad \lhd$$

Wir illustrieren diese Definition anhand von `rSAT`. Wir nehmen dafür an, dass `rSAT` als strenge NP-Relation gegeben ist. Die Formeln φ seien so codiert, dass in φ nur die Variablen $x_0, \dots, x_{|\varphi|-1}$ vorkommen. Zertifikate für φ sind dann Strings w der Form $\Sigma^{|\varphi|}$ sodass

(1) φ durch die Variablenbelegung $x_0 = w[0], x_1 = w[1], \dots$ erfüllt wird, und
(2) $w[j] = 0$ wenn x_j nicht in φ vorkommt.

Der *building block* ist im Wesentlichen dazu da, frische Klauseln einzuführen. Definiere zum Beispiel *block* als $(x_0 \vee x_1 \vee x_2)$. Wir haben dann z.B. $y = 001000\cdots \in \mathit{set}\text{-}\mathtt{rSAT}(\varphi)$, welches der Variablenbelegung $x_0 = 0, x_1 = 0, x_2 = 1, x_3 = 0, \dots$ entspricht. Mit der Projektion über $b_1 = 0, b_2 = 1, b_3 = 2$ erhalten wir dann die Menge $\{y[b_1, b_2, b_3] \mid y \in \mathit{set}\text{-}\mathtt{rSAT}(\varphi)\} = \Sigma^3 - \{000\}$. Das Wort 000 fehlt, denn (erweitert auf ein Zertifikat) erfüllt diese Belegung nicht φ.

Die Funktion $\mathit{join}(\varphi_1, \varphi_2, \dots) = (\psi, \alpha)$ definieren wir als die Konjunktion $\psi = \varphi_1 \wedge \varphi_2 \wedge \cdots$ zusammen mit der Sequenz α. Hierbei nummerieren wir dabei aber die Variablen davor um sodass die Variablen in φ_i und φ_j disjunkt sind. Die Sequenz α reflektiert diese Umbenennung, sodass aus einem Zertifikat y für ψ die entsprechenden erfüllende Belegungen für jedes φ_i zusammensetzt.

Die Funktion $\mathit{cpl}(\varphi, (i), (j))$ sorgt dafür, dass zwei Variablen x_i, x_j „gekoppelt" werden in dem Sinn, dass, unter allen erfüllenden Belegungen, $x_i \neq x_j$ gilt. Wir können das umsetzen indem wir $\mathit{cpl}(\varphi, (i), (j)) = (\psi, \beta)$ setzen mit $\psi = \varphi \wedge (x_i \vee x_j) \wedge (\neg x_i \vee \neg x_j)$. Die Sequenz β muss dann nur die $q(|\varphi|)$ ersten Bits eines Zertifikats auslesen; der Suffix ist redundant und entsteht nur weil ψ (und damit die Zertifikate für ψ) länger als φ ist. Es lässt sich leicht *cpl* auf längere Eingabesequenzen erweitern.

Mit dem *building block*, der Funktion *join* und der Funktion *cpl* ist es nun möglich, beliebige 3CNFSAT-Formeln zusammenzubauen. Das gilt nicht nur für die hier gewählten Definition konkret für `rSAT`, sondern sogar für alle universellen NP-Relationen. Ferner ist es sogar möglich, aus den Zertifikaten wieder eine erfüllende Belegung für die 3CNFSAT-Formel projektiv „herauszulesen". Intuitiv wird damit ersichtlich, dass die universellen NP-Relationen auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig sind. Die spezielle starke Form der Reduzierbarkeit, welche das einfache „Auslesen" aus den Zertifikaten zulässt, formalisieren Agrawal und Biswas wie folgt:

Definition 3.29 (Projektive Levin-Reduktion)**.** Seien Q und R zwei strenge NP-Relationen. Die NP-Relation Q lässt sich über eine *projektive Levin-Reduktion* auf R reduzieren, wenn eine Funktion $f \in \mathrm{FP}$ existiert, welche die folgenden Bedingungen erfüllen:

(1) $f(x) = (z, \alpha)$ wobei $x, z \in \Sigma^*$ und $\alpha \in \mathbb{N}_{>0}^{q(|x|)}$ ist eine Sequenz von positiven paarweise verschiedenen Indizes der Länge $q(|x|)$.
(2) $\{y[\alpha] \mid y \in \mathit{set}\text{-}R(z)\} = \mathit{set}\text{-}Q(x)$. $\lhd$

Es ist leicht zu sehen, dass projektive Levin-Reduktionen eine reflexive und transitive Ordnung auf den NP-Relationen bilden, und aus einer projektiven Levin-Reduktion von Q auf R auch $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ folgt.

Abschließend können universelle NP-Relationen als harte Relationen bezüglich projektiver Levin-Reduktion charakterisiert werden:

Satz 3.30 (Agrawal und Biswas 1992a)**.** *Sei R eine strenge NP-Relation. Folgende Aussagen sind äquivalent:*

(1) R ist eine universelle Relation, i.e. hat einen building block*, ist* joinable *und ist* couplable*.*

(2) jede NP-Relation lässt sich mittels einer projektiven Levin-Reduktion auf R reduzieren.

Diese Äquivalenz gilt nur im unrelativierten Fall.

Agrawal und Biswas (1992a) haben explizit verifiziert, dass rSAT, r3CNFSAT, rHAMCYCLE, rINDSET, rKNAPSACK und eine Variante rMAXCUT′ vom maximalen Schnitt alle universell sind.

Da aus projektiver Levin-Reduktion auch Levin-Reduktion folgt, sehen wir dass Universalität eine Verstärkung von $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit ist.

Korollar 3.31. *Wenn R eine universelle NP-Relation ist, dann ist R $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig für* FNP*. Diese Aussage gilt nur im unrelativierten Fall.*

An dieser Stelle sei auch noch einmal darauf hingewiesen, dass Agrawal und Biswas (1992a) das Ziel verfolgten, eine gemeinsame Struktur aller natürlichen NP-vollständigen Probleme zu erfassen. Im Speziellen vermuten sie, dass tatsächlich für alle natürlichen NP-vollständigen Entscheidungsprobleme L auch mindestens eine NP-Relation R für L existiert, die „hinreichend“ natürlich ist. Sie machen das an zwei Gründen fest: zum einen die intuitive Einsicht, dass *joinability* und *coupability* sehr natürlich wirkende strukturelle Eigenschaften sind. Zum anderen anhand empirischer Belege: In ihrer Arbeit präsentierten sie fünf konkrete Beispiele aus verschiedenen Problembereichen, darunter Logik (Erfüllbarkeit), Zahlentheorie (Knapsack) und Graphentheorie (Hamiltonkreis, Unabhängige Menge, Größter Schnitt). Diese Beispiele deuten den Autoren zufolge darauf hin, dass das Konzept universeller Beziehungen nicht auf bestimmte Kategorien von NP-vollständigen Problemen beschränkt ist. Konkret wurde aber nicht exhaustiv untersucht, welche der natürlichen NP-Relationen universell sind und welche nicht, und Agrawal und Biswas (1992a) lassen das als Frage offen. Unten werden wir sehen, dass Färbungsprobleme möglicherweise keine natürliche NP-Relationen zulassen.

Die Abbildung 1 fasst noch einmal die relative Stärke der Vollständigkeitsbegriffe zusammen. Die eingezeichneten Trennungen weren im nächsten Abschnitt erläutert.

Lemma 3.32. *Es gelten die in Abbildung 1 eingezeichneten Inklusionen.*

Beweis. Mit Lemma 3.26 bleiben nur noch drei nichttriviale Implikationen offen:

1. Falls R universell ist und die beteiligten Funktionen *join* und *clp* injektiv und p-invertierbar sind, dann ist schon aus Satz 3.30 klar, dass R auch projektiv Levin-vollständig ist. Es existiert also für NP-Relation Q eine Funktion $f \in \mathrm{FP}$, und es gilt für eine Q-Instanz x dass $f(x) = (z, \alpha)$. Hier ist z die R-Instanz, auf die reduziert wird. Aus dem Beweis des Satzes von Agrawal und Biswas (1992a) geht hervor, dass sich dieses z aus der kombinierten Anwendung von *join* und *cpl* entsteht, also lässt sich auch aus z wieder aufgrund p-Invertierbarkeit die Instanz x zurückgewinnen. Es lässt sich leicht sehen, dass sich so eine $\leq_{\mathrm{L,1,i}}^{\mathrm{p}}$-Reduktion von Q nach R konstruieren lassen kann.

2. Falls R universell ist und die beteiligte Funktione *join* injektiv und p-invertierbar ist, dann ist $\mathrm{Proj}(R)$ auch paddable, und damit p-isomorph zu SAT (Agrawal und Biswas 1992a, Thm. 8.2). Das lässt sich leicht nachvollziehen: durch durch an-*join*-en von Dummy-Instanzen an Instanz x lassen sich beliebige Werte in x hineincodieren, und durch die p-Invertierbarkeit wieder extrahieren.

3. Sei L eine Menge. Ist L p-isomorph zu SAT, dann existiert auch eine NP-Relation R' sodass $\mathrm{Proj}(R') = L$ und R' ist universell. Diese Aussage ist eine einfache Generalisierung von Beobachtung 3.22. □

Trennungen

Zunächst halten Agrawal und Biswas (1992a) fest, dass die Universalität eine Eigenschaft ist, die sogar bezüglich Problemen gilt, die mutmaßlich nicht p-isomorph sind. Angenommen, es existiert eine Einwegfunktion $f \in \mathrm{FP}$, das heißt f ist injektiv, aber f ist nicht p-invertierbar. Unter der *Encrypted Complete Set Conjecture* (ECSC) wird die Vermutung genannt, nach der die Menge

$$h(\mathtt{SAT}) = \{h(\varphi) \mid \varphi \in \mathtt{SAT}\}$$

nicht paddable ist, damit also auch nicht p-isomorph zu SAT ist. Gleichzeitig ist $\mathtt{SAT} \leq_{\mathrm{m}}^{\mathrm{p}} h(\mathtt{SAT})$ über Reduktionsfunktion h, und damit $h(\mathtt{SAT})$ auch $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig. Damit ist $h(\mathtt{SAT})$, zu verstehen als eine „verschlüsselte“ Variant zu SAT, ein vermutetes Gegenbeispiel für die Berman–Hartmanis-Isomorphievermutung IC. Gleichzeitig ist leicht zu sehen, dass eine entsprechende

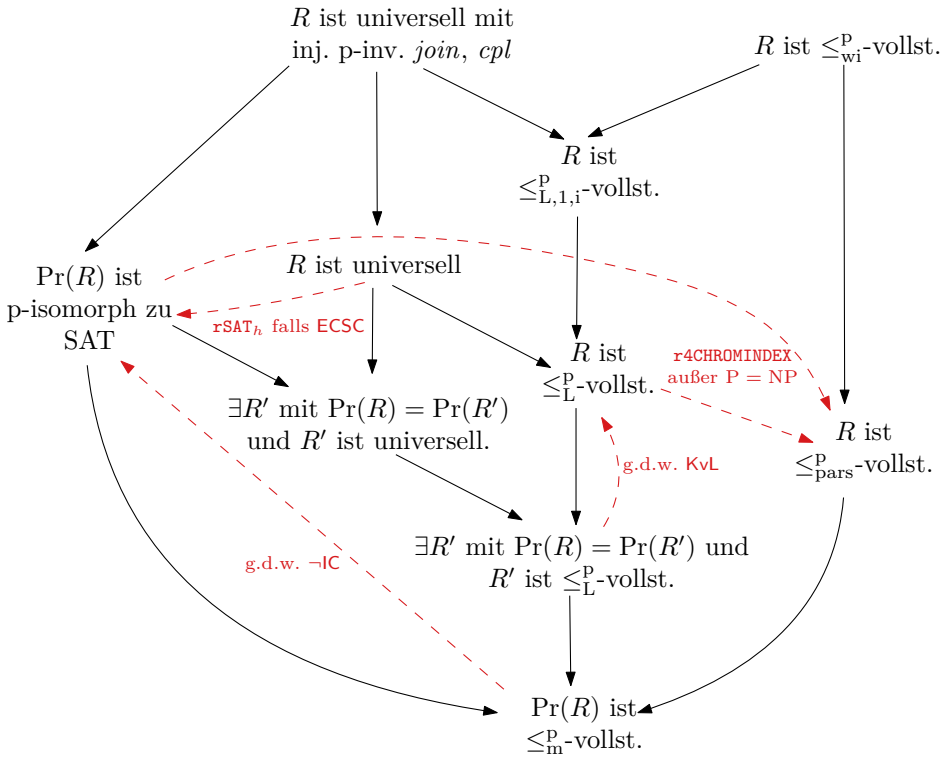


Abbildung 1: Implikationen zwischen den Vollständigkeitsbegriffen. Ein unterbrochener Pfeile von A nach B sagt aus, dass ein Gegenbeispiel R für die Implikation A $\Rightarrow$ B existiert, also eine NP-Relation R die A erfüllt und gleichzeitig $\neg$B erfüllt.

natürliche NP-Relation

$$\mathtt{rSAT}_h = \{(z,(\varphi,w)) \mid z = h(\varphi), \text{ und } w \text{ ist erfüllende Belegung für } \varphi\}$$

sogar universell ist. Wir haben also

Beobachtung 3.33. *Angenommen* ECSC *dann existiert eine NP-Relation R die universell ist, aber* $\mathrm{Proj}(R)$ *ist nicht p-isomorph zu* SAT.

Nun werden wir uns auf die sparsamen Reduktionen konzentrieren. Zu einem Graphen G mit Knotenmenge $\{0,1,\ldots,n-1\}$ können wir einen *Schnitt* als einen String $w \in \Sigma^*$ schreiben, wobei $V_0 = \{i \mid i < n, w[i] = 0\}$ und $V_1 = \{i \mid i < n, w[i] = 0\}$ den Graphen in zwei Teile partitioniert. Einem Schnitt w können wir dann ein Gewicht zuordnen: die Anzahl an Kanten in G die zwischen V_0 und V_1 laufen. Sei nun

$$\begin{aligned}\mathtt{rMAXCUT} = \{((G,r),w) \mid\ & G \text{ ist Graph mit Knotenmenge } \{0,1,\ldots,n-1\},\\ & \text{und } w \in \Sigma^n \text{ ist ein Schnitt mit Gewicht} \geq r\}.\end{aligned}$$

Diese natürliche NP-Relation ist ein Beispiel für eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Relation, die aber nicht $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig ist. Die $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit von lässt sich leicht aus den üblichen $\leq^{\mathrm{p}}_{\mathrm{m}}$-Reduktionen verstärken.

Wir behaupten nun dass $\mathtt{rSAT} \not\leq^{\mathrm{p}}_{\mathrm{pars}}$. Angenommen es existiert eine solche geizige Reduktion f. Beachte dass die SAT-Instanz $\varphi =$ „x_1“ genau eine erfüllende Belegung hat. Dann wäre

$$1 = |\mathit{set}\text{-}\mathtt{rSAT}(\varphi)| = |\mathit{set}\text{-}\mathtt{rMAXCUT}(f(\varphi))|.$$

Es lässt sich aber leicht sehen, dass $|\mathit{set}\text{-}\mathtt{rMAXCUT}(x)|$ für jede rMAXCUT-Instanz gerade sein muss: ist w Schnitt mit Gewicht $\geq r$, dann ist auch der komplementäre String $\overline{w}$ auch ein Schnitt mit Gewicht $\geq r$; die Mengen V_0 und V_1 werden einfach vertauscht. Damit erhalten wir den Widerspruch. Auf ähnliche Weise lässt sich zeigen, dass rMAXCUT auch nicht universell sein kann.

An dieser Stelle muss aber kritisch hervorgehoben werden, dass dieses Gegenbeispiel auf einem kontingenten „Hütchenspielertrick“ aufbaut: Die Schnitte w und $\overline{w}$ werden als unterschiedliche Zertifikate gehandhabt, *repräsentieren* doch aber die *identische* Partitionierung des Graphen. Das Problem löst sich auf, wenn anstelle der naiven Formulierung von rMAXCUT folgende Verfeinerung gewählt wird:

$$\begin{aligned}\mathtt{rMAXCUT}' = \{((G,r),w) \mid\ & G \text{ ist Graph mit Knotenmenge } \{0,1,\ldots,n-1\},\\ & \text{und } w \in \Sigma^n \text{ ist ein Schnitt mit Gewicht} \geq r, \text{ und startet mit } 0.\}.\end{aligned}$$

In anderen Worten, ein Schnitt für eine rMAXCUT′-Instanz hat immer den Knoten $0 \in V_0$. Dann ist auch möglich, eine sparsame Reduktion von rSAT auf rMAXCUT′ anzugeben, und auch möglich zu zeigen, dass rMAXCUT′ universell ist.

Ein filigraneres Beispiel ist Kantenfärbung: Wir werden zeigen dass das Problem der 4-Kantenfärbung nicht vollständig unter sparsamen Reduktionen ist, außer P = NP.

Zu einem Graphen $G = (V, E)$ mit Kantenmenge $E = \{0, 1, \dots, m-1\}$ können wir eine *k-Kantenfärbung* als String w der Länge m über dem Alphabet $\{1, 2, \dots k\}$ darstellen, wobei Kante j die Farbe $w[j]$ erhält. Wir wollen im Folgenden die Anzahl der möglichen Kantenfärbungen zählen, und sind dabei insbesondere nicht an redundanten Lösungen interessiert, die aus reiner Permutation der Farben entsteht. Wir setzen für eine *gültige* Färbung w daher voraus, dass w die unter Permutationen lexikographisch kleinste Färbung ist, in dem Sinne dass keine Permutation π auf $\{1, 2, \dots, k\}$ existiert sodass $\pi(w)$ lexikographisch kleiner ist als w. (Beachte: wir suchen *nicht* nach einer „global" lexikographisch kleinsten Färbung von G.) Definiere nun

$$\begin{aligned}\texttt{r4CHROMINDEX} = \{((G,k),w) \mid\ & G \text{ ist Graph mit Kantenmenge } \{0,1,\dots,m-1\} \\ & G \text{ hat maximalem Grad 4,} \\ & \text{und } w \in \{1,2\dots,4\}^m \text{ ist gültige Färbung mit 4 Farben}\}.\end{aligned}$$

Satz 3.34 (Cai und Govorov 2020 nach Edward und Welsh)**.** *Die NP-Relation* r4CHROMINDEX *ist nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig, außer* P = NP*.*

Skizze. Sei $\chi'(G)$ die minimale Anzahl an Farben, die zur Kantenfärbung eines Graphen G benötigt werden. Cai und Govorov können sämtliche Graphen charakterisieren, welche eine eindeutige (modulo Permutationen der Farben) 4-Kantenfärbung haben:

- Unter den Graphen mit $\chi'(G) = 4$ ist $K_{1,k}$ der einzige Graph mit eindeutiger Kantenfärbung. (Das ist der Satz von Thomason (1978).)
- Unter den Graphen mit $\chi'(G) = 3$ sind C_3 und $K_{1,3}$ die einzigen Graphen mit eindeutiger Kantenfärbung.
- Unter den Graphen mit $\chi'(G) = 2$ ist $K_{1,2}$ der einzige Graph mit eindeutiger Kantenfärbung.
- Unter den Graphen mit $\chi'(G) = 1$ ist $K_{1,1}$ der einzige Graph mit eindeutiger Kantenfärbung.

(In allen Fällen können isolierte Knoten ignoriert werden.) Damit kann also in Linearzeit überprüft werden, ob ein gegebener Graph G eine eindeutige Kantenfärbung mit 4 Farben zulässt. Sei $A \in P$ diese Menge der eindeutig färbbaren Graphen.

Mit diesem Fakt zeigen wir nun die Aussage. Angenommen, r4CHROMINDEX ist $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig, dann existiert auch eine sparsame Reduktion f von rSAT auf r4CHROMINDEX. Sei φ eine beliebige SAT-Formel, in der nur die Variablen $x_1, \dots, x_n$ vorkommen. Wir werden nun in Polynomialzeit entscheiden ob $\varphi \in$ SAT. Definiere eine zweite SAT-Formel

$$\varphi' = (\neg y \wedge \varphi) \vee (y \wedge \neg x_1 \wedge \neg x_2 \wedge \cdots \wedge x_n),$$

wobei y ein neues Variablensymbol ist. Es ist leicht zu sehen, dass φ' genau eine erfüllende Belegung mehr als φ hat.

Wir haben nun

$$\varphi \notin \texttt{SAT} \iff |\mathit{set}\text{-}\texttt{rSAT}(\varphi)| = 0 \iff |\mathit{set}\text{-}\texttt{r4CHROMINDEX}(f(\varphi))| = 1 \iff A \in \mathrm{P},$$

und damit SAT $\in$ P, P = NP. □

Leven und Galil (1983) zeigen, dass die Menge Proj(r4CHROMINDEX) $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig ist. Mit den Konstruktionen aus deren Beweis ist es leicht zu sehen, dass die NP-Relation r4CHROMINDEX auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. (Die wesentlichen Ideen werden unten kurz skizziert.) Es ist auch leicht zu sehen, dass Proj(r4CHROMINDEX) paddable ist, also auch p-isomorph zu SAT. Wir kommen zum Resultat:

Beobachtung 3.35. *Die NP-Relation* r4CHROMINDEX *ist* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig, und* Proj(r4CHROMINDEX) *ist p-isomorph zu* SAT*. Sie ist insbesondere nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig außer* P = NP*.*

Gleichzeitig ist nicht klar, ob sich dieses Ergebnis zur Universalität von r4CHROMINDEX verstärken kann. (Das würde Universalität von $\leq^{\mathrm{p}}_{\mathrm{pars}}$-Vollständigkeit trennen.) Weder ist ist klar, wie sich ein *building block* angeben kann, noch wie (für Aussage (2) von Satz **??**) sich eine projektive Levin-Reduktion von rSAT auf r4CHROMINDEX angeben kann. Die wesentliche Schwierigkeit liegt darin, die *projektive* Natur der projektiven Levin-Reduktion umzusetzen: aus den Färbungen bzw. Zertifikaten kann nicht Bit für Bit eine Lösung herausgelesen werden, wie sie die Definition **??** verlangt.

Dies sei im Folgenden am etwas einfacherem Fall der 3-Kantenfärbbarkeit (die auch $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig ist) illustriert; die wesentliche Idee überträgt sich auch auf k-Kantenfärbbarkeit, $k \geq 3$. Holyer (1981) zeigt die $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit der 3-Kantenfärbbarkeit, indem von 3SAT in CNF darauf reduziert wird. Das ist, gegeben eine 3CNFSAT-Formel φ in konjunktiver Normalform wird ein 3-regulärer Graph G konstruiert der 3-färbbar ist genau dann wenn φ erfüllbar ist. Wie üblich ist G aus einzelnen Gadgets zusammengesetzt welche spezielle (aussagenlogische) Aufgaben übernehen. Die „Verdrahtung" der einzelnen Gadgets erfolgt hierbei je über ein *Paar von zwei Kanten*. In einer 3-Kantenfärbung repräsentiert diese

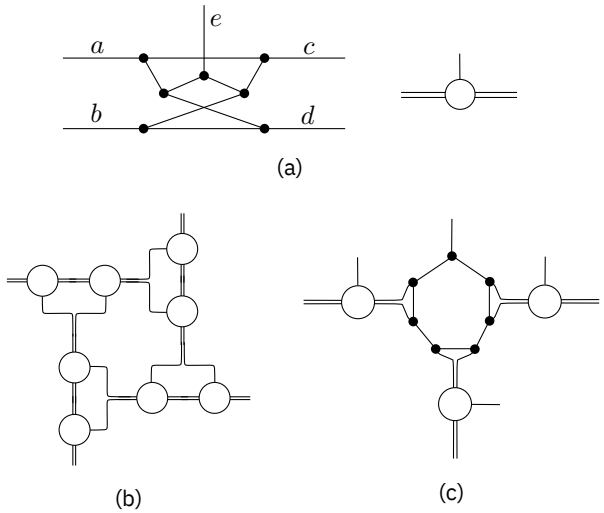


Abbildung 2: Die von Holyer (1981) verwendeten Gadgets um die NP-Vollständigkeit der 3-Kantenfärbbarkeit in 3-regulären Graphen zu zeigen.
(a) Das Gadget zum Invertieren. Beachte dass in einer gültigen Färbung die Kanten a und b die gleiche Farbe haben („wahr“) genau dann wenn c und d ungleiche Farben haben („falsch“). Außerdem haben entweder a, b, e oder c, d, e alle drei unterschiedliche Farben. Das Symbol rechts ist die schematische Darstellung dieses Gadgets in den Abbildungen (b) und (c).
(b) Gadget für je eine Variable. Beachte dass in einer gültigen Färbung alle Outputs entweder „wahr“ oder „falsch“ sind.
(c) Gadget für eine Klausel. In einer gültigen Färbung ist mindestens einer der drei Inputs „wahr“.

Paar den Wert „wahr“ wenn die zwei Kanten die gleiche Farbe haben, und „falsch“ wenn die zwei Kanten unterschiedliche Farben haben. Ein Gadget zum Invertieren konstruiert Holyer z.B. wie in Abbildung 2(a), wobei die Paare (a, b) und (c, d) die Belegungen übertragen. Aufbauend darauf lassen sich dann größere Gadgets konstruieren, welche Variablen bzw. Klauseln darastellen. Abbildung 2(b) zeigt z.B. ein Gadget für die Variablenbelegung, bei der jeder Output den gleichen Wahrheitswert (entweder alle „falsch“ oder alle „wahr“) hat.

Das zentrale Problem ist nun, dass sich selbst unter einer geeigneten Codierung der Färbungen in den Zertifikaten w nicht mit einem Bit aus dem Zertifikat w der von der Färbung „zugewiesener“ Wahrheitswert einer Variable ausgelesen werden kann. Mit einer flexibleren allgemeinen Levin-Reduktion lässt sich dies aber umsetzen (i.e. lese an zwei Stellen in w die zugewiesene Farbe von zwei Kanten aus und vergleiche die Farben). Ob `r4CHROMINDEX` universell im Sinne von Definition **??** ist, bzw. äquivalent vollständig bezüglich projektiven Levin-Reduktionen ist, sei hier offen gelassen und als Frage formuliert:

Frage 3.36. *Ist* `r4CHROMINDEX` *(bzw. eine geeignete natürliche Variante) universell?*

Die Frage lässt sich – im Hinblick auf die Separationen der Vollständigkeitsbegriffe – auch folgendermaßen verallgemeinern:

Frage 3.37. *Angenommen* P $\neq$ NP. *Existiert dann eine natürliche NP-Relation R die universell ist, aber nicht $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig ist?*

Je ein Argument spricht für bzw. gegen eine positive Beantwortung von Frage 3.36. Einerseits das oben schon skizzierte Argument, dass sich Färbbarkeiten offenbar nicht gut mit der projektiven Levin-Reduzierbarkeit verträgt. Es sei darauf hingewiesen, dass Agrawal und Biswas (1992a) in ihrer Arbeit zwar exemplarisch die Universalität vieler Suchprobleme aus verschiedensten kombinatorischen Bereichen gezeigt haben, Färbungsprobleme wurden hierbei aber nicht betrachtet. Inwiefern sich Universalität generalisieren lässt, indem das projektive „Auslesen“ von Werten aus den Zertifikaten abgeschwächt wird, z.B. über eine Art polynomialzeit-berechenbares Schema.

Andererseits ist es für Färbungsprobleme nicht *prinzipiell* unmöglich, Universalität zu zeigen. Beispielsweise ist das Problem `r3COL` der 3-Färbbarkeit eines Graphens durchaus als unverselle NP-Relation darstellbar, wenn wieder (wie schon bei `rMAXCUT'` oder `r4CHROMINDEX`) nach der unter Permutationen der Farben lexikographisch kleinsten Färbung (sortiert anhand der Knoten) gesucht wird. Ein Lehrbuch-Beweis der $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit über `3CNFSAT` $\leq^{\mathrm{p}}_{\mathrm{m}}$ `3COL` startet üblicherweise mit einem „Palette“-Gadget aus drei Knoten $v_{\mathrm{false}}, v_{\mathrm{true}} v_{\mathrm{base}}$. Vgl. Abbildung 3. In einer gültigen Färbung entspricht die Farbe von v_{true} dann der Farbe „wahr“. Nummeriert man in einer `3COL`-Instanz G_φ für `3CNFSAT`-Instanz φ die Knoten so um dass $v_{\mathrm{false}}, v_{\mathrm{true}} v_{\mathrm{base}}$ als erste Knoten G_φ sind, dann hat immer v_{false} die Farbe 1 und v_{true} die Farbe 2 (ansonsten existiert eine Permutation der Farben sodass die Färbung lexikographisch

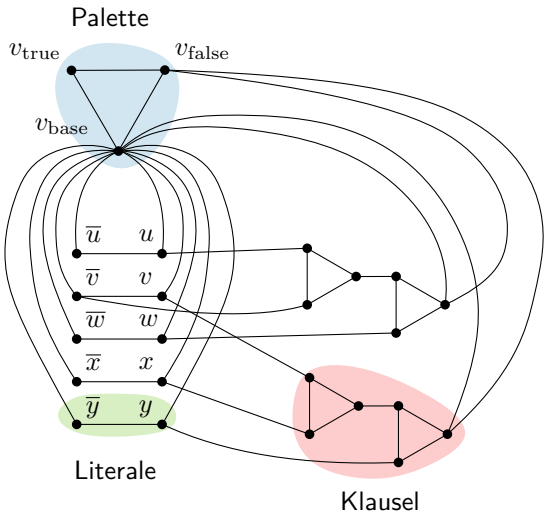


Abbildung 3: Reduktion der 3CNFSAT-Instanz $\varphi = (u \vee \overline{v} \vee \overline{w}) \wedge (v \vee x \vee y)$. Eine Dreifärbung dieses Graphen entspricht einer erfüllenden Belegung von φ. Beachte dass zu jeder Variable genau ein Knoten zu einem entsprechendem Literal (positiv bzw. negativ) die gleiche Farbe wie v_{true} hat, und der Knoten zum anderen Literal die gleiche Farbe wie v_{false} hat. Es ist leicht zu sehen, dass jedes Klausel-Gadget genau dann dreifärbbar ist, wenn mindestens eine der drei angeschlossenen Literale die gleiche Farbe wie v_{true} hat.

kleiner ist.) Eine projektive Levin-Reduktion kann also nun in einem Bit auslesen ob beliebiger Knoten v die Farbe „wahr“ hat, denn diese ist immer 1. Eine enstprechende Codierung des Zertifikats wäre z.B. von der Form $h(c_1)h(c_2)\cdots$ wobei $c_i \in \{1,2,3\}$ die Farbe des i-ten Knotens ist, und h eine one-hot-Codierung umsetzt, i.e. $h(1) = 100, h(2) = 010, h(3) = 001$. Es lässt sich dann leicht eine projektive Levin-Reduktion von `r3CNFSAT` auf `3COL` angeben.

Mit dieser letzten Beobachtung wollen wir dieses Kapitel über die NP-Suchprobleme, deren Beziehung zu Entscheidungsproblemen, und die zuletzt präsentierte Übersicht über die gemeinsamen Strukturen der vollständigen NP-Suchprobleme abschließen.

4 Suchprobleme und die Hypothese Q im Kontext des Pudlákschen Programms

In der Einleitung dieser Arbeit wurde bereits angedeutet, dass die Hypothese Q von Fenner u. a. große Nähe und Verwandtschaft zu Hypothesen hat, die Suchprobleme im Allgemeinen und Beweissystemen im Speziellen betreffen. Damit ergeben sich Beziehungen zu Hypothesen aus dem Pudlákschen Programm, insbesondere ¬SAT (also dass eine NP-vollständige Mengen mit p-optimalem Beweissystem für diese Menge existiert). In diesem Kapitel werden wir diese Beziehungen näher erarbeiten. Zur Erinnerung:

Vermutung 1.1 (Q, Fenner u. a. 2003)**.** *Für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ eine akzeptierende Berechnung von $N(x)$ ist.*

Im Kapitel werden wir uns grob folgenden drei Desiderata widmen: erstens, nähern wir uns in Abschnitt **??** erneut der Frage zwischen Levin- und Karp-Vollständigkeit bzw. der Hypothese KvL aus vorigem Kapitel. Insbesondere analysieren wir die Beziehungen von KvL zu Q und versuchen, KvL in das Pudlák'sche Programm einzuordnen.

Zweitens, in Abschnitt **??**, verallgemeinen wir Charakterisierungen Q, die sich insbesondere auf Suchprobleme und deren Beweissysteme bzw. „Zertifikatsschemata" beziehen. Insbesondere zeigen wir für eine große Klasse von vollständigen NP-Suchproblemen R (nämlich jene die Levin-paddable sind) dass das zu R assoziierte *Standardbeweissystem* ((x, y) mit $R(x, y)$ ist ein Beweis für x) p-optimal ist, genau dann wenn Q. Damit wird die p-Optimalität des entsprechenden Standardbeweissystems zu einer Invariante, die entweder für *alle* Levin-paddable NP-Suchprobleme zutrifft, oder für *keins*.

Drittens ergänzen wir im gesamten Verlauf dieses Kapitels das Pudlák'sche Programm um weitere Hypothesen, sodass Abbildung **??** der Beziehungen zwischen den Pudlákschen Hypothesen vergrößert wird. Damit erreichen wir den Stand, der in Abbildung **??** dargestellt wird.

Für alle dieser drei Desiderata ist es zunächst notwendig, auf die Hypothese Q einzugehen. Fenner u. a. (2003) beobachten, dass das Invertieren von surjektiven ehrlichen FP-Funktionen eine erstaunlich robuste Aussage ist, die eine Vielzahl von äquivalenten „fundamentalen" (Fenner u. a. 2003) Charakterisierungen aus der Komplexitätstheorie zulässt, so zum Beispiel die effiziente Lösbarkeit von TFNP-Suchproblemen, oder das effiziente Ausrechnen akzeptierender Rechenwege einer totalen NPTM. Wir können jetzt schon festhalten, dass die aktuelle Forschung diese Hypothese als sehr stark einschätzt, und eher die negative Beantwortung (i.e. ¬Q) vermutet.

Satz 4.1 (Äquivalente Formulierungen der Hypothese Q; Fenner u. a. 2003)**.** *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*.*

(2) $\mathrm{NPMV}_t \subseteq_{\mathrm{c}} \mathrm{FP}$*.*

(3) $\mathrm{TFNP} \subseteq_{\mathrm{c}} \mathrm{FP}$*.*

(4) $\mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$ *und* $\mathrm{NPMV}_t \subseteq_{\mathrm{c}} \mathrm{NPSV}_t$*.*

(5) Jede surjektive ehrlichen Funktion $f \in \mathrm{FP}$ ist p-invertierbar.

(6) Für jede Menge $L \in \mathrm{P}$ und jede NPTM N mit $L(N) = L$ existiert eine Funktion $h \in \mathrm{FP}$ mit

$$x \in L \implies N(x) \text{ akz. mit Rechenweg } h(x).$$

Fenner u. a. (2003) und Messner (2000) characterisieren Q noch durch zwei weitere Formen, diesmal über je eine Aussage über die Menge SAT:

Satz 4.2. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*.*

(2) (Fenner u. a. 2003) Für jede NPTM N mit $L(N) = \mathrm{SAT}$ existiert eine Funktion $h \in \mathrm{FP}$ sodass

$$N(\varphi) \text{ akz. mit Rechenweg } w \implies h(w) \text{ ist eine erfüllende Belegung für } \varphi.$$

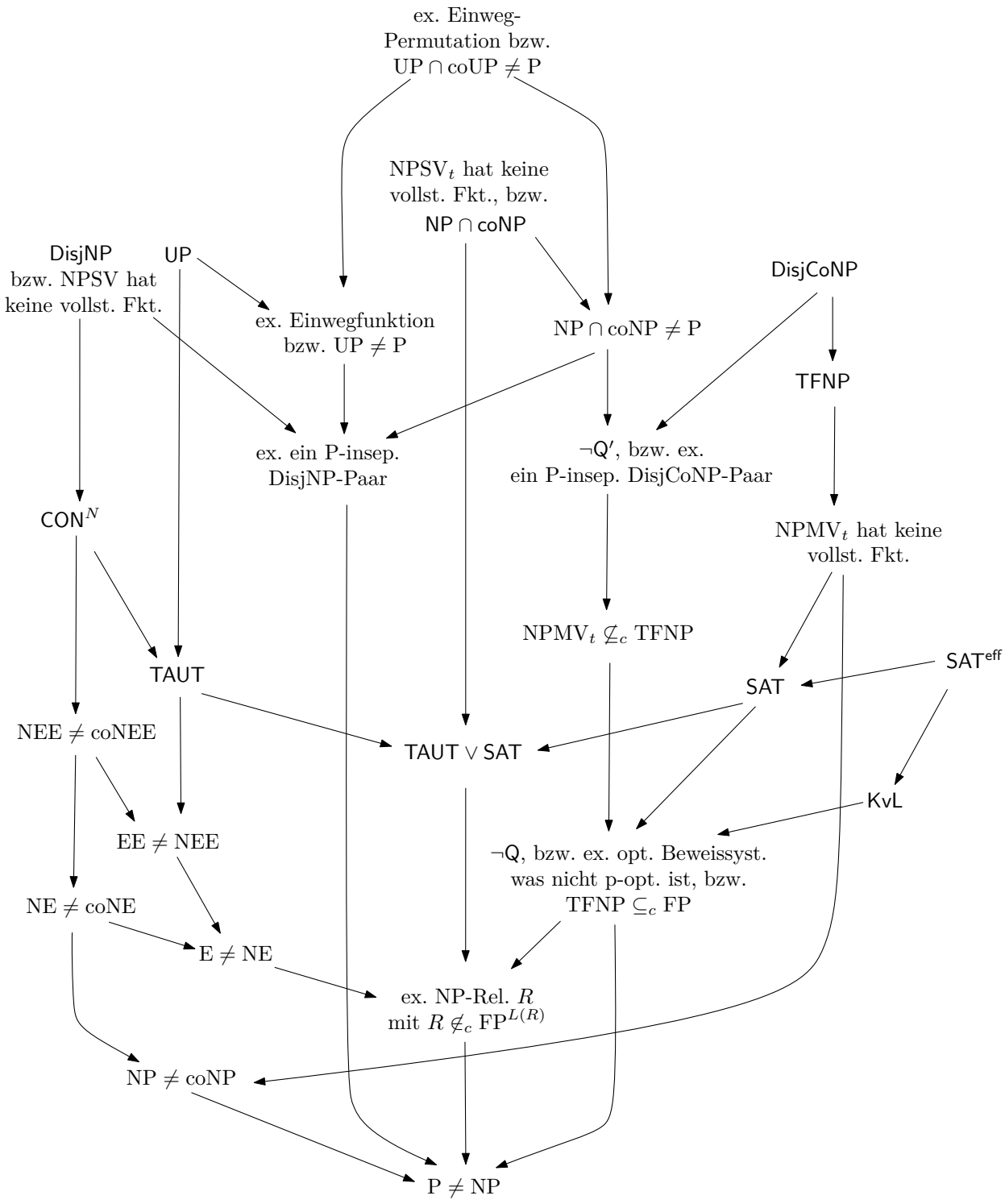


Abbildung 4: Bekannte (relativierenden) Implikationen zwischen den betrachteten Hypothesen und weiteren Aussagen. Satz 4.23 gibt Belegstellen für jede dieser Implikationen an.

(3) (Messner 2000) Das Standardbeweissystem sat

$$sat(\varphi, w) = \begin{cases} \varphi & \text{wenn } w \text{ eine erfüllende Belegung für } \varphi \text{ ist} \\ \bot & \text{sonst.} \end{cases}$$

für rSAT *ist p-optimal.*

Dieser Satz relativiert nicht.

In anderen Worten sagt Aussage (2) aus, dass es modulo Umcodieren nur einen einzigen SAT-Solver gibt, und insbesondere alle SAT-Solver äquivalent zum trivialen Solver ist, welcher nur alle möglichen Belegungen ausprobiert. Die Aussage (3) macht eine analoge Aussage über Beweissysteme: egal wie komplex ein Beweissystem h für SAT ist, wir können immer einen h-Beweis für φ in eine erfüllende Belegung für φ (quasi ein trivialer Beweis für $\varphi \in$ SAT) transformieren.

In Abschnitt **??** werden wir sehen, dass sich die obigen Charakterisierungen auf weitere (aber möglicherweise nicht alle) vollständigen NP-Relationen generalisiert, womit insbesondere auch die beiden Charakterisierungen von Fenner u. a. und Messner zu einer *relativierbaren* Variante verallgemeinert werden. Hierfür führen wir jetzt schon den Begriff eines Standardbeweissystems formal ein.

Definition 4.3 (Standardbeweissystem)**.** Sei R eine NP-Relation. Wir definieren bezüglich R das *Standardbeweissystem* std_R für $\mathrm{Proj}(R)$ wie folgt:

$$std_R(w) = \begin{cases} x & \text{wenn } w = (x, y) \text{ und } (x, y) \in R, \\ \bot & \text{sonst.} \end{cases}$$ ◁

Damit ist, wie durch die Formulierung oben suggeriert, $sat = std_{\text{rSAT}}$. Bevor wir nun mit einer Diskussion zwischen Karp-Vollständigkeit und Levin-Vollständigkeit fortsetzen, schließen wir diesen Einstieg mit folgender einfachen Beobachtung ab:

Beobachtung 4.4. *Für jede NP-Relation R ist das Standardbeweissystem std_R ehrlich.*

Beweis. Sei q die Zertifikatsschranke von R, und sei $w = (x, y)$ gegeben sodass $std_R(x, y) = x$. An dieser Stelle müssen wir auf die konkrete Codierung von Beweisen $w = (x, y)$ eingehen. Wie in **??** beschrieben, codieren wir Tupel in einer solchen Weise sodass

$$|w| = |(x, y)| = 2(|x| + |y| + 2) = 2|x| + 2|y| + 4.$$

Da $(x, y) \in R$ gilt für y auch $|y| \leq q(|x|)$. Damit also

$$|w| \leq 2|x| + 2q(|x|) + 4 \leq q'(|x|) = q'(|std_R(w)|).$$

für ein geeignetes Polynom q', wie gewünscht. □

4.1 Karp-Vollständigkeit vs. Levin-Vollständigkeit

Wir wiederholen hier erneut die zentrale offene Frage und Vermutung aus Abschnitt **??**:

Frage 3.20. *Wenn* $\mathrm{Proj}(R)$ *eine* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständige Menge für* NP *ist, ist dann auch R eine* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständige NP-Relation für* FNP*?*

Vermutung 3.21 (Karp-vs-Levin-Vermutung; KvL)**.** *Es existiert eine NP-Relation R sodass* $\mathrm{Proj}(R) \leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständig für* NP *ist, aber R ist nicht* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* FNP*.*

Ich möchte argumentieren, dass die obige Frage bzw. Vermutung eng mit der Hypothese Q zusammenhängt. Im speziellen werden wir sehen, dass die Hypothese Q so charakterisiert werden kann, dass sie *fast* der Vermutung KvL entspricht.

Satz 4.5. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*: für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in$ FP sodass $g(x)$ eine akzeptierende Berechnung von $N(x)$ ist.*

(2) Für jedes Paar von NP-Relationen A, B und jede Funktion $f \in$ FP gilt:

$$\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(B) \text{ via } f \iff A \leq^{\mathrm{p}}_{\mathrm{L}} B \text{ via Reduktionsfunktion } f.$$

Beweis. (1) $\Longrightarrow$ (2): Die Richtunng von rechts nach links ist klar. Für die andere Richtung sei $\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(B)$ mit A, B NP-Relationen. Sei q hierbei das Polynom was die Zertifikatslänge in A begrenzt. Wir wollen nun eine Levin-Reduktion von A auf B angeben. Sei $f \in$ FP die Funktion, welche die Reduktion $\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(B)$ realisiert.

Definiere folgende NPTM N, die wie folgt auf Eingabe w arbeitet:

```
1 wenn w nicht von der Form (x, y') dann akzeptiere
2 (x, y') ← w
3 wenn (f(x), y') ∉ B dann akzeptiere
4 sonst
    (Ab hier gilt f(x) ∈ Pr(B) und x ∈ Pr(A))
5   Rate nichtdeterministisch y ∈ Σ^{q(|x|)}
6   Akzeptiere genau dann wenn (x, y) ∈ A.
```

Es ist leicht zu sehen dass $L(N) = \Sigma^*$. Nach (1) existiert nun also eine Funktion g sodass, für alle w, $g(w)$ eine akzeptierender Rechwenweg von $N(w)$ ist.

Damit lässt die Levin-Reduktion von A auf B angeben: wähle f als Reduktionsfunktion, und definiere g' als Translationsfunktion, welche aus dem akzeptierenden Rechenweg $g(x, y')$ das geratene Zertifikat y von Zeile 5 ausliest. Dann gilt

$$\begin{aligned}&(f(x), y') \in B \implies N(x, y') \text{ akz. auf einem Rechenweg in Z. 6, ratet } y\\ &\implies (x, y) = (x, g'(x, y')) \in A\end{aligned}$$

wie gewünscht. Wir haben $A \leq_{\mathrm{L}}^{\mathrm{p}}$ via f, g'.

(2) $\implies$ (1): Sei n eine NPTM mit $L(N) = \Sigma^*$. Definiere die NP-Relation

$$A = \{(x, w) \mid x \in \Sigma^*,\ N(x) \text{ akz. auf Rechenweg } w\},$$

und die NP-Relation

$$B = \{(x, \varepsilon) \mid x \in \Sigma^*\}.$$

Es ist leicht zu sehen das $\mathrm{Proj}(A) = L(N) = \Sigma^* = \mathrm{Proj}(B)$ und dass $\mathrm{Proj}(A) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(B)$ über die Identitätsfunktion. Nach Annahme (2) lässt sich nun diese Reduktion zu einer Levin-Reduktion $A \leq_{\mathrm{L}}^{\mathrm{p}} B$ verstärken, wobei die Reduktionsfunktion die Identitätsfunktion ist, und $h \in \mathrm{FP}$ die Translationsfunktion. Wir haben also

$$(x, \varepsilon) \in B \implies (x, g(x, \varepsilon)) \in A \implies N(x) \text{ akz. auf Rechenweg } h(x, \varepsilon).$$

Definieren wir nun $g(x) = h(x, \varepsilon)$ haben wir (1) gezeigt: für alle x akzeptiert $N(x)$ auf $g(x)$. □

Beachte, dass in Aussage (2) die Implikation von rechts nach links ohnehin immer gilt. Damit lässt sich Aussage (2) auch so formulieren, dass jede Karp-Reduktion zu einer Levin-Reduktion verstärkt werden kann, indem zur Reduktionsfunktion f eine geeignete Translationsfunktion g hinzugefügt wird. Mit dieser Charakterisierung folgt auch unmittelbar, dass $\mathtt{Q}$ hinreichend für $\neg\mathtt{KvL}$ ist.

Satz 4.6. $\mathtt{KvL} \implies \neg\mathtt{Q}$.

Beweis. Wir zeigen die Kontraposition, und starten mit der Voraussetzung $\mathtt{Q}$. Wir wollen nun $\neg\mathtt{KvL}$ zeigen. Sei hierfür R eine beliebige NP-Relation sodass $\mathrm{Proj}(R)$ $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig ist. Damit gilt also schon für alle weiteren NP-Relationen A, dass $\mathrm{Proj}(A) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(R)$. Nach Satz 4.20 gilt also auch die Aussage 4.20(6), und damit $A \leq_{\mathrm{L}}^{\mathrm{p}} R$. Also ist R auch $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig, wie gewünscht und wir haben $\neg\mathtt{KvL}$ gezeigt. □

Was sind natürlich notwendige Bedingungen für die Hypothese $\mathtt{KvL}$? Diese Frage erscheint tatsächlich wesentlich schwieriger als gedacht. Insbesondere scheint es unklar, ob aus irgend einer von Pudláks Hypothesen die Aussage $\mathtt{KvL}$ folgt.

Besonders interessant erscheint aber die Beziehung zur Hypothese $\neg\mathtt{Q}$, also genau die Umkehrung von Satz 4.6, ist ja die Aussage (2) vom Satz **??** *fast* in der Form „Karp-Vollständigkeit = Levin-Vollständigkeit“ ist. Betrachten wir hier exemplarisch den Fall Relationen für $\mathtt{SAT}$. Ich vermute, dass $\neg\mathtt{Q} \Rightarrow \mathtt{KvL}$; um das zu plausibilisieren möchte ich zeigen, dass $\neg\mathtt{Q} \wedge \neg\mathtt{KvL}$ unwahrscheinlich ist.

Starten wir mit $\neg\mathtt{Q}$, dann gilt mit Satz 4.2 für alle Funktionen $h \in \mathrm{FP}$

$$N(\varphi) \text{ akz. mit Rechenweg } w \implies (\varphi, h(\varphi, w)) \in \mathtt{rSAT}. \tag{4.1}$$

In anderen Worten: es existiert zwar eine NPTM N welche $\mathtt{SAT}$ entscheidet, aber aus den akzeptierenden Rechenwegen w von $N(x)$ auf $x \in \mathtt{SAT}$ kann nicht effizient eine akzeptierende Belegung für x abgeleitet werden.

Wir können N äquivalent als NP-Relation R_N repräsentieren, mit $(\varphi, w) \in R_N$ genau dann wenn $N(x)$. mit Rechenweg w akzeptiert. Damit kann Gleichung 4.1 so verstanden werden, dass $\mathtt{rSAT} \not\leq_{\mathrm{L}}^{\mathrm{p}} R_N$ *falls die Reduktionsfunktion f die Identitätsfunktion ist.*

Unter der Annahme $\neg\mathtt{KvL}$ existiert nun eine Levin-Reduktion $\mathtt{rSAT} \leq_{\mathrm{L}}^{\mathrm{p}} R_N$ mit Reduktions- bzw. Translationsfunktion f, g. Das ist zunächst kein Widerspruch, denn es könnte ja $f \neq \mathrm{id}$. Gleichzeitig wäre die Existenz einer solchen Reduktion überraschend. Wir hätten nach Definition

$$N(f(\varphi)) \text{ akz. mit Rechenweg } w \implies \varphi \text{ wird von Belegung } g(\varphi, w) \text{ erfüllt}. \tag{4.2}$$

Einerseits ist es also nicht möglich, aus dem Rechenweg w effizient eine akzeptierende Belegung für $f(\varphi)$ zu bestimmen, obwohl w bezeugt dass $f(\varphi)$ erfüllbar ist. Andererseits reicht der „Beweis“ w aber aus, um (zusammen mit der Information φ) effizient wieder eine erfüllende Belegung für φ zu berechnen. Das plausibilisiert zwar einen Widerspruch, bzw. dass

$\neg\mathtt{Q} \wedge \neg\mathtt{KvL}$ wahrscheinlich falsch ist, ist aber natürlich kein solcher. Die Umkehrung von Satz 4.6 bleibt offen.

Dennoch vermute ich, dass solche Funktionen f, g nicht jeweils für alle NPTM N mit $L(N) = \mathtt{SAT}$ existieren können. Tatsächlich können wir die eben formulierte Vermutung auch in der Theorie der Beweissystemen formulieren: hierfür können wir die beiden Aussagen aus Gleichung 4.2 je als Aussagen über „Beweissysteme" verstehen. Links ist der Rechenweg w der „Beweis" für $f(\varphi) \in \mathtt{SAT}$ über den Verifiktor N, und rechts ist $g(\varphi, w)$ die erfüllende Belegung für φ, also ein *sat*-Beweis für $\varphi \in \mathtt{SAT}$.

Um diese Idee nun zu formalisieren, definieren wir zunächst eine abgeschwächte Variante der p-Simulation.

Definition 4.7. Seien h, h' Beweissysteme für L. Das Beweissystem h *p-simuliert effektiv* h' falls Funktionen $f, g \in \mathrm{FP}$ existieren sodass

(1) $x \in L \iff f(x) \in L$,

(2) $h'(w) = f(x) \implies h(g(x, w)) = x$.

Wir schreiben in diesem Fall auch $h' \leq^{\mathrm{p}}_{\mathrm{eff}} h$. $\lhd$

In anderen Worten, falls $h' \leq^{p}_{\mathrm{eff}} h$, dann kann h zwar nicht *jeden* h'-Beweis w für $x \in L$ in einen h-Beweis für (das gleiche) x effizient umrechnen, es kann aber zumindest alle *relevanten* h'-Beweise effizient umrechnen, nämlich für jedes $x \in L$ die h'-Beweise für $f(x)$ in h-Beweise für x. Klar ist: p-Simulation impliziert effektive p-Simulation impliziert Simulation unter Beweissystemen.

Die obige Intuition lässt sich also folgendermaßen formulieren: ich vermute, dass das Standardbeweissystem *sat* nicht jedes Beweissystem effektiv p-simulieren kann, insbesondere nicht jenes was von N induziert wird. Wir können diese Vermutung auch allgemeiner ohne Bezugnahme auf $\mathtt{SAT}$ bzw. *sat* formulieren:

Vermutung 4.8 ($\mathtt{KvL}$ formuliert unter Beweissystemen)**.** *Für jede NP-Relation R mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem* $\mathrm{Proj}(R)$ *kann std_R nicht alle anderen optimalen Beweissysteme für* $\mathrm{Proj}(R)$ *effektiv p-simulieren.*

Beachte, dass in der obigen Formulierung *std_R nicht* „effektiv p-optimal" ist in dem Sinn, dass *jedes* Beweissystem effektiv p-simuliert werden kann, sondern eben nur die optimalen Beweissysteme bzw. genau jene mit kurzen Beweisen.

Dass die Formulierung der Vermutungen 3.21 und 4.8 äquivalent sind, zeigt folgende Beobachtung:

Beobachtung 4.9. *Folgende Aussagen sind äquivalent:*

(1) Für jede NP-Relation R mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem $\mathrm{Proj}(R)$ *ist $R \leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig.*

(2) Für jede NP-Relation Q mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem $\mathrm{Proj}(Q)$ *kann std_Q jedes optimale Beweissystem h für* $\mathrm{Proj}(Q)$ *effektiv p-simulieren.*

Beweis. (1)⇒(2): Sei R eine NP-Relation mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem $\mathrm{Proj}(R)$. Wir zeigen, dass std_R jedes andere optimale Beweissystem h effektiv p-simulieren kann. Nachdem h optimal ist, hat es auch kurze Beweise (Beob. 2.13): für jedes $x \in \mathrm{Proj}(R)$ existiert ein h-Beweis w mit $|w| \leq q(|x|)$ für geeignetes Polynom q. Definiere

$$R_h = \{(x, w) \mid |w| \leq q(|x|), h(w) = x\}.$$

Diese Relation ist offenbar eine NP-Relation und $\mathrm{Proj}(R_h) = \mathrm{Proj}(R)$ und damit ist $\mathrm{Proj}(R_h)$ auch $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig. Nach Voraussetzung (1) ist also R_h auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig. Insbesondere gilt also auch $R \leq^{\mathrm{p}}_{\mathrm{L}} R_h$. Damit existieren also Funktionen $f, g \in \mathrm{FP}$ sodass $x \in \mathrm{Proj}(R) \leftrightarrow f(x) \in \mathrm{Proj}(R)$ und

$$(f(x), w) \in R_h \implies (x, g(x, w)) \in R.$$

Nach Definition gilt also

$$h(w) = f(x) \implies std_R(g(x, w) = x,$$

und damit ist $h \leq^{\mathrm{p}}_{\mathrm{eff}} std_R$.

(2)⇒(1): Sei R eine NP-Relation wobei $\mathrm{Proj}(R) \leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig ist. Wir zeigen nun, dass R auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Sei hierfür Q eine beliebige NP-Relation; wir wollen $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ zeigen.
Aus der $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit folgt unmittelbar die Existenz einer Reduktionsfunktion f mit

$$x \in \mathrm{Proj}(Q) \iff f(x) \in \mathrm{Proj}(R).$$

Definiere

$$h(w) = \begin{cases} x & \text{falls } w = (x, y) \text{ und } (f(x), y) \in R \\ \bot & \text{sonst.} \end{cases}$$

Wir zeigen, dass h ein Beweissystem für $\mathrm{Proj}(Q)$ ist. Es ist offenbar dass $h \in \mathrm{FP}$. Die Funktion h ist korrekt: wenn $h(x, y) = x$ dann ist $f(x) \in \mathrm{Proj}(R)$ und nach Eigenschaft von f auch $x \in \mathrm{Proj}(Q)$. Die Funktion h ist vollständig: Sei $x \in \mathrm{Proj}(Q)$. Dann ist schon $f(x) \in \mathrm{Proj}(R)$ und es gibt ein y mit $(f(x), y) \in R$. Also ist (x, y) ein h-Beweis für x.

Außerdem ist klar, dass h kurze Beweise hat, damit ist h auch optimal (Beob. 2.13). Damit gilt nach (2) nun, dass $h \leq^{\mathrm{p}}_{\mathrm{eff}} std_Q$. Also existieren Funktionen $f', g' \in \mathrm{FP}$ sodass

$$x \in \mathrm{Proj}(Q) \iff f'(x) \in \mathrm{Proj}(Q), \quad h(w) = f'(x) \implies std_Q(g'(x,w)) = x.$$

Das reicht aus, $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ zu zeigen: wähle $f''(x) = f(f'(x))$ als Reduktionsfunktion, dann gilt

$$(f''(x), y) \in R \implies (f(f'(x)), y) \in R \implies h(\underbrace{f'(x), y}_{w}) = f'(x)$$
$$\implies std_Q(g'(x,w)) = x \implies (x, g'(x,w)) \in Q.$$

Die Translationsfunktion g'', welche (x, y) zu $g'(x, w)$ übersetzt, lässt sich leicht angeben. □

Mit der Definition der effektiven p-Simulation und der eben beweisenen äquivalenten Formulierung der KvL-Vermutung lässt sich nun zumindest die Hypothese SAT so verstärken, dass diese hinreichend für KvL ist.

Vermutung 4.10 ($\mathsf{SAT}^{\mathsf{eff}}$)**.** *Keine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge $L \in$ NP hat ein optimales Beweissystem h, welches alle anderen optimalen Beweissysteme für L effektiv p-simulieren kann. In anderen Worten, für jedes optimales Beweissystem h für L existiert ein optimales Beweissysteme h' für L sodass $h' \not\leq^{\mathrm{p}}_{\mathrm{eff}} h$.*

Satz 4.11. *(1)* $\mathsf{SAT}^{\mathsf{eff}} \implies \mathsf{SAT}$

(2) $\mathsf{SAT}^{\mathsf{eff}} \implies \mathsf{KvL}$

Beweis. 1. Zu (1): Klar aus Kontraposition. Wenn für eine Menge $L \in$ NP ein p-optimales Beweissystem h für L existiert, dann kann dieses (optimale) h auch alle anderen Beweissysteme p-simulieren, und damit insbesondere auch alle optimalen Beweissyste h' effektiv p-simulieren.

2. Zu (2): Wieder klar aus Kontraposition. Unter $\neg\mathsf{KvL}$ folgt mit der Formulierung aus Vermutung 4.8 dass für eine NP-Relation R, Proj(R) vollständig, das (optimale) Standardbeweissystem std_R alle optimalen Beweissysteme für Proj(R) effektiv p-simulieren kann. Dann existiert also auch *ein* optimales Beweissystem für die $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge Proj(R) $\in$ NP welches dies leistet. □

Wir haben also je eine notwendige ($\neg$Q) und eine hinreichende Hypothese ($\mathsf{SAT}^{\mathsf{eff}}$) für KvL. Nichtsdestotrotz bleiben noch viele Fragen offen, die wir hier aus Platzgründen nicht weiter verfolgen werden. Gibt es natürliche (z.B. kryptographische) Annahmen die hinreichend für KvL sind? Wie ist die Beziehung zu den anderen Pudlákschen Hypothesen? Wie verhält sich insbesondere SAT zu $\mathsf{SAT}^{\mathsf{eff}}$? Kann in Beobachtung 4.9(2) so verstärkt werden, dass std_Q jedes (nicht nur die optimalen) Beweissystem effektiv p-simulieren kann, also gewissermaßen std_Q „effektiv p-optimal“ ist?

Insgesamt ist durch die vorherigen Überlegungen aber ein erster Schritt getan, die Beziehung zwischen Levin- und Many-one-Vollständigkeit über die Vermutung KvL im Kontext des Pudlákschen Programms einzuordnen. Weitere Forschung in diese Richtung erscheint vielversprechend.

4.2 Hypothese Q und Suchprobleme

Wie im Einstieg des Kapitels angesprochen, geben Fenner u. a. (2003) bzw. Messner (2000) äquivalente Charakterisierungen der Hypothese Q an, welche sich im Wesentlichen auf die $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit von rSAT beziehen. Wir wiederholen hier noch einmal die Aussage:

Satz 4.2. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*.*

(2) Für jede NPTM N mit $L(N) =$ SAT existiert eine Funktion $h \in$ FP sodass

$$N(\varphi) \text{ akz. mit Rechenweg } w \implies std_{\mathsf{rSAT}}(\varphi, h(w)) = \varphi,$$

i.e. $h(w)$ ist erfüllende Belegung für φ.

(3) Das Standardbeweissystem std_R (bzw. identisch sat) für rSAT *ist p-optimal.*

Dieser Satz relativiert nicht.

Diese beiden Charakterisierungen wollen wir im Folgenden verallgemeinern und auf beliebige $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relationen R übertragen. Hieraus ergibt sich schon unmittelbar der technische Beitrag, dass dann diese Charakterisierungen auch in einem relativierten Umgebung angewendet werden können, um z.B. ein Orakel zu konstruieren, was Q von anderen Hypothesen trennt.

Zweitens ergibt sich aus der Verallgemeinerung das überraschende Ergebnis, dass $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit allein nicht ausreicht. In den originalen Beweisen von Fenner u. a. und Messner wurden diese zusätzlichen ausgenutzten Eigenschaften von rSAT nur stillschweigend mitgedacht. Die folgende Generalisierung deckt diese Eigenschaften aus, und plausibilisiert dass diese womöglich nicht von allen $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständigen NP-Relationen geteilt werden.

Eine dieser stärkereren Eigenschaften von rSAT, welche im folgenden Beweis gebraucht wird, ist eine spezielle Form von Levin-Vollständigkeit. Für gegebene NP-Relation R verlangen wir nicht nur, dass $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ für alle Q, sondern auch dass diese Reduktion ehrlich ist. Dies gilt insbesondere für rSAT: es ist leicht zu sehen, dass die Reduktionsfunktionen im Satz von Cook–Levin, welche Instanzen x auf eine Formel φ_x reduzieren, verlängernd sind. Damit sind sie insbesondere ehrlich. Für welche $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen NP-Relationen das noch zutrifft, werden wir unten betrachten.

Mit dieser ehrlichen Levin-Vollständigkeit lässt sich nun die Charakterisierung von Fenner u. a. generalisieren:

Lemma 4.12. *Sei R eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation, mit der zusätzlichen Eigenschaft dass für die jeweilige entsprechende Problem-Reduktionsfunktion $f\colon Q \to R$ für $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ immer gilt, dass f ehrlich ist. Folgende Aussagen sind äquivalent:*

(1) Für alle NPTM N mit $L(N) = \Sigma^*$ lassen sich aus Eingabe x Rechenwege von $N(x)$ effizient bestimmen: es existiert $r \in \mathrm{FP}$ sodass $N(x)$ auf Rechenweg $r(x)$ akzeptiert. (Das ist die Aussage Q.*)*

(2) Für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ lassen sich akzepierende Rechenwege von N in Zertifikate umrechnen: es existiert eine Funktion $h \in \mathrm{FP}$ sodass

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x,\alpha)) \in R.$$

Beweis. (1)⇒(2): Sei R eine beliebige NP-Relation mit Zertifikatsschranke q, und sei N eine beliebige NPTM mit $L(N) = \mathrm{Proj}(R)$. Definiere nun die NPTM $N'(w)$ wie folgt:

```
1 wenn w nicht von der Form (x, α) dann akzeptiere
2 (x, α) ← w
3 wenn N(x) akzeptiert nicht auf Rechenweg α dann akzeptiere
4 sonst
  |  (Ab hier gilt x ∈ Pr(R), also auch set-R(x) ≠ ∅)
5 |  Rate nichtdeterministisch y ∈ Σ^{≤q(|x|)}
6 |  Akzeptiere genau dann wenn (x, y) ∈ A.
```

Es ist nun leicht zu sehen dass $L(N') = \Sigma^*$. Nach Voraussetzung (1) existiert eine Funktion $r \in \mathrm{FP}$ sodass für alle x die Maschine $N(w)$ auf Rechenweg $r(w)$ akzeptiert. Nun gilt

$$\begin{aligned} &N(x) \text{ akz. mit Rechenweg } \alpha \\ &\implies N'(x,\alpha) \text{ akz. in Z. 6} \\ &\implies N'(x,\alpha) \text{ akz. mit Rechenweg } r(x,\alpha) \text{ in Z. 6,} \end{aligned}$$

und aus diesem Rechenweg $r(x,\alpha)$ kann effizient der geratene Zeuge $y \in \mathit{set}\text{-}R(x)$ aus Z. 5 ausgelesen werden. Da $r \in \mathrm{FP}$ existiert also auch ein $h \in \mathrm{FP}$ sodass $h(x,\alpha)$ genau diesen geratenen Zeugen y berechnet. Wir haben dann also

$$\implies (x, h(x,\alpha)) \in R,$$

wie gewünscht.

(2)⇒(1): Sei R eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation unter ehrlichen Problem-Reduktionsfunktionen, und Zertifikatsschranke p. Sei nun N eine NPTM mit $L(N) = \Sigma^*$. Betrachte die entsprechende NP-Relation

$$R_N = \{(x,\alpha) \mid N(x) \text{ akz. mit Rechenweg } \alpha\}$$

Da R ja vollständig ist, gilt $R_N \leq_{\mathrm{L}}^{\mathrm{p}} R$ via $f, g \in \mathrm{FP}$ und (nach Voraussetzung) ist f ehrlich; es existiert ein Polynom q sodass $q(|f(x)|) \geq |x|$.
Definiere nun die folgende NPTM $N'(w)$:

```
1 Rate nichtdeterministisch x ∈ Σ^{≤q(|w|)}
2 wenn f(x) = w dann akzeptiere
  (Ab hier kann man x wegwerfen)
3 Rate nichtdeterministisch y ∈ Σ^{≤p(|w|)}
4 Akzeptiere genau dann wenn (w, y) ∈ R.
```

Wir zeigen nun, dass $L(N') = \mathrm{Proj}(R)$. Wir müssen hierfür nur die Fälle betrachten, wenn $N'(w)$ in Z. 2 akzeptiert. In diesem Fall gilt $f(x) = w$, und wir haben

$$x \in \Sigma^* \implies x \in \mathrm{Proj}(R_N) \implies f(x) \in \mathrm{Proj}(R) \implies w \in \mathrm{Proj}(R),$$

wie gewünscht.
Nach Voraussetzung (2) gilt nun also, dass eine Funktion $h \in \mathrm{FP}$ existiert sodass

$$N'(w) \text{ akz. mit Rechenweg } \alpha \implies (w, h(w,\alpha)) \in R.$$

Beobachte wie für $N'(f(x))$ immer ein trivialer akzeptierender Rechenweg α_x existiert: nämlich jener, welcher in Z. 1 das Urbild x rät. Beobachte dass die Umformung $x \mapsto \alpha_x$ in Polynomialzeit möglich ist.

Um nun (1) zu zeigen müssen wir aus $x \in \Sigma^*$ effizient einen akzeptierenden Rechenweg für N bestimmen. Wir haben

$$\begin{aligned} & N'(f(x)) \text{ akz. mit Rechenweg } \alpha_x \implies (f(x), h(f(x), \alpha_x)) \in R \\ & \implies (x, \underbrace{g(h(f(x), \alpha_x))}_{r(x)}) \in R_N \quad \text{nach Translationsfunktion } g \\ & \implies N(x) \text{ akz. mit Rechenweg } r(x) \end{aligned}$$

mit $r \in \mathrm{FP}$, $r(x) = g(h(f(x), \alpha_x))$, wie gewünscht. □

Wir wollen nun auch die zweite Charakterisierung von Messner generalisieren. Im originalen Beweis wurde erneut eine sekundäre stärkerere Eigenschaft von rSAT ausgenutzt, die einer schwachen Form von Paddability entspricht. Ähnlich wie bei der Berman–Hartmanis-Paddability wollen wir beliebige Instanzen x zu längeren Instanzen x' vergrößern. Zusätzlich verlangen wir, dass wir auch auf Zertifikaten y für x' wieder Zertifikate y für x zurückrechnen können. In anderen Worten: wir codieren „redundante Teile“ in x hinein, um x' zu erhalten. Für Zertifikate y' für x' können wir dann den Teil des Zertifikats wegwerfen, welcher sich ohnehin nur auf den redundanten Padding bezieht, und erhalten wieder ein Zertifikat für x.

Definition 4.13 (Levin-Paddability)**.** Eine NP-Relation R ist *Levin-paddable* wenn Funktionen $pad \in \mathrm{FP}$ und $padsol \in \mathrm{FP}$ existieren, sowie ein Polynom r sodass

(1) $x \in \mathrm{Proj}(R) \iff pad(x, 1^n) \in \mathrm{Proj}(R)$,

(2) $(pad(x, 1^n), y) \in R \implies (x, padsol(x, 1^n, y)) \in R$,

(3) $r(|pad(x, 1^n)|) \geq n$. (Funktion pad ist ehrlich bzgl. der zweiten Komponente.) ◁

Beachte dass wir im Gegensatz zur Berman–Hartmanis-Paddability keine Invertierbarkeit der Padding-Funktion verlangen. Später werden wir sehen, welche NP-Relationen alle diese Eigenschaft der Levin-Paddability erfüllen. Festhalten können wir aber, dass rSAT Levin-paddable ist. Das ist einfach zu sehen: padde Formeln φ auf, indem z.B. Disjunktionen neue Variablen hinzugefügt werden, i.e.

$$\varphi' = pad(\varphi, 1^n) = \varphi \vee x_k \vee x_{k+1} \vee \cdots \vee x_{k+n},$$

wobei k hinreichend groß sein soll, dass x_k nicht als Variable in φ vorkommt. Ist nun w' eine erfüllende Belegung für φ', dann entferne alle Variablenbelegungen $x_k, x_{k+1}, \ldots$ aus w'; es ergibt sich eine erfüllende Belegung w für φ.

Mit dieser Definition können wir nun einen Beweis von Messner (2000, Thm. 5.2) generalisieren. Beachte dass hier nicht notwendigerweise von vollständigen NP-Relationen gesprochen wird, und das im Beweis (3)⇒(1)die Levin-Paddability notwendig zu sein scheint, damit std_R auch nicht-ehrliche Beweissysteme p-simulieren kann.

Lemma 4.14. *Sei R eine NP-Relation die Levin-paddable ist. Folgende Aussagen sind äquivalent:*

(1) Das Standardbweissystem std_R bzgl. R ist p-optimal.

(2) Für alle NTM N (ohne Laufzeitbeschränkung) mit $L(N) = \mathrm{Proj}(R)$ lassen sich akzepierende Rechenwege von N in Zertifikate umrechnen: es existiert eine Funktion $h \in \mathrm{FP}$ sodass

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x, \alpha)) \in R.$$

(3) Für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ lassen sich akzepierende Rechenwege von N in Zertifikate umrechnen: es existiert eine Funktion $h \in \mathrm{FP}$ sodass

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x, \alpha)) \in R.$$

Beweis. (1)⇒(2): Für NTM f_N können wir das assoziierte Beweissystem

$$f_N(x, \alpha) = \begin{cases} x & N(x) \text{ akz. mit Rechenweg } \alpha \\ \bot & \text{sonst} \end{cases}$$

angeben. Es ist klar, dass f_N ein Beweissystem für $\mathrm{Proj}(R)$ ist. Aus der p-Optimalität von std_R gilt nun $std_R \leq^{\mathrm{p}} f_N$, bzw. p-simuliert das Standardbeweissystem das Beweissystem f_N. Damit existiert eine Funktion $h \in \mathrm{FP}$ sodass

$$std_R(h(x, \alpha)) = f_N(x, \alpha).$$

Diese Funktion h erfüllt genau die Eigenschaften von (2): Wir haben

$$\begin{aligned} & N(x) \text{ akz. mit Rechenweg } \alpha \implies h(x, \alpha) = x \\ & \implies std_R(h(x, \alpha)) = x \\ & \implies (x, h(x, \alpha)) \in R, \end{aligned}$$

wie gewünscht.

(2)⇒(3): Klar.

(3)⇒(1): Angenommen (3) gilt. Seien *pad*, *padsol* die entsprechenden Funktionen, welche die Levin-Paddability von R realisieren. Das Polynom r sei so gewählt dass $r(|pad(x, 1^n)|) \geq n$ (vgl. 4.13(3)).
Wir wollen nun zeigen, dass std_R auch p-optimal ist. Sei hierfür f ein beliebiges Beweissystem für Proj(R). Wir zeigen nun, dass $std_R \leq^{\mathrm{p}} f$. Seien *pad*, *padsol* die entsprechenden Padding-Funktionen von R. Definiere nun

$$f'(w) = \begin{cases} pad(x, 1^{|w|}) & \text{falls } w = 1z \text{ und } f(z) = x, \\ x & \text{falls } w = 0z \text{ und } std_R(z) = x, \\ \bot & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, dass f' ehrlich ist: es ist ehrlich für Eingaben $0z$, denn das Standardbeweissystem std_R ist ehrlich nach Beobachtung 4.4. Es ist ehrlich für Eingaben $w = 1z$, denn

$$|1z| = |w| \leq r(|\underbrace{pad(x, 1^{|w|})}_{f'(1z)}|) = r(|f'(|w|)|).$$

Sei im Folgenden dann das Polyom r' so gewählt, dass $|w| \leq r'(|f'(w)|)$ gilt.
Definiere nun die NPTM $N_{f'}$ welche auf Eingabe x erst nichtdeterministisch einen Beweis w, $|w| \leq r'(|x|)$ rät, und genau dann akzeptiert falls $f'(w) = x$. Es ist klar, dass $L(N_{f'}) = \text{Proj}(R)$. Nach Voraussetzung (3) gibt es also nun eine Funktion $h \in \text{FP}$ sodass

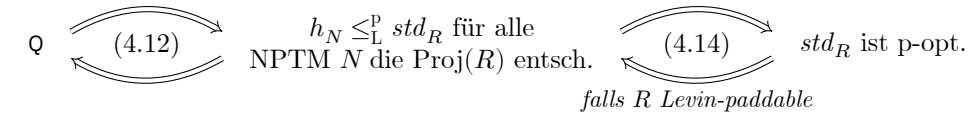
$$N_{f'}(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x, \alpha)) \in R. \tag{4.3}$$

Jetzt können wir $std_R \leq^{\mathrm{p}} f$ zeigen: sei z ein f-Beweis für x, d.h. $f(z) = x$. Wir wissen, dass $f'(1z) = pad(x, 1^{|1z|}) = x'$. Daher können wir aus z einen Rechenweg α_z konstruieren, sodass $N_{f'}(x')$ akzeptiert, nämlich jener der den f'-Beweis $1z$ rät. Die Abbildung $z \mapsto \alpha_z$ lässt sich in Polynomialzeit leisten.
Nun gilt

$$\begin{aligned} & N_{f'}(x') \text{ akz. mit } \alpha_z \implies (x', \underbrace{h(x', \alpha_z)}_{y'}) \in R \text{ nach (4.3)} \\ & \implies (pad(x, 1^{|1z|}), y') \in R \text{ mit } y' = h(x', \alpha_z) \text{ und obiger Def. von } x' \\ & \implies (x, \underbrace{padsol(x, 1^{|1z|}, y')}_{y}) \in R \\ & \implies std(x, y) = x \text{ mit } y = padsol(x, 1^{|1z|}, y') \end{aligned}$$

und wir haben aus dem f-Beweis z für x einen std_R-Beweis (x, y) für x bestimmt. Es ist klar, dass die Übersetzung $z \mapsto (x, y)$ in Polynomialzeit möglich ist. □

Wir fassen kurz den aktuellen Stand zusammen. Sei R eine NP-Relation. Für beliebige NPTM mit $L(N) = \text{Proj}(R)$ wollen wir hier mit h_N jenes Beweissystem verstehen für das $h_N(x, w) = x$ falls $N(x)$ mit Rechenweg w akzeptiert, und sonst undefiniert ist. Wir haben nun folgendes Bild:



Wir wollen nun eine möglichst breite Klasse an NP-Relationen angeben, für die diese beiden obigen Äquivalenzen gelten, also insbesondere diejenigen NP-Relationen, welche die selbe Charakterisierung wie rSAT im Fall der unvelativierbaren Charakterisierung von Fenner u. a. und Messner zulassen.

Wir überlegen uns hierzu zunächst, dass „$\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig sind und Levin-paddable“ ausreichend ist, da Levin-Paddability insbesondere zulässt, eine Levin-Reduktion so zu padden, dass die Reduktionsfuktion auch ehrlich ist.

Lemma 4.15. *Die in Lemma 4.12 und 4.14 genannten Voraussetzungen an die NP-Relation R werden von allen solchen R erfüllt, die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig sind und Levin-paddable sind.*

Beweis. Es ist sofort klar, dass R die Voraussetzungen von Lemma 4.14 erfüllt. Es bleibt nur zu zeigen, dass für jede NP-Relation Q eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-Reduktion angegeben werden kann, bei dem die Problem-Reduktionsfunktion ehrlich ist. Wir nutzen hierbei aus, dass R eine Levin-paddable Relation ist.

Nachdem R vollständig ist, gilt $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$; sei $f, g \in \text{FP}$ die Reduktions- bzw. Translationsfunktion welche diese Reduktion realisieren. Wir werden nun Funktionen $f', g' \in \text{FP}$ angeben, welche die gleiche Reduktion realisieren, aber f' ehrlich, wie gewünscht.

Sei *pad*, *padsol* die zu R zugehörigen Padding-Funktionen. Definiere

$$f'(x) = pad(f(x), 1^{|x|}).$$

Es gilt

$$x \in \text{Proj}(Q) \iff f(x) \in \text{Proj}(R) \iff pad(f(x), 1^{|x|}) = f'(x) \in \text{Proj}(R),$$

wobei erste Implikation die Eigenschaft der Reduktionsfunktion f ist, und die zweite aus der Definition von Levin-Paddability folgt. Aus der Definition von Levin-Paddability folgt auch $r(|f'(x)|) \geq |x|$ für ein geeignetes Polynom r, und damit ist auch f' ehrlich.

Definiere

$$g'(x,z) = g(x, padsol(f(x), 1^{|x|}, z)).$$

Sei nun $(f'(x), z) \in R$. Die Funktion g' berechnet nun ein Zertifikat y für x: Wir haben $(pad(f(x), 1^{|x|}), z) \in R$, also gilt nach Levin-Paddability dass

$$(f(x), padsol(f(x), 1^{|x|}, z)) \in R,$$

und nach Definition der Translationsfunktion g gilt dann

$$(x, g(x, padsol((f(x), 1^{|x|}, z))) \in Q,$$

und das ist genau $(x, g'(x,z)) \in Q$, wie gewünscht. □

Diese Eigenschaft lässt sich leicht für die kanonische NP-Relation rKAN überprüfen, und gilt insbesondere auch im relativierten Fall.

Beobachtung 4.16. *Die kanonische Levin-vollständige NP-Relation* rKAN *ist Levin-paddable.*

Folgende Beobachtung hilft uns, natürliche NP-Relationen zu identifizieren, welche Levin-vollständig als auch -paddable sind.

Beobachtung 4.17. *(1) Gilt* $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}} R$*, und ist die zugehörige Reduktionsfuktion f ehrlich, dann ist R Levin-paddable (und $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig).*

(2) Jede $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$-vollständige NP-Relation R ist auch Levin-paddable.

Damit können wir schon als Ergebnis festhalten, dass jede $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$-vollständige Relation R die in Lemma 4.14 und 4.12 genannten Voraussetzungen an die NP-Relation R erfüllt. Das sind nach Goldreich (2008) unrelativierten Fall u.a. rSAT, rSETCOVER, rVERTEXCOVER, rCLIQUE, r3COLORABILITY.

Beweis zu Beobachtung 4.17. Aussage (2) folgt unmittelbar aus (1): Wir haben $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}} R$ und damit ist die entsprechende Reduktionsfunktion f p-invertierbar, und damit ehrlich.

Für (1) nutzen wir die Levin-Paddability von rKAN aus: übersetze Instanz x von R nach rKAN, padde dort hoch, und überetze zu R-Instanz x' zurück. Ist dann y' ein Zertifikat für x', dann lässt sich dies auf ähnlichem Weg wieder zu einem Zertifikat für x zurückrechnen.

Seien f, g die Reduktions- bzw. Translationsfunktion, welche $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}} R$ bezeugen, und seinen analog f', g' jene Funktionen, welche $R \leq^{\mathrm{p}}_{\mathrm{L}} \mathtt{rKAN}$ bezeugen. Erstere existieren nach Voraussetzung, zweitere existieren weil rKAN $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Nach Voraussetzung ist f ehrlich. Und nach Beobachtung 4.16 existieren für rKAN Padding-Funktionen $pad_{\mathtt{rKAN}}$, $padsol_{\mathtt{rKAN}}$. Sei q ein entsprechendes Polynom mit $q(|pad_{\mathtt{rKAN}}(x, 1^n)|) \geq n$, $q(|f(x)|) \geq |x|$.

Definiere nun

$$pad_R(x, 1^n) = f(pad_{\mathtt{rKAN}}(f'(x), 1^n)).$$

Die Zugehörigkeit zu Proj(R) bleibt erhalten:

$$\begin{aligned} x \in \mathrm{Proj}(R) &\iff f'(x) \in \mathtt{KAN} \iff pad_{\mathtt{rKAN}}(f'(x), 1^n) \in \mathtt{KAN} \\ &\iff f(pad_{\mathtt{rKAN}}(f'(x), 1^n)) \in \mathrm{Proj}(R) \iff pad_R(x, 1^n) \in \mathrm{Proj}(R). \end{aligned}$$

Ferner gilt

$$\begin{aligned} & q(q(|pad_R(x, 1^n)|)) \\ &= q(q(|f(pad_{\mathtt{rKAN}}(f'(x), 1^n)|)) \\ &\geq q(|pad_{\mathtt{rKAN}}(f'(x), 1^n)|) \\ &\geq n. \end{aligned}$$

und damit ist pad_R wie gewünscht ehrlich bzgl. n (mit Polynom $q \circ q$).

Es verbleibt noch die Funktion $padsol_R$. Nehme hierfür an dass wir ein y' haben mit $(pad_R(x, 1^n), y') \in R$. Wir können über g, g' das Zertifikat y' zu Zertifikat y mit $(x, y) \in R$ zurück übersetzen: Sei $p = pad_{\mathtt{rKAN}}(f'(x), 1^n)$, dann gilt

$$(f(p), y') \in R \implies (p, \underbrace{g(p, y')}_{z}) \in \mathtt{rKAN}.$$

Definere $z = g(p, y')$. Nun haben wir

$$\begin{aligned} (p, z) &= (pad_{\mathtt{rKAN}}(f'(x), 1^n), z) \in \mathtt{rKAN} \\ &\implies (f'(x), \underbrace{padsol_{\mathtt{rKAN}}(f'(x), 1^n, z)}_{z'}) \in \mathtt{rKAN} \end{aligned}$$

und mit $z' = padsol_{\mathtt{rKAN}}(f'(x), 1^n, z)$ gilt

$$(f'(x), z') \in \mathtt{rKAN} \implies (x, \underbrace{g'(x, z')}_{y}) \in R.$$

Es ist leicht zu sehen, dass sich eine Funktion $padsol_R \in$ FP angeben kann, die aus $x, 1^n$ dieses entsprechende y berechnen kann. □

Anstelle der Betrachtung, wie die *Reduktionen* zwischen den einzelnen NP-Relationen aufgebaut sind, können wir auch strukturelle Eigenschaften von NP-Relationen ausnutzen, um Paddability zu zeigen. Hierbei macht die Definition von *Universalität* durch Agrawal und Biswas (1992a) aus dem Abschnitt **??** einen produktiven Beitrag. Ist eine NP-Relation

joinable, dann können wir auch zu einer Instanz beliebig viele Dummy-Instanzen anhängen. Aufgrund der speziellen Eigenschaften der *join*-Funktion können wir auch den relevanten Teil aus Zertifikaten für die verlängerten Instanz zielgenau auslesen.

Beobachtung 4.18. *Jede NP-Relation die* joinable *ist, ist auch Levin-paddable.*

Vor dem Beweis können wir mit dieser Aussage festhalten, dass jede universelle Relation R die in Lemma 4.14 und 4.12 genannten Voraussetzungen an die NP-Relation R erfüllt. Das sind nach Agrawal und Biswas (1992a) u.a. $\mathtt{rSAT}$, $\mathtt{rHAM}$, $\mathtt{rINDSET}$, $\mathtt{rKNAPSACK}$, $\mathtt{rMAXCUT}$.

Beweis zu Beobachtung 4.18. TODO: Alles überprüfen! Was meint streng monoton? Wir brauchen auch keinen Block! Sei R eine NP-Relation, mit zugehörigem Polynom q, welches die Zertifikatsgröße spezifiziert. Zur Erinnerung, dieses Polynom ist streng monoton steigend, und aus $(x, y) \in R$ folgt $|y| = q(|x|)$. Wir zeigen zunächst, wie wir für beliebige Instanz x und $n \in \mathbb{N}$ auf eine Instanz x' hochpadden, in dem Sinne dass $q(|x'|) \geq n$.

Nach Voraussetzung hat die Relation R einen *building block block*. Es lässt sich leicht aus der Definition eines *building block* ableiten, dass $|block| > 0$ und $block \in \mathrm{Proj}(R)$. Damit gilt auch dass die Zertifikate y zu *block* die Länge $l = q(|block|) \geq |block| \geq 1$ haben.

Nach Voraussetzungen ist die Relation R auch *joinable*, das heißt wir haben eine Funktion $join \in \mathrm{FP}$. Sei

$$(x', \delta) = join(x, \underbrace{block, block, \dots, block}_{n \text{ mal}}).$$

Wir werden nun über die Länge $|\delta|$ auf die Länge von Zertifikaten zu x' schließen, und damit $|x'|$ beschränken. Nach Definition ?? gilt

$$|\delta| = q(|x|) + q(n) \cdot q(|block|) = q(|x|) + q(n) \cdot l \geq n.$$

Beob. dass unter Definition ?? alle Zertifikate y' für x' die feste Länge $q(|x'|)$ haben. Zur Erinnerung: wir haben

$$\{y'[\delta] \mid y' \in \Sigma^{q(|x'|)}, (x', y') \in R\} = \{yy_1y_2 \cdots y_n \mid y \in \Sigma^{q(|x|)}, y_1, y_2, \dots \in \Sigma^l, \\ (x, y), (block, y_1), (block, y_2), \dots \in R\} \quad (4.4)$$

Die Sequenz δ besteht nach Definition aus paarweise verschiedenen Indizes, daher können wir argumentieren, dass auch alle Zertifikate y' (mit vorgegebener Länge $q(|x'|)$) mindestens die Länge $|\delta|$ haben. Damit gilt

$$q(|x'|) \geq |\delta| \geq n$$

wie gewünscht.

Sei nun *pad* genau jene polynomialzeit-berechenbare Funktion, die aus x und 1^n die Instanz x' konstruiert:

$$pad(x, 1^n) = x' \quad \text{wobei } (x', \delta) = join(x, \underbrace{block, block, \dots, block}_{q(n) \text{ mal}}).$$

Dann gilt schon sofort, dass $q(|pad(x, 1^n)|) = q(|x'|) \geq n$ wie gewünscht.

Wir zeigen jetzt, dass die Zugehörigkeit zu $\mathrm{Proj}(R)$ erhalten bleibt: Gilt $x \notin \mathrm{Proj}(R)$, dann ist die rechte Menge in (4.4) leer, also auch die linke Menge und damit $x' = pad(x, 1^n) \notin \mathrm{Proj}(R)$. Falls anders herum $x \in \mathrm{Proj}(R)$, dann ist die rechte Menge nicht leer, existiert ja ein Zertifikat y für x und je ein weiters y_i für *block*. Also ist auch die linke Menge nicht leer, damit $pad(x, 1^n) \in \mathrm{Proj}(R)$.

Die noch verbleibende Funktion *padsol* ist durch die bitweise Projektion durch δ leicht möglich:

$$padsol(x, 1^n, y') = y'[\delta[1 : q(|x|)]] \quad \text{wobei } (\cdot, \delta) = join(x, \underbrace{block, block, \dots, block}_{n \text{ mal}}).$$

Wir verifizieren: Sei $(pad(x, 1^n), y') \in R$, dann ist nach (4.4) $y'[\delta] = yy_1y_2 \cdots$ wobei $y \in \Sigma^{q(|x|)}$, $(x, y) \in R$. Wir haben

$$padsol(x, 1^n, y') = y'[\delta[1 : q(|x|)]] = (yy_1y_2 \cdots)[1 : q(|x|)] = y$$

und damit $(x, padsol(x, 1^n, y')) = (x, y) \in R$, wie gewünscht. □

Es bleibt die Frage offen, ob Levin-Paddability für *alle* vollständigen NP-Relationen zutrifft. Unter Annahmen einer geeigneten Einwegfunktion vermute ich, dass dies nicht der Fall ist. Die Argumentation verläuft hier ähnlich zur *Encrypted Complete Set Conjecture*. Wir setzen hier eine stärkere *secure one-way function* (Grollmann und Selman 1988) f voraus, die selbst mithilfe funktionaler Orakel-Queries nur auf einer dünnen Menge p-invertierbar ist. Präzise meinen wir damit folgendes: sei A ein beliebiger Polynomialzeit-Algorithms, der auf Eingabe w versucht, das Urbild $f^{-1}(w)$ zu berechnen. Zusätzlich darf A das Urbild $f^{-1}(w')$ von einem Wort $w' \neq w$ erfragen. Dann wird A nur auf einer dünnen Menge $W \subseteq \Sigma^*$ das korrekte Urbild aller $w \in W$ bestimmen können. (Vgl. die Ähnlichkeit zur Selbstreduzierbarkeit aus Abschnitt **??** und effektiver p-Simulation aus Abschnitt **??**.) Die Existenz einer solchen Einwegfunktion erscheint aus kryptographischer Perspektive naheliegend.

Betrachte nun die Menge

$$Q = \{(f(x), (x, z)) \mid x, z \in \Sigma^*, (x, z) \in \mathtt{rKAN}\}.$$

Es ist leicht zu sehen dass $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}} Q$ und damit ist $Q \leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig. Gleichzeitig kann dann Q nicht Levin-paddable sein. Denn angenommen, Q ist Levin-paddable, dann lässt sich f mit einem funktionalen Orakel-Query *zumindest auf den Werten* $f(\mathtt{SAT})$ p-invertieren: gegeben $w \in f(\mathtt{SAT})$, berechne erst eine zweite Instanz $w' = pad(w, 1^n) \in \mathrm{Proj}(Q)$ mit hinreichend langem n sodass $w' \neq w$. Frage dann an das Orakel und erhalte $(x', z') \in set\text{-}Q(w')$. Dann gilt $(x, z) = padsol(w, 1^n, (x', z'))$ mit $f(x) = w$, i.e. x ist das gesuchte Urbild von w. Aus der Existenz der Einwegfunktion f folgt $\mathrm{P} \neq \mathrm{NP}$, und damit ist insbesondere $f(\mathtt{SAT})$ eine

nicht-dünne Menge (Mahaney 1982). Das widerspräche nun den Eigenschaften von f, also ist Q nicht Levin-paddable.

Dennoch bleibt die allgemeine Frage zwischen $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit und Levin-Paddability offen, die wir im Folgenden nicht weiter bearbeiten werden:

Frage 4.19. *Ist jede $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation R auch Levin-paddable? Existiert ggf. ein Gegenbeispiel in einer geeigneten relativierten Umgebung?*

Unabhängig von dieser Frage können wir nun aber abschließend die vorigen Ergebnisse zur Beziehung zwischen Suchproblemen und der Hypothese `Q` zusammenfassen in folgendem Satz zusammenfassen. Beachte dass diese Charakterisierungen relativieren. Die Äquivalenz zu Aussage (7) bzw. (10) ist hierbei eine einfache relativierbare Generalisierung von Beweisen durch Messner (2000, Thm. 5.3) bzw. Fenner u. a. (2003).

Satz 4.20 (Äquivalente Formulierungen der Hypothese `Q`)**.** *Folgende Aussagen sind äquivalent:*

(1) Hypothese `Q`: Für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ eine akzeptierende Berechnung von $N(x)$ ist.

(2) $\mathrm{NPMV}_t \subseteq_{\mathrm{c}} \mathrm{FP}$

(3) $\mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$ *und* $\mathrm{NPMV}_t \subseteq_{\mathrm{c}} \mathrm{NPSV}_t$

(4) Jede surjektive ehrliche Funktion $f \in \mathrm{FP}$ ist p-invertierbar.

(5) Für jede Menge $L \in \mathrm{P}$ und jede NPTM N mit $L(N) = L$ existiert eine Funktion $h \in \mathrm{FP}$ mit

$$x \in L \implies N(x) \text{ akz. mit Rechenweg } h(x).$$

(6) Für jedes Paar von NP-Relationen A, B und jede Funktion $f \in \mathrm{FP}$ gilt:

$$\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(B) \text{ via } f \iff A \leq^{\mathrm{p}}_{\mathrm{L}} B \text{ via Reduktionsfunktion } f.$$

(7) Für jedes Beweissystem h gilt: h ist optimal $\iff$ h ist p-optimal.

(8) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable NP-Relation R sodass für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $h \in \mathrm{FP}$ mit

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x, \alpha)) \in R.$$

(9) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable NP-Relation R für welche das Standardbeweissystem std_R p-optimal ist.

(10) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständige NP-Relation R sodass für jede Menge $S \in \mathrm{P}$ mit $S \subseteq \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $g \in \mathrm{FP}$ sodass

$$x \in S \implies (x, g(x)) \in R.$$

Beweis. 1. (1) $\iff$ (2) $\iff$ (3) $\iff$ (4) $\iff$ (5): nach Fenner u. a. (2003, Thm. 2).

2. (1) $\iff$ (6): nach Lemma 4.5.

3. (1) $\iff$ (8) $\iff$ (9): nach Lemma 4.12 und 4.14.

4. (5) $\implies$ (10): Wir zeigen eine stärkere Variante von (10), welche sich über *alle* NP-Relationen R estreckt (und damit auch über $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständige `rKAN`, wie von (10) gefordert). Sei R eine beliebige NP-Relation, wobei Polynom q die Zertifikatsgröße beschränkt. Sei nun $S \subseteq \mathrm{Proj}(R)$ mit $S \in \mathrm{P}$. Definiere die NPTM N, welche auf Eingabe x folgendes leistet: teste zuerst ob $x \in S$; falls nicht, lehne sofort ab. Rate dann ein $y \in \Sigma^{\leq q(|x|)}$ und akzeptiere genau dann wenn $(x, y) \in R$.
Klar ist, dass $L(N) = S$. Nach (5) existiert nun eine Funktion $h \in \mathrm{FP}$, die für $x \in S$ einen akzeptierenden Rechenweg $h(x)$ von $N(x)$ ausgibt. Wir können sogar aus $h(x)$ das geratene Zertifikat y extrahieren. Es ist daher leicht eine Funktion $g \in \mathrm{FP}$ anzugeben für die $(x, g(x)) \in R$ für alle $x \in S$.

5. (10) $\implies$ (5): Sei $L \in \mathrm{P}$ und sei N eine NPTM mit $L(N) = L$, wobei das Polynom q die Laufzeit beschränkt. Wir wollen eine Funktion $h \in \mathrm{FP}$ definieren sodass $h(x)$ ein akzeptierender Rechenweg von $N(x)$ für $x \in L$ ist. Definiere die NP-Relation

$$Q = \{(x, y) \mid N(x) \text{ akzeptiert mit Rechenweg } y \in \Sigma^{\leq q(|x|)}\}.$$

Nachdem (10) gilt, haben wir eine $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständige NP-Relation R. Damit gilt $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ mittels Reduktions- bwz. Translationsfunktion $f, k \in FP$. Insbesondere existiert eine Inverse $f^{-1} \in \mathrm{FP}$ zu f.
Sei $S = f(L)$ die Bildmenge der Elemente aus L, also

$$S = \{f(x) \mid x \in L\}.$$

Es ist leicht zu sehen dass $S \subseteq \mathrm{Proj}(R)$. Außerdem ist $S \in \mathrm{P}$: teste $z \in S$ indem getestet wird ob $f^{-1}(z) = x \neq \perp$ und ob $x \in L$.
Damit sind die Voraussetzungen von (10) erfüllt, und es existiert eine Funktion $g \in \mathrm{FP}$ sodass $(z, g(z)) \in R$ für

alle $z \in S$. Damit gilt

$$\begin{aligned} & x \in L \implies f(x) \in S \implies (f(N,x), g(f(x))) \in R \\ & \implies ((x), k(g(f(x)))) \in Q \\ & \implies N(x) \text{ akz. mit Rechenweg } \underbrace{k(g(f(x)))}_{h(x)}. \end{aligned}$$

Definiere nun die gesuchte Funktion $h \in \mathrm{FP}$ mit $h(x) = k(g(f(x)))$. Damit gilt für alle $x \in L$ dass $N(x)$ mit Rechenweg $h(x)$ akzeptiert, wie gewünscht.

6. (2) $\implies$ (7): Die Richtung von rechts nach links ist klar. Sei für die andere Richtung h ein optimales Beweissystem für eine Menge L. Wir wollen zeigen, dass h auch p-optimal ist. Sei dafür g ein weiteres Beweissystem für L. Nach Voraussetzung haben wir $g \leq h$, das heißt es existiert eine (nicht notwendigerweise effiziente) Funktion f sodass $g(w) = h(f(w))$, und gleichzeitig ist $|f(w)| \leq q(|w|)$ für ein geeignetes Polynom q.
Betrachte folgende Multifunktion f':

$$f'(w) \mapsto y \iff \exists y \in \Sigma^{\leq q(|w|)}, g(w) = h(y).$$

Es lässt sich leicht zeigen, dass $f' \in \mathrm{NPMV}$, über einen geeigneten NPTM-Transduktor. Es ist sogar $f' \in \mathrm{NPMV}_t$, denn für jedes w mindestens $f(w) \in \mathit{set}\text{-}f'(w)$.
Nach (2) gilt also $f' \in \mathrm{NPMV}_t \subseteq_{\mathrm{c}} \mathrm{FP}$, also existiert eine Funktion $f'' \in \mathrm{FP}$ welche eine Verfeinerung von f' ist. Diese Funktion übersetzt g-Beweise w für x effizient in h-Beweise für x: Sei $g(w) = x$, dann gilt

$$f''(w) = y \quad \text{mit } y \in \Sigma^{\leq q(|w|)}, x = g(w) = h(y)$$

also ist $h(f''(w)) = x$ bzw. $f''(w)$ ein h-Beweis für x, wie gewünscht.

7. (7) $\implies$ (9): klar, denn rKAN ist $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig, ist Levin-paddable, und das Standardbeweissystem $\mathit{std}_{\mathsf{rKAN}}$ ist (wie jedes Standardbeweissystem einer NP-Relation) optimal. Zusammen mit (7) ist es also auch p-optimal. □

Analysiert man die Beweise bezüglicher der Äquvalenz von Aussage Q zu (8)–(10) können wir sogar feststellen, dass die Wahl der Relation R beliebig ist. Wir können daher Q über universell quantifizierte Varianten von (8)–(10) charakterisieren.

Satz 4.21. *Entweder gelten die Aussagen (1)–(4) oder die Aussagen (1')–(4'):*

(1) Q.

(2) Für alle $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständigen Levin-paddable NP-Relationen R, alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $h \in \mathrm{FP}$ mit

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x,\alpha)) \in R.$$

(3) Für alle $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständigen Levin-paddable NP-Relationen R ist das Standardbeweissystem std_R p-optimal.

(4) Für alle $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständigen NP-Relationen R, für alle Mengen $S \in \mathrm{P}$ mit $S \subseteq \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $g \in \mathrm{FP}$ sodass

$$x \in S \implies (x, g(x)) \in R.$$

(1') ¬Q.

(2') Es existiert keine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable NP-Relation R, sodass für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $h \in \mathrm{FP}$ mit

$$N(x) \text{ akz. mit Rechenweg } \alpha \implies (x, h(x,\alpha)) \in R.$$

(3') Es existiert keine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable NP-Relation R ist das Standardbeweissystem std_R p-optimal.

(4') Es existiert keine $\leq^{\mathrm{p}}_{\mathrm{L,1,inv}}$-vollständige NP-Relation R, sodass für alle Mengen $S \in \mathrm{P}$ mit $S \subseteq \mathrm{Proj}(R)$ gilt: es existiert eine Funktion $g \in \mathrm{FP}$ sodass

$$x \in S \implies (x, g(x)) \in R.$$

Beachte dass (2') nicht die negierte Version von (2) ist, für (3) und (4) gilt dies analog.

4.3 Bekannte Implikationen, Offene Orakel

Im letzten Abschnitt dieses Kapitels werden wir nun die in Abbildung 4 abgebildeten Implikationen und Äquivalenzen nachweisen. Damit werden insbesondere auch die Hypothesen Q und KvL in das Pudláksche Programm eingeordnet. Zum Schluss wird noch angegeben, welche der Hypothesen im (vergrößtertem) Pudlákschen Programm durch ein Orakel separiert sind, und welche Separierungen noch offen sind.

Zunächst führen wir noch eine abgeschwächte Variante von Q ein, die von Fenner u. a. (2003) vorgeschlagen wurde.

Vermutung 4.22 (Q', Fenner u. a. 2003)**.** *Für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x) \in \{0,1\}$ das erste Bit einer akzeptierende Berechnung von $N(x)$ ist.*

Jetzt können wir auch die in Abbildung 4 abgebildeten Implikationen und Äquivalenzen nachweisen.

Satz 4.23. *Es gelten die in Abbildung 4 abgebildeten Implikationen und Äquivalenzen.*

Beweis. Es gelten die notiertern Äquivalenzen:
1. $\neg\mathsf{Q} \Leftrightarrow$ *sat* ist p-optimal, nach Satz 4.20.
2. NP $\neq$ coNP $\Leftrightarrow$ existiert keine NP-harte Funktion in TFNP, nach Satz **??**.
3. $\neg\mathsf{Q}' \Leftrightarrow \exists$ P-inseparierbares DisjCoNP-Paar, nach Fortnow und Rogers (2002).
4. NP $\cap$ coNP $\neq$ P $\Leftrightarrow$ NPSV$_t \not\subseteq$ FP, nach Selman (1994).
5. UP $\neq$ P $\Leftrightarrow \exists$ Einwegfunktionen, nach Grollmann und Selman (1988, Thm. 10).
6. UP $\cap$ coUP $\neq$ P $\Leftrightarrow \exists$ Einwegpermutationen, nach Homan und Thakur (2003).
7. $\mathsf{NP} \cap \mathsf{coNP} \Leftrightarrow$ NPSV$_t$ hat keine vollständige Funktion, nach Beyersdorff, Köbler und Messner (2009, Prop. 3).
8. $\mathsf{DisjNP} \Leftrightarrow$ NPSV hat keine vollständige Funktion, nach Glaßer, Selman und Sengupta (2005, Thm. 9).

Es gelten die eingezeichneten Implikationen:
1. $\mathsf{DisjNP} \Rightarrow \mathsf{CON}^{\mathsf{N}}$ nach Köbler, Messner und Torán (2003).
2. $\mathsf{UP} \Rightarrow \mathsf{TAUT}$ nach Köbler, Messner und Torán (2003, Cor. 4.1).
3. $\mathsf{CON}^{\mathsf{N}} \Rightarrow$ NEE $\neq$ coNEE nach Köbler, Messner und Torán (2003).
4. NP $\cap$ coNP $\neq$ P $\Rightarrow \neg\mathsf{Q}' \Rightarrow$ NPMV$_{\mathrm{t}} \not\subseteq_{\mathrm{c}}$ TFNP $\Rightarrow \neg\mathsf{Q}$ nach Fenner u. a. (2003).
5. E $\neq$ NE $\Rightarrow \exists$ NP-Relation die nicht auf Entscheidung reduzierbar ist, nach Impagliazzo und Sudan (1991).
6. UP $\neq$ P $\Rightarrow \exists$ P-inseparierbares DisjNP-Paar, nach Grollmann und Selman (1988, Thm. 5).
7. $\mathsf{NP} \cap \mathsf{coNP} \Rightarrow \mathsf{TAUT} \vee \mathsf{SAT}$ nach Köbler, Messner und Torán (2003, Cor. 5.1).
8. NPMV$_{\mathrm{t}}$ hat keine vollständige Funktion $\Rightarrow$ SAT nach Beyersdorff, Köbler und Messner (2009, Thm. 25). Es ist leicht zu sehen, dass der Beweis auch auf unsere relativierte Variante von SAT generalisiert.
9. NPMV$_{\mathrm{t}}$ hat keine vollständige Funktion $\Rightarrow$ NP $\neq$ coNP nach Satz **??**.
10. $\mathsf{SAT}^{\mathsf{eff}} \Rightarrow \mathsf{SAT}, \mathsf{SAT}^{\mathsf{eff}} \Rightarrow \mathsf{KvL}$, nach Satz 4.11.
11. $\mathsf{KvL} \Rightarrow \neg\mathsf{Q}$, nach Satz 4.6.
12. $\neg\mathsf{Q} \Rightarrow \exists$ NP-Relation die nicht auf Entscheidung reduzierbar ist, denn unter $\neg\mathsf{Q}$ gilt mit Satz 4.20 auch die Negation von 4.20(1), also eine NPTM N mit $L(N) = \Sigma^*$ wobei keine Funktion $g \in \mathrm{FP}$ existiert, welche für alle x durch $g(x)$ einen akzeptierenden Rechenweg von $N(x)$ bestimmt. Definiere die NP-Relation R_N mit $(x,\alpha) \in R_N$ genau dann wenn $N(x)$ mit Rechenweg α existiert. Nun gilt nach Vorigem auch $R \notin_{\mathrm{c}} \mathrm{FP} = \mathrm{FP}^{\Sigma^*} = \mathrm{FP}^{L(R)}$.
13. $\mathsf{DisjCoNP} \Rightarrow \mathsf{TFNP} \Rightarrow$ NPMV$_{\mathrm{t}}$ hat keine vollständig Funktion, nach Pudlák (2017).
14. NP $\cap$ coNP $\neq$ P $\Rightarrow \exists$ P-inseparierbares DisjNP-Paar, denn wenn alle DisjNP-Paare P-separierbar, dann ist auch für jede Menge $L \in$ NP $\cap$ coNP jeweils das DisjNP-Paar $(L, \overline{L})$ P-separierbar und damit $L \in$ P.
15. $\mathsf{DisjNP} \Rightarrow \exists$ P-inseparierbares DisjNP-Paar; ist klar, denn wenn alle DisjNP-Paare P-separierbar wären, dann wären auch alle Paare $\leq^{\mathrm{pp}}_{\mathrm{m}}$-vollständig.
16. $\mathsf{DisjCoNP} \Rightarrow \exists$ P-inseparierbares DisjCoNP-Paar; ist aus selben Gründen klar.
17. $\mathsf{CON}^{\mathsf{N}} \Rightarrow \mathsf{TAUT}$ klar, weil aus p-Optimalität auch Optimalität folgt.
18. $\mathsf{SAT} \Rightarrow \neg\mathsf{Q}$ klar: wen Q, dann ist nach Satz 4.20 jedes optimale Beweissystem auch p-optimal. Dann gilt auch $\neg$SAT: jede Menge $L \in$ NP hat ein optimales Beweissystem h (Beobachtung 2.12) und das ist nach Voraussetzung p-optimal.
19. $\mathsf{UP} \Rightarrow$ UP $\neq$ P klar.
20. $\mathsf{NP} \cap \mathsf{coNP} \Rightarrow$ NP $\cap$ coNP $\neq$ P klar.
21. $\exists$ P-inseparierbares DisjNP-Paar $\Rightarrow$ P $\neq$ NP klar.
22. UP $\cap$ coUP $\Rightarrow$ UP $\neq$ P, UP $\cap$ coUP $\Rightarrow$ NP $\cap$ coNP $\neq$ P klar.
23. NEE $\neq$ coNEE $\Rightarrow$ NE $\neq$ coNE $\Rightarrow$ NP $\neq$ coNP $\Rightarrow$ P $\neq$ NP klar.
24. NEE $\neq$ coNEE $\Rightarrow$ EE $\neq$ NEE $\Rightarrow$ E $\neq$ NE klar. □

Der verbleibende Beweis ist eine Generalisierung von Dingel (2022).

Satz 4.24. *Wenn* NP = coNP *dann existiert eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Multifunktion f für* NPMV$_t$.

Beweis. Nach Voraussetzung können wir in NP testen, ob ein Wort x im Urbild einer beliebigen NPMV-Multifuktion liegt. Es gilt $\mathsf{KAN} \in$ NP und damit $\mathsf{KAN} \in$ coNP. Insbesondere ist dann die Menge

$$U = \{(i, x, 1^n) \mid T_i \text{ akz. auf keinem Rechenweg der Länge} \leq n\} \in \mathrm{NP},$$

und wird von der NPTM N_u in Laufzeit $q(|(i,x,1^n)|)$ entschieden.

Betrachte nun die Multifunktion f, die durch folgenden nichtdeterministischen Transduktor $T'(i,x,1^n)$ berechnet wird:

wenn T kein Transduktor ist oder $n \neq |x|^i + i$ **dann**
  akzeptiere
Rate nichtdeterministisch einen Rechenweg α von T_i der Länge $\leq n$
Rate nichtdeterministisch einen Rechenweg β von N_u der Länge $\leq q(|i,x,1^n|)$
wenn Falls $T_i(x)$ mit α akzeptiert **dann**
  $y \leftarrow$ Ausgabe von $T_i(x)$ auf α
  Gebe y aus
sonst wenn Falls $N_u(i,x,1^n)$ mit β akzeptiert **dann**
  Gebe ε aus
sonst
  Lehne ab

Es ist leicht zu sehen dass T' in Polynomialzeit arbeitet. Wir betrachten nun Eingaben $(i,x,1^n), n = |x|^i + i$. Es gilt nun:

- Entweder ist *set*-$T_i(x) \neq \emptyset$, dann existiert für jedes $y \in$ *set*-$T_i(x)$ ein akzeptierender Rechenweg α der Länge $\leq n$ auf T_i der y ausgibt, und damit wird auch f dieses y in Z. 5 ausgeben. Gleichzeitig ist damit

$(i, x, 1^n) \notin U$ und Z. 7 niemals erreicht. Es gilt also $\textit{set-}f(i, x, 1^n) = \textit{set-}T_i(x)$.

- Oder es gilt $\textit{set-}T_i(x) = \emptyset$. Dann wird jeder Rechenweg der Länge $\leq n$ von $T_i(x)$ ablehnen, und f definitiv nicht in Z. 5 akzeptieren. Andererseits gilt dann $(i, x, 1^n) \in U$ und Z. 7 wird auf mindestens einem Rechenweg von f erreicht. Es gilt also $\textit{set-}f(i, x, 1^n) = \{\varepsilon\}$.

Damit ist klar, dass $f \in \text{NPMV}_t$. Wir zeigen nun, dass f auch NPMV_t-vollständig ist. Sei hierfür g eine beliebige Multifunktion aus NPMV_t. Dann existiert auch ein i sodass der nichtdeterministische Transduktor T_i diese Multifunktion g in berechnet, und dabei terminiert $T_i(x)$ in $\leq |x|^i + i$ vielen Schritten.

Nun gilt nach obiger Beobachtung schon dass

$$\textit{set-}g(x) = \textit{set-}T_i(x) \neq \emptyset \implies \textit{set-}f(\underbrace{i, x, 1^{|x|^i+i}}_{h(x)}) = \textit{set-}T_i(x) = \textit{set-}g(x)$$

und $h(x) = (i, x, 1^{|x|^i+i})$ realisiert die Reduktion von g auf f, wie gewünscht. □

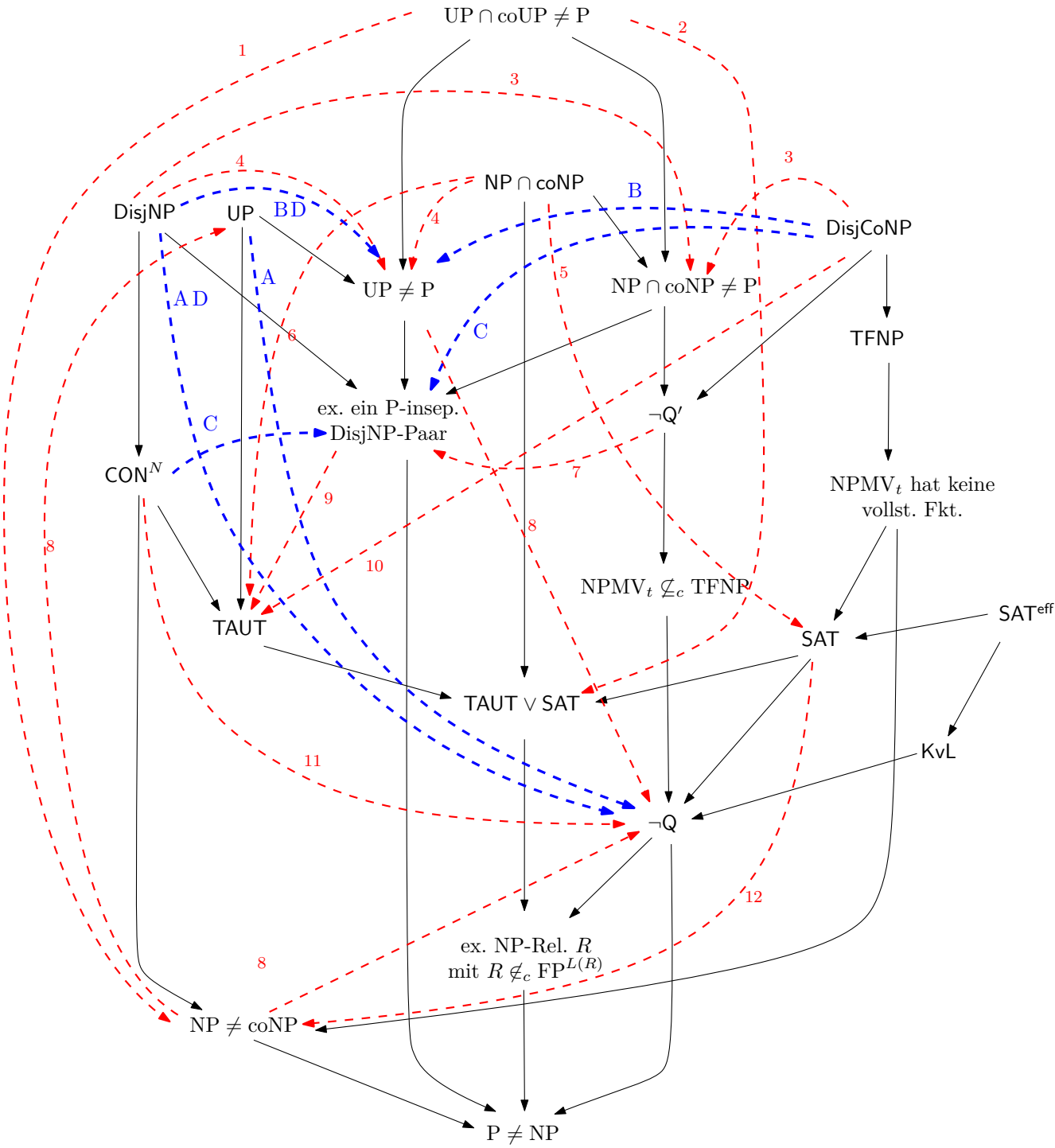

UP ∩ coUP ≠ P
NP ∩ coNP
DisjNP
UP
DisjCoNP
UP ≠ P
NP ∩ coNP ≠ P
TFNP
ex. ein P-insep.
DisjNP-Paar
¬Q′
CON^N
NPMV_t hat keine
vollst. Fkt.
NPMV_t ⊈_c TFNP
TAUT
SAT
SAT^eff
TAUT ∨ SAT
KvL
¬Q
ex. NP-Rel. R
mit R ∉_c FP^{L(R)}
NP ≠ coNP
P ≠ NP
A
B
C
D
1
2
3
4
5
6
7
8
9
10
11
12

Abbildung 5

5 Orakel

Abschließend erweitern wir noch unsere Notation, die uns insbesondere bei der Orakelkonstruktion helfen wird. Diese folgt Überlegungen von Dose und Glaßer (2019). Anstelle von Orakeln als Menge zu verstehen, können wir äquivalent Orakel auch als unendlich lange Wörter $u \in \Sigma^\omega$ formulieren, die wir als das Orakel $\{i \mid w[i] = 1\} \subseteq \mathbb{N}$ interpretieren. Mit der obigen Identifikation von Wörtern und natürlichen Zahlen beschreibt nun u sowohl ein Orakel über $\mathbb{N}$ als auch über Σ^*; wir können also z.B. von der relativen Berechnung $M^w(x)$ sprechen. Analog fassen wir endlich lange Wörter $w \in \Sigma^*$ als *partielles* Orakel $\{i \mid w[i] = 1\}$, welches die Zugehörigkeit der Wörter $x < |w|$ festlegt, aber die Zugehörigkeit aller Wörter $y \geq |w|$ noch nicht endgültig festlegt. Auf dieser Idee der endgültigen bzw. noch nicht endgültigen Zugehörigkeit aufbauend können wir auch von *definiten* Berechnungen sprechen: Eine Rechnung $M^w(x)$ ist *definit* wenn auf allen Rechenwegen von $M^w(x)$ nur Orakelfragen gestellt werden, welche eine Länge $< |w|$ haben.

Literatur

Adleman, Leonard und Kenneth Manders. 1977. „Reducibility, Randomness, and Intractibility (Abstract)“. In: *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing (STOC '77)*. Boulder, Colorado, USA: Association for Computing Machinery, 1977, S. 151–163. DOI: 10.1145/800105.803405.

Agrawal, Manindra und Somenath Biswas. 1992a. *Universal relations*. Techn. Ber. Kanpur, Indien: Department of Computer Science und Engineering, Indian Institute of Technology Kanpur, 1992. URL: http://repository.ias.ac.in/92033/. Eine überarbeitete Fassung der Proceedings-Version (1992b).

Agrawal, Manindra und Somenath Biswas. 1992b. „Universal relations“. In: *Proceedings of the Seventh Annual Structure in Complexity Theory Conference*. Seventh Annual Structure in Complexity Theory Conference. Juni 1992, S. 207–220. DOI: 10.1109/SCT.1992.215395.

Agrawal, Manindra, Neeraj Kayal und Nitin Saxena. 2004. „PRIMES Is in P“. In: *Annals of Mathematics* 160.2 (2004), S. 781–793. DOI: 10.4007/annals.2004.160.781.

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

Baker, Theodore, John Gill und Robert Solovay. 1975. „Relativizations of the P=?NP Question“. In: *SIAM Journal on Computing* 4.4 (Dez. 1975), S. 431–442. DOI: 10.1137/0204037.

Balcázar, José L. 1989. „Self-reducibility structures and solutions of NP problems.“ In: *Revista Matemática de la Universidad Complutense de Madrid* 2.2-3 (1989), S. 175–184. URL: http://eudml.org/doc/43531.

Bellare, Mihir und Shafi Goldwasser. 1994. „The Complexity of Decision Versus Search“. In: *SIAM Journal on Computing* 23.1 (Feb. 1994), S. 97–119. DOI: 10.1137/S0097539792228289.

Beyersdorff, Olaf, Johannes Köbler und Jochen Messner. 2009. „Nondeterministic functions and the existence of optimal proof systems“. In: *Theoretical Computer Science* 410.38 (6. Sep. 2009), S. 3839–3855. DOI: 10.1016/j.tcs.2009.05.021. (Besucht am 23. 06. 2019).

Borodin, Allan B. und Alan J. Demers. 1976. *Some Comments on Functional Self-Reducibility and the NP Hierarchy*. Techn. Ber. 76-284. Ithaca, New York, USA: Department of Computer Science, Cornell University, 1976. URL: https://hdl.handle.net/1813/6540.

Buhrman, H., J. Kadin und T. Thierauf. 1998. „Functions Computable with Nonadaptive Queries to NP“. In: *Theory of Computing Systems* 31.1 (1. Feb. 1998), S. 77–92. DOI: 10.1007/s002240000079. (Besucht am 30. 09. 2022).

Cai, Jin-Yi und Artem Govorov. 2020. *The Complexity of Counting Edge Colorings for Simple Graphs*. version: 1. 10. Okt. 2020. DOI: 10.48550/arXiv.2010.04910. arXiv: 2010.04910[cs]. (Besucht am 02. 11. 2022).

Cook, Stephen A. 1971. „The Complexity of Theorem-Proving Procedures“. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC'71)*. Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971, S. 151–158. DOI: 10.1145/800157.805047.

Cook, Stephen A. und Robert A. Reckhow. 1979. „The relative efficiency of propositional proof systems“. In: *The Journal of Symbolic Logic* 44.1 (März 1979), S. 36–50. ISSN: 0022-4812, 1943-5886. DOI: 10.2307/2273702.

Dingel, David. 2022. „Separation der relativierten Vermutungen SAT und TFNP“. Bachelorarbeit. Universität Würzburg, 22. Okt. 2022.

Dose, Titus. 2020. „An oracle separating conjectures about incompleteness in the finite domain“. In: *Theoretical Computer Science* 809 (24. Feb. 2020), S. 466–481. DOI: 10.1016/j.tcs.2020.01.003. (Besucht am 16. 03. 2021).

Dose, Titus und Christian Glaßer. 2019. *NP-Completeness, Proof Systems, and Disjoint NP-Pairs*. 050. 2019. URL: https://eccc.weizmann.ac.il/report/2019/050/ (besucht am 03. 04. 2019).

Fenner, Stephen A., Lance Fortnow, Ashish V. Naik und John D. Rogers. 2003. „Inverting onto functions“. In: *Information and Computation* 186.1 (Okt. 2003), S. 90–103. DOI: 10.1016/S0890-5401(03)00119-6. (Besucht am 04. 01. 2022).

Fischer, Sophie, Lane A. Hemaspaandra und Leen Torenvliet. 1995. „Witness-isomorphic reductions and the local search problem“. In: *Mathematical Foundations of Computer Science 1995*. Hrsg. von Jiří Wiedermann und Petr Hájek. Bd. 969. Lecture Notes in

Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, S. 277–287. DOI: 10.1007/3-540-60246-1_134. (Besucht am 03. 10. 2022).

Fortnow, Lance und John D. Rogers. 2002. „Separability and one-way functions“. In: *Computational Complexity* 11.3 (1. Juni 2002), S. 137–157. DOI: 10.1007/s00037-002-0173-4. (Besucht am 13. 09. 2022).

Glaßer, Christian, Alan L. Selman und Samik Sengupta. 2005. „Reductions between disjoint NP-Pairs“. In: *Information and Computation* 200.2 (1. Aug. 2005), S. 247–267. ISSN: 0890-5401. DOI: 10.1016/j.ic.2005.03.003.

Glaßer, Christian, Alan L. Selman, Samik Sengupta und Liyu Zhang. 2004. „Disjoint NP-Pairs“. In: *SIAM Journal on Computing* 33.6 (Jan. 2004), S. 1369–1416. DOI: 10.1137/S0097539703425848.

Goldreich, Oded. 2008. *Computational Complexity: a Conceptual Perspective*. Cambridge: Cambridge University Press, 2008. 606 S. ISBN: 978-0-521-88473-0.

Grollmann, Joachim und Alan L. Selman. 1988. „Complexity Measures for Public-Key Cryptosystems“. In: *SIAM Journal on Computing* 17.2 (Apr. 1988), S. 309–335. DOI: 10.1137/0217018. (Besucht am 05. 03. 2023).

Harsha, Prahladh, Daniel Mitropolsky und Alon Rosen. 2023. „Downward Self-Reducibility in TFNP“. In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Hrsg. von Yael Tauman Kalai. Bd. 251. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 67:1–67:17. DOI: 10.4230/LIPIcs.ITCS.2023.67.

Hartmanis, J. und L. Berman. 1976. „On isomorphisms and density of NP and other complete sets“. In: *Proceedings of the eighth annual ACM symposium on Theory of computing*. STOC '76. New York: Association for Computing Machinery, 3. Mai 1976, S. 30–40. DOI: 10.1145/800113.803628. (Besucht am 23. 08. 2023).

Hemaspaandra, Lane A. 1998. „Complexity Theory Column 20: Take-home complexity“. In: *ACM SIGACT News* 29.2 (Juni 1998), S. 9–13. DOI: 10.1145/288079.288080. (Besucht am 03. 10. 2022).

Holyer, Ian. 1981. „The NP-Completeness of Edge-Coloring“. In: *SIAM Journal on Computing* 10.4 (Nov. 1981), S. 718–720. DOI: 10.1137/0210055. (Besucht am 03. 09. 2023).

Homan, Christopher M. und Mayur Thakur. 2003. „One-way permutations and self-witnessing languages“. In: *Journal of Computer and System Sciences* 67.3 (1. Nov. 2003), S. 608–622. DOI: 10.1016/S0022-0000(03)00068-0. (Besucht am 30. 09. 2023).

Impagliazzo, Russel und Mahdu Sudan. 1991. Private Kommunikation. Nicht Publiziert. Berichtet von Bellare und Goldwasser 1994, S. 102. Mai 1991.

Karp, Richard M. 1972. „Reducibility among Combinatorial Problems“. In: *Complexity of Computer Computations*. Hrsg. von Raymond E. Miller, James W. Thatcher und Jean D. Bohlinger. Boston, MA: Springer, 1972, S. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

Köbler, Johannes und Jochen Messner. 2000. „Is the Standard Proof System for SAT P-Optimal?“ In: *FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science*. Hrsg. von Sanjiv Kapoor und Sanjiva Prasad. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, S. 361–372. DOI: 10.1007/3-540-44450-5_29.

Köbler, Johannes, Jochen Messner und Jacobo Torán. 2003. „Optimal proof systems imply complete sets for promise classes“. In: *Information and Computation* 184.1 (10. Juli 2003), S. 71–92. DOI: 10.1016/S0890-5401(03)00058-0. (Besucht am 23. 06. 2019).

Krajíček, Jan und Pavel Pudlák. 1989. „Propositional proof systems, the consistency of first order theories and the complexity of computations“. In: *Journal of Symbolic Logic* 54.3 (Sep. 1989), S. 1063–1079. DOI: 10.2307/2274765.

Leven, Daniel und Zvi Galil. 1983. „NP completeness of finding the chromatic index of regular graphs“. In: *Journal of Algorithms* 4.1 (1. März 1983), S. 35–44. DOI: 10.1016/0196-6774(83)90032-9.

Levin, Leonid A. 1973. „Универсальные задачи перебора [‚Universelle Suchprobleme‘]“. In: *Проблемы Передачи Информации [„Problemy Peredachi Informatsii“]* 9.3 (1973), S. 115–116. URL: https://www.mathnet.ru/ppi914. Eine Übersetzung erscheint bei Trakhtenbrot (1984, S. 399–400).

Lynch, Nancy und Richard J. Lipton. 1978. „On Structure Preserving Reductions“. In: *SIAM Journal on Computing* 7.2 (Mai 1978), S. 119–126. DOI: 10.1137/0207010. (Besucht am 25. 08. 2023).

Mahaney, Stephen R. 1982. „Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis“. In: *Journal of Computer and System Sciences* 25.2 (1. Okt. 1982), S. 130–143. ISSN: 0022-0000. DOI: 10.1016/0022-0000(82)90002-2.

Megiddo, Nimrod und Christos H. Papadimitriou. 1991. „On total functions, existence theorems and computational complexity“. In: *Theoretical Computer Science* 81.2 (30. Apr. 1991), S. 317–324. DOI: 10.1016/0304-3975(91)90200-L.

Messner, Jochen. 2000. „On the simulation order of proof systems“. Diss. Universität Ulm, 2000. URL: https://citeseerx.ist.psu.edu/pdf/bec3958d845653cfa73493258d3b550a17e8defd (besucht am 03. 07. 2022). Archivierte Fassung des Originals.

Papadimitriou, Christos H. 1994. *Computational complexity*. Reading, Massachusets, USA: Addison-Wesley, 1994. ISBN: 978-0-201-53082-7.

Pudlák, Pavel. 2017. „Incompleteness in the finite domain“. In: *The Bulletin of Symbolic Logic* 23.4 (2017), S. 405–441. DOI: 10.1017/bsl.2017.32. (Besucht am 01. 04. 2019).

Selman, Alan L. 1988. „Natural Self-Reducible Sets“. In: *SIAM Journal on Computing* 17.5 (Okt. 1988), S. 989–996. DOI: 10.1137/0217062.

Selman, Alan L. 1994. „A taxonomy of complexity classes of functions“. In: *Journal of Computer and System Sciences* 48.2 (Apr. 1994), S. 357–381. DOI: 10.1016/S0022-0000(05)80009-1. (Besucht am 08. 01. 2022).

Simon, Janos. 1975. *On Some Central Problems in Computational Complexity*. Techn. Ber. 75-224. Ithaca, New York, USA: Department of Computer Science, Cornell University, 1975. URL: https://hdl.handle.net/1813/6975.

Thomason, A. G. 1978. „Hamiltonian Cycles and Uniquely Edge Colourable Graphs“. In: *Annals of Discrete Mathematics*. Hrsg. von B. Bollobás. Bd. 3. Advances in Graph Theory. Elsevier, 1. Jan. 1978, S. 259–268. DOI: 10.1016/S0167-5060(08)70511-9. (Besucht am 08. 11. 2023).

Trakhtenbrot, B.A. 1984. „A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms“. In: *Annals of the History of Computing* 6.4 (Okt. 1984), S. 384–400. DOI: 10.1109/MAHC.1984.10036.

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

Wechsung, Gerd. 2000. *Vorlesungen zur Komplexitätstheorie*. Bd. 32. Teubner-Texte zur Informatik. Wiesbaden: Vieweg+Teubner Verlag, 2000. DOI: 10.1007/978-3-322-80024-4.