Masterarbeit

KOMPLEXITÄT VON SUCHPROBLEMEN UND BEWEISSYSTEMEN

ANTON EHRMANNTRAUT

20. FEBRUAR 2024

BETREUER: PROF. DR. CHRISTIAN GLASSER

JULIUS-MAXIMILIANS-UNIVERSITÄT WÜRZBURG
LEHRSTUHL FÜR INFORMATIK I
ALGORITHMEN UND KOMPLEXITÄT

Inhaltsverzeichnis

1 Einleitung

Diese Arbeit beschäftigt sich mit algorithmischen Suchproblemen. Hierbei ist eine Eingabeinstanz gegeben, für welche eine entsprechende Lösung gesucht wird. Beispiele für solche Suchprobleme sind:

1. Gegeben eine positive Zahl n, berechne die Primfaktorzerlegung von n.
2. Sei G eine kontextfreie Grammatik. Gegeben ein Wort w, berechne einen Ableitungsbaum von w über G, oder gebe sonst „w nicht von G generiert" aus.
3. Gegeben ein Graph, berechne das größte Matching in diesem Graphen.
4. Gegeben ein Graph, berechne eine Knotenfärbung mit drei Farben, oder gebe sonst „nicht färbbar mit drei Farben" aus.
5. Gegeben ein Graph, berechne eine größte Clique in diesem Graphen.
6. Gegeben ein Graph und eine positive Zahl k, berechne eine Clique mit $\geq k$ Knoten in diesem Graphen, oder gebe sonst „keine Clique mit k Knoten möglich" aus.
7. Gegeben eine aussagenlogische Formel φ, bestimme eine erfüllende Belegung für φ, oder gebe sonst „unerfüllbar" aus.
8. Gegeben eine aussagenlogische Formel φ, bestimme einen Beweis (unter einem geeigneten Beweissystem, z.B. Resolution) für die Gültigkeit von φ, oder gebe sonst eine Belegung an, welche φ nicht erfüllt.
9. Gegeben einen prädikatenlogischen Satz φ in der Sprache der Arithmetik[1], bestimme einen Beweis (unter einem geeigneten vollständigem Kalkül, z.B. Sequenzenkalkül) für die Gültigkeit von φ (φ ist wahr in jeder Struktur), oder gebe sonst „φ ungültig" aus.

1. Gemeint ist die prädikatenlogische Sprache mit einer Konstante 0, einer unären Nachfolgerfunktion, je binären Funktionen $+$, $\times$, und binärer Relation $\leq$.

Innerhalb des Forschungsbereichs der theoretischen Informatik beschäftigt sich die Berechenbarkeitstheorie mit der Frage, welche dieser Aufgaben überhaupt algorithmisch berechenbar sind. Das Beispiel (9) ist beispielsweise überhaupt nicht berechenbar, in dem Sinn, dass kein Algorithmus existiert, welcher für jeden Satz φ nach endlicher Zeit mit der korrekten Lösung antwortet. Alle anderen Probleme (1)–(8) sind im Prinzip algorithmisch lösbar, indem alle möglichen Lösungskandidaten durchsucht werden.

Der Unterbereich der algorithmischen Komplexitätstheorie ist weniger an den prinzipiellen Grenzen von Berechenbarkeit interessiert, sondern fokussiert sich unter den berechenbaren Aufgaben damit, welche Ressourcen (Rechenzeit, Speicherplatz, zugeführte Zufälligkeit) hierfür notwendig sind. Die Komplexitätstheorie interessiert sich also dafür, welche dieser Aufgaben effizient durchgeführt werden können, und somit als umsetzbar für Computer angesehen werden.

In der Disziplin hat sich für „Effizienz" bzw. „Umsetzbarkeit" insbesondere folgender Konsens durchgesetzt: ein Algorithmus ist „effizient" wenn die Laufzeit des Algorithmus polynomiell mit der Eingabegröße wächst. Ein Algorithmus, bspw. für das Suchproblem

(3), muss also so beschaffen sein, dass eine Konstante $c > 0$ existiert, sodass bei der Verarbeitung eines doppelt so großen Graphen dieser Algorithmus nur c-mal so lang rechnet.

Unter den oben genannten Suchproblemen ist ein solcher Algorithmus mit polynomieller Laufzeit nur für Probleme (2) und (3) bekannt. Für die Suchprobleme (1), (4), (5), (6), (7), (8) lässt sich aber ein trivialer Suchalgorithmus mit exponentieller Laufzeit angeben. Für Suchproblem (4) bedeutet das z.B., alle möglichen exponentiell vielen Zuweisungen von Farben auszuprobieren.

In der Komplexitätstheorie wurden solche *Such*probleme wie (1)–(8) sehr früh in den Hintergrund verschoben, und stattdessen wurden die korrespondierenden *Entscheidungs*probleme in den Blick genommen. Anstelle nach einer Lösung zu suchen, wird sich darauf beschränkt zu entscheiden, *ob* eine Lösung existiert. Der Algorithmus muss also nur die Antwort „ja" oder „nein" ausgeben. Zugehörige Entscheidungsprobleme zu den oben genannten Suchproblemen wären:

- 2′. Sei G eine kontextfreie Grammatik. Gegeben ein Wort w, entscheide ob w aus G generiert werden kann.
- 4′. Gegeben ein Graph, entscheide ob dieser Graph mit drei Farben färbbar ist.
- 6′. Gegeben ein Graph und eine positive Zahl k, entscheide ob eine Clique mit $\geq k$ Knoten in diesem Graphen existiert.
- 7′. Gegeben eine aussagenlogische Formel φ, entscheide ob φ erfüllbar ist.
- 8′. Gegeben eine aussagenlogische Formel φ, entscheide ob φ gültig ist.
- 9′. Gegeben einen prädikatenlogischen Satz φ in der Sprache der Arithmetik, entscheide ob φ gültig ist.
- 10′. Gegeben eine Turing-Maschine M und eine Eingabe x, entscheide ob M auf Eingabe x nach endlich vielen Rechenschritten terminiert.

Beachte dass zu Suchproblemen (1), (3) und (5) keine unmittelbare Variante als Entscheidungsproblem existiert: es existiert immer eine Primfaktorzerlegung von n, analog existiert immer ein größtes Matching bzw. eine größte Clique in einem Graphen. Zusätzlich führen wir auch noch das Entscheidungsproblem 10′ ein, für das analog keine sinnvolle Variante als Suchproblem angegeben werden kann.

Auf dem ersten Blick erscheint dieser Schwerpunkt unnatürlich. In der Praxis sind wir interessiert, effizient Lösungen zu finden, um z.B. eine Karte einzufärben (Suchproblem 4), oder um einen Sourcecode zu parsen (Suchproblem 2). Die Feststellung „Karte ist dreifärbbar", „Sourcecode ist wohlgeformt" der Entscheidungsalgorithmen erscheint auf dem ersten Blick wenig hilfreich.

Für diese Fokussierung auf Entscheidungsprobleme gibt es durchaus Gründe. Zum einen ist klar, dass das Suchproblem nicht einfacher sein kann, als das zugehörige Entscheidungsproblem. Die Unberechenbarkeit eines Entscheidungsproblems schließt also auch die Berechenbarkeit des Suchproblems aus. Das entspricht genau der historischen Forschungsentwicklung zum „*Hilbertschen Entscheidungsproblem*" (9′), worauf Turing das *Halteproblem* (10′) reduziert hat. Mit der Unentscheidbarkeit des Halteproblems folgt die Unentscheidbarkeit des Entscheidungsproblems (9′), und damit der Unentscheidbarkeit des entsprechenden Suchproblems (9). Das Argument lässt sich auch auf die potentiell effizient lösbaren Entscheidungsprobleme übertragen. Die Komplexitätstheorie gibt Indizien, dass die Entscheidungsprobleme (4′), (6′)–(8′) wahrscheinlich nicht in Polynomialzeit lösbar sind, womit unmittelbar folgt, dass auch die Suchprobleme (4)–(8) nicht in Polynomialzeit lösbar sind.

Für die Fokussierung auf Entscheidungsprobleme innerhalb der Komplexitätstheorie gibt es zweitens auch fachgeschichtliche Gründe: zunächst war die Trennung *Entscheidungsproblem vs. Suchproblem* innerhalb der Berechenbarkeitstheorie meist nicht strikt notwendig, da „Berechenbarkeit“ zwischen den beiden Varianten meist äquivalent war. So ist die Berechenbarkeit des Hilbertschen Entscheidungsproblems (9′) tatsächlich äquivalent zur Berechenbarkeit des Suchproblems (9). (Falls der Entscheidungsalgorithmus „φ ist gültig“ ausgibt, dann enumeriere so lange alle Sequenzbeweise, bis einer φ beweist. Das terminiert nach Vollständigkeit des Sequenzkalküls.) Dann stand die Komplexitätstheorie der späten 1950er Jahre nah an der Automatentheorie und der Theorie der formalen Sprachen, als diese einen ersten Vorschlag zur Unterteilung der berechenbaren Aufgaben in „einfach“ und „schwer“ machten (vgl. Koucký, 2023). Eine zentrale Unterteilung in „Schwierigkeit“ bzw. Komplexität war z.B. die Hierarchie der formalen Sprachen von Chomsky (die Regulären als sehr einfach, die Kontextfreien als etwas komplexer, die Kontextsensitiven als noch komplexer). Die einzig relevanten Suchprobleme – Parsing wie in (2) – haben sich dann aber auch relativ schnell geklärt (z.B. CYK-Parsing für die Kontextfreien), womit die zentralen Untersuchungsfragen wohl eher waren, welche Sprachen durch welche Grammatiken (nicht) generiert werden können, bzw. welche Automaten welche Sprachen (nicht) erkennen können. Dafür ist die Beschränkung auf die Entscheidungsvariante („Generiert die Grammatik G genau die Sprache L? Erkennt der Kellerautomat A genau die Sprache L?“) ausreichend zur Etablierung unterer Schranken, und ist insbesondere auch dienlich im pragmatischen Sinn. Stellvertretend sei hier Kozen zitiert: „We do this for mathematical simplicity and because the behavior we want to study is already present at this level“ (1997, S. 7).

Dieser Pragmatismus setzt sich in der ressourcenfokussierten Komplexitätstheorie fort, die seit den 1960er Jahren den Ressourcenverbrauch von Algorithmen als zentraler Indikator für „Schwierigkeit“ versteht. Das betrifft insbesondere die Klassen P und NP; hierzu lassen sich die Aufgaben (1)–(8) und (2′)–(8′) zählen. Wieder reicht es in den meisten Fällen aus, sich auf die Entscheidungsprobleme zu beschränken. Diese Fokussierung ist durchaus fundiert: Einerseits, weil die zentralen algorithmischen Herausforderungen schon bei der Entscheidungsvariante auftreten („behavior we want to study is already present at this level“). So kann zum Beispiel die P-NP-Frage äquivalent als Frage über Entscheidungsprobleme auch als Frage über Suchprobleme formuliert werden. Andererseits lässt sich zeigen, dass für viele relevante Aufgaben das Suchproblem nicht schwerer ist als das Entscheidungsproblem (unter polynomieller Unschärfe). Dieses Argument wird üblicherweise als *search reduces to decision* formuliert: gegeben ein effizienter Algorithmus, welcher das Entscheidungsproblem löst, kann auch ein effizienter Algorithmus angegeben werden, welcher das Suchproblem löst. Mit diesem Argument kann z.B. die Aussage „Suchproblem (7) ist effizient lösbar“ äquivalent zu „Entscheidungsproblem (7′) effizient lösbar“ gesetzt werden. Die Konzentration auf Entscheidungsprobleme kommt dann unter anderem auch mit dem Vorteil, dass viele theoretische Konzepte einfacher zu fassen sind und kompakter zu formulieren sind („mathematical simplicity“). Wir können uns zum Beispiel auf (laufzeitbeschränkte) Algorithmen ohne Ausgabe konzentrieren, die Eingaben nur akzeptieren und ablehnen müssen.

NP-Suchprobleme als Forschungsgegenstand

Diese Arbeit setzt genau an dieser Fokussierung an Entscheidungsproblemen an. Die zentrale Motivation dieser Arbeit besteht darin, diese identifizierte Leerstelle aufseiten der Suchprobleme zu adressieren. Vor diesem Hintergrund wird sich diese Arbeit in vier Forschungsdesideraen den *NP-Suchproblemen* im Gegensatz zu den sonst üblichen NP-Entscheidungsproblemen nähern. Diese können als die Suchprobleme verstanden werden, die zu NP-Sprachen korrespondieren. Als Suchprobleme lässt sich eine einfache Charakterisierung formulieren: NP-Suchprobleme sind solche Suchprobleme, bei denen

- die Lösung – falls sie existieren sollte – höchstens polynomiell länger als die Eingabe ist, und
- effizient (d.h. in Polynomialzeit) verifiziert werden kann, ob ein fraglicher Lösungskandidat tatsächlich eine korrekte Lösung für eine Eingabe darstellt.

Das entspricht der sonst auch üblichen „Zertifikats-Definition“ der Komplexitätsklasse NP. Insbesondere induziert jede nichtdeterministische Polynomialzeit-Turing-Maschine ein NP-Suchproblem („gegeben Eingabe, finde einen akzeptierenden Rechenweg, oder gebe ‚lehnt ab‘ aus“) und umgekehrt.

Viele der anfangs genannten Suchprobleme bilden NP-Suchprobleme. Hierbei werden die „negativen Antworten“ als „es existiert keine Lösung“ verstanden. Dann ist beispielsweise das Suchproblem (4) ein NP-Suchproblem: die Färbung (Zuordnung von Knoten zu einer der drei Farben) ist höchstens so lang wie der Eingabegraph, und zu einer beliebigen Färbung (valide oder nicht) kann in Polynomialzeit überprüft werden, ob diese Färbung tatsächlich jeden zwei adjazenten Knoten eine unterschiedliche Farbe zugewiesen wird. Die Suchprobleme (1)–(4), (6), (7) sind ebenso NP-Suchprobleme.

Das Suchproblem (5) ist dagegen mutmaßlich kein NP-Suchproblem, denn es ist nicht bekannt wie verifiziert werden kann, dass eine Teilmenge C an Knoten in einem Graph tatsächlich eine *größte* Clique ist.[2] Das Suchproblem (8) ist auch mutmaßlich kein NP-Suchproblem, denn kein Beweissystem ist bekannt, dass Gültigkeit mit polynomiell langen Beweisen ausdrücken kann. Zumindest für das Resolutionskalkül existieren spezielle gültige Formeln φ mit exponentiell langen Resolutionsbeweisen.

Die Einschränkung auf NP-Suchprobleme ist im Wesentlichen eine Konsequenz der hohen Wichtigkeit und Relevanz der Komplexitätsklasse NP, sowohl theoretisch innerhalb der Komplexitätstheorie („Wie viel hilft Nichtdeterminismus den Polynomialzeit-Berechnungen?“), als auch in der Praxis, da sehr viele interessante und in der industriellen Anwendung aufkommenden Berechnungsaufgaben als NP-Suchprobleme formuliert werden können. Hinzu kommt die Beobachtung, dass jene Suchprobleme, welche nicht den Bedingungen von NP-Suchproblemen genügen, so gut wie definitiv zu komplex und schwer sind, um zu erwarten dass sie überhaupt effizient gelöst werden können. Für die NP-Suchprobleme (und insbesondere die nicht-vollständigen NP-Suchprobleme) ist es zumindest noch plausibel, effiziente Suchalgorithmen entwickeln zu können.

Wie aber bereits oben angesprochen, werden üblicherweise in der Literatur nicht die NP-Suchprobleme untersucht, sondern meist nur die entsprechenden NP-Entscheidungsprobleme. Das geschieht mit der Begründung, dass sich die meisten Suchprobleme auf das jeweilige Entscheidungsproblem reduzieren lassen können (*search reduces to decision*).

Als erstes Forschungsdesiderat möchte diese Arbeit genau jene Beziehung zwischen

2. Suchproblem (3) fragt auch nach einer optimalen Lösung, ist aber ein pathologisches NP-Suchproblem, denn ein größtes Matching kann ohnehin in Polynomialzeit berechnet werden. Die „Verifikation“ besteht also darin zu überprüfen, ob die fragliche Lösung genau so viele Pärchen bildet wie die ad hoc berechnete optimale Lösung.

NP-Suchproblemen und NP-Entscheidungsproblemen näher untersuchen. Insbesondere wollen wir das *search-reduces-to-decision*-Argument präzise einordnen und auch zeigen, dass dieses Argument nicht immer zutrifft, also Situationen in denen eine reine Betrachtung der Entscheidungsvarianten eigentlich nicht ausreicht.

Tatsächlich gilt das für viele interessante Suchprobleme. Das sind zum Beispiel schon jene NP-Suchprobleme, die immer eine Lösung haben; hier kann zunächst nicht unmittelbar ein entsprechendes Entscheidungsproblem formuliert werden. Das haben wir bereits bei den Suchproblemen (1) und (3) gesehen. Diese *totalen* NP-Suchprobleme sind insofern interessant, da viele effizient lösbar sind (z.B. Suchproblem 3), andererseits für viele die effiziente Lösbarkeit noch offen ist. Gleichzeitig wird erwartet, dass die totalen NP-Suchprobleme nicht NP-hart sind; damit ist die effiziente Lösbarkeit zumindest dieser Suchprobleme durchaus in Reichweite. Das trifft zum Beispiel auf das Suchproblem (1) der Faktorisierung zu. Beachte, dass die Sicherheit des RSA-Kryptosystems maßgeblich von der Unlösbarkeit der Faktorisierung abhängt. Dieses totale NP-Suchproblem ist momentan nicht effizient lösbar, aber gleichzeitig auch nicht NP-hart. Die Untersuchung solcher totalen NP-Suchprobleme geht im Wesentlichen auf Johnson, Papadimitriou und Yannakakis (1988) und Megiddo und Papadimitriou (1991) zurück.

Fenner, Fortnow, Naik und Rogers (2003) können die Vermutung „alle totalen NP-Suchprobleme sind effizient lösbar" in verschiedenen äquivalenten Formulierungen charakterisieren, so zum Beispiel als Invertierbarkeit von surjektiven Funktionen, oder als das effiziente Lösen des Suchproblems (7) unter Angabe einer nichtdeterministischen Turing-Maschine, welche die Menge $\mathtt{SAT}$ der erfüllbaren aussagenlogischen Formeln erkennt. Sie fassen diese jeweils äquivalenten Charakterisierungen unter der Hypothese Q zusammen. Erste Nennungen finden sich bereits in Arbeiten von Fortnow und Rogers (1993). Für diese Arbeit wird hier folgende Formulierung von Q angewendet, die über nichtdeterministische Turing-Maschinen spricht, welche jede Eingabe akzeptieren:

Vermutung 1.1 (Q). *Für jede nichtdeterministische Turing-Maschine N mit polynomieller Laufzeitbeschränkung, und $L(N) = \Sigma^*$ existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ eine akzeptierender Rechenweg von $N(x)$ ist.*

In dieser Formulierung wird auch die Schwierigkeit dieser Vermutung klar: wir bekommen zwar versprochen, dass $N(x)$ akzeptiert, wir wissen aber nicht, *welcher* nichtdeterministische Rechenweg x akzeptiert, und nicht wie wir einen solchen Rechenweg effizient berechnen könnten.

Obwohl Q zunächst nur über totale Suchprobleme spricht, hat die Hypothese Q große Nähe und Verwandtschaft zu analogen Aussagen, die NP-Suchprobleme einerseits und die sogenannten Beweissysteme andererseits betreffen. Als zweites Desiderat will daher diese Arbeit an den Charakterisierungen von Q weiter arbeiten, sowie die Beziehung zwischen Q und NP-Suchproblemen, den Beweissystemen (nach Cook und Reckhow, 1979) bzw. dem *Pudlákschen Programm* (2017) näher untersuchen.

Beweissysteme und das Pudláksche Programm

NP-Suchprobleme wie oben eingeführt korrespondieren auf natürliche Weise mit „Beweissystemen" im intuitiven Sinn. Wir gehen das beim Suchproblem (7) durch: sollte eine Formel φ erfüllbar sein, dann existiert ein „Beweis" für die Erfüllbarkeit von φ, nämlich eben eine Belegung w welche φ erfüllt. Somit ist dieses Beweissystem gewissermaßen vollständig. Dieser Beweis ist nicht nur kurz, sondern kann effizient (gemeint ist: mit einem Algorithmus in Polynomialzeit) überprüft werden, ob der Beweis w tatsächlich zu φ „passt", also ob w die Formel φ erfüllt. Damit ist dieses Beweissystem auch korrekt.

Jedes NP-Suchproblem nach der obigen Definition induziert dann ein solches korrektes und vollständiges Beweissystem. Diese Beweissysteme sind sogar insofern besonders stark, als da zu jeder korrekten Instanz ein Beweis existiert, der sogar nur polynomiell länger ist. Insbesondere induziert ein solches Beweissystem mit polynomiell kurzen Beweisen ein NP-Suchproblem (gegeben Instanz, suche einen korrekten Beweis für die Instanz) und umgekehrt.

Das bei Suchproblem (8) angedeutete Beweissystem der Resolution für die Gültigkeit aussagenlogischer Formeln ist zwar ein Beweissystem für die Tautologien, jedoch wie bereits angesprochen, eines ohne *polynomiell lange* Beweise. Zumindest für die Resolution bildet damit (8) kein NP-Suchproblem. Existiert ein polynomiell beschränktes Beweissystem für die aussagenlogischen Tautologien?

Dieser Frage gingen Cook und Reckhow (1979) nach, und erarbeiten hierfür zunächst eine knappe und elegante Definition von aussagenlogischen Beweissystemen: *Eine Polynomialzeit-berechenbare Funktion f ist ein* aussagenlogisches Beweissystem, *wenn der Bildbereich von f mit der Menge* TAUT *der Tautologien übereinstimmt*. Wenn $f(w) = \varphi$, dann wissen wir dass φ eine Tautologie ist, und dieser Fakt wird insbesondere über den Beweis w im Beweissystem f erfasst. Diese Definition erfasst damit genau die oben genannten intuitiven Eigenschaften:

- Die Relation *„w ist ein Beweis für φ"* ist in Polynomialzeit entscheidbar.
- Das Beweissystem ist korrekt: f beweist nur Tautologien.
- Das Beweissystem ist vollständig: zu jeder Tautologie φ existiert ein Beweis w, d.h. $f(w) = \varphi$.

Für das Resolutionskalkül könnte ein solches aussagenlogisches Beweissystem in dieser Form so aufgeschrieben werden:

$$h(\varphi, w) \stackrel{\text{df}}{=} \begin{cases} \varphi & w \text{ ist Resolutionsbeweis für die Gültigkeit von } \varphi, \\ \perp & \text{sonst.} \end{cases}$$

Hat ein aussagenlogisches Beweissystem f für jede Tautologie φ einen höchstens polynomiell längeren Beweis w für φ, sagen wir dass *f kurze Beweise hat*. Das aussagenlogische Beweissystem h hat *keine* kurzen Beweise. Die oben gestellte Frage, ob (8) ein NP-Suchproblem ist, lässt sich äquivalent mit der Frage charakterisieren, ob ein aussagenlogisches Beweissystem mit kurzen Beweisen existiert. Tatsächlich beobachten Cook und Reckhow sogar, dass diese Existenz äquivalent zur Aussage NP = coNP ist.

Diese Einsicht motivierte das sogenannte *Cook–Reckhow-Programm* (vgl. Buss, 1996): Hierbei nähern wir uns der Frage NP vs. coNP durch Untersuchen immer stärkere aussagenlogische Beweissysteme. Um NP $\neq$ coNP zu erreichen, könnten wir entweder zeigen dass kein (längen-)optimales aussagenlogisches Beweissystem (d.h. ein Beweissystem welches höchstens polynomiell längere Beweise als jedes andere Beweissystem hat) existiert, oder ein optimales aussagenlogisches Beweissystem angeben, sodass dieses keine kurzen Beweise hat. Aufbauend auf dieser Verbindung wurden zunehmend auch untere und obere Schranken von speziellen aussagenlogischen Beweissystemen untersucht,

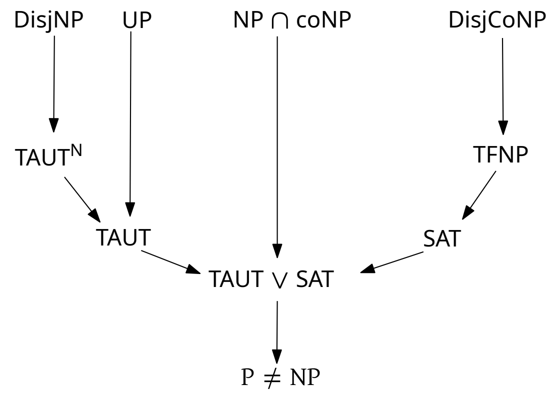

Abbildung 1: Implikationen zwischen Pudláks Hypothesen (2017). Beachte dass diese Implikationen relativieren.

sowie auch Beweissysteme allgemein für beliebige Mengen (und nicht nur Tautologien) betrachtet.

Die Existenz von optimalen Beweissystemen bzw. P-optimalen (d.h. optimal in dem Sinn dass sogar die Beweise zwischen den Beweissystemen effizient übersetzt werden können) Beweissystemen wurde von Krajíček und Pudlák (1989) in Beziehung gesetzt mit endlicher Konsistenz von mathematischen Theorien. Darauf aufbauend zeigt Pudlák (2013, Kap. 6, 2017) ferner Verbindungen zwischen P-optimalen bzw. optimalen Beweissystemen, Arithmetik mit exponentiell beschränkten Quantoren („*bounded arithmetic*") und der Existenz von vollständigen Elementen sogenannter *Promise-Klassen*. Promise-Klassen sind solche Komplexitätsklassen, deren Mengen durch speziell operierende Turing-Maschinen mit speziellen Eigenschaften erkannt werden können, wobei diese Eigenschaften üblicherweise über (Nicht)determinismus und polynomielle Laufzeit hinaus gehen. Für die Klasse $\mathrm{UP} \subseteq \mathrm{NP}$ bedeutet das z.B., dass die Sprache (wie bei NP) von einer nichtdeterministischen Polynomialzeit-Turing-Maschine erkannt werden muss, die aber – das ist der Promise – auch nur auf höchstens einem nichtdeterministischen Rechenweg akzeptieren darf.

Razborov (1994) zeigt hierbei als erstes eine Verbindung zwischen der Promise-Klasse DisjNP und der Existenz von optimalen aussagenlogischen Beweissystemen. Viele weitere Beziehungen zur Existenz vollständiger Elemente der Promise-Klassen UP, $\mathrm{NP} \cap \mathrm{coNP}$, DisjCoNP wurden ausgemacht (vgl. auch Messner, 2000; Köbler, Messner und Torán, 2003; Beyersdorff und Sadowski, 2011). Beyersdorff, Köbler und Messner (2009) und Pudlák (2017) zeigen ferner analoge Verbindungen zu den Funktionenklassen NPMV_t und TFNP.

Motiviert durch Fragen der endlichen Widerspruchsfreiheit von Theorien und *bounded arithmetic* formuliert Pudlák (2017) folgende Hypothesen, die hier in ihrer komplexitätstheoretischen Fassung genannt werden:

SAT	:	es ex. keine $\leq_m^p$-vollst. Menge für NP mit P-opt. Beweissystem
TAUT	:	es ex. keine $\leq_m^p$-vollst. Menge für coNP mit P-opt. Beweissystem[3]
TAUT^N	:	es ex. keine $\leq_m^p$-vollst. Menge für coNP mit opt. Beweissystem
$\mathrm{NP} \cap \mathrm{coNP}$	:	es ex. keine $\leq_m^p$-vollst. Menge für UP
UP	:	es ex. keine $\leq_m^p$-vollst. Menge für $\mathrm{NP} \cap \mathrm{coNP}$
DisjNP	:	es ex. kein $\leq_m^{pp}$-vollst. disjunktes NP-Paar für DisjNP
DisjCoNP	:	es ex. kein $\leq_m^{pp}$-vollst. disjunktes coNP-Paar für DisjCoNP

3. In der Notation von Pudlák CON bzw. CON^N für die nichtuniforme Variante. Beachte auch, dass Pudlák die Hypothese TAUT nicht so wie hier formuliert hat, sondern als Aussage über die Nicht-Existenz eines P-optimalen aussagenlogischen Beweissystems (also speziell für die Menge TAUT), genau wie anfangs des Abschnitts gefragt wurde. Die beiden Charakterisierungen sind aber äquivalent. Gesagtes gilt analog auch für SAT und TAUT^N, vgl. Abschnitt 2.5.

Seien an dieser Stelle auch schon die folgenden zwei wichtigen natürlichen Mengen definiert:

$$\begin{aligned}\texttt{SAT} &\stackrel{\text{df}}{=} \{\varphi \mid \varphi \text{ ist eine erfüllbare aussagenlogische Formel}\},\\ \texttt{TAUT} &\stackrel{\text{df}}{=} \{\varphi \mid \varphi \text{ ist eine gültige aussagenlogische Formel}\}.\end{aligned}$$

Zur Notation: natürliche Mengen wie TAUT werden in Schreibmaschinenschrift notiert, während Hypothesen in serifenloser Schrift notiert werden.

Insbesondere arbeitet Pudlák die Beziehung zwischen diesen einzelnen Hypothesen heraus, und kommt hierbei zu Abbildung 1, die anzeigt, wie die bekannten Implikationen zwischen den oberen Hypothesen verlaufen. Auf der Seite der Komplexitätstheorie fragt Pudlák im Speziellen nach natürlichen plausiblen stärkeren Hypothesen (die z.B. sowohl TAUT als auch TFNP implizieren, wobei Pudlák diese beiden Hypothesen jeweils als plausibel ansieht), sowie im Allgemeinen nach Separationen zwischen diesen Hypothesen. Zum Beispiel kann durch Angabe von Orakeln gezeigt werden, dass zwei Hypothesen unter relativierbaren Beweisen nicht gleich sind, oder stärker sogar unabhängig unter relativierbaren Beweisen sind. Dieses allgemeine Forschungsdesiderat fasse ich für diese Arbeit lose als *Pudlácksches Programm* zusammen.

Die vorliegende Arbeit will drittens an genau diesem Pudlákschen Programm beitragen, indem im Wesentlichen die Übersicht in Abbildung 1 verfeinert wird, und dabei stärkere (und schwächere) Hypothesen eingeordnet werden. Hierbei fokussiere ich mich insbesondere auf jene Hypothesen, die mit NP-Suchproblemen im Zusammenhang stehen, wie z.B. Q. Im Folgenden sei noch auf den „Orakel“-Teil des Pudlákschen Programms eingegangen.

Orakel und Relativierungen

Orakel und Orakel-Turing-Maschinen sind Begriffe aus der Berechenbarkeitstheorie, um die relative Schwierigkeit von algorithmischen Entscheidungsproblemen zu untersuchen, die über *berechenbar vs. unberechenbar* hinausgehen. Wenn eine Turing-Maschine eine Abstraktion eines Computers darstellt, dann ist ein Orakel eine Abstraktion einer Datenbank in der Cloud, die vom Computer angerufen werden kann, um zu fragen, ob ein gewisser Eintrag in der Datenbank liegt. Dieses Abfragen des Entscheidungsproblems („Ist Eintrag in Datenbank?“) kann der Computer gewissermaßen gratis durchführen.

Formal werden Orakel als Mengen A realisiert, und Orakel-Turing-Maschinen dürfen für beliebige aufgeschriebene Wörter w abfragen, ob $w \in A$ liegt. Die Orakel-Turing-Maschine erhält dann sofort die Antwort auf diese Frage. Ist nun A insbesondere eine unberechenbare Menge, dann kann die zugehörige Turing-Maschine auch komplexere Mengen entscheiden, die sonst unberechenbar wären. Post (1944) arbeitet aus diesem Begriff die Turing-Reduzierbarkeit aus: A ist auf B Turing-reduzierbar wenn A über eine Orakel-Turing-Maschine mit Orakel B entschieden werden kann. In anderen Worten: A kann mittels Hilfe von Orakel B entschieden werden. Damit können sonst unberechenbare Mengen A und B nach ihrer relativen Schwierigkeit über einfache Berechenbarkeit hinaus geordnet werden können. Diese Ordnung ermöglicht zum Beispiel die Unterteilung der unentscheidbaren Mengen in *Grade der Unlösbarkeit*.

Cook (1971) überträgt diese Form von Reduzierbarkeit auf den polynomiellen Bereich der Komplexitätstheorie, um so die relative Komplexität zwischen zwei Mengen A und B unter polynomieller Unschärfe einzuschätzen: A ist auf B Cook-reduzierbar wenn A mit einer Orakel-Turing-Maschine mit Orakel B in Polynomialzeit entschieden werden kann. Auf ähnliche Weise wurde der Begriff von Orakeln im polynomiellen Bereich eingesetzt, um die Polynomialzeit-Hierarchie zu definieren, womit NP generalisiert wird, und der Komplexitätsraum zwischen P und PSPACE verfeinert werden kann.

Neben diesen deskriptiven Eigenschaften haben sich Orakel als nützliches beweistheoretisches Werkzeug in der Komplexitätstheorie erwiesen. Die zentrale Einsicht hierbei ist, dass viele der üblichen mathematischen Beweismethoden, welche in der Komplexitätstheorie eingesetzt werden, *relativieren*. Das bedeutet, dass diese mathematischen Beweise nicht nur die eigentliche Aussage (wie z.B. der Hierarchiesatz $\mathrm{P} \neq \mathrm{E}$) beweisen, sondern für jedes Orakel A diese Beweise auch die *relativierte* Aussage beweisen, bei der alle beteiligten Turing-Maschinen Zugriff auf das Orakel A bekommen. Die Aussage $\mathrm{P} \neq \mathrm{E}$ relativiert so zur Aussage $\mathrm{P}^A \neq \mathrm{E}^A$ für jedes A, das bedeutet dass eine Menge L existiert, die von einer Exponentialzeit-Orakel-Turing-Maschine mit Zugriff auf A erkannt wird, aber keine Polynomialzeit-Turing-Maschine kann L entscheiden, selbst mit Orakel-Zugriff auf A.

Damit werden speziell konstruierte Orakel zu einem Indiz, 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. Tatsächlich zeigen Baker, Gill und Solovay sogar zusätzlich, dass $\mathrm{P}^B = \mathrm{NP}^B$ für ein zweites Orakel B. Damit kann also auch die Aussage „$\mathrm{P} \neq \mathrm{NP}$“ nicht mit relativierbaren Methoden -- also den üblichen Methoden -- bewiesen werden. Nimmt man diese beiden Indizien zusammen, ergibt sich dass die P-NP-Frage *unabhängig* unter relativierbaren Beweisen ist.

Im Kontext des Pudlákschen Programms wurde für viele potentielle Implikationen (wie z.B. $\mathsf{DisjCoNP} \Rightarrow \mathsf{TAUT}$) ein Orakel konstruiert, relativ zu diesem diese Implikationen nicht gelten (es existiert ein Orakel, relativ zu dem $\mathsf{DisjCoNP}$ gilt aber nicht TAUT). Entsprechende Konstruktionen wurden unter anderem von Glaßer, Selman, Sengupta und Zhang (2004), Dose und Glaßer (2019), Dose (2020b,c), Dingel (2022), Ehrmanntraut, Egidy und Glaßer (2022) und Khaniki (2022) entwickelt. Damit wird plausibilisiert, dass gewisse Hypothesen des Pudlákschen Programms tatsächlich unterschiedlich sind.

Diese Arbeit reiht sich in dieses Arbeitsvorhaben direkt ein, und wird viertens ein weiteres Orakel konstruieren, um Hypothesen (unter relativierbaren Beweisen) zu trennen.

Beitrag und Überblick

Der Aufbau der Arbeit und die einzelnen Beiträge sind hier noch einmal zusammengefasst.

Im nächsten Kapitel 2 klären wir die notwendigen mathematischen Grundlagen. Insbesondere definieren wir präzise den Begriff des (Cook–Reckhow-)Beweissystems und den der Relativierungen, welche bereits oben angesprochen wurden.

Im Kapitel 3 formalisieren wir den oben bereits intuitiv erfassten Begriff der NP-Suchprobleme und totalen NP-Suchprobleme. Einerseits wird die Beziehung zwischen NP-Suchproblemen und NP-Entscheidungsproblemen erläutert, einschließlich des *search-reduces-to-decision*-Arguments. Es werden Ergebnisse zusammengetragen, die aufzeigen, wann dieses Argument zutrifft, und wann es insbesondere nicht zutrifft. Andererseits werden wir die NP-Suchprobleme mittels der Levin-Reduzierbarkeit, ähnlich zu Karp-Reduzierbarkeit auf den Entscheidungsproblemen, ordnen, und so einen Vollständigkeitsbegriff erarbeiten. Im Speziellen werden wir die NP-vollständigen Suchprobleme betrachten und mit den NP-vollständigen Entscheidungsproblemen gegenüberstellen.

In Kapitel 4 werden wir Hypothesen zu NP-Suchproblemen und die Aussage Q in das

Pudláksche Programm einordnen. Erstens untersuchen wir, ob die (Levin-)Vollständigkeit eines NP-Suchproblems mit der (Karp-)Vollständigkeit des entsprechenden NP-Entscheidungsproblems übereinstimmt, und stellen diese als offene Frage anderen Hypothesen gegenüber. Zweitens werden wir (nicht-relativierbare) Charakterisierungen von Q durch Fenner, Fortnow, Naik und Rogers (2003) und Köbler und Messner (2000) verallgemeinern und relativieren, womit wir als Nebeneffekt auch präzise zu (relativierbaren) Implikationen zwischen Q und den Pudlákschen Hypothesen kommen. Drittens wird zum Abschluss des Kapitels, wieder in Form eines Surveys, das Pudláksche Programm durch das Hinzufügen weiterer Hypothesen erweitert und verfeinert, welche in der Literatur diskutiert werden. Zudem erarbeiten wir eine Übersicht über die Implikationen zwischen diesen Hypothesen, sowie Orakeln welche diese Implikationen unter relativierbaren Beweisen trennen. Im Wesentlichen wird Abbildung 1 zu Abbildungen 7 und 8 (S. 45, 62) vergrößert.

Im Sinne dieser Trennungen unter relativierbaren Beweisen wird in Kapitel 5 ein Orakel konstruiert, relativ zu diesem DisjNP und UP und Q gilt. Damit gibt es also keinen relativierbaren Beweis für $\neg$Q, selbst wenn man beide Pudlákschen Hypothesen DisjNP und UP annimmt. Insbesondere wird damit der erste Schritt unternommen, die relative Stärke der Hypothese Q gegenüber den anderen der Pudlákschen Hypothesen unter relativierbaren Beweismethoden abzugrenzen.

Die Arbeit endet mit einer abschließenden Diskussion in Kapitel 6, erläutert die Ergebnisse noch einmal, und trägt die offenen Fragen zusammen, welche in den vorigen Kapiteln gestellt wurden.

2 Grundlagen

Dieses Kapitel legt die Grundlagen für die folgenden Kapitel fest. Abschnitt 2.1 erläutert mathematische Notationen für diese Arbeit. 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^* nennen wir (endliche) *Wörter*, sind also endliche Sequenzen von Zeichen aus Σ. Die Menge Σ^ω entspricht der Menge der ω-unendlichen Wörter. 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 $w \in \Sigma^* \cup \Sigma^\omega$ für $0 \leq i < |w|$ identifizieren wir mit $w[i]$. Falls w ein (echter) Präfix von v ist dann schreiben wir $w \sqsubseteq v$ (bzw. $w \sqsubsetneq v$). Gelegentlich schreiben wir auch $\Sigma^{\leq n}$ für die Menge $\{w \in \Sigma^* \mid |w| \leq n\}$. In manchen Fällen identifizieren wir Mengen $A \subseteq \mathbb{N}$ mit ihrem charakteristischem ω-Wort aus Σ^ω, wobei $A = a_0a_1a_2\cdots$, alle $a_i \in \Sigma$ und $i \in A$ genau dann wenn $a_i = 1$. Heißt, es gilt $A[i] = 1$ genau dann wenn $i \in A$.

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 $|A|$. Außerdem bezeichnet $\ell(A) \overset{\text{df}}{=} \sum_{w \in A} |w|$. Für solche Teilmengen A von Σ^* verstehen wir das Komplement $\overline{A}$ als $\Sigma^* - A$. Wir sagen, dass zwei Mengen A, B *auf Menge D übereinstimmen* wenn $x \in A$ genau dann wenn $x \in B$ für alle $x \in D$, bzw. äquivalent dazu $A \cap D = B \cap D$.

Für zwei Mengen $A, B \subseteq \Sigma^*$ bezeichnet der *Join* $A \oplus B$ die Menge

$$A \oplus B \overset{\text{df}}{=} \{0a \mid a \in A\} \cup \{1b \mid b \in B\}.$$

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 B mit genau einem Element aus A reliert, also genau dann wenn R surjektiv und injektiv ist.

Binäre Relationen nennen wir eine (partielle) *Funktion* wenn diese Relation funktional ist. Eine Funktion sei im Folgenden also 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 sein müssen, verstehen wir manchmal auch aus historischen Gründen als (partielle) Multifunktionen, dem Begriff der „*partial multivalued function*" nachempfunden.

Für eine binäre Relation $R \subseteq A \times B$ schreiben wir $\mathrm{Proj}(R)$ für die Menge $\{x \mid (x, y) \in R\}$. Für ein Element $x \in A$ schreiben wir $\textit{set-}R(x) = \{y \mid (x, y) \in R\}$ für die Bildmenge von x auf R Manchmal werden wir binäre Relationen auch über die Spezifikation der jeweiligen Bildmengen definieren, also z.B. $Q \subseteq \mathbb{N} \times \mathbb{N}$, $\textit{set-}Q(n) \overset{\text{df}}{=} \{0, 1, \dots, n\}$ 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, und nicht die *Bildmenge*.

Für eine Funktion f bezeichnen wir die Urbild- bzw. Bildmenge (*domain* und *image*) mit $\mathrm{dom}(f)$ und $\mathrm{img}(f)$. (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 binäre Umkehrrelation. Beobachte dass f^{-1} funktional ist, wenn f injektiv ist. Ist f zusätzlich surjektiv, dann ist die Umkehrfunktion f^{-1} eine totale Funktion.

Eine Funktion $f\colon \Sigma^* \rightarrow \Sigma^*$ nennen wir *verlängernd* wenn $|f(x)| \geq |x|$ für alle $x \in \mathrm{dom}(f)$. Wir nennen f *polynomiell längenbeschränkt* wenn ein Polynom p existiert sodass $|f(x)| \leq p(|x|)$ für alle $x \in \mathrm{dom}(f)$. Wir nennen f *ehrlich* wenn ein Polynom q existiert sodass $q(|f(x)|) \geq |x|$ für alle $x \in \mathrm{dom}(f)$. Manchmal übertragen wir diese Begrifflichkeiten auf allgemeine binäre Relationen, und sprechen z.B. von polynomiell längenbeschränkten Relationen R wenn $|y| \leq p(|x|)$ für alle $(x, y) \in R$.

Im Folgenden definieren wir noch den Begriff der *Verfeinerung* auf Multifunktionen. Seien F, G zwei Multifunktionen. Wir nennen G eine *Verfeinerung* von F wenn $\mathrm{Proj}(F) = \mathrm{Proj}(G)$ und $\textit{set-}G(x) \subseteq \textit{set-}F(x)$ für alle $x \in \mathrm{Proj}(F)$ (bzw. äquivalent $\in \mathrm{Proj}(G)$). Ist F eine Multifunktion, und $\mathcal{G}$ eine Klasse von Multifunktion, 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 $0 \prec_{\mathrm{lex}} 1$ fordern. Unter dieser Definition existiert ein Ordnungsisomorphismus zwischen $(\Sigma^*, \prec_{\mathrm{lex}})$ und $(\mathbb{N}, <)$, welcher insbesondere eine totale bijektive Abbildung zwischen Σ^* und $\mathbb{N}$ induziert, die 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 aus $\mathbb{N}$ behandeln und umgekehrt. Es können 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 $\preceq_{\text{lex}}$ 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 $a \preceq_{\text{lex}} b$) 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 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: $|\langle u_1, \dots, u_n\rangle| = 2(|u_1| + \cdots + |u_n| + n)$. Eine solche Paarungsfunktion kann beispielsweise über $\langle u_1, \dots, u_n\rangle \mapsto f(\#u_1\#\cdots\#u_n)$ 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.

Wenn also f eine Funktion ist, meinen wir mit „$f \in \mathrm{P}$“ dass *der Graph* von f, also die Menge $\{\langle x, y\rangle \mid f(x) = y\}$ in Polynomialzeit entschieden werden kann. 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.

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 ein Eingabeband, ein Arbeitsband, und ein Ausgabeband. Jede TM besitzt einen ausgezeichneten Zustand zum Akzeptieren. In jedem Rechenschritt einer TM geht die TM von einer Konfiguration in eine nächste Konfiguration über (wobei diese in der nichtdeterministischen Variante nicht eindeutig sein muss). Mit Konfiguration meinen wir hier eine Beschreibung der Bandinhalte, der Kopfpositionen und dem Zustand.

Für (nichtdeterministische oder deterministische) TM M und Eingabe $x \in \Sigma^*$ sagen wir, dass *α ein Rechenweg der Berechnung $M(x)$* ist, wenn α eine Sequenz $C_0, C_1, \dots$ von Konfigurationen ist, diese mit der initialen C_0 Konfiguration startet, und in der die Konfiguration C_{i+1} aus C_i über einen (bzgl. M) legalen Rechenschritt hervorgeht.

Wir sagen, dass ein Rechenweg $\alpha = (C_0, \dots, C_m)$ *terminierend* ist, wenn kein weiterer legaler Rechenschritt (bzgl. M) mehr aus C_m möglich ist. Ein Rechenweg ist *akzeptierend* wenn dieser terminierend ist, und der Zustand der letzten Konfiguration C_m der akzeptierende Zustand ist. Entsprechend sagen wir auch, dass $M(x)$ *bezüglich α mit Ausgabe y akzeptiert* wenn α ein akzeptierender Rechenweg ist, und y auf dem Ausgabeband steht. Analog sagen wir, dass ein Rechenweg *ablehnt* wenn dieser terminiert, aber nicht akzeptiert.

Wir betrachten nun konkret deterministische TM. Nachdem höchstens ein terminierender Rechenweg α einer Berechnung $M(x)$ einer deterministischen TM M auf Eingabe x existiert, können wir verkürzend auch sagen, dass $M(x)$ *mit Ausgabe y akzeptiert* oder kurz $M(x)$ *akzeptiert*, wenn $M(x)$ bezüglich α mit Ausgabe y akzeptiert. Sollte α dagegen ablehnend sein, oder die Berechnung $M(x)$ nicht terminieren, dann sagen wir, dass $M(x)$ *ablehnt*.[4]

4. Im Verlauf dieser Arbeit müssen wir unbeschränkt lang rechnenden TM nicht berücksichtigen, und identifizieren daher „Ablehnen" als Negation von „Akzeptieren" also mit „ablehnend terminieren oder nicht terminieren".

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

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

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

Wenn wir M im Kontext der zweiten Berechnungsweise 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 Berechnungsweisen einer TM erweitern wir nun auf nichtdeterministische TM. Sei N eine nichtdeterministische TM, und x eine Eingabe. Dann existieren bezüglich Berechnung $N(x)$ nicht nur eine, sondern ggf. mehrere terminierende Rechenwege. Im Sinne eines existentiellen Akzeptierverhaltens sagen wir dass $N(x)$ *akzeptiert* wenn *mindestens* ein akzeptierender Rechenweg α auf $N(x)$ existiert. Ansonsten sagen wir, dass $N(x)$ *ablehnt*.

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

- Die von N *entschiedene Sprache* ist die Menge

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

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

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

Die berechnete Multifunktion 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 jenes Wort, was auf dem Ausgabeband steht. Wie im deterministischen Fall können wir von nichtdeterministischen Turing-Transduktoren sprechen, wenn wir die zweite Berechnungsweise betonen wollen. Ebenso können wir wieder abkürzend von „der Multifunktion N“ sprechen.

Sowohl im deterministischen als auch dem nichtdeterministischen Fall können wir Berechnungen eine *Laufzeit* zuordnen: für eine TM M und Eingabe $x \in \Sigma^*$ sei

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

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

Eine TM bzw. NTM M nennen wir *total* falls $L(M) = \Sigma^*$, heißt $M(x)$ akzeptiert jede Eingabe x.

2.3 Komplexitätsklassen

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

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

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

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

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

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

$$\begin{aligned}
\mathrm{FP} &\overset{\text{df}}{=} \{f \subseteq \Sigma^* \times \Sigma^* \mid f \text{ ist funktional, ex. PTM-Transduktor } M, \text{der } f \text{ berechnet}\},\\
\mathrm{NPSV} &\overset{\text{df}}{=} \{f \subseteq \Sigma^* \times \Sigma^* \mid f \text{ ist funktional, ex. NPTM-Transduktor } M, \text{der } f \text{ berechnet}\},\\
\mathrm{NPMV} &\overset{\text{df}}{=} \{f \subseteq \Sigma^* \times \Sigma^* \mid \text{ex. NPTM-Transduktor } N, \text{der } f \text{ berechnet}\}.
\end{aligned}$$

Gegeben eine Funktionenklasse $\mathcal{F}$ definieren wir $\mathcal{F}_t$ als diejenige Teilmenge von $\mathcal{F}$ der Multifunktionen $f \in \mathcal{F}$, die total sind. Analog sei $\mathcal{F}_g$ die Teilmenge der Multifunktionen $f \in \mathcal{F}$, für die (der Graph) f in P liegt, also für die in Polynomialzeit überprüft werden kann, ob für (x, y) auch $f(x) = y$ gilt.

Ist für eine Funktion $f \in \mathrm{FP}$ auch $f^{-1} \in_c \mathrm{FP}$, also eine (funktionale) Verfeinerung g von f^{-1} in FP, dann sagen wir auch, dass f P-*invertierbar* ist. Falls f injektiv ist, dann ist insbesondere f^{-1} funktional, und Ehrlichkeit von f ist ferner eine notwendige Bedingung für die P-Invertierbarkeit von f.

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

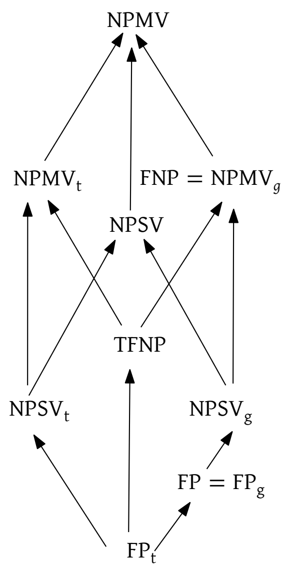

Abbildung 2: Inklusionen zwischen den in dieser Arbeit definierten Funktionenklassen. Eine Pfeil von $\mathcal{F}$ nach $\mathcal{G}$ sagt aus dass $\mathcal{G} \subseteq \mathcal{F}$.

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 als DisjNP.

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

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

Entsprechend 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 $A \subseteq S$ und $B \subseteq \overline{S}$.

Reduktionen

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

- $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ wenn A mittels einer Orakel-PTM mit Orakel B entschieden werden kann, bzw. $A \in \mathrm{P}^B$ (Turing- bzw. Cook-Reduzierbarkeit; siehe unten für einen präzisen Orakel-Begriff).
- $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (Many-one- bzw. Karp-Reduzierbarkeit).
- $A \leq_{1}^{\mathrm{p}} B$ wenn eine injektive Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (One-one-Reduzierbarkeit).
- $A \leq_{1,\mathrm{i}}^{\mathrm{p}} B$ 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 (vgl. Köbler und Messner, 2000; Beyersdorff, Köbler und Messner, 2009; Pudlák, 2017). Seien g, h zwei Multifunktionen:

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

Insbesondere für zwei Funktionen $g, h \in \mathrm{FP}$ gilt

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

Auf den Paaren aus DisjNP und DisjCoNP hat sich folgender Begriff von Reduzierbarkeit herausgebildet.[5] Seien $(A, B), (C, D)$ zwei disjunkte NP-Paare (bzw. zwei disjunkte coNP-Paare):

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

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

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

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

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 $A \in \mathcal{C}$). Geht die Klasse $\mathcal{C}$ aus dem Kontext hervor, lassen wir manchmal auch die Angabe „*für $\mathcal{C}$*“ weg. Auf Mengen betrachten wir üblicherweise die Reduktionsordnung $\leq_{\mathrm{m}}^{\mathrm{p}}$, und nennen daher manchmal eine Sprache A kurz *$\mathcal{C}$-vollständig* wenn A $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig für $\mathcal{C}$ ist. Auf Grundlage der Existenz universeller effizienter Turing-Maschinen können für die Klassen P und NP jeweils eine kanonische $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge angegeben werden. Für NP ist diese

Definition 2.2.

$$\mathtt{KAN} \stackrel{\mathrm{df}}{=} \{(N, x, 1^n) \mid N \text{ ist eine NTM und ex. Rechenweg } \alpha \text{ auf } N(x) \text{ mit} \leq n \text{ vielen Schritten}\}. \qquad \lhd$$

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

Polynomialzeit-Isomorphie

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

- $A \equiv_{\mathrm{m}}^{\mathrm{p}} B \stackrel{\mathrm{df}}{\iff} A \leq_{\mathrm{m}}^{\mathrm{p}} B$ und $B \leq_{\mathrm{m}}^{\mathrm{p}} A$.
- $A \equiv_{1}^{\mathrm{p}} B \stackrel{\mathrm{df}}{\iff} A \leq_{1}^{\mathrm{p}} B$ und $B \leq_{1}^{\mathrm{p}} A$.
- $A \equiv_{1,\mathrm{i}}^{\mathrm{p}} B \stackrel{\mathrm{df}}{\iff} A \leq_{1,i}^{\mathrm{p}} B$ und $B \leq_{1,i}^{\mathrm{p}} A$.

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

- $A \equiv^{\mathrm{p}} B \stackrel{\mathrm{df}}{\iff}$ 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:

Berman und Hartmanis (1977) zeigen dass $\equiv_{1,\mathrm{i}}^{\mathrm{p}}$-äquivalente Sprachen dann P-isomorph sind, wenn die jeweiligen Reduktionsfunktionen verlängernd sind.

Satz 2.4 (Berman und Hartmanis, 1977)**.** *Gilt $A \leq_{1,\mathrm{i}}^{\mathrm{p}} B$ via f und $B \leq_{1,\mathrm{i}}^{\mathrm{p}} A$ 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 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. $\lhd$

Paddability erlaubt es, Many-one-Reduktionen zu verlängernden Reduktionen zu verstärken:

Satz 2.6 (Berman und Hartmanis, 1977, Thm. 5, 7). *(1) Ist A paddable so gibt es für jedes B mit $B \leq_m^p A$ eine injektive* P*-invertierbare verlängernde Funktion $f \in$* FP*, welche $B \leq_{1,i}^p$ realisiert.*

(2) Sind A, B paddable, so folgt aus $A \equiv_m^p$ stets $A \equiv^p$.

Mit obigem Begriff von Paddability lässt sich die $\equiv^p$-Äquivalenzklasse von KAN insbesondere folgendermaßen charakterisieren:

Satz 2.7. *Eine Menge $A \in$* NP *ist genau dann* P*-isomorph zu* KAN *wenn $A \leq_m^p$-vollständig und paddable ist.*

Berman und Hartmanis machen die empirische Beobachtung, dass die ihnen bekannten bekannten $\leq_m^p$-vollständigen Mengen für NP alle paddable sind, also auch paarweise P-isomorph sind. Aufbauend auf dieser Beobachtung vermuteten Berman und Hartmanis, dass das für *alle* NP-vollständigen Mengen gilt.

Vermutung 2.8 (IC). *Alle $\leq_m^p$-vollständigen Mengen für* NP *sind* P*-isomorph. In anderen Worten: die $\equiv_m^p$-Äquivalenzklasse der vollständigen Mengen ist gleich der $\equiv^p$-Äquivalenzklasse von* KAN*.*

2.4 Orakel und Relativierungen

Wie in der Einleitung schon angesprochen, ist die Idee hinter Orakel-Berechnungen die Untersuchung, welche Probleme B effizient(er) durch einen Algorithmus gelöst werden können, wenn der Algorithmus eine (fiktive) Möglichkeit hat, ein (ggf. sehr schweres) Problem A ohne Rechenaufwand zu lösen. Der Zugriff auf A kann also wie ein „Nachschlagewerk“ oder „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 $q_?, q_{yes}, q_{no}$.

Gegeben ein Orakel $A \subseteq \Sigma^*$ können OTM nun Fragen der Form $x \overset{?}{\in} A$ 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 der 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 $\mathrm{time}_M^A(x)$ definieren, und von (deterministischen bzw. nichtdeterministischen) Polynomialzeit-Orakel-Turing-Maschinen (POTM, NPOTM) sprechen, wenn die Laufzeit auf allen Eingaben und für alle Orakel polynomiell durch die Eingabelänge beschränkt ist.

Wir können nun die relativierten Komplexitätsklassen $\mathrm{P}^O, \mathrm{NP}^O, \mathrm{FP}^O, \mathrm{NPMV}^O, \ldots$ relativ zu einem gegebenen Orakel O definieren, wobei in der jeweiligen Definition die

TM durch OTM ersetzt werden, welche 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. $A \leq_{\mathrm{m}}^{\mathrm{p},O} B$ wenn eine Funktion $f \in \mathrm{FP}^O$ existiert mit $x \in A \leftrightarrow f(x) \in B$. Die kanonische NP-vollständige Menge KAN kann ebenso zu $\mathtt{KAN}^O$ relativiert werden. Zur Vollständigkeit

$$\mathtt{KAN}^O \stackrel{\mathrm{df}}{=} \{(N, x, 1^n) \mid N \text{ ist eine NOTM und akz. } x \text{ relativ zu } O \\ \text{auf einem RW mit} \leq n \text{ vielen Schritten}\}.$$

Die Vollständigkeit relativiert insbesondere, heißt wir haben $A \leq_{\mathrm{m}}^{\mathrm{p},O} \mathtt{KAN}^O$ für alle Orakel O und Mengen $A \in \mathrm{NP}^O$. Für *natürliche* Mengen wie SAT usw. werden wir in dieser Arbeit dagegen keine relativierte Variante definieren. In diesem Sinne ist SAT im Allgemeinen *nicht* $\leq_{\mathrm{m}}^{\mathrm{p},O}$-vollständig, in dem Sinne dass ein Orakel O existiert und eine Menge $A \in \mathrm{NP}^O$ sodass $A \not\leq_{\mathrm{m}}^{\mathrm{p},O} \mathtt{SAT}$.[6]

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 haben wir z.B. schon bei der relativierenden $\leq_{\mathrm{m}}^{\mathrm{p}}$-Vollständigkeit von KAN gesehen. Auch das Diagonalargument in einem typischen Beweis des Hierarchiesatzes $\mathrm{P} \subsetneq \mathrm{E}$ relativiert beispielsweise. Das lässt sich intuitiv so erklären, dass bei diesem Argument Polynomialzeit-Turing-Maschinen durch eine Exponentialzeit-Turing-Maschine simuliert werden. Es ist aber klar, dass eine solche Simulation auch analog mit Orakeln funktioniert: NPOTMs relativ zu O lassen sich auch durch eine Exponentialzeit-Turing-Maschine relativ zu O simulieren. Damit sehen wir, dass auch auch $\mathrm{P}^O \subsetneq \mathrm{E}^O$ für jedes beliebige Orakel O gilt.

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.

6. Es ist durchaus möglich, SAT so zu verallgemeinern dass eine entsprechende Variante auch relativiert vollständig bleibt. Um das zu erreichen, ergänzt man üblicherweise die SAT-Formeln durch n-stellige Prädikate $X_n(x_1, x_2, \dots, x_n)$ und interpretiert diesen Term als wahr genau dann wenn $x_1 x_2 \cdots x_n$ im Orakel liegt. So zum Beispiel bei Dingel (2022).

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 $f \in \mathrm{FP}$ ist ein *Beweissystem für L* wenn $\mathrm{img}(f) = L$. Gilt $f(w) = x$, sagen wir auch, dass w ein *f-Beweis* für x ist.

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

Beachte dass „f hat kurze Beweise" nicht bedeutet, dass f *nur* kurze Beweise hat: im Allgemeinen ist f nicht ehrlich.

Cook und Reckhow stellen die Frage, welche Mengen Beweissysteme mit kurzen Beweisen haben. Ein einfaches Beweissystem für die Menge $\mathtt{SAT} \in \mathrm{NP}$ wäre das *Standardbeweissystem sat* für SAT:

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

Dieses Beweissystem hat kurze Beweise.

Cook und Reckhow machen dagegen die Beobachtung, dass TAUT genau dann ein Beweissystem mit kurzen Beweisen hat, wenn $\mathrm{NP} = \mathrm{coNP}$. Diese Einsicht motivierte das sogenannte *Cook–Reckhow-Programm* (vgl. Buss, 1996): man nähert sich der Frage $\mathrm{NP} \neq \mathrm{coNP}$ mittels Untersuchung immer stärkerer 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 effizient oder berechenbare) 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 sagen wir, dass f das Beweissystem g P-*simuliert.* ◁

Wenn klar ist, dass f und g Beweissysteme für die gleiche Menge sind, können wir abkürzend alternativ auch äquivalent $g \leq_{\mathrm{m}}^{\mathrm{p}} f$ schreiben, um zu sagen, dass f das Beweissystem g P-simuliert

Die Beweissysteme mit kurzen Beweisen sind unter Simulation abgeschlossen, und die P-Invertierbarkeit von Beweissystemen sind unter P-Simulation abgeschlossen:

Beobachtung 2.11. *Seien h, h' zwei Beweissysteme für L.*

(1) Wenn h kurze Beweise hat, und h' das Beweissystem h simulieren kann, dann hat auch h' kurze Beweise.

(2) Wenn h P-invertierbar ist, und h' das Beweissystem h P-simulieren kann, dann ist auch h' P-invertierbar.

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

Definition 2.12. Ein Beweissystem f für L ist *(P-)optimal* wenn es jedes Beweissystem 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 $f \in \mathrm{FP}^O$ sprechen, wenn für jedes Beweissystem $g \in \mathrm{FP}^O$ eine Funktion $h \in \mathrm{FP}^O$ existiert mit $f(h(w)) = g(w)$.

Jedes Beweissystem mit kurzen Beweisen ist auch optimal.

Beobachtung 2.13. *Sei $f \in \mathrm{FP}$ ein Beweissystem für L. Hat f kurze Beweise, dann ist f optimal.*

Beweis. Sei $g \in \mathrm{FP}$ ein Beweissystem für L. Wir zeigen, dass f das Beweissystem g simulieren kann. Nachdem f kurze Beweise hat, existiert auch eine (nicht notwendigerweise berechenbare) Funktion μ und ein Polynom q, sodass μ jedem $x \in L$ einen f-Beweis $\mu(x)$ mit $|\mu(x)| \leq q(|x|)$ zuweist. Damit gilt insbesondere $f(\mu(x)) = x$.

Sei nun $\pi(w) \stackrel{\mathrm{df}}{=} \mu(g(w))$. Zum einen gilt

$$f(\pi(w)) = f(\mu(g(w))) = g(w),$$

heißt g-Beweise können in f-Beweise umgeschrieben werden. Außerdem ist π polynomiell längenbeschränkt: es gilt

$$|\pi(w)| = |\mu(g(w))| \leq q(|g(w)|) \leq q(p(|w|)),$$

wobei p das Polynom ist, welches die Laufzeit von g beschränkt. □

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 diese Voraussetzung auf NEE = coNEE ab.

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

Beobachtung 2.14. *(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. Zu (1): Betrachte die Funktion

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

Diese Funktion ist definitiv ein Beweissystem in FP für A. Klar ist auch dass h kurze Beweise hat. Dieses Beweissystem ist P-optimal, denn wenn g ein Beweissystem für A ist, gilt $g(w) = h(g(w))$, also $g \leq^{\mathrm{p}}_{\mathrm{m}} h$.

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, welche A in polynomieller Laufzeit p entscheidet. Definiere nun

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

Nachdem $L(N) = A$ ist h definitiv ein Beweissystem für A. Es ist auch klar, dass $f \in$ FP. Nachdem akzeptierende Rechenwege höchstens polynomiell länger als x sind, hat h kurze Beweise. Also ist h nach Beobachtung 2.13 auch optimal. □

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

Beobachtung 2.15. *Sei $A \in \mathrm{NP}$ und f ein Beweissystem für A. Das Beweissystem f ist optimal genau dann wenn f kurze Beweise hat.*

Beweis. Richtung von rechts nach links klar, das gilt schon im allgemeinen Fall nach Beobachtung 2.13. 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 2.16 (Messner, 2000, Thm. 3.2). *Hat A ein* (P-)*optimales Beweissystem und* $B \leq_{\mathrm{m}}^{\mathrm{p}} A$*, dann hat auch B ein* (P-)*optimales Beweissystem.*

Der Beweis von Messner relativiert dabei insbesondere. In Kombination mit vollständigen Mengen erhalten wir hieraus folgendes Korollar:

Korollar 2.17. *Folgende Aussagen sind äquivalent:*

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

(2) Für jede Menge $B \in \mathrm{NP}$ *existiert eine* (P-)*optimales Beweissystem.*

Analoge Äquivalenzen gelten für coNP*.*

Das erklärt auch die Form, in der wir in der Einleitung die Hypothese SAT, TAUT gewählt haben. Ursprünglich sagte die Hypothese TAUT aus, dass kein P-optimales aussagenlogische Beweissystem existiert, also kein Beweissystem für die coNP-vollständige Menge TAUT existiert. Mit dem vorigen Korollar ist klar, dass diese Aussage im unrelativierten Fall äquivalent zu unserer hier gewählten Definition von TAUT ist: *Keine* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-vollständige Menge A für* NP *hat ein* P*-optimales Beweissystem*. Denn wenn ein P-optimales Beweissystem für TAUT existiert, dann auch für *eine* coNP-vollständige Menge. Für die andere Richtung, wenn für *eine* coNP-vollständige Menge ein P-optimales Beweissystem existiert, dann auch für alle Mengen in coNP, also insbesondere auch für TAUT.

Die hier gewählte Charakterisierung hat vor allem den Vorteil, dass SAT, TAUT, $\mathrm{TAUT}^{\mathrm{N}}$ auf natürliche Weise relativieren. So relativiert beispielsweise SAT auf Orakel O zur Aussage „kein $\leq_{\mathrm{m}}^{\mathrm{p},O}$-vollständiges $L \in \mathrm{NP}^O$ für 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 (2020a) vorgeschlagen wurde.

3 Zur Konzeptualisierung und Ordnung von Suchproblemen

In diesem Kapitel werden wir grundsätzlich überlegen, wie NP-Suchprobleme in der Komplexitätstheorie erfasst werden können, wie wir diese in ihrer Schwierigkeit untereinander vergleichen können und in welcher Beziehung sie zu den NP-Entscheidungsproblemen stehen. In Abschnitt 3.1 werden wir eine formal präzise Definition von NP-Suchproblemen erarbeiten, wie sie bereits in der Einleitung intuitiv vorgestellt wurden. Das umfasst auch die Unterklasse TFNP der totalen NP-Suchprobleme.

In Abschnitt 3.2 gehen wir auf die Beziehung zwischen NP-Suchproblemen und den entsprechenden NP-Entscheidungsproblemen ein. Insbesondere zeigen wir, in welchen Situationen das Entscheidungsproblem „gleich schwer" wie das Suchproblem ist (das ist das Argument *search reduces to decision*), und in welchen nicht.

Um die Schwierigkeit der unterschiedlichen NP-Suchproblemen zu vergleichen, werden wir – analog wie auf den Entscheidungsproblemen – ein Begriff der *Levin*-Reduzierbarkeit definieren. In Abschnitt 3.3 definieren wir diesen Reduzierbarkeits-Begriff präzise und betrachten Eigenschaften des entsprechenden Vollständigkeitsbegriffs. In diesem Zusammenhang betrachten wir auch die bekannteren verwandten *sparsamen* („parsimonious") Reduktionen.

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 $x \in \Sigma^*$, entweder eine entsprechende positive Lösungsinstanz $y \in \Sigma^*$ zu berechnen, oder negativ abzulehnen. Hier noch einmal das Beispiel (7) 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*. Zur Erinnerung: 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 der Länge von φ. Zum anderen kann effizient geprüft werden, ob y tatsächlich eine erfüllbare Belegung von φ ist.

Wir können die beiden obigen Punkte noch einmal in eine formale Definition gießen:

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

(1) in Polynomialzeit entscheidbar ist, d.h. $R \in \mathrm{P}$, bzw. genauer $\{\langle x,y\rangle \mid (x,y) \in R\} \in \mathrm{P}$, und

(2) polynomiell längenbeschränkt 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 *R-Zertifikate* (oder manchmal *R-Lösungen*). Wir sagen dann für $(x,y) \in R$, dass y ein R-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 allgemeine Suchproblem zu einer NP-Relation R kann jetzt wie folgt formal formuliert werden:

Suchproblem zur NP-Relation R:

Gegeben: Instanz $x \in \Sigma^*$.
Gesucht: Zertifikat $y \in \Sigma^*$ 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 mindestens 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 NP-Relation R lautet

Entscheidungsproblem zur NP-Relation R:

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

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

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 NP-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: wenn das Suchproblem lösbar ist, dann ist auch das Entscheidungsproblem lösbar.

Im Folgenden werden eine Beispiele von natürlichen NP-Relationen angegeben. Um diese von den sonst 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.

- `rLARGESTMATCHING` $\stackrel{\text{df}}{=} \{(G, M) \mid G$ ist ein Graph, M ein größtes Matching auf $G\}$. Das korrespondiert zum Suchproblem (3) aus der Einleitung.
- `rSAT` $\stackrel{\text{df}}{=} \{(\varphi, w) \mid \varphi$ ist eine aussagenlogische Formel, w erfüllende Belegung für $\varphi\}$. Das korrespondiert zum Suchproblem (7) aus der Einleitung.
- `rCLIQUE` $\stackrel{\text{df}}{=} \{((G, k), C) \mid G$ ist ein Graph, C eine Clique mit $\leq k$ Knoten$\}$. Das korrespondiert zu Suchproblem (6) aus der Einleitung.
- `rHAMCYCLE` $\stackrel{\text{df}}{=} \{(G, P) \mid G$ ist ein Graph, P ein Zyklus, welcher jeden Knoten von G genau einmal berührt$\}$.
- `rANOTHERHAMCYCLE` $\stackrel{\text{df}}{=} \{((G, P), P') \mid G$ ist ein Graph, P, P' je ein Zyklus der jeden Knoten genau einmal berührt, $P \neq P'\}$.
- `rFACTORIZATION` $\stackrel{\text{df}}{=} \{(n, (p_1, p_2, \dots, p_k)) \mid n \in \mathbb{N}, n > 1$, alle $p_i > 1$ Primzahlen, und $n \stackrel{\text{df}}{=} p_1 \cdots p_k\}$. Das korrespondiert zum Suchproblem (1) aus der Einleitung.
- `rFACTOR` $\stackrel{\text{df}}{=} \{(n, p) \mid n \in \mathbb{N}$, $n > 1$ ist nicht prim, und p ist ein Primfaktor von $n\}$.
- `rSMALLFACTOR` $\stackrel{\text{df}}{=} \{((n, a), p) \mid n \in \mathbb{N}$, $n > 1$ ist nicht prim, und $p \leq a$ ist ein Primfaktor von $n\}$.
- `rGI` $\stackrel{\text{df}}{=} \{((G, H), \sigma) \mid G, H$ sind Graphen mit gleicher Knotenmenge, und σ ist ein Graphisomorphismus von G nach $H\}$.

Jede dieser Relationen ist auch eine NP-Relation. Beachte, dass die Menge der Primzahlen in Polynomialzeit entscheidbar ist (Agrawal, Kayal und Saxena, 2004). Bei jeder der obigen natürlichen Relationen gilt, dass die Projektion auch der sonst üblichen Sprache aus NP zum Entscheidungsproblem entspricht. Wir haben z.B.

$$\mathrm{Proj}(\texttt{rCLIQUE}) = \{(G, k) \mid \text{ex. Clique } C \text{ von Graph } G \text{ mit} \leq k \text{ Knoten}\} \stackrel{\text{df}}{=} \texttt{CLIQUE}.$$

Die Definition von 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 $f \in$ 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 Multifunktionen mit binären Relationen identifiziert.

Mit dieser Perspektivierung lässt sich auch formal 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 $R \in_\mathrm{c}$ FP. Diese Aussage bedeutet ja, dass eine Verfeinerung f von R existiert, und f ist dabei eine (partielle) in Polynomialzeit berechenbare Funktion. Es existiert also ein deterministischer Polynomialzeit-Transduktor T, welcher f berechnet. Für eine Eingabeinstanz x wird also entweder $T(x)$ einen Wert y ausgeben für den $y = f(x) \in$ *set*-$R(x)$ gilt, bzw. in anderen Worten, eine R-Lösung y für x im Sinne der Definition 3.1. Oder, falls $T(x)$ ablehnt, dann ist $x \notin \mathrm{dom}(f) = \mathrm{Proj}(R)$, heißt „$f(x)$ lehnt ab“ bedeutet, dass x keine Lösung hat.

Unter dieser Definition von Lösrbarkeit ist klar, dass das NP-Suchproblem „schwieriger“ ist als das entsprechende NP-Entscheidungsproblem, in dem Sinne dass sich das NP-Entscheidungsproblem auf das NP-Suchproblem reduzieren lässt:

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

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

- $\mathtt{rLARGESTMATCHING} \in_c \mathrm{FP}$.
- $\mathrm{NP} = \mathrm{P} \iff \mathtt{rSAT} \in_c \mathrm{FP} \iff \mathtt{rCLIQUE} \in_c \mathrm{FP} \iff \mathtt{rHAMCYCLE} \in_c \mathrm{FP} \iff \mathtt{rANOTHERHAMCYCLE} \in_c \mathrm{FP}$.
- Unklar, ob $\mathtt{rSMALLFACTOR}$, $\mathtt{rFACTOR}$, $\mathtt{rFACTORIZATION} \overset{?}{\in}_c \mathrm{FP}$. Wir haben aber $\mathrm{UP} \cap \mathrm{coUP} = \mathrm{P} \implies \mathtt{rSMALLFACTOR} \in_c \mathrm{FP} \iff \mathtt{rFACTOR} \in_c \mathrm{FP} \iff \mathtt{rFACTORIZATION} \in_c \mathrm{FP}$.
- Unklar, ob $\mathtt{rGI} \overset{?}{\in}_c \mathrm{FP}$.

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

Totale NP-Suchprobleme

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

Das sind – entsprechend dieser Definition von FNP bzw. Konzeptualisierung von Suchproblemen – genau jene NP-Relationen, welche (links-)total sind: für jedes $x \in \Sigma^*$ existiert ein $y \in \Sigma^*$ mit $(x, y) \in R$. In anderen Worten, $\mathrm{Proj}(R) = \Sigma^*$. Die oben definierten NP-Relationen $\mathtt{rFACTORIZATION}$ und $\mathtt{rFACTOR}$ sind nicht total; nachdem die negativen Instanzen aber einfach zu entscheiden sind, können für beide NP-Relationen effektiv äquivalente Relationen angegeben werden, die total sind:

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

Herbei soll „*ungültig*“ ein String sein, welcher eine Lösung darstellt. 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 oder diskreter Logarithmus. Beachte, dass TFNP nicht identisch ist zur Klasse $\mathrm{NPMV_t}$: Die Klasse TFNP ist eine Teilmenge von $\mathrm{NPMV_t}$ jener totalen Multifunktionen $f \in \mathrm{NPMV_t}$, für die der (der Graph) f in P liegt. Man könnte also TFNP äquivalent als $(\mathrm{NPMV_t})_\mathrm{g}$ schreiben. Beachte, dass TFNP sogar eine echte Teilmenge von $\mathrm{NPMV_t}$ ist, außer $\mathrm{P} = \mathrm{NP}$:

Beobachtung 3.4 (vgl. Fenner, Fortnow, Naik und Rogers, 2003, Prop. 5). *Wenn für alle $f \in \mathrm{NPMV_t}$ auch (der Graph) $f \in \mathrm{P}$ ist, dann gilt $\mathrm{P} = \mathrm{NP}$.*

Wie in der Einleitung angesprochen, lässt sich die Aussage „alle Suchprobleme in TFNP sind effizient lösbar" auf einfache Weise äquivalent als Hypothese Q formulieren.

Beobachtung 3.5 (Fenner, Fortnow, Naik und Rogers, 2003). *Folgende Aussagen sind äquivalent:*

(1) Aussage Q*: für jede totale NPTM N (d.h. $L(N) = \Sigma^*$) existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ ein akzeptierender Rechenweg von $N(x)$ ist.*

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

Beweis. (1) $\Longrightarrow$ (2): Sei $R \in \mathrm{TFNP}$ mit Zertifikatsschranke p. Definiere die NPTM N die auf Eingabe x erst ein Zertifikat $y \in \Sigma^{\leq p(|x|)}$ rät, und genau dann akzeptiert wenn $(x, y) \in R$. Es ist klar, dass $L(N) = \mathrm{Proj}(R) = \Sigma^*$. Nach (1) existiert also $g \in \mathrm{FP}$ sodass $g(x)$ ein akzeptierender Rechenweg von $N(x)$ ist. Aus diesem Rechenweg lässt sich nun effizient das geratene Zertifikat extrahieren: sei $g'(x)$ definiert als der geratene Zeuge y. Dann ist $(x, g'(x)) \in R$ und damit g' eine Verfeinerung von R. Da $g' \in \mathrm{FP}$ ist $R \in_\mathrm{c} \mathrm{FP}$.

(2) $\Longrightarrow$ (1): Sei N eine NPTM mit $L(N) = \Sigma^*$. Klar ist, dass die Relation $R \stackrel{\mathrm{df}}{=} \{(x, \alpha) \mid N(x)$ akz. auf Rechenweg $\alpha\}$ eine totale NP-Relation ist. Mit (2) existiert also eine Verfeinerung $g \in \mathrm{FP}$ von R. Für $x \in \Sigma^*$ ist also $(x, g(x)) \in R$ und nach Definition akzeptiert also $N(x)$ auf Rechenweg $g(x)$, wie gewünscht. □

Aus der Beschäftigung mit TFNP-Problemen kam es ferner zu einer umfassenden Theoriebildung. So kam z.B. eine verfeinerte Betrachtung durch Aufmachen bestimmter Unterklassen von TFNP hinzu. Diese Unterklassen verinnerlichen hierbei jeweils das kombinatorische Prinzip, „warum" ein Suchproblem total ist (vgl. den Überblick von Goldberg und Papadimitriou, 2018). Exemplarisch seien hier zwei Unterklassen skizziert:

- Die Unterklasse PLS („polynomial local search") umfasst die Suchprobleme, welche in die Form eines Suchgraphen mit polynomiellen Grad gebracht werden können, worauf ein lokales Optimum gesucht ist. Das zugrunde liegende kombinatorische 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. Beachte außerdem, dass lokale Optimalität in Polynomialzeit überprüft werden kann, denn es muss nur getestet werden, ob einer der linear vielen möglichen Flips eines Knotens zu einer Verbesserung führt.

- 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. Diese zwei Teilmengen existieren immer 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 3.3 noch Reduktionen auf NP-Relationen definieren; dieser Reduktionsbegriff identisch zu dem sonst üblichen Reduktionsbegriff 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 geht mit einer Vereinfachung der Konzepte, Definitionen und Theorien einher, und gleichzeitig lassen sich viele NP-Suchproblem auf das entsprechende Entscheidungsproblem reduzieren. Gemeint ist damit: gegeben ein effizienter Algorithmus A, welcher das korrespondierende NP-Entscheidungsproblem löst, kann auch ein effizienter Algorithmus A' angegeben werden, welcher das NP-Suchproblem löst. Dieses Argument wird als *search reduces to decision* beschrieben. In diesen Fällen ist dann die Lösbarkeit der Entscheidungsvariante äquivalent zur Lösbarkeit der Suchvariante.

In diesem Abschnitt werden detailliert Suchprobleme und Entscheidungsprobleme gegenübergestellt 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.

Zunächst sei auf die Beziehungen zwischen NP-Relationen und NP-Sprachen hingewiesen. Tatsächlich haben wir bereits gesehen, dass wir über die Projektion jedem NP-Suchproblem bzw. NP-Relation ein korrespondierendes Entscheidungsproblem aus NP zuordnen konnten. Tatsächlich lässt sich diese Zuordnung auch umkehren: zu jeder Sprache bzw. Entscheidungsproblem $L \in \mathrm{NP}$ existiert eine NPTM N, welche L entscheidet. Diese induziert eine NP-Relation für L (akz. Rechenweg auf $N(x)$ ist Lösung für x). Das ist die übliche „Zertifikats-Charakterisierung" von NP aus den Lehrbüchern.

Beobachtung 3.6 (Zertifikats-Definition von NP)**.**

$$\mathrm{NP} = \{\mathrm{Proj}(R) \mid R \textit{ ist eine NP-Relation}\}.$$

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.

Auch UP lässt eine solche Zertifikats-Definition zu, in der UP mit der Menge der Projektionen von rechtseindeutigen NP-Relationen übereinstimmt.

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

Beweis. Die Richtung von links nach rechts ist klar, folgt ja aus der Lösbarkeit des Suchproblems die Lösbarkeit des Entscheidungsproblems (Beobachtung 3.2 mit Beobachtung 3.6).

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

$$A_R \overset{\mathrm{df}}{=} \{(x,z) \mid \exists y \in \Sigma^{\leq q(|x|)}, (x,y) \in R, z \sqsubseteq y\}.$$

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

$z \leftarrow \varepsilon$
solange $|z| \leq q(|x|)$ **tue** *(Invariante: wenn eine R-Lösung für x ex., dann existiert eine R-Lösung y mit z $\sqsubseteq$ y)*
    wenn $(x,z) \in R$ **dann**
        akzeptiere mit z
    sonst wenn $(x, z0) \in A_R$ **dann**
        $z \leftarrow z0$
    sonst wenn $(x, z1) \in A_R$ **dann**
        $z \leftarrow z1$
    sonst
        ablehnen
ablehnen

Es ist klar, dass der obige Algorithmus eine Funktion definiert, welche eine Verfeinerung von R ist. Unter Annahme $A_R \in \mathrm{P}$ 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_R auch Orakelfragen an ein (externes) Orakel A_R gestellt werden können, d.h. das Suchproblem von R kann *à la Cook* auf das Entscheidungsproblem von A_R reduziert werden. In anderen Worten, $R \in_{\mathrm{c}} \mathrm{FP}^{A_R}$. Das generalisiert sogar, wenn statt A_R ein beliebiges Orakel gewählt wird, welches $\leq_{\mathrm{m}}^{\mathrm{T}}$-vollständig für NP ist. Ist also $\mathrm{Proj}(R) \leq_{\mathrm{m}}^{\mathrm{T}}$-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 über ein Algorithmus A effizient lösbar, dann kann dieser Algorithmus so eingesetzt werden, dass auch das Suchproblem zu R effizient durch einen Algorithmus A' gelöst werden kann.

Korollar 3.8 (*Search reduces to decision* für die NP-Vollständigen). *Sei R eine NP-Relation, für die* $\mathrm{Proj}(R)$ *auch* $\leq_{\mathrm{m}}^{\mathrm{T}}$*-vollständig für* NP *ist. Dann gilt* $R \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(R)}$.

Damit bleibten insbesondere diejenigen Situationen offen, in denen R eine NP-Relation ist, aber $\mathrm{Proj}(R)$ nicht $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig für NP ist, also $\mathrm{Proj}(R)$ ein NP-Intermediate ist.

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

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

und

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

jeweils mit einer polynomiellen Anzahl an Orakelfragen.

Konkret ist das zum Beispiel der Fall bei der NP-Relation $\mathtt{rSMALLFACTOR}$. Zur Erinnerung, wir haben

$$\mathrm{Proj}(\mathtt{rSMALLFACTOR}) = \{(n,a) \mid n > 1 \text{ nicht prim, ex. Faktor } p > 1 \text{ von } n \text{ mit } p \leq a\}.$$

Wollen wir für gegebene n, a einen Faktor $p \leq a$ finden, können wir diesen mit binärer Suche un polynomiell vielen Orakelfragen an $\mathrm{Proj}(\mathtt{rSMALLFACTOR})$ finden. In anderen Worten, es gilt $\mathtt{rSMALLFACTOR} \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(\mathtt{rSMALLFACTOR})}$.

Mittels der Eindeutigkeit der Primfaktorzerlegun lässt sich im Übrigen zeigen, dass $\mathrm{Proj}(\mathtt{rSMALLFACTOR}) \in \mathrm{UP} \cap \mathrm{coUP}$. Damit ergibt sich auch der auf S. 26 angegebene Stand, dass ein Kollaps von $\mathrm{UP} \cap \mathrm{coUP}$ mit P zur Folge hätte, dass $\mathtt{rSMALLFACTOR} \in_{\mathrm{c}} \mathrm{FP}$.

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

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

Unter Annahme von $\mathrm{NP} \neq \mathrm{coNP}$ gilt $\mathrm{Proj}(\mathtt{rLINDIV}) \notin \mathrm{P}$ (Adleman und Manders, 1977); ob $\mathrm{Proj}(\mathtt{rLINDIV})$ auch $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig für NP 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 bzw. 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 meint $\varphi[x_1/0]$ die Formel, welche 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 daher $\mathtt{rSAT} \in \mathrm{FP}^{\mathrm{Proj}(\mathtt{rSAT})}$. (Beachte aber, dass $\mathrm{Proj}(\mathtt{rSAT}) = \mathtt{SAT}$ schon $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig ist. Damit folgt $\mathtt{rSAT} \in \mathrm{FP}^{\mathtt{SAT}}$ schon aus Korollar 3.8.)

Ein weiteres nichttriviales Beispiel wäre die NP-Relation $\mathtt{rGI}$. Zur Erinnerung: dieses Suchproblem sucht nach einem Graphisomorphismus zwischen zwei gegebenen Graphen. Deren Projektion ist mutmaßlich nicht $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig für NP. Gleichzeitig gilt

$\texttt{rGI} \in_c \mathrm{FP}^{\mathrm{Proj}(\texttt{rGI})}$: 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 solche nichttriviale Reduktion von $\texttt{rLINDIV}$ auf $\mathrm{Proj}(\mathrm{rLINDIV})$ möglich ist, scheint in der Literatur nicht untersucht zu sein.

Abschließend wollen wir noch drei theoretische Resultate diskutieren. Das erste charakterisiert diejenigen Sprachen, für welche sich das Suchproblem auf das Entscheidungsproblem reduzieren lässst. Das zweite und drite Resultat gibt hinreiche plausible Bedingungen an, unter denen eine NP-Relation R existiert, für welche sich das Suchproblem nicht auf das Entscheidungsproblem reduzieren lässt.

Zunächst folgende Definition:

Definition 3.9. (1) Eine deterministische OTM heißt *robust für A* falls $L(M^O) = A$ für alle Orakel O.

(2) Eine Menge A heißt *selbsthelfend* falls eine OTM M existiert, welche robust für A ist, und für die $\mathrm{time}_M^A(x)$ polynomiell in abh. von $|x|$ wächst, d.h. zusammen mit dem Orakel A ist M^A ist eine POTM. ◁

Balcázar fasst die Intuition hinter dieser Definition wie folgt zusammen: man will die Situation abbilden, dass ein Entscheidungsalgorithmus 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.

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.

Satz 3.10 (Balcázar, 1989)**.** *Sei $A \in \mathrm{NP}$. Folgende Aussagen sind äquivalent:*

(1) A ist selbsthelfend.

(2) Es existiert eine NP-Relation R sodass $\mathrm{Proj}(R) = A$ und $R \in_c \mathrm{FP}^A$.

Andererseits existieren unter geeigneten Bedingungen eine NP-Relation R, für welche *search reduces to decision* fehlschlägt.

Satz 3.11 (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 R mit $\mathrm{Proj}(R) \in \mathrm{NP} - \mathrm{P}$, für die $R \notin_c \mathrm{FP}^{\mathrm{Proj}(R)}$ gilt.*

Unter stärkeren Bedingungen lässt sich 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 ein „Zertifikatssystem“ für L aussieht, ist keins so einfach dass Zertifikate mit Hilfe eines Orakels für das Suchproblem L gefunden werden können.

Satz 3.12 (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 $R \notin_c \mathrm{FP}^L$ für jede NP-Relation R für L, d.h. für die $\mathrm{Proj}(R) = L$ gilt. (Bzw. ist L nicht selbsthelfend.)*

Beachte, dass für jede dieser Relationen R aus den beiden vorigen Sätzen die entsprechende Projektion $\mathrm{Proj}(R)$ ein NP-Intermediate ist; $\mathrm{Proj}(R)$ kann nicht $\leq_m^p$-vollständig für NP sein, denn das wäre ein Widerspruch zu Korollar 3.8.

Selbstreduzierbarkeit in TFNP

Für *totale* Suchprobleme, also genau jene aus TFNP, kann nicht sinnvoll gefragt werden, ob hier das Suchproblem auf das Entscheidungsproblem reduziert werden kann, ist ja für $R \in \text{TFNP}$ das entsprechende Entscheidungsproblem $\text{Proj}(R) = \Sigma^*$ 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 schlagen Harsha, Mitropolsky und Rosen folgenden Begriff der Selbstreduzierbarkeit vor:

Betrachte hierbei zunächst 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 Sinn dass die PTM Orakelfragen der Form „*gib mir ein Zertifikat y für x′*“ stellen kann. Das Orakel antwortet dann mit einem solchen Zertifikat 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 Einschränkung 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 einen solchen Transduktor relativ zu R gelöst werden kann, sagen wir, dass R *nach unten selbstreduzierbar ist*.

Zum Verständnis erinnern wir uns an die TFNP-Relation $\texttt{rFACTOR}'$, welche nach einem nichttrivialem Faktor für n sucht, oder „n ist prim“ ausgibt. Wäre nun $\texttt{rFACTOR}'$ nach unten selbstreduzierbar, dann würde das bedeuten, dass ein Faktor von zusammengesetztem $n \in \mathbb{N}$ 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 erst durch Harsha, Mitropolsky und Rosen (2023) angetreten. Sie zeigen die Selbstreduzierbarkeit nach unten für das TFNP-Problem „Iterate with source“, welches als ein kanonischer Repräsentant für die Unterklasse PLS (zur Erinnerung: *polynomial local search*) gilt.

Iterate with Source:

Gegeben: $n \in \mathbb{N}$, und ein Schaltkreis $S\colon \Sigma^n \to \Sigma^n$ polynomieller Größe abhängig von n, und ein $u \in \Sigma^n$ mit $u < S(u)$.

Gesucht: Knoten $v \in \Sigma^n$ sodass $v < S(v) \not< S(S(v))$ gilt.

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

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

Es ist offen ob $\texttt{rFACTOR}' \stackrel{?}{\in} \text{PLS}$ und zumindest unplausibel, weil unklar ist wie Faktorisierung als Suche nach einem lokalen Optimum repräsentiert werden kann. Tatsächlich zeigen die Autorinnen sogar die stärkere Konsequenz $\texttt{rFACTOR}' \in \text{UEOPL}$, was auch noch $\texttt{rFACTOR}' \in \mathrm{PPAD}$ zur Folge hätte. Auch die Frage, ob $\texttt{rFACTOR}' \stackrel{?}{\in} \mathrm{PPAD}$ gilt, ist offen und wurde breit untersucht. Eine positive Antwort wäre daher 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 für Suchprobleme, sowohl TFNP-Probleme als auch FNP-Probleme, sehr klein. Harsha, Mitropolsky und Rosen stellen fest: „Almost all the study of downward self-reducibility, to date, has been focused in the decisional landscape" (2023). Es bedarf auf jeden Fall weiterer Untersuchungen, unter anderem auch im Richtung einer Konzeptualisierung von „kleinerer Instanz", die robuster als „kürzerer String" ist. Hier könnte eine Konzeptualisierung wie bei der *disjunktiver Selbstreduzierbarkeit* auf Entscheidungsproblemen (vgl. Meyer und Paterson, 1979; Balcázar, 1989; Selman, 1988; Wechsung, 2000, Abschn. 9.5) produktiv gemacht werden, welche die Ordnungsrelation „kürzer" über eine beliebige polynomiell wohlfundierte und längenbeschränkte Halbordnung auf den Wörtern verallgemeinert.

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 als Analog zur Many-one-Reduktion herausgebildet (vgl. Papadimitriou, 1994, S. 229; Goldreich, 2008, S. 61; Arora und Barak, 2009, S. 50):

Definition 3.14 (Levin-Reduzierbarkeit[7]). Seien Q, R zwei NP-Relationen. Wir sagen dass sich Q *auf* R *(Polynomialzeit-)Levin-reduzieren lässt*, bzw. $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$, wenn zwei Funktionen $f : \Sigma^* \to \Sigma^*, g : \Sigma^* \times \Sigma^* \to \Sigma^*, f, g \in \mathrm{FP}$ existieren sodass

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

(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 R-Zertifikat für die Instanz $f(x)$ ist, dann lässt sich aus y wieder ein Q-Zertifikat $g(x, y)$ für die originale Instanz x berechnen.

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

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

$$Q \leq_{\mathrm{L},1,\mathrm{i}}^{\mathrm{p}} R \implies Q \leq_{\mathrm{L},1}^{\mathrm{p}} R \implies Q \leq_{\mathrm{L}}^{\mathrm{p}} R. \qquad \triangleleft$$

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

Beobachtung 3.15. *Seien Q und R zwei NP-Relationen. Wenn $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ dann gilt $\mathrm{Proj}(Q) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(R)$.*

Die Relationen $\leq_{\mathrm{L}}^{\mathrm{p}}$, $\leq_{\mathrm{L},1}^{\mathrm{p}}$ und $\leq_{\mathrm{L},1,\mathrm{i}}^{\mathrm{p}}$ sind reflexiv und transitiv, bilden also eine Quasiordnung. Genau so wie wir es bei der üblichen $\leq_{\mathrm{m}}^{\mathrm{p}}$-Reduktion auf den Suchproblemen gewohnt sind, ordnet $\leq_{\mathrm{L}}^{\mathrm{p}}$ die Suchprobleme der NP-Relationen nach ihrer „Schwierigkeit": wenn $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ 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 einzigen

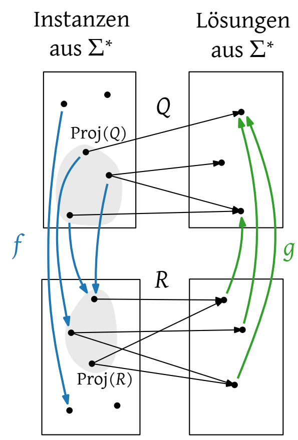

Abbildung 3: Schematische Skizze einer Levin-Reduktion $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ über Reduktionsfunktion f und Translationsfunktion g. Beachte, wie f Instanzen ohne Lösungen in R zu Instanzen ohne Lösungen in Q reduziert.

7. Die Bezeichnung *Levin*-Reduktion ist hier in Anlehnung 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 3.14) 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.

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 $\leq_{\mathrm{L}}^{\mathrm{p}}$-Ordnung:

Lemma 3.16. *Seien Q und R zwei NP-Relationen. Wenn $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ und $R \in_{\mathrm{c}}$ FP dann ist ist $Q \in_{\mathrm{c}}$ FP.*

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

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

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

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

Hierbei folgt die erste Äquivalenz nach Definition 3.14(1). Zum anderen haben wir

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

wobei dritte Implikation nach Definition 3.14(2) folgt. Also ist $q(x)$ ein R-Zertifikat für x, wie gewünscht. □

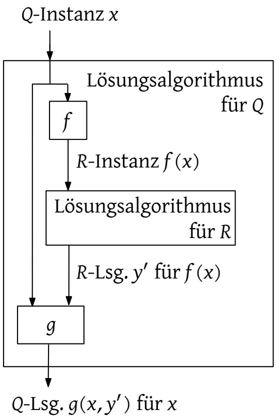

Abbildung 4: Skizze, wie eine Levin-Reduktion $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ von NP-Relation Q auf NP-Relation R genutzt werden kann, einen Algorithmus für Q mithilfe eines Algorithmus für R anzugeben. Hierbei ist f die Reduktions- und g die Translationsfunktion.

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

Definition 3.17. Sei $\mathcal{F}$ eine Klasse von Multifunktionen (z.B. FNP oder TFNP), und sei $R \in \mathcal{F}$. Wir nennen $R \in \mathcal{F}$ *$\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig für $\mathcal{F}$* wenn R ein größtes Element von $\mathcal{F}$ geordnet über $\leq_{\mathrm{L}}^{\mathrm{p}}$ ist: Für alle $Q \in \mathcal{F}$ gilt $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$.

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

Wie schon bei $\leq_{\mathrm{m}}^{\mathrm{p}}$-Reduktionen geschehen, lassen wir die Angabe der Klasse $\mathcal{F}$ gelegentlich auch weg, wenn $\mathcal{F}$ klar aus dem Kontext hervorgeht.

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

Definition 3.18.

$$\begin{aligned} \mathtt{rKAN} \overset{\mathrm{df}}{=} \{((N, x, 1^n), \alpha) \mid\ & N \text{ ist NPTM}, \\ & \alpha \text{ ist ein akz. Rechenweg auf } N(x) \text{ mit} \leq n \text{ Schritten}\}. \end{aligned}$$ ◁

Beachte dass $\mathrm{Proj}(\mathtt{rKAN}) = \mathtt{KAN}$. Über die Identifikation von NP-Relationen mit NPTMs (Beobachtung 3.6) ist klar, dass rKAN vollständig ist.

Satz 3.19. *Die kanonische NP-Relation* rKAN *ist $\leq_{\mathrm{L},1,\mathrm{i}}^{\mathrm{p}}$-vollständig für* FNP.

Die Ordnung $\leq_{\mathrm{L}}^{\mathrm{p}}$ und dessen Vollständigkeitsbegriff verhält sich auch sonst wie bei dem Analog $\leq_{\mathrm{m}}^{\mathrm{p}}$ gewohnt.

Lemma 3.20. *Sei R eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation für* FNP. *Es gelten folgende Aussagen:*

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

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

(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 $\leq_\mathrm{L}^\mathrm{p}$-vollständig für FNP sind. Das Bekannteste ist $\mathtt{rSAT}$. Zur Erinnerung:

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

Die Aussage „$\mathtt{rSAT}$ ist $\leq_\mathrm{L}^\mathrm{p}$-vollständig“ entspricht dem üblichen Cook–Levin-Satz.

Satz 3.21 (Satz von Cook und Levin). *Die NP-Relation* $\mathtt{rSAT}$ *ist* $\leq_{\mathrm{L},1,\mathrm{i}}^\mathrm{p}$*-vollständig für* FNP*.*

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

Vermutung 3.22 (TFNP). *Es existiert keine NP-Relation* $R \in \mathrm{TFNP}$ *die* $\leq_\mathrm{L}^\mathrm{p}$*-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 Kapitel 4 erarbeitet.

Bevor wir uns der Beziehung zwischen Karp- und Levin-Reduktion nähern, sei an dieser Stelle noch kurz hingeweisen, dass sich die Levin-Reduzierbarkeit auf natürliche Weise abschwächen lassen kann, so wie Many-one- auf Turing-Reduzierbarkeit. Intuitiv ist mit einer solchen Turing-Reduktion von NP-Suchproblem Q auf R gemeint, dass sich Q effizient lösen lassen kann, wenn ein effizienter Algorithmus für eine Verfeinerung von R existiert. Das lässt sich zum Beispiel formal über die Idee der funktionalen Orakel umsetzen, ähnlich wie bei der Selbstreduzierbarkeit nach unten (S. 33). Insbesondere dürfen hierbei mehrere Orakelfragen gestellt werden. (Eine Beschränkung auf höchstens eine Orakelfrage wäre äquivalent zur Levin-Reduzierbarkeit.)

Klar ist, dass diese Turing-Reduzierbarkeit auf Suchproblemen die Lösbarkeit nach unten überträgt. Zum Beispiel lässt sich $\mathtt{rFACTORIZATION}$ auf $\mathtt{rFACTOR}$ Turing-reduzieren: gegeben n, frage nach einem Primfaktor p_1 von n, frage dann nach einem Primfaktor p_2 von n/p_1, usw. Damit folgt $\mathtt{rFACTOR} \in_\mathrm{c} \mathrm{FP} \implies \mathtt{rFACTORIZATION} \in_\mathrm{c} \mathrm{FP}$, womit dann auch die angegebenen Äquivalenzen auf S. 26 gelten.

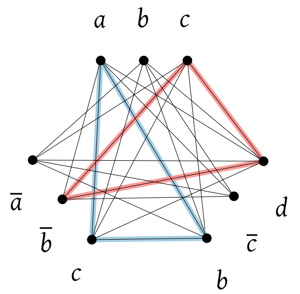

Abbildung 5: Schema der Reduktion von $\mathtt{SAT}$ auf $\mathtt{CLIQUE}$, hier für die CNF-Formel $\varphi = (a \vee b \vee c) \wedge (\bar{a} \vee \bar{b} \vee c) \wedge (b \vee \bar{c} \vee d)$. Diese wird auf eine $\mathtt{CLIQUE}$-Instanz $f(\varphi) = (G, 3)$ reduziert. Die Knoten von G entsprechen hierbei den Literalen der Klauseln, und genau dann inzident wenn diese in unterschiedlichen Klauseln sind, außer x und $\bar{x}$. Damit hat G eine Clique C der Größe 3 genau dann wenn φ erfüllbar ist. Aus der Konstruktion wird klar, dass sich aus jeder solchen Clique C auch eine erfüllende Belegung für φ bestimmen lassen kann, je nach dem welche Knoten bzw. Klauseln in C enthalten sind.

Karp- vs. Levin-Reduktionen

Die bekannten NP-Relationen R mit $\leq_\mathrm{m}^\mathrm{p}$-vollständiger Projektion $\mathrm{Proj}(R) \in \mathrm{NP}$ sind auch Levin-vollständig. Typische Präsentationen von $\leq_\mathrm{m}^\mathrm{p}$-Vollständigkeit erfolgten über die $\leq_\mathrm{m}^\mathrm{p}$-Reduktion von einer bereits bekannten NP-vollständige Menge, z.B. $\mathtt{SAT} \leq_\mathrm{m}^\mathrm{p} \mathtt{CLIQUE}$. Diese Beweise geben uns üblicherweise nicht nur eine $\leq_\mathrm{m}^\mathrm{p}$-Reduktionsfunktion f von Instanzen x der einen Menge zu Instanzen $f(x)$ der anderen Menge, 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 Zertifikate für $f(x)$ nach Zertifikaten für x reichen dann aus, um eine $\leq_\mathrm{L}^\mathrm{p}$-Reduktion zu realisieren, und damit Levin-Vollständigkeit zu zeigen.

Beispielsweise gilt im Lehrbuch-Beweis von $\mathtt{SAT} \leq_\mathrm{m}^\mathrm{p} \mathtt{CLIQUE}$ (siehe Abbildung 5), dass einerseits jede erfüllende Belegung w der (CNF-)Formel φ einer Clique C im Graph der Instanz $f(\varphi)$ entspricht, und umgekehrt lässt sich an jeder Clique C eine erfüllende Belegung w für φ trivial ablesen. Damit lässt sich leicht eine entsprechende Translationsfunktion g angeben, sodass f, g die Levin-Reduktion $\mathtt{rSAT} \leq_\mathrm{L}^\mathrm{p} \mathtt{rCLIQUE}$ auf den entsprechenden NP-Relationen realisieren.

Nach Goldreich (2008, S. 104) sind die folgenden NP-Relationen jedenfalls definitiv $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{i}}$-vollständig: $\mathtt{rSAT}$, $\mathtt{rSETCOVER}$, $\mathtt{rCLIQUE}$, $\mathtt{rVERTEXCOVER}$, $\mathtt{r3COLORABILITY}$. Die von Papadimitriou (1994, S. 193–198) angegebene Reduktion $\mathtt{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} \mathtt{HAMCYCLE}$ lässt sich leicht zu $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} \mathtt{rHAMCYCLE}$ erweitern, womit $\mathtt{rHAMCYCLE}$ auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Ebenso ist $\mathtt{rANOTHERHAMCYCLE}$ auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig (1994, S. 232). Damit gilt mit Lemma 3.20(3) auch die auf S. 25 angegebene Äquivalenz „$\mathrm{NP} = \mathrm{P}$ genau dann wenn $\mathtt{rSAT}, \mathtt{rCLIQUE}, \ldots \in_{\mathrm{c}} \mathrm{FP}$“.

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

Frage 3.23. *Sei R eine NP-Relation. 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 wollen wir die pessimistische negative Antwort auf diese Frage als Vermutung formulieren:

Vermutung 3.24 (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 $\Rightarrow \mathrm{P} \neq \mathrm{NP}$. In Abschnitt 4.1 werden wir diese Hypothese und dessen Beziehung zu anderen Hypothesen erarbeiten.

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 (d.h. $\mathrm{Proj}(R_L) = L$) sodass R_L $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist? In anderen Worten, existiert ein hinreichend ausdrucksstarkes „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 3.25 (Buhrman, Kadin und Thierauf, 1998)**.** *Für jede Menge $L \in$* NP *die* P-*isomorph zu* $\mathtt{SAT}$ *ist, existiert eine NP-Relation R_L sodass* $\mathrm{Proj}(R_L) = L$ *und R_L auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig für* FNP *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

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

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 $\mathtt{rSAT}$ ja $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist, existieren Reduktions- und Translationsfunktionen f, g die $Q \leq^{\mathrm{p}}_{\mathrm{L}} \mathtt{rSAT}$ realisieren. Definiere nun

$$f'(x) \stackrel{\mathrm{df}}{=} h^{-1}(f(x)).$$

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

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

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

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

Damit haben (im unrelativierten Fall) insbesondere alle *bekannten* NP-vollständigen Mengen, d.h. die zu $\mathtt{SAT}$ P-isomorphen Mengen, eine entsprechende $\leq_{\mathrm{L}}^{\mathrm{p}}$-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 zur Interpretierbarkeit gegenüber der Instanz x.

Sparsame Reduktionen

Wir wollen hier noch auf den Begriff der *sparsamen („parsimonious“)* Reduktionen eingehen. Diese ist einerseits verwandt mit der oben betrachteten Levin-Reduktion, andererseits aber wesentlich breiter bekannt und untersucht, insbesondere aufgrund der Relevanz von sparsamen Reduktion in der Komplexitätstheorie des Zählens.

Aus den Erfahrungen in der Entwicklung der NP-Vollständigkeit von Entscheidungsproblmen wurde schnell deutlich, dass die NP-vollständigen Mengen sich viele nicht-offensichtliche Eigenschaften teilen. In stärkster Ausprägung ist das z.B. die (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. Diese Forschung betrachtete aber insbesondere primär die NP-Entscheidungsprobleme, und nicht NP-Suchprobleme. Nichtsdestotrotz wurde intuitiv aber die Beobachtung gemacht, dass auch viele der entsprechenden *Suchprobleme* eine inhärente strukturelle Ähnlichkeit untereinander haben, auch wenn diese oft nicht explizit mitgedacht wurden (vgl. auch die Diskussion von Hemaspaandra, 1998).

Hieraus entwickelte sich u.a. der Begriff der sparsamen Reduktionen. 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 unausgesprochen mit.

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

Definition 3.26 (Sparsame Reduktionen)**.** Seien Q, R NP-Relationen. Wir sagen dass sich Q auf R (in Polynomialzeit) *sparsam* (*„parsimonious“*) reduzieren lässt, bzw. $Q \leq_{\mathrm{pars}}^{\mathrm{p}} R$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit

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

Beachte, dass sparsame Reduktionen eine Many-one-Reduktion realisieren: wir haben

$$x \in \mathrm{Proj}(Q) \iff |set\text{-}Q(x)| > 0 \iff |set\text{-}R(f(x))| > 0 \iff f(x) \in \mathrm{Proj}(R).$$

Sparsame 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 wie vielen Rechenwegen akzeptiert die Berechnung $N(x)$ der NPTM N?*“. Es ist einfach zu sehen, dass sich sparsame Reduktionen $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{pars}} \mathtt{rKAN}$ und $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{pars}} \mathtt{rSAT}$ angeben lassen können. Damit kann das Zählproblem zur $\mathtt{rSAT}$-Instanz x als $\mathtt{rKAN}$-Instanz $f(x)$ repräsentiert werden und umgekehrt – die beiden Zählprobleme sind relativ zum jeweils anderem gleich schwer. Tatsächlich sind $\mathtt{rSAT}$ und $\mathtt{rKAN}$ sogar $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig für FNP. Nach Goldreich lassen sich auch die üblichen Reduktionen der natürlichen NP-vollständigen Suchprobleme als sparsame Reduktion umformulieren; entsprechend sind alle bekannten NP-vollständigen Suchprobleme auch $\leq^{\mathrm{p}}_{\mathrm{pars}}$-vollständig Goldreich (2008, S. 204). Auf eine weitere Präsentation der Komplexitätstheorie des Zählens muss hier aber verzichtet werden (siehe Wechsung, 2000, Kap. 7; Arora und Barak, 2009, Chap. 17).

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

An dieser Stelle sei noch auf zwei weitere verwandte Arbeiten verwiesen, die hier nur kurz skizziert werden sollen: zum einen gehen Fischer, Hemaspaandra und Torenvliet (1995) noch einen Schritt über sparsame Reduktionen hinaus, und erforschen „*witness-isomorphic reductions*“ zwischen NP-Relationen. Hier erhält die Reduktionsfunktion von den A-Instanzen auf die B-Instanzen nicht nur die *Anzahl* der Zertifikate, sondern es werden zusätzlich die A-Zertifikate für $x \in \mathrm{Proj}(A)$ mit den B-Zertifikaten für $f(x) \in \mathrm{Proj}(B)$ in eine effiziente in Polynomialzeit berechenbare und invertierbare Eins-zu-Eins-Korrespondenz gesetzt. Damit sind sind *witness-isomorphic reductions* insbesondere eine Verstärkung von Levin-Reduktionen und sparsamen Reduktionen.

Zum anderen geben Agrawal und Biswas (1992a) strukturelle Kriterien an, die zur $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit von NP-Relationen R ausreichen. Diese Kritieren lauten intuitiv, dass sich bezogen auf das Suchproblem R gewisse „Gadgets“ mit geeigneten Eigenschaften konstruieren lassen können, wie wir sie auch aus einigen NP-Vollständigkeitsbeweisen kennnen. Agrawal und Biswas nennen dann R *universell*. Rein aus diesen *strukturellen* Eigenschaften lässt sich dann nachweisen, dass Universalität von R hinreichend für $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit von R ist. Agrawal und Biswas (1992a) überprüfen insbesondere die Universalität den ihnen bekannten vollständigen Entscheidungsproblemen, und können so in einer uniformen Weise die $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit vieler natürlicher NP-Suchbrobleme ableiten.

Nun wollen wir abschließend noch die einzelnen Vollständigkeitsbegriffe in Beziehung setzen. Zusammen mit Beobachtung 3.25 sehen wir bereits die eingezeichneten Implikationen aus Abbildung 6.

Nun zu den eingezeichneten Trennungen: zunächst halten wir erstens fest, dass die $\leq^{\mathrm{p}}_{\mathrm{L}}$-Vollständigkeit eine Eigenschaft ist, welche sogar bezüglich Problemen gilt, die mutmaßlich nicht P-isomorph sind. Angenommen, es existiert eine Einwegfunktion $f \in \mathrm{FP}$,

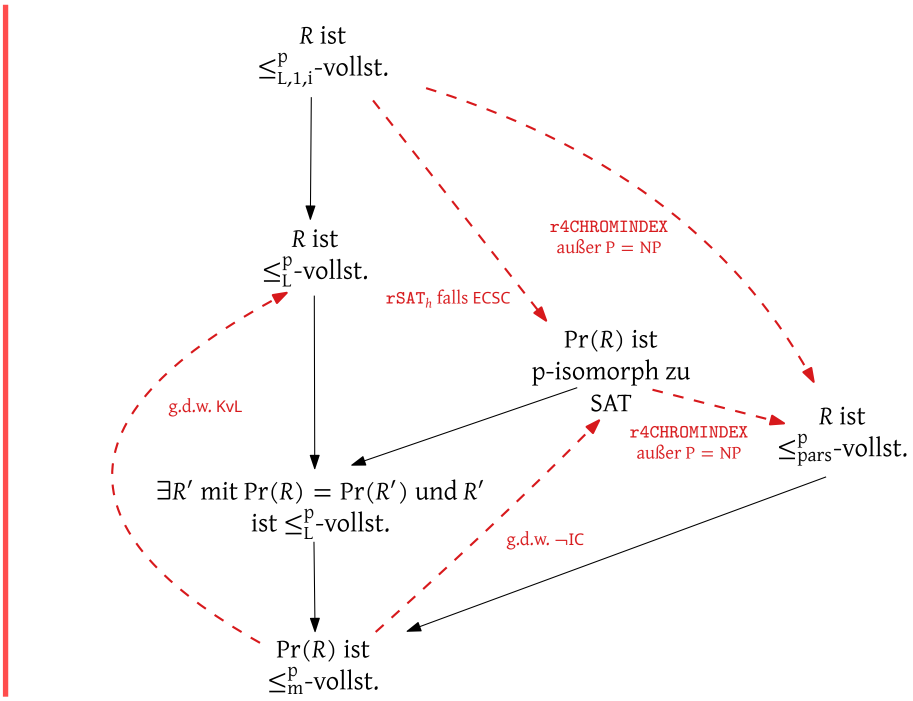

Abbildung 6: Implikationen zwischen den Vollständigkeitsbegriffen, wobei R eine beliebige aber feste NP-Relation ist. Ein unterbrochener Pfeile von A nach B sagt aus, dass ein Gegenbeispiel Q für die Implikation A $\Rightarrow$ B existiert, also eine NP-Relation Q die A erfüllt und gleichzeitig $\neg$B erfüllt.

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

$$f(\mathtt{SAT}) \stackrel{\text{df}}{=} \{f(\varphi) \mid \varphi \in \mathtt{SAT}\} \in \mathrm{NP}$$

nicht paddable ist, damit also auch nicht P-isomorph zu $\mathtt{SAT}$ ist. Gleichzeitig ist $\mathtt{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} f(\mathtt{SAT})$ über Reduktionsfunktion f, und damit $f(\mathtt{SAT})$ auch $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig für NP. Damit ist $f(\mathtt{SAT})$, zu verstehen als eine „verschlüsselte" Variante zu $\mathtt{SAT}$; ein vermutetes Gegenbeispiel für die Berman–Hartmanis-Isomorphievermutung IC. Gleichzeitig ist leicht zu sehen, dass eine entsprechende natürliche NP-Relation

$$\mathtt{rSAT}_f \stackrel{\text{df}}{=} \{(z, (\varphi, w)) \mid z = f(\varphi), \text{und } w \text{ ist erfüllende Belegung für } \varphi\}$$

$\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Wir haben $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} \mathtt{rSAT}_f$ über Reduktionsfunktion f und über Translationsfunktion $g(\varphi, (\varphi, w)) = w$.

Nun werden wir uns auf die sparsamen Reduktionen konzentrieren. Die Suche nach einem maximalen Schnitt ist ein triviales Beispiel einer $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständigen NP-Relation, welche nicht unter sparsamen Reduktionen vollständig ist.

Zu einem Graphen G mit Knotenmenge $\{0, 1, \dots, n-1\}$ können wir einen *Schnitt* als einen String $w \in \Sigma^n$ schreiben, wobei $V_0 \stackrel{\text{df}}{=} \{i \mid i < n, w[i] = 0\}$ und $V_1 \stackrel{\text{df}}{=} \{i \mid i < n, w[i] = 0\}$ 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

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

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 $\mathtt{rSAT} \not\leq^{\mathrm{p}}_{\mathrm{pars}} \mathtt{rMAXCUT}$. Angenommen es existiert eine solche sparsame Reduktion f. Beachte dass die SAT-Instanz $\varphi =$ „x_1" genau eine erfüllende

Belegung hat. Dann wäre

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

Es lässt sich aber leicht sehen, dass $|set\text{-}\mathtt{rMAXCUT}(x)|$ 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.

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:

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

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.

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

Zu einem Graphen G mit Kantenmenge $\{0, 1, \dots, m-1\}$ 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 *als Partitionierung* zählen, und sind dabei insbesondere nicht an redundanten Lösungen interessiert, die aus reiner Permutation der Farben entsteht. Ähnlich zu rMAXCUT′ setzen wir 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

$$\begin{aligned}\mathtt{r4CHROMINDEX} \overset{\text{df}}{=} \{((G,k),w) \mid\ & G \text{ ist Graph mit Kantenmenge } \{0,1,\dots,m-1\}\\ & G \text{ hat maximalem Grad } 4,\\ & \text{und } w \in \{1,2,3,4\}^m \text{ ist gültige Färbung mit 4 Farben}\}.\end{aligned}$$

8. Dieses Beispiel geht auf ein unpubliziertes Preprint von Edward und Welsh mit dem Titel „On the Complexity of Uniqueness Problems" welches offenbar in den 1980ern zirkuliert ist; viele der Arbeiten aus diesem Abschnitt nehmen auf genau dieses Preprint Bezug. Tatsächlich ist überliefert, dass die Autoren über die Kantenfärbbarkeit sogar ein „Gegenbeispiel" zur Berman–Hartmanis-Isomorphievermutung gefunden hätte. Hierbei gingen Edward und Welsh aber von einer wesentlichen stärkeren abweichenden Interpretation der Isomorphievermutung aus: neben der Isomorphie zwischen allen NP-vollständigen Entscheidungsproblemen würde ihre Interpretation der Isomorphievermutung auch eine Isomorphie auf den jeweiligen Zertifikatsmengen umfassen. Das würde (mindestens) eine sparsame Interreduzierbarkeit zwischen allen NP-vollständigen Suchproblemen implizieren. Diese Aussage ist nun aber so stark, dass diese durch eben das Beispiel der Kantenfärbbarkeit widerlegt werden kann.

Vgl. Hemaspaandra (1998), Fischer, Hemaspaandra und Torenvliet (1995), Cai und Govorov (2021) und Welsh (1993, S. 118).

Satz 3.27 (Cai und Govorov, 2021 nach Edward und Welsh[8]). *Die NP-Relation* r4CHROMINDEX *ist nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig, außer* $\mathrm{P} = \mathrm{NP}$.

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 4-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 4-Kantenfärbung.
- Unter den Graphen mit $\chi'(G) = 2$ ist $K_{1,2}$ der einzige Graph mit eindeutiger 4-Kantenfärbung.
- Unter den Graphen mit $\chi'(G) = 1$ ist $K_{1,1}$ der einzige Graph mit eindeutiger 4-Kantenfärbung.

In allen Fällen können isolierte Knoten ignoriert werden. Sei hier nur der Beweis für den Fall $\chi'(G) = 3$ skizziert. Sei hierfür G ein solcher Graph, dann existiert also mindestens eine Kantenfärbung C von G mit drei Farben. Sei C_i die Teilmenge der Kanten in Farbe i. Wir haben ohne Beschränkung also $C_1, C_2, C_3 \neq \emptyset, C_4 = \emptyset$. In je C_1, C_2, C_3 ist dann auch nur genau eine Kante enthalten, denn andernfalls könnte die zweite Kante auch in Farbe 4 gefärbt sein; das widerspräche der eindeutigen 4-Kantenfärbung. Damit folgt schon mal, dass G aus genau drei Kanten besteht. Gleichzeitig müssen alle Kanten paarweise zueinander inzident sein: wenn $e \in C_i$ nicht mit $f \in C_j$ inzident ist, könnten wir auch e mit der Farbe j färben; wieder Widerspruch zur eindeutigen 4-Kantenfärbung. Also kann G nur die Form eines Kreises C_3 oder eines Sterns $K_{1,3}$ haben.

Die Fälle $\chi'(G) = 2$ und $\chi'(G) = 1$ gehen analog. Insgesamt ergibt sich also, dass in Linearzeit überprüft werden, ob ein gegebener Graph G eine eindeutige 4-Kantenfärbung zulässt. Sei $A \in \mathrm{P}$ 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 $\varphi \in \mathtt{SAT}$. Definiere eine zweite SAT-Formel

$$\varphi' \overset{\mathrm{df}}{=} (\neg y \wedge \varphi) \vee (y \wedge \neg x_1 \wedge \neg x_2 \wedge \cdots \wedge x_n),$$

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

Wir haben nun

$$\begin{aligned} &\varphi \notin \mathtt{SAT} \iff |\mathit{set}\text{-}\mathtt{rSAT}(\varphi)| = 0 \iff |\mathit{set}\text{-}\mathtt{rSAT}(\varphi')| = 1 \\ &\iff |\mathit{set}\text{-}\mathtt{r4CHROMINDEX}(f(\varphi'))| = 1 \iff f(\varphi') \in A, \end{aligned}$$

und damit $\mathtt{SAT} \in \mathrm{P}$. □

Leven und Galil (1983) zeigen, dass die Menge Proj(`r4CHROMINDEX`) $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig für NP 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. Es ist auch leicht zu sehen, dass Proj(`r4CHROMINDEX`) paddable ist, also auch P-isomorph zu `SAT` ist. Wir kommen zum Resultat:

Korollar 3.28. *Die NP-Relation* `r4CHROMINDEX` *ist* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* FNP*, und die zugehörige Projektion* Proj(`r4CHROMINDEX`) *ist* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständig für* NP*, und* P*-isomorph zu* `SAT`*. Diese NP-Relation ist insbesondere nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig für* FNP*, außer* $\mathrm{P} = \mathrm{NP}$*.*

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, Fortnow, Naik und Rogers 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 $\leq_m^p$-vollständige Menge L mit P-optimalem Beweissystem für L existiert). In diesem Kapitel werden wir diese Beziehungen näher erarbeiten. Zur Erinnerung:

Vermutung 1.1 (Q). *Für jede totale NPTM N (d.h. $L(N) = \Sigma^*$) existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ ein akzeptierender Rechenweg von $N(x)$ ist.*

In diesem Kapitel werden wir uns grob folgenden drei Desideraten widmen: erstens, nähern wir uns in Abschnitt 4.1 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 Pudláksche Programm einzuordnen.

Zweitens, in Abschnitt 4.2, verallgemeinern wir Charakterisierungen von Q, die sich insbesondere auf Suchprobleme und deren assoziierte Beweissysteme beziehen. Insbesondere zeigen wir für die vollständigen NP-Suchprobleme R dass das zu R assoziierte *Standardbeweissystem* ((x, y) mit $(x, y) \in R$ ist ein Standardbeweis für $x \in \mathrm{Proj}(R)$) P-optimal ist, genau dann wenn Q gilt. Damit wird die P-Optimalität des entsprechenden Standardbeweissystems zu einer Invariante, die entweder für *alle* vollständigen NP-Suchprobleme zutrifft, oder für *keins*.

Drittens ergänzen wir im gesamten Verlauf dieses Kapitels das Pudláksche Programm um weitere Hypothesen (KvL, Q, …), sodass Abbildung 1 der Beziehungen zwischen den Pudlákschen Hypothesen vergrößert und verfeinert wird. Wir erreichen damit den Stand, der in Abbildung 7 dargestellt wird. Damit einher wird abschließend ein Überblick über existierende Orakelkonstruktionen angegeben, welche Hypothesen des Pudlákschen Programms (ergänzt um Q, KvL, …) trennen.

Für alle dieser drei Desiderate ist es zunächst notwendig, auf die Hypothese Q einzugehen. Fenner, Fortnow, Naik und Rogers (2003) beobachten, dass das Invertieren von surjektiven ehrlichen FP-Funktionen eine erstaunlich robuste Aussage ist, die eine Vielzahl von äquivalenten „fundamentalen“ (2003, S. 91) Charakterisierungen aus der Komplexitätstheorie zulässt, so zum Beispiel die effiziente Lösbarkeit von TFNP-Suchproblemen, oder die P-Invertierbarkeit von surjektiven FP-Funktionen. Wir können jetzt schon festhalten, dass die aktuelle Forschung diese Hypothese als sehr stark einschätzt, und eher die negative Beantwortung ¬Q vermutet.

Satz 4.1 (Äquivalente Formulierungen der Hypothese Q; Fenner, Fortnow, Naik und Rogers, 2003). *Folgende Aussagen sind äquivalent:*

(1) *Hypothese* Q.

(2) $\mathrm{NPMV}_t \subseteq_c \mathrm{FP}$.

(3) $\mathrm{TFNP} \subseteq_c \mathrm{FP}$.

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

(5) *Jede surjektive ehrliche 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*

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

Dieser Satz relativiert insbesondere.

Fenner, Fortnow, Naik und Rogers (2003) sowie Köbler und Messner (2000) charakterisieren Q noch durch zwei weitere Formen, diesmal über je eine Aussage über die Menge $\mathtt{SAT}$:

Satz 4.2 (Fenner, Fortnow, Naik und Rogers, 2003). *Es gilt* Q *genau dann wenn Folgendes gilt: Für jede NPTM* N *mit* $L(N) = \mathtt{SAT}$ *existiert eine Funktion* $h \in \mathrm{FP}$ *sodass*

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

Satz 4.3 (Köbler und Messner, 2000). *Es gilt* Q *genau dann wenn das Standardbeweissystem*

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

für $\mathtt{SAT}$ P*-optimal ist.*

Diese zwei Sätze relativieren nicht.

In anderen Worten sagt Satz 4.2 aus, dass es unter Annahme von Q modulo Umcodieren nur einen einzigen SAT-Solver gibt, und insbesondere alle SAT-Solver äquivalent zum trivialen Solver sind, welcher nur alle möglichen Belegungen ausprobiert. Satz 4.3 macht eine analoge Aussage über Beweissysteme: egal wie komplex ein Beweissystem h für $\mathtt{SAT}$ ist, wir können immer einen h-Beweis für φ in eine erfüllende Belegung für φ (quasi ein trivialer Beweis für $\varphi \in \mathtt{SAT}$) transformieren. Damit ist auch leicht zu sehen, dass Q ⇒ ¬SAT, zumindest im unrelativierten Fall.

In Abschnitt 4.2 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, Fortnow, Naik und Rogers bzw. Köbler und Messner zu einer *relativierbaren* Variante verallgemeinert werden. Mit dieser Verallgemeinerung ist es dann auch für uns möglich, Q formal in das Pudláksche Programm (u.a. durch Q ⇒ ¬SAT) einzuordnen. Hierfür führen wir jetzt schon den Begriff eines Standardbeweissystems bezüglich einer NP-Relation formal ein.

Definition 4.4 (Standardbeweissystem einer NP-Relation). Sei R eine NP-Relation. Wir definieren bezüglich R das *Standardbeweissystem* std_R für $\mathrm{Proj}(R)$ wie folgt:

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

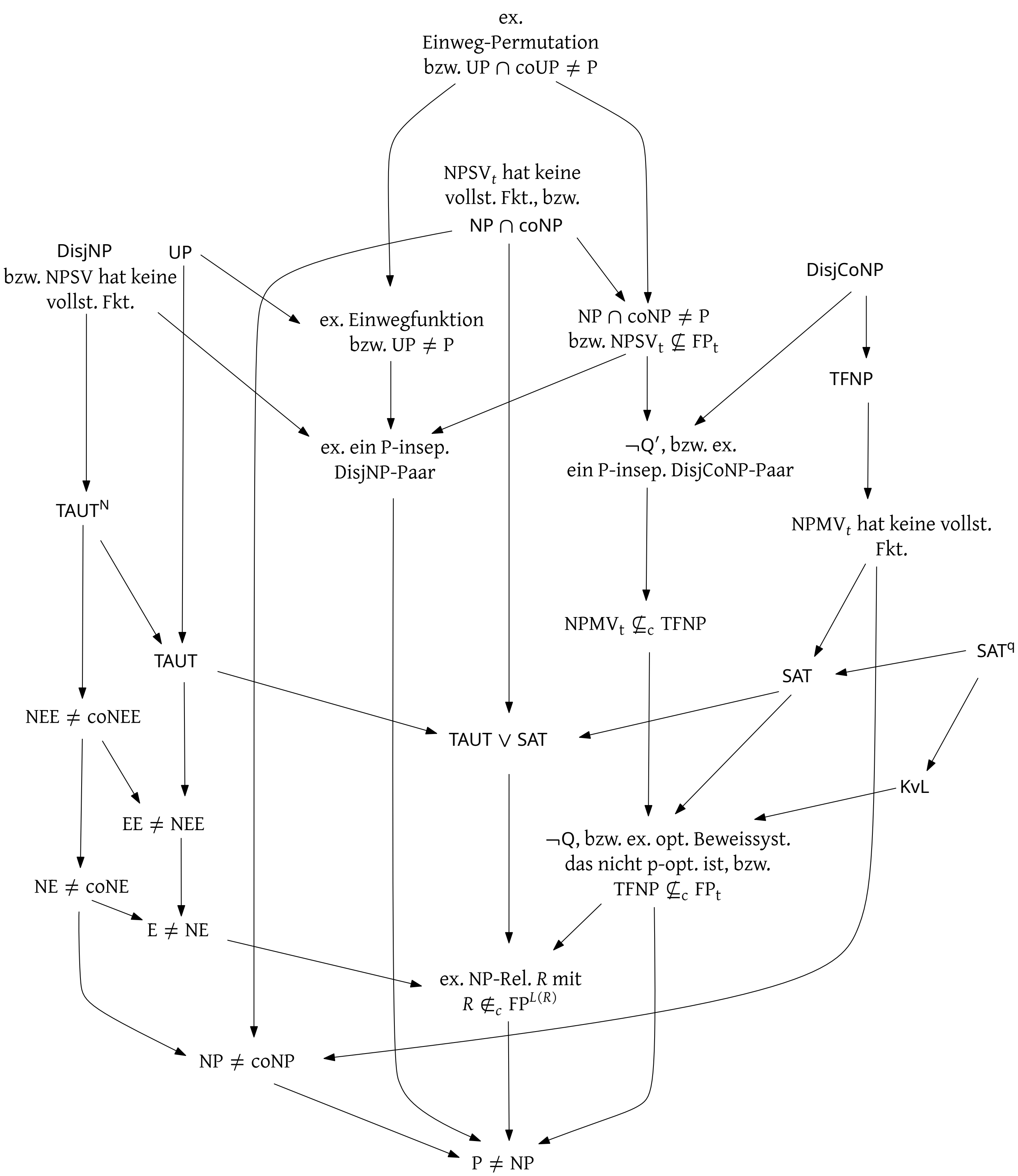

Abbildung 7: Bekannte (relativierenden) Implikationen zwischen den betrachteten Hypothesen und weiteren Aussagen. Satz 4.27 gibt Belegstellen für jede dieser Implikationen an.

Damit ist, wie durch die Formulierung oben suggeriert, $sat = std_{\mathtt{rSAT}}$. Insbesondere ist für jede NP-Relation R das Standardbeweissystem std_R für $\mathrm{Proj}(R)$ ehrlich, optimal, und hat kurze Beweise. Beachte, dass sich aufgrund der speziellen Form der Beweise von Standardbeweissystemen die P-Simulation knapper formulieren lässt:

Beobachtung 4.5. *Sei R eine NP-Relation, und h ein Beweissystem für $\mathrm{Proj}(R)$. Folgende Aussagen sind äquivalent:*

(1) *std_R P-simuliert h.*

(2) *Es existiert eine polynomiell längenbeschränkte Funktion $g \in \mathrm{FP}$, sodass*

$$h(w) = x \implies std_R(x, g(w)) = x \quad \text{(bzw. äquivalent } (x, g(w)) \in R\text{)}$$

gilt.

Beweis. (1) $\implies$ (2): Wenn std_R das Beweissystem h P-simuliert, dann existiert eine Funktion $\pi \in \mathrm{FP}$ sodass $h(w) = x \rightarrow std_R(w) = x$ gilt. Haben wir also einen h-Beweis w für x gegeben, dann muss $\pi(w)$ von der Form (x, z) mit $z \in \Sigma^*$ und $(x, z) \in R$ sein. Insbesondere können wir dann eine Funktion $g \in \mathrm{FP}$ angeben, sodass $g(w)$ die zweite Komponente von $\pi(w)$ ausgibt. Dann gilt $std_R(x, g(w)) = x$.

(2) $\implies$ (1): Haben wir eine solche Funktion $g \in \mathrm{FP}$ gegeben, dann realisiert $\pi(w) = (h(w), g(w))$ die P-Simulation bzw. Reduktion $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$. $\square$

Bevor wir nun mit einer Diskussion zwischen Karp-Vollständigkeit und Levin-Vollständigkeit fortsetzen, machen wir folgende Beobachtung über Q und der Ordnung der Simulation vs. P-Simulation. Diese Beobachtung machen schon Köbler und Messner (2000), aber deren Beweis (vgl. Messner, 2000, Thm. 5.2) relativiert insbesondere nicht. Wir zeigen die Aussage hier über eine nichttriviale Verallgemeinerung, welche insbesondere relativiert.

Satz 4.6. *Folgende Aussagen sind äquivalent:*

(1) Q.

(2) *Sei L eine Menge und h, g je Beweissysteme für L. Es gilt*

$$h \text{ simuliert } g \iff h \text{ P-simuliert } g.$$

(3) *Für jedes Beweissystem h gilt: h ist optimal $\iff$ h ist P-optimal.*

(4) *Jedes ehrliche Beweissystem h ist P-optimal.*

Beweis. (1) $\implies$ (2): Die Richtung von rechts nach links ist klar. Wir zeigen die andere Richtung. Nach Voraussetzung kann h das Beweissystem g simulieren, das heißt es existiert eine (nicht notwendigerweise effiziente) Funktion π sodass $g(w) = h(\pi(w))$, und gleichzeitig ist $|\pi(w)| \leq q(|w|)$ für ein geeignetes Polynom q.

Betrachte folgende Multifunktion f:

$$set\text{-}f(w) \stackrel{\text{df}}{=} \{y \mid \exists y \in \Sigma^{\leq q(|w|)}, g(w) = h(y)\}.$$

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 $\pi(w) \in set\text{-}f(w)$.

Mit Q gilt nach Satz 4.1 also $f \in \mathrm{NPMV}_t \subseteq_c \mathrm{FP}$, 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

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

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

(2) $\Longrightarrow$ (3): Klar.

(3) $\Longrightarrow$ (4): Klar, denn wenn h ehrlich ist, dann hat es insbesondere auch kurze Beweise. Also ist h auch optimal (Beobachtung 2.13), und nach (3) dann auch P-optimal.

(4) $\Longrightarrow$ (1): Wir zeigen, dass mit Voraussetzung (4) auch TFNP $\subseteq_c$ FP gilt, was nach Satz 4.1 äquivalent zu Q ist. Sei hierfür $R \in$ TFNP gegeben.

Das Standardbeweissystem std_R für Σ^* ist ehrlich, also nach (4) auch ein P-optimales Beweissystem für Σ^*. Beachte, dass auch die Identitätsfunktion id ein Beweissystem für Σ^* ist. Also kann std_R das Beweissystem id P-simulieren. Nach Beobachtung 4.5(2) existiert also eine Funktion $g \in$ FP sodass $\mathrm{id}(w) = x \to (x, g(w)) \in R$. Da nun aber $\mathrm{id}(x) = x$ gilt $(x, g(x)) \in R$, also ist g eine Verfeinerung von R. Damit $R \in_c$ FP, wie gewünscht. □

4.1 Karp-Vollständigkeit vs. Levin-Vollständigkeit

Wir wiederholen hier erneut die zentrale offene Frage und Vermutung aus Abschnitt 3.3:

Frage 3.23. *Sei R eine NP-Relation. Wenn* $\mathrm{Proj}(R)$ *eine* $\leq_m^p$*-vollständige Menge für* NP *ist, ist dann auch R eine* $\leq_L^p$*-vollständige NP-Relation für* FNP*?*

Vermutung 3.24 (KvL)**.** *Es existiert eine NP-Relation R sodass* $\mathrm{Proj}(R)$ $\leq_m^p$*-vollständig für* NP *ist, aber R ist nicht* $\leq_L^p$*-vollständig für* FNP.

Zunächst sei hier noch einmal hervorgehoben, dass eine negative Beantwortung der Frage 3.23, also ein Beweis von KvL schwer ist. Zum einen haben wir bereits gesehen, dass ein Beweis KvL auch sofort P $\neq$ NP beweisen würde. Insbesondere ist ein relativierender Beweis von KvL ausgeschlossen, denn existiert ein Orakel, relativ zu diesem $\neg$KvL (z.B. ein PSPACE-vollständiges Orakel, welches NP auf P kollabiert). Auch für eine positive Beantwortung der Frage (also ein Beweis für $\neg$KvL) fehlen uns konkrete Indizien.

Wir werden uns daher im Folgenden insbesondere auf Beziehungen zwischen KvL und gewissen anderen Hypothesen konzentrieren. In diesem Sinne möchte ich 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 einer Verstärkung der Vermutung $\neg$KvL entspricht.[9]

9. Fenner, Fortnow, Naik und Rogers (2003) gaben hierbei eine ähnliche Aussage an (Cor. 3: „Q holds iff every Karp reduction from A to B can be extended to a Levin reduction“), es ist aber hervorzuheben, dass die Autoren von einem unüblichen Begriff von Levin-Reduktionen ausgehen, der sich von dem hier verwendeten unterscheidet. Dieser umfasst nicht eine „Rückwärts-Translation“ von Zertifikaten für B-Instanzen zu A-Instanzen, sondern eine „Vorwärts-Translation“ von Zertifikaten für A-Instanzen zu B-Instanzen.

Satz 4.7. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q.

(2) Für jedes Paar von NP-Relationen A, B gilt:

$$\mathrm{Proj}(A) \leq_m^p \mathrm{Proj}(B) \iff A \leq_L^p B.$$

Beweis. (1) $\Longrightarrow$ (2): Die Richtung von rechts nach links ist klar. Für die andere Richtung sei $\mathrm{Proj}(A) \leq_m^p \mathrm{Proj}(B)$ mit A, B NP-Relationen. Sei q hierbei die Zertifikatsschranke von A. Wir wollen nun eine Levin-Reduktion von A auf B angeben. Sei $f \in$ FP die Funktion, welche die Reduktion $\mathrm{Proj}(A) \leq_m^p \mathrm{Proj}(B)$ realisiert. Zunächst halten wir fest, dass unter (1) das Standardbeweissystem std_R P-optimal ist, denn es ist ehrlich, also nach Satz 4.6 auch P-optimal.

Definiere folgendes Beweissystem

$$h(w) = \begin{cases} x & \text{falls } w = 0\langle x, y\rangle \text{ und } (x, y) \in A \\ x & \text{falls } w = 1\langle x, z\rangle \text{ und } (f(x), z) \in B \\ \perp & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, dass h ein Beweissystem für $\mathrm{Proj}(A)$ ist. Also kann std_R das Beweissystem h P-simulieren. Mit Beobachtung 4.5 folgt, dass eine Funktion g existiert mit

$$h(w) = x \implies (x, g(w)) \in A.$$

Wir zeigen nun, dass $A \leq_{\mathrm{L}}^{\mathrm{p}} B$. Sei hierfür f von oben die entsprechende Reduktionsfunktion. Es gilt

$$(f(x), z) \in B \implies h(1\langle x, z\rangle) = x \implies (x, \underbrace{g(1\langle x, z\rangle)}_{g'(x,z)}) \in A,$$

heißt mit der Translationsfunktion $g' \in \mathrm{FP}, g'(x, z) = g(1\langle x, z\rangle)$ können wir B-Zertifikate für $f(x)$ in A-Zertifikate für x umrechnen, wie gewünscht.

(2) $\implies$ (1): Wir zeigen $\mathrm{TFNP} \subseteq_c \mathrm{FP}$; das ist nach Satz 4.1 äquivalent zu (1). Sei nun A eine totale NP-Relation Definiere nun die NP-Relation

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

Es ist leicht zu sehen das $\mathrm{Proj}(A) = \Sigma^* = \mathrm{Proj}(B)$ 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, mit Reduktionsfunktion $f \in \mathrm{FP}$ und Translationsfunktion $g \in \mathrm{FP}$.

Sei nun ein Wort $x \in \Sigma^*$ gegeben. Hieraus können wir effizient ein A-Zertifikat berechnen: es gilt $(f(x), \varepsilon) \in B$, also auch $(x, g(x, \varepsilon)) \in A$. Heißt, $g(x, \varepsilon)$ ist unser gesuchtes A-Zertifikat, welches wir auch effizient berechnen können. $\square$

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 Q hinreichend für ¬KvL ist: ist $\mathrm{Proj}(A)$ $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig, dann lässt sich jede Reduktion $\mathrm{Proj}(B) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(A)$ zu einer $\leq_{\mathrm{L}}^{\mathrm{p}}$-Reduktion $B \leq_{\mathrm{L}}^{\mathrm{p}} A$ verstärken; also ist A $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig.

Korollar 4.8. Q $\implies$ ¬KvL.

P-Quasi-Simulation

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

Um uns dieser Frage dennoch zu nähern, beginnen wir zunächst, die Hypothese KvL auf weitere Weisen zu charakterisieren. Es ist intuitiv ersichtlich, dass für eine feste Sprache L die NP-Relationen für L, die ehrlichen Beweissysteme für L, und die NPTMs, welche L entscheiden, alle im breitesten Sinn „Zertifikatsschemata“ sind, und alle diese Berechnungsmodelle untereinander umgeschrieben werden können: jede NP-Relation R induziert ein ehrliches Beweissystem std_R, jedes ehrliche Beweissystem h induziert eine

NPTM N mit $L(N) = L$ (rate einen h-Beweis polynomieller Länge), und jede NPTM N mit $L(N) = L$ induziert eine NP-Relation für L (w ist eine Lösung für x wenn w ein akzeptierender Rechenweg auf $N(x)$ ist). Insbesondere ist für ein ehrliches Beweissystem h die Umkehrrelation $h^{-1} = \{(x, w) \mid h(w) = x\}$ eine NP-Relation für $\mathrm{img}(h)$.

Wir wollen nun die existierende $\leq_{\mathrm{L}}^{\mathrm{p}}$-Ordnung auf den NP-Relationen aufgreifen, und diese auf die ehrlichen Beweissysteme übertragen. Wir definieren hierfür eine abgeschwächte Variante der P-Simulation, welche der $\leq_{\mathrm{L}}^{\mathrm{p}}$-Reduktion nachempfunden ist.

Definition 4.9. Seien h, h' Beweissysteme für L. Das Beweissystem h P-*quasi-simuliert* h' falls Funktionen $f, g \in \mathrm{FP}$ existieren sodass

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

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

In anderen Worten, falls h das Beweissystem h' P-quasi-simuliert, 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. Die P-Quasi-Simulation ist reflexiv und transitiv, und sie liegt insbesondere zwischen Simulation und P-Simulation: P-Simulation impliziert P-Quasi-Simulation impliziert Simulation. Ferner ist diese Art von Simulation *effektiv*, in dem Sinn dass die P-invertierbaren Beweissysteme selbst unter P-Quasi-Simulation abgeschlossen sind.

Es ist intuitiv ersichtlich, dass P-quasi-Simulation auf ehrlichen Beweissystemen das Analog der $\leq_{\mathrm{L}}^{\mathrm{p}}$-Reduktion auf NP-Relationen darstellt. Wir können folgende Beobachtung festhalten:

Beobachtung 4.10. *Sei $L \in \mathrm{NP}$, seien h, h' zwei ehrliche Beweissysteme für L, und seien R, R' zwei NP-Relationen für L.*

(1) $h^{-1} \leq_{\mathrm{L}}^{\mathrm{p}} h'^{-1} \iff$ *h kann h'* P-*quasi-simulieren.*

(2) $R \leq_{\mathrm{L}}^{\mathrm{p}} R' \iff$ *std_R kann $std_{R'}$* P-*quasi-simulieren.*

Beweis. Zu (1): Wir haben

$$\begin{aligned}
&h^{-1} \leq_{\mathrm{L}}^{\mathrm{p}} h'^{-1} \iff \exists f, g \in \mathrm{FP}.\, [(f(x), w) \in h'^{-1} \to (x, g(x, w)) \in h^{-1}] \\
&\iff \exists f, g \in \mathrm{FP}.\, [h'(w) = f(x) \to h(g(x, w)) = x] \\
&\iff h \text{ kann } h' \text{ P-quasi-simulieren,}
\end{aligned}$$

wobei hier die existentizell quantifizierte Variable f nur über Reduktionsfunktionen von L auf L reichen soll.

Zu (2): Ähnlich wie (1), nur etwas aufwändiger. Wir haben

$$\begin{aligned}
&R \leq_{\mathrm{L}}^{\mathrm{p}} R' \iff \exists f, g \in \mathrm{FP}.\, [(f(x), w) \in R' \to (x, g(x, w)) \in R] \\
&\iff \exists f, g \in \mathrm{FP}.\, [std_{R'}(f(x), w) = f(x) \to std_R(x, g(x, w)) = x \\
&\iff \exists f, g' \in \mathrm{FP}.\, [std_{R'}(w') = f(x) \to std_R(g'(x, w')) = x \\
&\iff std_R \text{ kann } std_{R'} \text{ P-quasi-simulieren,}
\end{aligned}$$

wobei wieder die existentizell quantifizierte Variable f nur über Reduktionsfunktionen von L auf L reichen soll. $\square$

Die intuitive Aussage von ¬KvL „Alle NP-Relationen für NP-vollständige Sprache L sind im Wesentlichen gleich“ überträgt sich nun auch auf ehrliche Beweissysteme:

Satz 4.11. *Folgende Aussagen sind äquivalent:*

(1) *Aussage* $\neg$KvL.

(2) *Für jede* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-vollständige Menge* $L \in \mathrm{NP}$ *gilt: Wenn* R, R' *zwei NP-Relationen für* L *sind, dann*

$$R \leq_{\mathrm{L}}^{\mathrm{p}} R'.$$

(3) *Für jede* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-vollständige Menge* $L \in \mathrm{NP}$ *gilt: Wenn* h, h' *zwei ehrliche Beweissysteme für* L *sind, dann*

$$h \text{ P-}\textit{quasi-simuliert } h'.$$

(4) *Für eine* $\leq_{\mathrm{L}}^{\mathrm{p}}$*-vollständige NP-Relation* R *gilt: Wenn* R' *eine NP-Relation für* $\mathrm{Proj}(R)$ *ist, dann*

$$R \leq_{\mathrm{L}}^{\mathrm{p}} R'.$$

Beweis. (1) $\Longrightarrow$ (2): Klar, denn mit (1) ist R' $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig.

(2) $\Longrightarrow$ (4): Klar, ist ja rSAT $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig.

(4) $\Longrightarrow$ (1): Sei Q eine NP-Relation, wobei $\mathrm{Proj}(Q)$ $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständig für NP ist. Wir werden zeigen, dass $R \leq_{\mathrm{L}}^{\mathrm{p}} Q$, wobei dann auch Q $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig ist (Beob. 3.20), wie gewünscht. Die Argumentation verläuft hierbei ähnlich wie bei Satz 4.7. Sei $f \in \mathrm{FP}$ die Funktion, welche $\mathrm{Proj}(R) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(Q)$ realisiert. Sei nun

$$R' = \{(x, 0y) \mid (x, y) \in R\} \cup \{(x, 1z) \mid (f(x), z) \in Q\}.$$

Es lässt sich leicht zeigen, dass R' eine NP-Relation für $\mathrm{Proj}(R)$ ist. Nach (4) gilt nun $R \leq_{\mathrm{L}}^{\mathrm{p}} R'$. Ebenso gilt $R' \leq_{\mathrm{L}}^{\mathrm{p}} Q$, denn wir haben

$$(f(x), z) \in Q \implies (x, \underbrace{1z}_{g(x,z)}) \in R',$$

also realisiert f als Reduktionsfunktion und $g(x, z) \stackrel{\mathrm{df}}{=} 1z$ als Translationsfunktion die Levin-Reduktion $R' \leq_{\mathrm{L}}^{\mathrm{p}} Q$. Damit gilt $R \leq_{\mathrm{L}}^{\mathrm{p}} R' \leq_{\mathrm{L}}^{\mathrm{p}} Q$ wie gewünscht.

(2) $\Longleftrightarrow$ (3): Einfach mit Beobachtung 4.10 nachweisbar. $\square$

Als Korollar erhalten wir dabei insbesondere folgende Aussage, welche KvL mit Standardbeweissystemen von $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen NP-Relationen in Beziehung setzt:

Korollar 4.12. *Sei* R *eine NP-Relation, sodass* R $\leq_{\mathrm{L}}^{\mathrm{p}}$*-vollständig ist. Folgende Aussagen sind äquivalent:*

(1) *Aussage* $\neg$KvL.

(2) *Für alle ehrlichen Beweissysteme* h *mit* $\mathrm{img}(h) = \mathrm{Proj}(R)$ *gilt*

$$std_R \text{ P-}\textit{quasi-simuliert } h.$$

Beweis. (1) $\Longrightarrow$ (2): Klar, denn std_R ist ehrlich, und mit Satz 4.11(2) kann std_R damit h auch P-quasi-simulieren.

(2) $\Longrightarrow$ (1): Wir zeigen die Aussage von Satz 4.11(4), womit auch $\neg$KvL bzw (1) gilt. Sei hierfür R' eine NP-Relation für $\mathrm{Proj}(R)$. Mit (2) wissen wir dass std_R das Beweissystem $std_{R'}$ P-quasi-simulieren kann. Mit Beobachtung 4.10(2) folgt $R \leq_{\mathrm{L}}^{\mathrm{p}} R'$. $\square$

Unter dieser Charakterisierung erhalten wir einen zweiten trivialen Beweis von Korollar 4.8: Q ist hinreichend für ¬KvL, denn unter Annahme von Q ist jedes ehrliche Beweissystem P-optimal, heißt auch das Standardbeweissystem $std_{\mathtt{rKAN}}$ für KAN ist P-optimal (Satz 4.6), kann also insbesondere auch alle ehrlichen Beweissysteme *h* P-quasi-simulieren. Damit auch ¬KvL (Korollar 4.12).

Mittels unserem Begriff der P-Quasi-Simulation können wir nun auch die Hypothese SAT auf natürliche Weise verstärken:

Vermutung 4.13 ($\mathsf{SAT^q}$). *Es existiert keine $\leq_m^p$-vollständige Menge L für* NP *mit einem Beweissystem h, welches alle anderen ehrlichen Beweissysteme für L* P-*quasi-simulieren kann.*

Diese Verstärkung ist am sichtbarsten, wenn wir uns überlegen, wie die Negation ¬SAT zu ¬$\mathsf{SAT^q}$ in zwei Aspekten abgeschwächt wird. Erstens der Umfang: es wird nicht mehr verlangt, dass ein Beweissystem *h* existiert, welche *alle* Beweissysteme P-simulieren kann, sondern nur noch alle *ehrlichen* Beweissysteme. Zweitens die Beziehung: es wird nicht mehr P-Simulation, sondern nur noch P-*Quasi*-Simulation vorausgesetzt.

Wir sehen nun, dass $\mathsf{SAT^q}$ eine Verstärkung von sowohl SAT als auch KvL ist:

Korollar 4.14. *(1)* $\mathsf{SAT^q} \implies \mathsf{SAT}$

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

Beweis. Für (1) haben wir bereits argumentiert. Wir zeigen nur noch (2) mittels Kontraposition. Unter ¬KvL folgt mit der Formulierung aus Korollar 4.12(2), dass das Standardbeweissystem der $\leq_L^p$-vollständigen NP-Relation rSAT alle ehrlichen Beweissysteme P-quasi-simulieren kann. Also hat die Menge KAN *ein* Beweissystem, welches alle ehrlichen Beweissysteme P-quasi-simulieren kann. □

Wir haben damit also je eine notwendige (¬Q) und eine hinreichende Hypothese ($\mathsf{SAT^q}$) für KvL. Durch die Charakterisierung von ¬KvL über Beweissysteme ergibt sich durch Korollar 4.12 eine interessante Ähnlichkeit zur Hypothese Q. Wir wollen diese beiden Hypothesen im Folgenden kurz gegenüberstellen. Zur Anschaulichkeit beziehe ich mich hierbei auf die NP-Relation rSAT. (Durch Ergebnisse im nächsten Abschnitt wird klar, dass sich dieses Argument auch auf rKAN relativierend überträgt.) Wir haben

- Q genau dann wenn *sat* jedes ehrliche Beweissystem P-simulieren kann (Satz 4.2),[10]
- ¬KvL genau dann wenn *sat* jedes ehrliche Beweissystem P-quasi-simulieren kann (Korollar 4.12).

10. Satz 4.2 macht zwar nur Aussagen über NPTMs, aber es ist leicht zu sehen, dass diese mit den ehrlichen Beweissystemen identifiziert werden können. Vgl. auch Satz 4.17.

Damit wird deutlich, dass die die Hypothesen Q und ¬KvL äquivalent sind, falls die P-Quasi-Simulation genauso stark ist, wie die P-Simulation.

Dennoch bleiben weitere Fragen offen, die wir hier aus Platzgründen nicht weiter verfolgen werden. Zum einen zur Charakterisierung von KvL: Wir konnten zwar KvL u.a. als Aussage über Beweissysteme formulieren, aber sind auch noch weitere äquivalente Charakterisierungen möglich, z.B. ähnlich wie bei Q?

Zum anderen die Beziehungen zwischen KvL und anderen Hypothesen bzw. Annahmen. Gibt es natürliche (z.B. kryptographische) Annahmen, welche hinreichend für KvL sind? Wie ist die Beziehung zu den anderen Pudlákschen Hypothesen? Wie verhält sich insbesondere SAT zu $\mathsf{SAT^q}$? Diese Fragen werden wir zum Teil in Kapitel 5 klären; dort wird ein Orakel konstruiert, welches zeigt, dass selbst unter der Annahme von DisjNP ∧

UP es nicht möglich ist, mit relativierenden Beweismethoden auf KvL zu schließen. Wir kommen hierauf am Ende dieses Kapitels noch einmal zurück.

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, Fortnow, Naik und Rogers (2003) bzw. Köbler und Messner (2000) äquivalente Charakterisierungen der Hypothese Q an, welche sich im Wesentlichen auf auf der $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit von $\mathtt{rSAT}$ aufbauen (Satz 4.2 und 4.3). Wir wiederholen hier noch einmal die Aussagen, aber mit einer etwas abstrakteren Notation. Wie schon angesprochen, können für eine feste Menge L die ehrlichen Beweissysteme für L mit den NPTMs identifiziert werden, welche L entscheiden. So können wir die anfangs genannten Charakterisierung von Fenner, Fortnow, Naik und Rogers stattdessen folgendermaßen notieren:

Satz 4.2 (Fenner, Fortnow, Naik und Rogers, 2003). *Es gilt* Q *genau dann wenn* $h \leq_{\mathrm{m}}^{\mathrm{p}} sat$ *für jedes ehrliche Beweissystem* h *für* $\mathtt{SAT}$*.*

Die Charakterisierung von Köbler und Messner können wir auf ähnliche Weise über Standardbeweissysteme von NP-Relationen schreiben.

Satz 4.3 (Köbler und Messner, 2000). *Es gilt* Q *genau dann wenn das Standardbeweissystem* $std_{\mathtt{rSAT}}$ P*-optimal ist.*

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

Das ist für den zweiten Satz 4.3 ohne weitere zusätzliche Bedingungen möglich. Beachte, dass sich der hier präsentierte Beweis maßgeblich von dem ursprünglichen Beweis von Messner (vgl. 2000, Thm. 5.2) unterscheidet, und insbesondere ohne sein (nichtrelativierendes) Padding-Argument auskommt.

Satz 4.15. *Sei* R *eine beliebige* $\leq_{\mathrm{L}}^{\mathrm{p}}$*-vollständige NP-Relation. Folgende Aussagen sind äquivalent:*

(1) *Aussage* Q*.*

(2) *Das Standardbeweissystem* std_R *ist* P*-optimal.*

Beweis. (1) $\Longrightarrow$ (2): Das Standardbeweissystem std_R ist ehrlich. Nach Satz 4.6 ist std_R damit P-optimal.

(2) $\Longrightarrow$ (1): Wir zeigen die Aussage, dass TFNP $\subseteq_{\mathrm{c}}$ FP. Nach Satz 4.1 ist das äquivalent zu Q. Sei hierfür ein solches $Q \in$ TFNP gegeben

Nachdem R $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig ist, haben wir Funktionen $f, g \in$ FP mit

$$x \in \mathrm{Proj}(Q) \iff f(x) \in \mathrm{Proj}(R), \quad (f(x), z) \in R \implies (x, g(x,z)) \in Q.$$

Definiere nun

$$h(w) = \begin{cases} f(x) & \text{wenn } w = 0x \\ x & \text{wenn } w = 1\langle x, y\rangle \text{ und } (x,y) \in R \\ \bot & \text{sonst} \end{cases}.$$

Es ist leicht zu sehen, dass h ein Beweissystem für $\mathrm{Proj}(R)$ ist, denn insbesondere gilt $f(x) \in \mathrm{Proj}(R)$ für jedes $x \in \Sigma^*$. Nach (2) gilt also $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$. Mit Beobachtung 4.5 haben wir eine Funktion $r \in \mathrm{FP}$, sodass

$$h(w) = x \implies (x, r(w)) \in R.$$

Damit gilt insbesondere

$$h(0x) = f(x) \implies (f(x), r(0x)) \in R \implies (x, \underbrace{g(r(0x))}_{q(x)}) \in Q.$$

Damit ist $q(x) = g(r(0x))$, $q \in \mathrm{FP}$ eine Verfeinerung von Q, wie gewünscht. □

Die andere Charakterisierung von Fenner, Fortnow, Naik und Rogers ist umfangreicher. Zunächst ist klar, dass „$h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$ für alle ehrlichen Beweissysteme“ schwächer ist als „$h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$ für alle Beweissysyteme“, was ja nach vorigem Satz äquivalent zu Q ist. Tatsächlich gilt die Umkehrung scheinbar nicht für alle $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen NP-Relationen R. Die Autoren zeigen diese Umkehrung für die NP-Relation $\mathtt{rSAT}$, und nutzen hierbei die Eigenschaft aus, dass die entsprechenden Reduktionsfunktionen alle P-invertierbar sind, bzw. in anderen Worten, die $\leq_{\mathrm{L,1,i}}^{\mathrm{p}}$-Vollständigkeit von $\mathtt{rSAT}$.

Wir werden im Folgenden diese Voraussetzung abschwächen, und betrachten dabei folgende Verstärkung der $\leq_{\mathrm{m}}^{\mathrm{p}}$-Reduktion.

Definition 4.16 (Ehrliche Levin-Reduzierbarkeit)**.** Seien Q, R zwei NP-Relationen. Wir sagen dass Q *sich auf* R *ehrlich Levin-reduzieren lässt*, bzw. $Q \leq_{\mathrm{L,h}}^{\mathrm{p}} R$ wenn $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$, und die zugehörige Reduktionsfunktion f ehrlich ist.

Definiere $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-Vollständigkeit entsprechend. ◁

Beachte, dass $\leq_{\mathrm{L,h}}^{\mathrm{p}}$ zwischen $\leq_{\mathrm{L}}^{\mathrm{p}}$ und $\leq_{\mathrm{L,1,i}}^{\mathrm{p}}$ liegt:

$$Q \leq_{\mathrm{L,1,i}}^{\mathrm{p}} R \implies Q \leq_{\mathrm{L,h}}^{\mathrm{p}} R \implies Q \leq_{\mathrm{L}}^{\mathrm{p}} R.$$

Damit impliziert $\leq_{\mathrm{L,1,i}}^{\mathrm{p}}$-Vollständigkeit eine $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-Vollständigkeit, und diese eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit.

Mit dem Begriff der ehrlichen Reduzierarkeit können wir den vorigen Beweis von Satz 4.15 erweitern, womit wir Satz 4.2 generalisieren und relativieren.

Satz 4.17. *Sei R eine beliebige $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständige NP-Relation. Folgende Aussagen sind äquivalent:*

(1) Aussage Q.

(2) Das Standardbeweissystem std_R ist P*-optimal.*

(3) Für jedes ehrliche Beweissystem h für $\mathrm{Proj}(R)$ *gilt $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$.*

Beweis. (1) $\implies$ (2): Klar, nach Satz 4.6 ist mit (1) das Beweissystem std_R P-optimal.

(2) $\implies$ (3): Klar.

(3) $\implies$ (1): Analog wie bei Satz 4.15. Wir zeigen wieder die Aussage, dass $\mathrm{TFNP} \subseteq_c \mathrm{FP}$. Sei hierfür ein solches $Q \in \mathrm{TFNP}$ gegeben. Nachdem R $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig ist, haben wir

Funktionen $f, g \in \mathrm{FP}$, welche $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ realisieren. Definiere nun

$$h(w) = \begin{cases} f(x) & \text{wenn } w = 0x \\ x & \text{wenn } w = 1\langle x, y\rangle \text{ und } (x, y) \in R \\ \bot & \text{sonst} \end{cases}.$$

Es ist leicht zu sehen, dass h ein Beweissystem für $\mathrm{Proj}(R)$ ist. Insbesondere ist h ehrlich, denn f ist ehrlich. Damit gilt mit (3), dass $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$; mit Beobachtung 4.5 existiert dann eine Funktion $r \in \mathrm{FP}$, sodass $h(w) = x \rightarrow (x, r(w)) \in R$. Damit ist $q(x) = g(r(0x))$, $q \in \mathrm{FP}$ eine Verfeinerung von Q, wie gewünscht. □

Wir fassen kurz den aktuellen Stand zusammen. Sei R eine NP-Relation. Wir haben nun folgendes Bild:

$$
\begin{tikzcd}
\mathsf{Q} \arrow[r, Rightarrow, bend right, "(4.15)"] & std_R \text{ ist P-opt.} \arrow[l, Rightarrow, bend right, "\text{falls } R \text{ Levin-vollst.}"'] \arrow[r, Rightarrow, bend right, "(4.17)"] & \begin{array}{c} h \leq_{\mathrm{L}}^{\mathrm{p}} std_R \text{ für alle} \\ \text{ehrl. Beweissyst. } h \text{ für } \mathrm{Proj}(R) \end{array} \arrow[l, Rightarrow, bend right, "\text{falls } R \text{ ehrlich Levin-vollst.}"']
\end{tikzcd}
$$

Levin-Paddability

Die stärkere Voraussetzung an die NP-Relation R, dass diese *ehrlich* Levin-vollständig sein muss, wollen wir hier noch einmal kurz beleuchten. Ganz ähnlich zur Berman–Hartmanis-Paddability lässt sich auch die ehrliche Levin-Vollständigkeit von R äquivalent als strukturelle Eigenschaft von R formulieren. Hierbei machen wir einen sehr schwachen Begriff von Paddabilty auf Suchproblemen produktiv, welcher von Messner (2000) konkret auf `rSAT` eingesetzt wurde.

Ähnlich wie bei der Berman–Hartmanis-Paddability wollen wir beliebige Instanzen x zu längeren Instanzen x' vergrößern. Zusätzlich verlangen wir, dass wir auch Zertifikate y für x' wieder zu Zertifikaten 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 das redundanten Padding bezieht, und erhalten wieder ein Zertifikat y für x.

Definition 4.18 (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. Wieder im Gegensatz zur Berman–Hartmanis-Paddability verlangen wir aber ein effizientes Rückrechnen der Zertifikate von gepaddeten Instanzen.

Wir sehen einfach, dass `rSAT` Levin-paddable ist: padde Formeln φ auf, indem z.B. Disjunktionen neuer Variablen hinzugefügt werden, also z.B.

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

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

Diese Eigenschaft lässt sich auch leicht für die kanonische NP-Relation $\mathtt{rKAN}$ überprüfen, indem z.B. für Eingabeinstanz $x = (N, x, 1^n)$ der NPTM N weitere Zustände hinzugefügt werden, welche über die Transitionsrelation von N nicht erreichbar sind. Diese Beobachtung gilt insbesondere auch im relativierten Fall.

Beobachtung 4.19. *Die kanonische $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{i}}$-vollständige NP-Relation $\mathtt{rKAN}$ ist Levin-paddable.*

Folgendes Lemma zeigt nun, dass Levin-Paddability unter den vollständigen Relationen hinreichend für *ehrliche* Vollständigkeit ist.

Lemma 4.20. *Sei R eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation. Wenn R Levin-paddable ist, dann ist R auch $\leq^{\mathrm{p}}_{\mathrm{L,h}}$-vollständig.*

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

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

$$f'(x) \stackrel{\mathrm{df}}{=} pad(f(x), 1^{|x|}).$$

Es gilt

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

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.

Wir zeigen nun, wie sich aus R-Zertifikaten für $f'(x)$ wieder Q-Zertifikate für x ausrechen lassen können. Sei hierfür ein z gegeben mit $(f'(x), z) \in R$. Nach Definition gilt also $(pad(f(x), 1^{|x|}), z) \in R$, also gilt nach Levin-Paddability dass

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

und nach Definition der Translationsfunktion g gilt dann

$$(x, \underbrace{g(x, w)}_{g(x,z)}) \in Q,$$

und wir haben unser Q-Zertifikat für x. Dieses können wir effizient mittels geeignetem $g' \in \mathrm{FP}$ berechnen. □

Für die andere Richtung sehen wir, dass sich Levin-Paddability auf $\leq^{\mathrm{p}}_{\mathrm{L,h}}$ nach oben überträgt.

Lemma 4.21. *(1) Gilt $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L,h}} R$, dann ist R Levin-paddable (und $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig für* FNP*).*

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

Beweis. Aussage (2) folgt unmittelbar aus (1).

Für (1) nutzen wir die Levin-Paddability von $\mathtt{rKAN}$ aus: übersetze Instanz x von R nach $\mathtt{rKAN}$, padde dort hoch, und übersetze 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 $\mathtt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L,h}} R$ 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 $\mathtt{rKAN}$ $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Nach Voraussetzung ist f ehrlich. Und nach Beobachtung 4.19 existieren für $\mathtt{rKAN}$ 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

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

Die Zugehörigkeit zu $\mathrm{Proj}(R)$ 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

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

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

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

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

Definiere $z \overset{\mathrm{df}}{=} g(p, y')$. 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 $z' \overset{\mathrm{df}}{=} padsol_{\mathtt{rKAN}}(f'(x), 1^n, z)$ gilt

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

Damit haben wir ein R-Zertifiakt y für x. Es ist leicht zu sehen, dass sich eine Funktion $padsol_R \in \mathrm{FP}$ angeben kann, die aus $x, 1^n, y'$ dieses entsprechende y berechnen kann. □

Wir erhalten damit als Korollar folgende Äquivalenz:

Korollar 4.22. *Sei R eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation. Folgende Aussagen sind äquivalent:*

(1) R ist $\leq^{\mathrm{p}}_{\mathrm{L,h}}$-vollständig.

(2) R ist Levin-paddable.

Für die üblichen $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen natürlichen NP-Relationen (wie rSETCOVER, rCLIQUE, rCLIQUE, r3COLORABILITY, rHAMCYCLE, rMAXCUT) ist leicht zu sehen, dass diese alle Levin-paddable sind, also auch $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständig sind. Es ist unklar, ob Levin-Paddability für *alle* vollständigen NP-Relationen zutrifft. Diese allgemeine Frage zwischen $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit und Levin-Paddability bzw. $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-Vollständigkeit werden wir aber im Folgenden nicht weiter bearbeiten:

Frage 4.23. *Wenn eine NP-Relation R $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig für* FNP *ist, ist dann R auch Levin-paddable? Existiert ggf. ein Gegenbeispiel in einer geeigneten relativierten Umgebung?*

Charakterisierungen von Q

Unabhängig von voriger Frage können wir nun aber abschließend die vorigen Ergebnisse in folgendem Satz zusammenfassen. Beachte dass diese Charakterisierungen relativieren.

Satz 4.24 (Äquivalente Formulierungen der Hypothese Q). *Folgende Aussagen sind äquivalent:*

(1) *Hypothese* Q*: Für jede totale NPTM N existiert eine Funktion $g \in \mathrm{FP}$ sodass für alle x das Bild $g(x)$ ein akzeptierender Rechenweg von $N(x)$ ist.*

(2) $\mathrm{TFNP} \subseteq_c \mathrm{FP}$

(3) $\mathrm{NPMV}_t \subseteq_c \mathrm{FP}$

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

(5) *Jede surjektive ehrliche Funktion $f \in \mathrm{FP}$ ist* P*-invertierbar, heißt die Umkehrrelation f^{-1} hat eine Verfeinerung in* FP*.*

(6) *Für jede Menge $L \in \mathrm{P}$ und jede NPTM N mit $L(N) = L$ existiert eine Funktion $h \in \mathrm{FP}$ mit*
$$x \in L \implies N(x) \text{ akz. mit Rechenweg } h(x).$$

(7) *Für jedes Paar von NP-Relationen A, B gilt:*
$$\mathrm{Proj}(A) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(B) \iff A \leq_{\mathrm{L}}^{\mathrm{p}} B.$$

(8) *Für jedes Beweissystem h gilt: h ist optimal $\iff$ h ist* P*-optimal.*

(9) *Jedes ehrliche Beweissystem h ist* P*-optimal.*

(10) *Es existiert eine $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständige NP-Relation R sodass für alle ehrlichen Beweissysteme h für $\mathrm{Proj}(R)$ auch $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$ gilt.*

(11) *Es existiert eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation R für welche das Standardbeweissystem std_R* P*-optimal ist.*

Beweis. 1. (1) $\iff$ (2) $\iff$ (4) $\iff$ (5) $\iff$ (6): nach Fenner, Fortnow, Naik und Rogers (2003, Thm. 2).
2. (1) $\iff$ (3): nach Beobachtung 3.5.
3. (1) $\iff$ (7): nach Satz 4.7.
4. (1) $\iff$ (8) $\iff$ (9): nach Satz 4.6.
5. (1) $\iff$ (10): nach Satz 4.17 und der $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-Vollständigkeit von rKAN.
6. (1) $\iff$ (11): nach Satz 4.15 und der $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit von rKAN. □

Analysiert man die Beweise bezüglicher der Äquvalenz von Aussage Q zu (10) bzw. (11) können wir sogar feststellen, dass die Wahl der $\leq_{\mathrm{L}}^{\mathrm{p}}$- bzw. $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständigen Relation R beliebig ist. Wir können daher Q über universell quantifizierte Varianten von (10) und (11) charakterisieren.

Satz 4.25. *Entweder gelten die Aussagen (1), (10), (11) oder die Aussagen (1'), (10'), (11'):*

(1) Q.

(10) *Für alle $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständigen NP-Relationen R, alle ehrlichen Beweissysteme h für* $\mathrm{Proj}(R)$ *gilt auch $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$.*

(11) *Für alle $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen NP-Relationen R ist das Standardbeweissystem std_R* P*-optimal.*

(1') ¬Q.

(10') *Es existiert keine $\leq_{\mathrm{L,h}}^{\mathrm{p}}$-vollständige NP-Relation R, sodass für alle ehrlichen Beweissysteme h für* $\mathrm{Proj}(R)$ *auch $h \leq_{\mathrm{m}}^{\mathrm{p}} std_R$ gilt.*

(11') *Es existiert keine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation R, sodass das Standardbeweissystem std_R* P-*optimal ist.*

Beachte dass (10') bzw. (11') nicht die negierte Version von (10) bzw. (11) ist.

4.3 Bekannte Implikationen und Orakel, offene Trennungen

Im letzten Abschnitt dieses Kapitels werden wir nun die in Abbildung 7 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 (vergrößertem) Pudlákschen Programm durch ein Orakel separiert sind, und welche Separationen noch offen sind.

Zunächst führen wir noch eine abgeschwächte Variante von Q ein, die von Fenner, Fortnow, Naik und Rogers (2003) in ihrer Untersuchung zur Hypothese Q vorgeschlagen wurde.

Vermutung 4.26 (Q′, Fenner, Fortnow, Naik und Rogers, 2003)**.** *Für jede totale NPTM N existiert eine Funktion $g \in \mathrm{FP}$, $\mathrm{img}(g) \subseteq \{0,1\}$ sodass für alle x das Bild $g(x) \in \{0,1\}$ das erste Bit eines akzeptierenden Rechenwegs von $N(x)$ ist.*

Die Hypothese Q′ lässt eine einfache komplexitätstheoretische Charakterisierung zu: es gilt Q′ genau dann wenn alle disjunkten coNP-Paare P-separierbar sind.

Nun können wir die in Abbildung 7 abgebildeten Implikationen und Äquivalenzen nachweisen.

Satz 4.27. *Es gelten die in Abbildung 7 abgebildeten Implikationen und Äquivalenzen.*

Beweis. Es gelten die notierten Äquivalenzen:

1. ¬Q $\Leftrightarrow$ $\exists$ optimales Beweissystem was nicht P-optimal ist $\Leftrightarrow$ TFNP $\not\subseteq_c$ FP, nach Satz 4.24.
2. ¬Q′ $\Leftrightarrow$ $\exists$ P-inseparierbares DisjCoNP-Paar, nach Fortnow und Rogers (1993, Lemma 2.12, vgl. Appendix; vgl auch Fenner, Fortnow, Naik und Rogers, 2003, Thm. 4).
3. $\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P} \Leftrightarrow \mathrm{NPSV_t} \not\subseteq \mathrm{FP}$, nach Fenner, Fortnow, Naik und Rogers (2003, Prop. 1).
4. $\mathrm{UP} \neq \mathrm{P} \Leftrightarrow \exists$ Einwegfunktionen, nach Grollmann und Selman (1988, Thm. 10).
5. $\mathrm{UP} \cap \mathrm{coUP} \neq \mathrm{P} \Leftrightarrow \exists$ Einwegpermutationen, nach Homan und Thakur (2003).

6. NP ∩ coNP ⇔ NPSV_t hat keine vollständige Funktion, nach Beyersdorff, Köbler und Messner (2009, Prop. 3).
7. DisjNP ⇔ NPSV hat keine vollständige Funktion, nach Glaßer, Selman und Sengupta (2005, Thm. 9).

Es gelten die eingezeichneten Implikationen:

1. DisjNP ⇒ $\mathsf{TAUT}^{\mathsf{N}}$ nach Köbler, Messner und Torán (2003, Cor. 6.1).
2. UP ⇒ TAUT nach Köbler, Messner und Torán (2003, Cor. 4.1).
3. TAUT ⇒ EE ≠ NEE, $\mathsf{TAUT}^{\mathsf{N}}$ ⇒ NEE ≠ coNEE nach Köbler, Messner und Torán (2003, Cor. 7.1).
4. NP ∩ coNP ≠ P ⇒ $\neg\mathsf{Q}'$ ⇒ $\mathrm{NPMV}_t \not\subseteq_c \mathrm{TFNP}$ ⇒ ¬Q nach Fenner, Fortnow, Naik und Rogers (2003, Prop. 9, Thm. 6).
5. E ≠ NE ⇒ ∃ NP-Relation, welche nicht auf Entscheidung reduzierbar ist, nach Impagliazzo und Sudan (1991), vgl. Satz 3.11.
6. UP ≠ P ⇒ ∃ P-inseparierbares DisjNP-Paar, nach Grollmann und Selman (1988, Thm. 5).
7. NP ∩ coNP ⇒ TAUT ∨ SAT nach Köbler, Messner und Torán (2003, Cor. 5.1).
8. NPMV_t hat keine vollständige Funktion ⇒ 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_t hat keine vollständige Funktion ⇒ NP ≠ coNP nach Satz 4.28.
10. $\mathsf{SAT}^{\mathsf{q}}$ ⇒ SAT, $\mathsf{SAT}^{\mathsf{q}}$ ⇒ KvL, nach Korollar 4.14.
11. KvL ⇒ ¬Q, nach Korollar 4.8.
12. ¬Q ⇒ ∃ NP-Relation, welche nicht auf Entscheidung reduzierbar ist, denn wenn ¬Q, dann gilt mit Satz 4.24 auch $\mathrm{TFNP} \not\subseteq_c \mathrm{FP}$, heißt es existiert NP-Relation $R \in \mathrm{TFNP}$ und $R \notin_c \mathrm{FP} = \mathrm{FP}^{\mathrm{Proj}(R)}$.
13. DisjCoNP ⇒ TFNP ⇒ NPMV_t hat keine vollständig Funktion, nach Pudlák (2017, Prop. 5.6, 5.10).
14. NP ∩ coNP ≠ P ⇒ ∃ P-inseparierbares DisjNP-Paar, denn wenn alle DisjNP-Paare P-separierbar wären, dann ist auch für jede Menge $L \in \mathrm{NP} \cap \mathrm{coNP}$ jeweils das DisjNP-Paar $(L, \overline{L})$ P-separierbar und damit $L \in \mathrm{P}$.
15. DisjNP ⇒ ∃ P-inseparierbares DisjNP-Paar; ist klar, denn wenn alle DisjNP-Paare P-separierbar wären, dann wären auch alle Paare trivialerweise $\leq_m^{pp}$-vollständig.
16. DisjCoNP ⇒ ∃ P-inseparierbares DisjCoNP-Paar; ist aus selben Gründen klar.
17. $\mathsf{TAUT}^{\mathsf{N}}$ ⇒ TAUT klar, weil aus P-Optimalität auch Optimalität folgt.
18. SAT ⇒ ¬Q klar: wenn Q, dann ist nach Satz 4.24 jedes optimale Beweissystem auch P-optimal. Dann gilt auch ¬SAT: jede Menge $L \in \mathrm{NP}$ hat ein optimales Beweissystem h (Beobachtung 2.14) und das ist nach Voraussetzung P-optimal.
19. UP ⇒ UP ≠ P klar.
20. NP ∩ coNP ⇒ NP ∩ coNP ≠ P klar.
21. NP ∩ coNP ⇒ NP ≠ coNP klar, denn wenn NP = coNP dann ist NP ∩ coNP = NP und damit existiert auch eine vollständige Menge.
22. ∃ P-inseparierbares DisjNP-Paar ⇒ P ≠ NP klar.
23. UP ∩ coUP ≠ P ⇒ UP ≠ P, UP ∩ coUP ≠ P ⇒ NP ∩ coNP ≠ P klar mittels Translationstechnik.
24. NEE ≠ coNEE ⇒ NE ≠ coNE ⇒ NP ≠ coNP ⇒ P ≠ NP klar mittels Translationstechnik.
25. NEE ≠ coNEE ⇒ EE ≠ NEE ⇒ E ≠ NE klar. □

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

Satz 4.28. *Wenn* NP $=$ coNP *dann existiert eine* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-vollständige Multifunktion* f *für* NPMV_t.

Beweis. Betrachte die Menge

$$U \stackrel{\text{df}}{=} \{(T, x, 1^n) \mid T \text{ ist NTM-Transduktor und für jeden Rechenweg } \alpha \text{ von } T(x) \text{ der Länge} \leq n \text{ gilt: } \alpha \text{ ist nicht terminierend oder } \alpha \text{ terminiert ablehnend}\}.$$

Nach Definition ist klar, dass $U \in$ coNP: $(T, x, 1^n) \notin U$ wenn ein Rechenweg α der Länge $\leq n$ existiert sodass $T(x)$ auf α akzeptiert. Nach Voraussetzung gilt also $U \in$ NP und es existiert eine NPTM N_U welche U entscheidet, und zwar in Laufzeit $q(|(T, x, 1^n)|)$ für geeignetes Polynom q.

Zur Erinnerung: wir sagen für eine NTM N, dass ein Rechenweg α einer Berechnung $N(x)$ ein *terminierender* Rechenweg ist, wenn kein weiterer legaler Rechenschritt (bzgl. N) mehr an α angehängt werden kann. Der Rechenweg α akzeptiert, wenn α terminiert und in der letzten Konfiguration in einem akzeptierendem Zustand steht.

Betrachte nun die Multifunktion f, die durch folgenden nichtdeterministischen Transduktor $F(T, x, 1^n)$ berechnet wird:

1 **wenn** T kein NTM-Transduktor ist **dann**
2 Gebe ε aus
3 Rate nichtdeterministisch einen Rechenweg α von $T(x)$ der Länge $\leq n$
4 Rate nichtdeterministisch einen Rechenweg β von $N_U(T, x, 1^n)$ der Länge $\leq q(|(T, x, 1^n)|)$
5 **wenn** α hat Länge n und α nicht terminiert **dann**
6 Gebe ε aus
7 **sonst wenn** $T(x)$ mit α akzeptiert **dann**
8 $y \leftarrow$ Ausgabe von $T(x)$ auf α
9 Gebe y aus
10 **sonst wenn** $N_U(T, x, 1^n)$ mit β akzeptiert **dann**
11 Gebe ε aus
12 **sonst**
13 Lehne ab

Es ist leicht zu sehen dass F in Polynomialzeit arbeitet. Wir bestimmen nun die Outputs von F in Abhängigkeit der Eingabe $(T, x, 1^n)$. Wir untersuchen hierfür drei Fälle:

1. Es existiert ein Rechenweg von $T(x)$ mit $> n$ Schritten. In diesem Fall existiert dann auch Rechenweg α mit genau n Schritten, welcher nicht terminiert. Damit terminiert F auf mindestens einem Rechenweg in Z. 6 und wir haben $\varepsilon \in$ *set-f*$(T, x, 1^n)$.
2. Jeder Rechenweg von $T(x)$ hat nur $\leq n$ viele Schritte, und *set-T*$(x) = \emptyset$. Dann wird jeder Rechenweg der Länge n von $T(x)$ terminieren, heißt F wird definitiv nicht in Z. 6 eine Ausgabe tätigen. Gleichzeitig wird kein Rechenweg der Länge $\leq n$ von $T(x)$ akzeptieren, heißt F wird auch nicht in Z. 9 eine Ausgabe tätigen. Zusätzlich gilt $(T, x, 1^n) \in U$, heißt F wird auf mindestens einem Rechenweg in Z. 11 mit Ausgabe ε akzeptieren. Es gilt also *set-f*$(T, x, 1^n) = \{\varepsilon\}$.
3. Jeder Rechenweg von $T(x)$ hat nur $\leq n$ viele Schritte, und *set-T*$(x) \neq \emptyset$. Wieder gilt, dass jeder Rechenweg der Länge n von $T(x)$ terminiert, heißt F wird definitiv nicht in Z. 6 eine Ausgabe tätigen. Gleichzeitig existiert für jedes $y \in$ *set-T*(x) ein

akzeptierender Rechenweg α_y mit $\leq n$ Schritten, welcher auf $T(x)$ den Output y macht. Dieser ist insbesondere terminierend. Damit wird auch F dieses y in Z. 9 ausgeben. Zusätzlich gilt $(T, x, 1^n) \notin U$, heißt Z. 11 wird daher niemals erreicht. Es gilt also $\textit{set-f}(T, x, 1^n) = \textit{set-T}(x)$.

Damit ist klar, dass $f \in \mathrm{NPMV}_t$. Wir zeigen nun, dass f auch NPMV_t-vollständig ist. Sei hierfür g eine beliebige Multifunktion aus NPMV_t. Dann existiert auch ein NPTM-Transduktor T_g sodass dieser g berechnet, und dabei terminiert $T_g(x)$ in $\leq p(|x|)$ vielen Schritten für geeignetes Polynom p.

Nun gilt nach obiger Beobachtung schon dass

$$\textit{set-g}(x) = \textit{set-T}_g(x) \neq \emptyset \implies \textit{set-f}(\underbrace{T_g, x, 1^{p(|x|)}}_{h(x)}) = \textit{set-T}(x) = \textit{set-g}(x)$$

und $h(x) = (T_g, x, 1^{p(|x|)})$ realisiert die Reduktion $g \leq^{\mathrm{p}}_{\mathrm{m}} f$, wie gewünscht. □

Orakeltrennungen

Abbildung 8 zeigt, welche Hypothesen unter relativierbaren Beweisen durch ein Orakel getrennt sind. Beachte, dass im Gegensatz zur Abbildung 7 aus Übersichtlichkeit einige Hypothesen ausgelassen wurden, nämlich jene der Exponentialzeitklassen, und jene der Reduzierbarkeit von Such- auf Entscheidungsprobleme. Das blau hervorgehobene Orakel O wird im folgenden Kapitel 5 konstruiert, vgl. Satz 5.1.

Für viele Paare von Hypothesen A, B haben wir damit entweder eine relativierende Implikation A ⇒ B oder es existiert ein Orakel welches A und B trennt, also ein Orakel relativ zu diesem A ∧ ¬B gilt, und damit einen relativierenden Beweis für diese Implikation ausschließt.

Die Stellung von Q sei hier hervorgehoben: mittels Orakelkonstruktionen erkennen wir eine weitestgehende Unabhängigkeit zwischen der Hypothese ¬Q auf der einen Seite und Hypothesen des Pudlákschen Programms auf der anderen Seite. Zum einen wissen wir über ein Orakel von Fenner, Fortnow, Naik und Rogers (2003, Thm. 12.3, Nr. 8 in der Abb.), dass die Annahme von UP ≠ P ∧ NP ≠ coNP unter relativierenden Beweisen nicht ausreicht, um ¬Q zu zeigen, also diese Annahme auch nicht ausreicht um KvL zu zeigen. Auf ähnliche Weise ergibt sich aus dem in Kapitel 5 konstruierten Orakel (O in der Abb.), dass auch die Annahme DisjNP ∧ UP nicht ausreicht, um ¬Q zu zeigen. Symmetrisch sehen wir über zwei Orakel von Dose (2020a, Cor. 3.3, 2020c, Thm. 3.2, siehe Nr. 5 u. 6 in der Abb.) dass die Annahme ¬Q (bzw. sogar die stärkere Annahme ¬Q′) nicht ausreicht, um TAUT bzw. SAT zu zeigen, also auch nicht um die stärkeren ursprünglichen Pudlákschen Hypothesen (DisjNP, UP, DisjCoNP, usw.) zu beweisen.

Teile dieser Beobachtung übertragen sich entsprechend auch auf die Verstärkung KvL von ¬Q. Weder UP ≠ P ∧ NP ≠ coNP noch DisjNP ∧ UP reicht als Annahme aus, um relativierend KvL zu beweisen. Damit wissen wir zumindest schon, dass KvL (wieder unter relativierenden Beweisen) nicht äquivalent zu einer der Aussagen „UP ≠ P“, NP ≠ coNP“, DisjNP, UP sein kann. Dennoch wären einige weitere Orakel wünschenswert, um KvL besser zu situieren, z.B. ein Orakel, relativ zu diesem ¬Q ∧ ¬KvL gilt (und damit KvL von ¬Q trennt)?

In diesem Sinne können wir auch allgemein fragen, für welche Paare A, B von Hypothesen sowohl unbekannt ist, ob $A \Rightarrow B$ über einen relativierbaren Beweis, noch ob ein Orakel existiert relativ zu diesem A ∧ ¬B. Das sei im Folgenden nun abschließend noch zusammengefasst (vgl. Tabelle 1).

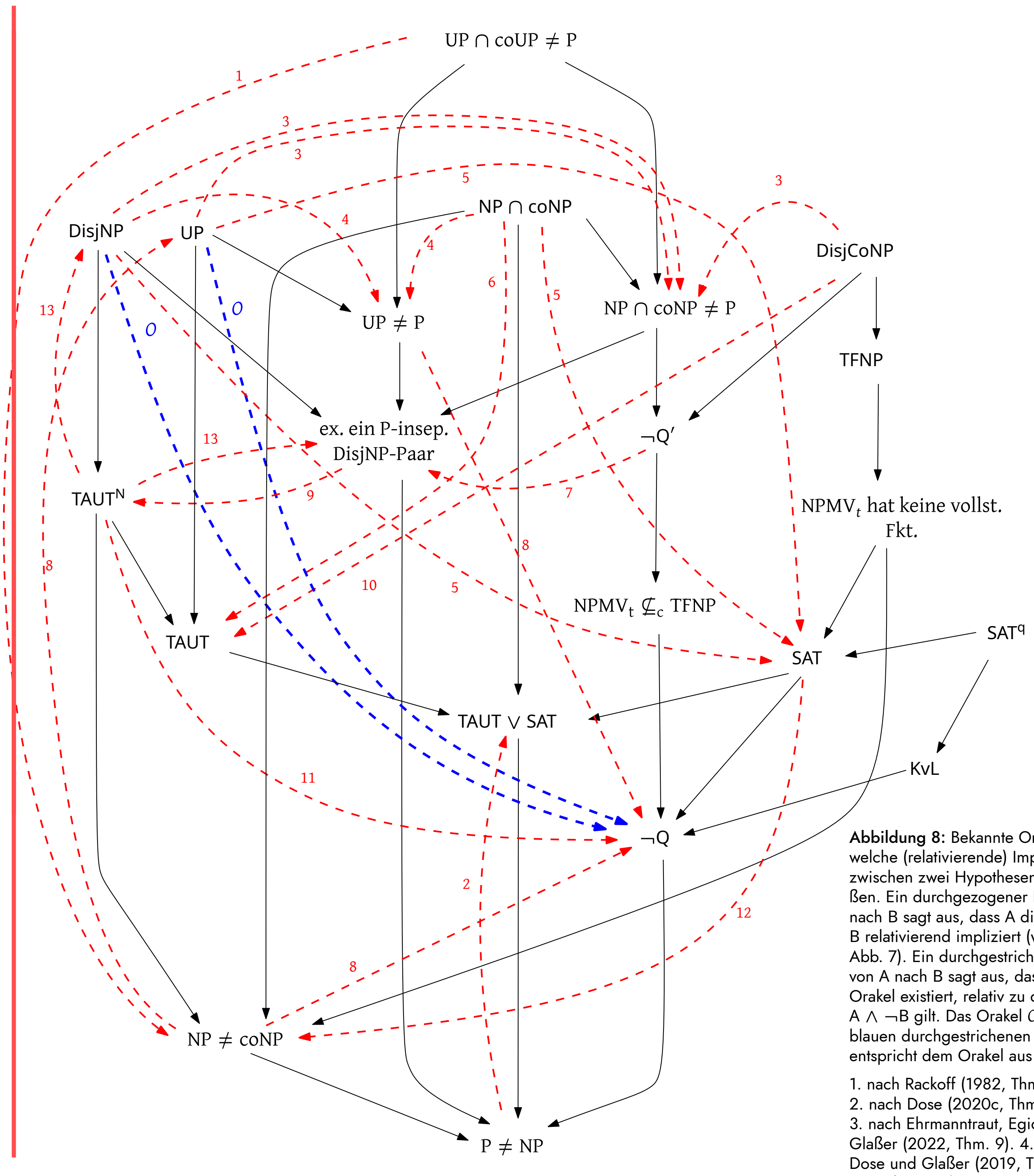

Abbildung 8: Bekannte Orakel, welche (relativierende) Implikationen zwischen zwei Hypothesen ausschließen. Ein durchgezogener Pfeil von A nach B sagt aus, dass A die Aussage B relativierend impliziert (wie in Abb. 7). Ein durchgestrichener Pfeil von A nach B sagt aus, dass ein Orakel existiert, relativ zu diesem $A \wedge \neg B$ gilt. Das Orakel O an den blauen durchgestrichenen Pfeilen entspricht dem Orakel aus Satz 5.1.

1. nach Rackoff (1982, Thm. 5).
2. nach Dose (2020c, Thm. 4.1).
3. nach Ehrmanntraut, Egidy und Glaßer (2022, Thm. 9). 4. nach Dose und Glaßer (2019, Thm. 4.1).
5. nach Dose (2020a, Cor. 3.3).
6. nach Dose (2020c, Thm. 3.2).
7. nach Fortnow und Rogers (2002, Thm. 3.2). 8. nach Fenner, Fortnow, Naik und Rogers (2003, Thm. 12.3).
9. nach Glaßer, Selman, Sengupta und Zhang (2004, Cor. 6.6).
10. nach Khaniki (2022, Thm. 5.1).
11. nach Khaniki (2022, Thm. 5.2).
12. nach Dingel (2022, Satz 3.12).
13. nach Glaßer, Selman, Sengupta und Zhang (2004, Cor. 6.34).

- Unter dem ursprünglichen Hypothesen des Pudlákschen Programms (SAT, TAUT, $\mathsf{TAUT}^{\mathsf{N}}$, Vollständigkeit von DisjNP, DisjCoNP, UP, NP∩coNP, TFNP, NPMV_t, Kollaps NP = coNP, NP ∩ coNP = P; die zusammengesetzte Hypothese SAT ∨ TAUT wird hier dagegen noch nicht diskutiert) kennen wir für fast alle Paare an Hypothesen eine relativierende Implikation oder ein entsprechendes Orakel, relativ zu diesem die Implikation nicht gilt. Offen bleiben nur diese neun Paare:

  (i) $\mathsf{TAUT} \overset{?}{\Rightarrow} \mathsf{TAUT}^{\mathsf{N}}$,
  (ii) $\mathsf{TAUT} \overset{?}{\Rightarrow} \mathrm{NP} \neq \mathrm{coNP}$,
  (iii) $\mathsf{UP} \overset{?}{\Rightarrow} \mathsf{TAUT}^{\mathsf{N}}$,
  (iv) $\mathsf{UP} \overset{?}{\Rightarrow} \mathsf{DisjNP}$,
  (v) $\mathsf{UP} \overset{?}{\Rightarrow} \mathrm{NP} \neq \mathrm{coNP}$,
  (vi) „NPMV_t hat keine vollst. Fkt.“ $\overset{?}{\Rightarrow}$ TFNP,
  (vii) $\mathsf{TFNP} \overset{?}{\Rightarrow} \mathsf{DisjCoNP}$,
  (viii) „NPMV_t hat keine vollst. Fkt.“ $\overset{?}{\Rightarrow}$ DisjCoNP.

  Ein Orakel für (v) wäre insbesondere auch ein Orakel für (i)–(vi); ein Orakel für (vi) oder (vii) wäre insbesondere auch ein Orakel für (viii).

- Erweitern wir den Blick um Q und um die verwandten Hypothesen Q′, „$\mathrm{NPMV}_t \not\subseteq_c \mathrm{TFNP}$“, so entstehen einige neue Paare A, B von Hypothesen, für die unbekannt ist, ob ein Orakel diese trennt, oder ob ein Beweis einer relativierbare Implikation existiert, z.B. $\mathsf{TFNP} \overset{?}{\Rightarrow} \neg\mathsf{Q}'$. Beachte aber, dass für jedes dieser offenen Paare A, B (mit A oder B in {Q, Q′, „$\mathrm{NPMV}_t \not\subseteq_c \mathrm{TFNP}$“}) die Konstruktion eines Orakels O mit $\mathsf{A} \wedge \neg\mathsf{B}$ höchstwahrscheinlich sehr schwierig ist: für jedes offene Paar A, B lässt sich verifizieren, dass relativ zu einem trennenden Orakel O auch $\mathsf{Q}' \wedge \neg\mathsf{Q}$ gilt. Damit trennt O insbesondere Q und Q′ unter relativierenden Beweisen, und würde eine seit 28 Jahren offene Frage von Fenner, Fortnow, Naik und Rogers (2003, vgl. auch 1996) beantworten. Das entspricht genau jenen offenen Orakelkonstruktionen, die in Tabelle 1 mit † markiert sind.

- Ergänzen wir weiter mit dem Kollaps „UP = P“ und der P-Separierbarkeit von DisjNP-Paaren entstehen weiter neue offenen Paare A, B von Hypothesen, unter anderem

  (ix) $\mathsf{DisjCoNP} \overset{?}{\Rightarrow}$ „DisjNP inseparierbar“,
  (x) $\mathrm{UP} \neq \mathrm{P} \overset{?}{\Rightarrow} \mathsf{TAUT}$

  Dies sind die zwei „stärksten“ Implikationen bzw. diejenigen Paare an Hypothesen, deren Orakelkonstruktionen gegen diese Implikationen am „schwierigsten“ sind. Gemeint ist, dass sämtliche anderen offenen Paare A, B von Hypothesen, wobei A oder B in {„UP ≠ P“, „DisjNP insep.“}, dann auch durch eines dieser Orakel getrennt wird, die (ix) und (x) trennen.

- Ergänzen wir nun abschließend mit den hier neu definierten Hypothesen KvL und $\mathsf{SAT}^{\mathsf{q}}$ entstehen wieder neue Paare A, B für die offen ist, ob A die Hypothese B impliziert, oder ob ein Orakel gegen diese Implikation existiert. Das gilt für so gut wie alle möglichen Paare zwischen KvL bzw. $\mathsf{SAT}^{\mathsf{q}}$ und den übrigen betrachteten

Hypothesen. Besonders im Hinblick auf den Schwerpunkt dieser Arbeit auf Suchprobleme und auf die Hypothese KvL sind die folgenden Paare interessant:

(xi) $\mathsf{KvL} \overset{?}{\Rightarrow} \mathsf{SAT}^{\mathsf{q}}$ (oder schwächer SAT),
(xii) $\mathsf{KvL} \overset{?}{\Rightarrow} \mathsf{TAUT}$,
(xiii) $\mathsf{DisjCoNP} \overset{?}{\Rightarrow} \mathsf{KvL}$ (oder schwächer $\neg\mathsf{Q} \overset{?}{\Rightarrow} \mathsf{KvL}$).

Diese Fragen werden im Folgenden nicht weiter untersucht. Stattdessen seien sie hier als zukünftige Forschungsdesiderate formuliert: zeige dass eine der obigen Implikationen gilt, gebe ein Gegenbeispiel an, oder konstruiere ein Orakel relativ zu diesem eine der obigen Implikationen nicht gilt.

- Schließlich gehen wir noch auf die zusammengesetzte Hypothese $\mathsf{TAUT} \vee \mathsf{SAT}$ ein. Zur Erinnerung: diese Hypothese besagt (in Verbindung mit Korollar 2.17), dass eine Menge $L \in \mathrm{NP} \cup \mathrm{coNP}$ existiert für die kein P-optimales Beweissystem existiert. Trotz der zusammengesetzten Natur dieser Hypothese lässt sich zeigen, dass $\mathsf{TAUT} \vee \mathsf{SAT}$ äquivalent zu weiteren natürlichen Hypothesen ist. Zum einen zeigt Khaniki (2022, Thm. 3.2) die Äquivalenz zur Hypothese RFN_1 betreffend der Beweisbarkeit endlicher Widerspruchsfreiheit (vgl. Pudlák, 2017). Zum anderen geben Egidy, Glaßer und Herold (2023) eine Verbesserung der ersten genannten Charakterisierung an, hierbei bezogen auf die Mengen der Booleschen Hierarchie (BH, vgl. Cai und Hemachandra, 1986; Cai, Gundermann u. a., 1988, 1989). Aufbauend auf Ergebnissen von Köbler, Messner und Torán (2003) ergibt sich, dass $\mathsf{TAUT} \vee \mathsf{SAT}$ genau dann gilt, wenn sogar eine Menge $L \in \mathrm{BH} \supseteq \mathrm{NP} \cup \mathrm{coNP}$ ohne P-optimales Beweissystem existiert.

  Auf diese beiden Charakterisierungen soll hier aber nicht weiter eingegangen werden. Trotzdem sei hier noch knapp die Beziehung der Hypothese $\mathsf{TAUT} \vee \mathsf{SAT}$ zu den anderen (unter relativierbaren Beweisen) erläutert. Einerseits existieren Orakel, sodass $(\mathsf{TAUT} \vee \mathsf{SAT}) \wedge \neg\mathsf{A}$ für alle bisher genannten Hypothesen A (außer $\mathrm{P} \neq \mathrm{NP}$, was trivialerweise eine notwendige Bedingung ist) . Andererseits bleibt für viele Hypothesen noch offen, ob diese hinreichend für $\mathsf{TAUT} \vee \mathsf{SAT}$ sein könnten. Insbesondere sind folgende Paare noch offen:

  (xiv) $\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P} \overset{?}{\Rightarrow} \mathsf{TAUT} \vee \mathsf{SAT}$,
  (xv) $\mathsf{KvL} \overset{?}{\Rightarrow} \mathsf{TAUT} \vee \mathsf{SAT}$.

Hiermit wollen wir die Diskussion über die Beziehungen der Hypothesen des erweiterten Pudláksche Programms abschließen. Im folgenden Kapitel werden wir nun noch das angekündigte Orakel O konstruieren.

Antezedens A \ Konsequent B	TAUT	$\mathrm{TAUT}^{\mathrm{N}}$	DisjNP	UP	SAT	$\mathrm{NPMV_t}$ unvollst.	TFNP	DisjCoNP	$\mathrm{NP} \cap \mathrm{coNP}$	$\mathrm{NP} \neq \mathrm{coNP}$	$\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P}$	$\neg\mathrm{Q}$	$\neg\mathrm{Q}'$	$\mathrm{NPMV_t} \not\subseteq_t \mathrm{TFNP}$	$\mathrm{UP} \neq \mathrm{P}$	DisjNP unsep.	KvL	$\mathrm{SAT}^{\mathrm{q}}$	$\mathrm{TAUT} \vee \mathrm{SAT}$
TAUT		?	13	4	O	O	O	O	O	?	O	O	O	O	4	13	O	O	
$\mathrm{TAUT}^{\mathrm{N}}$			13	4	O	O	O	O	O		O	O	O	O	4	***13***	O	O	
DisjNP				4	O	O	O	O	O		O	***O***	O	O	***4***		O	O	
UP		?	?		O	O	O	O	O	?	O	***O***	O	O			O	O	
SAT	10	10	10	10		12	12	12	3	***12***	3		†	†	?	?	?	?	
$\mathrm{NPMV_t}$ unvollst.	10	10	10	10			?	?	3		3		†	†	?	?	?	?	
TFNP	10	10	10	10				?	3		3		†	†	?	?	?	?	
DisjCoNP	***10***	10	10	10					3		***3***				?	?	?	?	
$\mathrm{NP} \cap \mathrm{coNP}$	***6***	6	6	4	***5***	5	5	5							***4***		?	5	
$\mathrm{NP} \neq \mathrm{coNP}$	6	6	6	4	O	O	O	O	O		O	O	O	O	4	13	O	O	?
$\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P}$	6	1	1	4	5	1	1	1	1	1					4		?	5	?
$\neg\mathrm{Q}$	6	1	1	4	5	1	1	1	1	1	3		†	†	4	7	?	5	?
$\neg\mathrm{Q}'$	6	1	1	4	5	1	1	1	1	1	3				4	***7***	?	5	?
$\mathrm{NPMV_t} \not\subseteq_t \mathrm{TFNP}$	6	1	1	4	5	1	1	1	1	1	3		†		4	7	?	5	?
$\mathrm{UP} \neq \mathrm{P}$	?	1	1	?	O	O	O	O	O	1	O	O	O	O			O	O	?
DisjNP unsep.	6	1	1	4	O	O	O	O	O	1	O	O	O	O	4		O	O	?
KvL	?	?	?	?	?	?	?	?	?	?	?		†	†	?	?		?	?
$\mathrm{SAT}^{\mathrm{q}}$	?	?	?	?		?	?	?	?	?	?		†	†	?	?			
$\mathrm{TAUT} \vee \mathrm{SAT}$	6	6	6	4	O	O	O	O	O	12	O	O	O	O	4	13	O	O	

Tabelle 1: Überblick über Orakel, welche Implikationen $\mathrm{A} \Rightarrow \mathrm{B}$ zwischen ausgewählten Hypothesen unter relativierbaren Beweisen trennen, in dem Sinn dass ein Orakel existiert relativ zu diesem $\mathrm{A} \wedge \neg\mathrm{B}$ gilt. Jede Zelle entspricht hierbei einer solchen Implikation. Die Hypothese A links ist hierbei der Antezedens, die Hypothese B oben der Konsequent.

Leere Zellen bedeuten, dass kein Orakel mit $\mathrm{A} \wedge \neg\mathrm{B}$ existiert, denn es existiert ein relativierender Beweis für $\mathrm{A} \Rightarrow \mathrm{B}$.

Eine Zahl (bzw. *O*) in der Zelle bedeutet, dass relativ zu einem Orakel $\mathrm{A} \wedge \neg\mathrm{B}$ gilt, also ein Orakel gegen die Implikation $\mathrm{A} \Rightarrow \mathrm{B}$. Die Zahl gibt hierbei an, um welches Orakel es sich aus Abb. 8 handelt, das Label *O*, dass es sich um das konstruierte Orakel aus Kapitel 5 handelt. Ist die Zahl (bzw. *O*) fett gedruckt, dann entspricht das Orakel genau dem Eingezeichneten, ansonsten folgt die behauptete Eigenschaft aus relativierbaren Implikationen zwischen den Hypothesen.

Ein rotes ? bedeutet, dass unbekannt ist, ob ein solches Orakel existiert.

Ein rotes † bedeutet, dass auch unbekannt ist, ob ein solches Orakel D existiert. Hierbei ist aber die Konstruktion besonders schwierig: wenn nämlich ein solches Orakel existiert, also $\mathrm{A} \wedge \neg\mathrm{B}$ relativ zu D gilt, dann muss auch $\mathrm{Q}' \wedge \neg\mathrm{Q}$ relativ zu D gelten.

5 Orakel mit DisjNP, UP und Q

Ziel dieses Kapitels ist die Konstruktion eines Orakels O, relativ zu dem einerseits die Hypothesen DisjNP und UP gelten, andererseits auch die Hypothese Q gilt. Hieraus ergibt sich das Resultat, dass $\neg$Q nicht mit relativierenden Mitteln beweisen werden kann, selbst unter Annahme der starken, aber wahrscheinlichen Annahme DisjNP $\wedge$ UP. Dies verbessert auch ein Ergebnis von Dose (2020a, Cor. 3.3), welcher ein ähnliches Orakel konstruiert, relativ zu diesem DisjNP $\wedge$ UP $\wedge$ $\neg$SAT gilt. Wir werden folgenden präzisen Satz beweisen:

Satz 5.1. *Es existiert ein Orakel O sodass folgende Aussagen gelten:*

(1) Für jede totale NPTM N existiert eine Funktion $g \in \mathrm{FP}^O$ sodass $g(x)$ ein akzeptierender Rechenweg von $N^O(x)$ ist (was äquivalent zu Q *relativ zu O ist).*

(2) Es existiert kein $\leq_\mathrm{m}^{\mathrm{pp},O}$-hartes Paar in DisjNP^O *für* DisjUP^O *(was* DisjNP *relativ zu O impliziert).*

(3) Es existiert keine $\leq_\mathrm{m}^{\mathrm{p},O}$-vollständige Menge für UP^O *(was äquivalent zu* UP *relativ zu O ist).*

Die Orakelkonstruktion bedient sich zwei zentralen Ideen. Die erste Idee betrifft vor allem die Konstruktion. Wir wenden hierbei ein Verfahren von Dose und Glaßer (2019) an, welche sich als ein starkes Framework erwiesen hat, schrittweise Orakel zu konstruieren, die simultan mehrere Eigenschaften erfüllen. Dieses Framework baut maßgeblich auf drei Zutaten auf: erstens, partiell definierten Orakeln, zweitens, ein Begriff von Gültigkeit, und drittens, eine stufenweise nicht-konstruktive Definitionsvorschrift des Orakels. Das wird später beim Einsatz in der Konstruktion näher erläutert.

Die zweite Idee besteht intuitiv darin, von einem PSPACE-vollständigem Orakel D auszugehen und dieses zu modifizieren, um die gewünschten Eigenschaften zu erhalten. Durch geschicktes Rückgreifen auf D lassen sich dann gewisse nichttriviale Berechnungen effizient umsetzen. Diese Idee geht auf Baker, Gill und Solovay (1975) zurück, die ein Orakel konstruiert haben, relativ zu diesem $\mathrm{P} = \mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{NP}$. Hierbei wird das PSPACE-vollständige Orakel D mit einem zweiten, speziell konstruierten Orakel B kombiniert; in der Originalfassung zu $D \cup B$. Das Orakel B sorgt für die Trennung zwischen P und NP, ist aber gleichzeitig so einfach zu entscheiden, dass ein komplementär akzeptierendes Maschinenpaar einer Sprache $L \in \mathrm{NP} \cap \mathrm{coNP}$ eine relevante Portion von B entscheiden kann, und dann mittels dem PSPACE-vollständige Orakel D dann abfragen kann, ob $x \in L$ gilt.

Wir werden im Folgenden etwas Ähnliches mittels *Relativierungen* von Orakelkonstruktionen umsetzen. In der Literatur hat es sich als dienlich gezeigt, anstelle mit einem konkreten PSPACE-vollständigem Orakel zu starten und zu ergänzen, stattdessen mit der *Annahme* $\mathrm{P} = \mathrm{PSPACE}$ zu starten. (Dieses Vorgehen findet sich zum Beispiel bei Blum und Impagliazzo, 1987; Fortnow und Rogers, 2002; Fenner, Fortnow, Kurtz und Li, 2003.) Wir können dann mit den gewohnten Methoden ein Orakel E konstruieren, relativ zu diesem die oberen Eigenschaften gelten. Im Verlauf dieses Kapitels zeigen wir die folgende Implikation:

Satz 5.2. *Angenommen* $\mathrm{P} = \mathrm{PSPACE}$. *Dann existiert ein Orakel E sodass folgende Aussagen gelten:*

(1) *Für jede totale NPTM N existiert eine Funktion $g \in \mathrm{FP}^E$ sodass $g(x)$ ein akzeptierender Rechenweg von $N^E(x)$ ist (was äquivalent zu* Q *relativ zu E ist).*

(2) *Es existiert kein $\leq_{\mathrm{m}}^{\mathrm{pp},E}$-hartes Paar in* DisjNP^E *für* DisjUP^E *(was* DisjNP *relativ zu E impliziert).*

(3) *Es existiert keine $\leq_{\mathrm{m}}^{\mathrm{p},E}$-vollständige Menge für* UP^E *(was äquivalent zu* UP *relativ zu E ist).*

Die Konstruktion von E ist dabei solcherart, dass diese Konstruktion auf ein zweites Orakel relativiert. Intuitiv gemeint ist damit, dass wir der Aussage ein zweites Orakel hinzufügen können, ohne die Gültigkeit der Implikation zu verlieren. Hieraus ergibt sich insbesondere, dass für jedes Orakel X folgende Implikation gilt:

$$\mathrm{P}^X = \mathrm{PSPACE}^X \overset{\text{5.2, rel.}}{\Longrightarrow} \text{ex. } E \text{ sodass } E \oplus X \text{ den Eig. von Satz 5.1 genügt.}$$

Diese Implikation ist geeignet, um Satz 5.1 zu beweisen. Mit Wahl von X als dem PSPACE-vollständigen D erreichen wir wie gewünscht, dass $O = E \oplus D$ den Eigenschaften von Satz 5.1 genügt. Diese Relativierungen von Orakelkonstruktionen werden im Verlauf dieses Kapitels noch formal präzise spezifiziert.

Die Hauptarbeit liegt also in der Konstruktion eines Orakels E aus Satz 5.2. Die zentrale Idee sei hier kurz skizziert: Wir starten mit einem leeren Orakel. Für die Aussage UP werden wir eine Familie von Zeugensprachen $C_m \in \mathrm{UP}^O$ definieren, welche sensitiv gegenüber dem Inhalt des Orakels E sind. Damit lassen sich dann Wörter so zu E hinzufügen, dass für jede UP^E-Maschine und jeden FP^E-Transduktor die Reduktion auf eine entsprechende Zeugensprache $C_m \in \mathrm{UP}^E$ ausgeschlossen wird. Das läuft über ein klassisches Diagonalargument, ähnlich wie die $\mathrm{P} \neq \mathrm{NP}$-Konstruktion von Baker, Gill und Solovay (1975), und kann analog für die Aussage DisjNP umgesetzt werden.

Nun zur Aussage Q. Unter Annahme von $\mathrm{P} = \mathrm{PSPACE}$ gilt trivialerweise Q relativ zum leeren Orakel. Hätten wir nun die Möglichkeit, effizient E auszurechnen, hätten wir auch Q relativ zu E. Das ist zwar nicht möglich, aber für jede totale NPTM N ist es zumindest möglich, eine relevante Portion von E zu berechnen, sodass sich diese Portion in N hineincodieren lässt. Wir erhalten dann eine äquivalente NPTM N' welche keine Orakelfragen stellen muss. Unter Annahme $\mathrm{P} = \mathrm{PSPACE}$ lässt sich dann ein akzeptierender Rechenweg für N' (bzw. N) finden.

Im folgenden Abschnitt 5.1 erarbeiten wir detailliert die Relativierungen von Orakelkonstruktionen. Anschließend spezifizieren wir in Abschnitt 5.2 noch einige spezielle Notationen, die wir zur Orakelkonstruktion benötigen. Aufbauend auf der obigen Skizze können wir dann in Abschnitt 5.3 die konkrete Konstruktionsvorschrift des Orakels formulieren. In Abschnitt 5.4 wird abschließend nachgewiesen, dass diese Konstruktionsvorschrift zum einen wohldefiniert ist, und zum anderen, dass sich hieraus die gewünschten Eigenschaften aus Satz 5.2 ergeben.

5.1 Relativierende Orakelkonstruktionen

Um den Begriff der relativierenden Orakelkonstruktion zu definieren, generalisieren wir zunächst den Begriff der (1-)Orakel-Turing-Maschine M^A auf 2-Orakel-Turing-Maschinen $M^{A,B}$. Anstelle eines Fragezustands haben 2-Orakel-TM zwei ausgezeichnete Fragezustände $q_{q?}$ und $q_{2?}$. Im Zustand $q_{1?}$ wird getestet ob der Inhalt w des Orakelbands in A liegt (und entsprechend zu q_{yes} oder q_{no} übergegangen), und analog in Zustand $q_{2?}$ getestet, ob w in B liegt. Damit rechnet M relativ dem Orakel-Paar (A, B). Definitionen und Aussagen können damit nicht nur, wie bisher, auf *ein* Orakel relativiert werden, sondern auch auf zwei Orakel (A, B) relativiert werden. Zum Beispiel meint $\mathsf{Q}^{A,B}$, dass zu jedem 2-Orakel-NPTM N, welcher total relativ zu (A, B) arbeitet, auch ein 2-Orakel-PTM-Transduktor existiert, der auf Eingabe x relativ zu (A, B) einen akzeptierenden Rechenweg der Berechnung $N^{A,B}(x)$ ausgibt.

Die zentrale Einsicht bei den üblichen Orakelkonstruktionen ist nun, dass diese *relativieren*. Das Argument ist das gleiche, wie bei der Relativierung der üblichen Beweismethoden. Während der Konstruktion von Orakeln wird nicht in die diskutierten TM „hineingeschaut“, sondern werden die Orakel nur anhand des Akzeptierverhaltens einiger TM definiert. Es lässt sich also in solchen relativierenden Orakelkonstruktionen jede 1-Orakel-TM durch eine 2-Orakel-TM austauschen, wobei das hinzugefügte Orakel fest, aber frei wählbar ist. Die Konstruktion selbst ist ja „blind“ gegenüber dem zweiten Orakel. Hätten wir also beispielsweise durch eine relativierende Orakelkonstruktion von A die Aussage Q^A, dann gilt auch $\mathsf{Q}^{A,X}$ für jedes Orakel X.

Eine solches Paar (A, B) von Orakeln können wir dann wieder zurück in *ein* äquivalentes Orakel $A \oplus B \subseteq \Sigma^*$ umwandeln. Zur Erinnerung:

$$A \oplus B = \{0a \mid a \in A\} \cup \{1b \mid b \in B\}.$$

Es ist leicht zu sehen, dass sich zu jeder 2-Orakel-(N)PTM N sich eine 1-Orakel-(N)PTM N' angeben lassen kann, sodass sich für alle A, B die TM N relativ zu (A, B) äquivalent verhält wie N' zu $A \oplus B$, und umgekehrt. In diesem Sinne hätten wie mit obigen Beispiel, dass auch $\mathsf{Q}^{A \oplus X}$ für alle X gilt.

Vor diesem Hintergrund können wir noch einmal den Plan der nun folgenden Orakelkonstruktion angeben. Wir zeigen erstens die Implikation von Satz 5.2 bzw.

$$\mathrm{P}^{\emptyset} = \mathrm{PSPACE}^{\emptyset} \implies \text{ex. } E \text{ sodass } E \text{ den Eig. von Satz 5.1 genügt.}$$

Diese Konstruktion relativiert insbesondere. Wir können daher auf beliebige Orakel X relativieren:

$$\mathrm{P}^{\emptyset,X} = \mathrm{PSPACE}^{\emptyset,X} \implies \text{ex. } E \text{ sodass } (E, X) \text{ den relativierten Eig. von Satz 5.1 genügt.}$$

Wählen wir nun ein PSPACE-vollständiges D als X, dann gilt trivialerweise die Voraussetzung $\mathrm{P}^{\emptyset,D} = \mathrm{PSPACE}^{\emptyset,D}$. Dann existiert also auch ein E, sodass (E, D) den Eigenschaften von Satz 5.1 genügt, wobei die Eigenschaften auf 2-Orakel-TM relativiert wurden. Wie bereits diskutiert, erfüllt $E \oplus D$ dann auch diese Eigenschaften, relativiert auf 1-Orakel-TM. Setzen wir also $O = E \oplus D$ sind wir fertig und haben Satz 5.1 bewiesen.

5.2 Notation zur Orakelkonstruktion

Nun definieren wir noch einige Notationen, die wir bei der Orakelkonstruktion, und insbesondere für den Beweis von Satz 5.2 benötigen.

Zunächst seien $\{M_i\}_{i\in\mathbb{N}}, \{N_i\}_{i\in\mathbb{N}}, \{F_i\}_{i\in\mathbb{N}}$ sogenannte *Standardaufzählungen* der Orakel-

PTMs, Orakel-NPTMs, bzw. Orakel-PTM-Tranduktoren, welche folgende Eigenschaften haben

1. Für jedes Orakel D terminiert $M_i^D(x)$ nach höchstens $p_i(|x|) \stackrel{\text{df}}{=} |x|^i + i$ Schritten; analog für N_i und F_i.

2. Für jede Orakel-PTM M existiert ein i sodass $L(M_i^D) = L(M^D)$, für jede Orakel-NPTM N existiert ein i sodass $L(N_i^D) = L(N^D)$, und für jeden Orakel-PTM-Transduktor F existiert ein i sodass $F^D(x) = F_i^D(x)$.

   Insbesondere kann jeder akzeptierender Rechenweg α der Berechnung $N_i(x)$ in einen akzeptierenden Rechenweg α' der Berechnung $N(x)$ effizient übersetzt werden.

Solche Standardaufzählungen existieren. Eine Möglichkeit, diese zu konstruieren, ist beispielsweise, jede nichtdeterministische TM mit einem Timer auszustatten, die nach polynomieller Laufzeit die Berechnung abbricht. Konkret: gegeben i, sei zunächst N die nichtdeterministische TM mit Codierung i. Die TM N_i führt dann auf Eingabe x parallel zur Rechnung $N(x)$ auf jedem Rechenweg einen Timer aus, der nach $> |x|^i + i$ Schritten den Rechenweg ablehnend abbricht. Damit ist die Laufzeit von N_i polynomiell beschränkt auf p_i.

Wie bereits angesprochen, werden wir ein Konstruktionsverfahren von Dose und Glaßer anwenden. Die erste wesentliche Zutat dieses Frameworks ist der Begriff von „partiell definierten“ Orakeln, bei denen also für gewisse x noch nicht endgültig festgelegt ist, ob $x \in O$ oder $x \notin O$ gelten soll. Diese werden mittels finiten Wörtern $w \in \Sigma^*$ formalisiert: ein finites Wort $w \in \Sigma^*$ können wir im Folgenden auch als die Menge $\{i \mid i < |w|, w[i] = 1\}$ verstehen (aber $|w|$ immer als die Länge von w). Die intendierte Interpretation ist, dass gegenüber der *Menge* w die Zugehörigkeit aller Zahlen x (bzw. äquivalent Wörtern) mit $x < |w|$ final spezifiziert ist, nicht dagegen für $x \geq |w|$ (und nur ersatzweise $x \notin w$ gilt).

Diese Interpretation von finiten Wörtern w als Orakel macht es einfacher, unsere Orakelkonstruktionen präzise und knapp zu beschreiben. Üblicherweise werden wir w so erweitern, dass die Zugehörigkeit des kleinsten $x \in \mathbb{N}$ spezifiziert wird, welche noch nicht final spezifiziert ist. Dieses x ist genau $|w|$ und wir legen die Zugehörigkeit final fest, indem wir an w entweder 0 anhängen (und x ist final nicht im Orakel $w0$) oder 1 anhängen (und x ist final im Orakel $w1$).

Wir können TMs relativ zu partiell definierten Orakeln auch rechnen lassen. Für $w \in \Sigma^*$ definieren wir $M^w(x)$ entsprechend als $M^{\{i|w(i)=1\}}(x)$ (heißt, Orakelanfragen, für die w nicht definiert ist, werden negativ beantwortet). Dies ermöglicht es uns auch, folgende Schreibweise zu definieren: Wir sagen, dass (N)TM $M^w(x)$ *definit* ist, wenn für alle Anfragen q auf allen Rechenwegen $q < |w|$ gilt (oder äquivalent: $w[q]$ ist für alle Anfragen q auf allen Rechenwegen definiert); wir sagen, dass $M^w(x)$ *definitiv akzeptiert* (bzw. *definitiv ablehnt*), wenn $M^w(x)$ definit ist und akzeptiert (bzw. ablehnt). Intuitiv beschreibt der Begriff „definit“ Berechnungen, für die alle Anfragen durch das Orakel schon final spezifiziert sind, die sich also nicht ändern, wenn das jeweilige partiell definierte Orakel erweitert wird.

Beobachtung 5.3. *Sei $w \in \Sigma^*$ und $v \in \Sigma^* \cup \Sigma^\omega$, und N eine O(N)TM.*

(1) *Wenn $M^w(x)$ eine definite Berechnung ist und $v \sqsupseteq w$ ist, dann ist $M^v(x)$ definit. Die Berechnung $M^v(x)$ akzeptiert genau dann, wenn $M^w(x)$ akzeptiert.*

(2) *Wenn w für alle Wörter der Länge $p_i(|x|) = |x|^i + i$ definiert ist, dann ist $M_i^w(x)$, $N_i^w(x)$, $F_i^w(x)$ definit.*

(3) *Wenn $M^w(x)$ auf einem Rechenweg mit der Menge Q an Orakelanfragen akzeptiert, und w, v auf Q übereinstimmen, dann akzeptiert $M^v(x)$ auf dem gleichen Rechenweg und mit der gleichen Menge an Anfragen Q.*

Für ein partielles Orakel w, einen PTM-Transduktor F und eine (N)PTM M schreiben wir manchmal $M^w(F^w(x))$ als die *eine* Berechnung der (N)PTM $M \circ F$ auf Eingabe x relativ zu w. Entsprechend sagen wir dann auch, dass $M^w(F^w(x))$ definit ist (bzw. definitiv akzeptiert, oder definitiv ablehnt) wenn $M \circ F$ definit auf Eingabe x relativ zu w ist (bzw. definitiv akzeptiert, definitiv ablehnt). Sollte es aus dem Kontext klar hervorgehen, lassen wir üblicherweise auch den Zusatz „partiell" weg, wenn wir z.B. von einem (partiellen) Orakel $w \in \Sigma^*$ sprechen.

Vor der Konstruktion machen wir nun noch folgende bekannte kombinatorische Aussage:

Lemma 5.4. *Sei G ein gerichteter bipartiter Graph mit den Knotenmengen A und B. Das heißt, jede Kante in G führt entweder von einem Knoten in A zu einem Knoten in B oder umgekehrt. Sei Δ eine obere Schranke für den Ausgangsgrad jedes Knotens in G.*

Wenn $|A|, |B| > 2\Delta$ gilt, dann gibt es ein $a \in A$ und $b \in B$, sodass weder (a, b) noch (b, a) eine Kante in G ist.

Beweis. Sei $n = \min\{|A|, |B|\} > 2\Delta$. Entferne Knoten aus A und B, bis beide Knotenmengen jeweils genau n Knoten haben, um einen gerichteten bipartiten Graphen G' mit den Knotenmengen A' und B' zu bilden. Sei G'' der zugrunde liegende ungerichtete Graph von G'. In G' gibt es $\leq |A'| \cdot \Delta + |B'| \cdot \Delta < n^2$ viele ungerichtete Kanten, aber es gibt n^2 viele ungerichtete Kanten im vollständigen bipartiten ungerichteten Graphen $K_{n,n}$.

Das bedeutet, dass es $a \in A' \subseteq A$, $b \in B' \subseteq B$ gibt, die in G'' nicht adjazent sind; damit sind sowohl $(a, b) \notin E(G')$ als auch $(b, a) \notin E(G')$ im induzierten gerichteten bipartiten Teilgraphen G'. Also gilt für den ursprünglichen Graphen G sowohl $(a, b) \notin E(G)$ als auch $(b, a) \notin E(G)$, wie gewünscht. $\square$

5.3 Definition

Nun werden wir den Beweis von Satz 5.2 erarbeiten. Dafür starten wir mit der Annahme P = PSPACE.

Wir möchten in unseren Orakelkonstruktion abzählbar unendlich viele *Ebenen* n, das heißt, Wörter gleicher Länge n, für eine abzählbar unendliche Familie von Zeugensprachen mit zunehmend großen Lücken injektiv reservieren und zuordnen. Hierfür sei $e(0) \overset{\text{df}}{=} 2$, $e(i) \overset{\text{df}}{=} 2^{e(i-1)}$. Es gibt eine polynomialzeit-berechenbare, polynomialzeit-invertierbare injektive Funktion f, die von $N \times \mathbb{N}$ nach $\mathbb{N}$ abbildet. Definiere nun $H_m \overset{\text{df}}{=} \{e(f(m, h)) \mid h \in \mathbb{N}\}$ als die Menge der für die Zeugensprache m reservierten Ebenen. Diese Definition stellt nun Folgendes sicher:

Beobachtung 5.5. *(1) Die Menge H_m ist abzählbar unendlich, eine Teilmenge der geraden Zahlen, und alle $H_0, H_1, \dots$ sind paarweise disjunkt.*

(2) Die Folge $\min H_0, \min H_1, \dots$ ist nach oben unbegrenzt.

(3) Wenn $n \in H_m$, dann gilt für jedes $a \in \mathbb{N}$: $n < a < 2^n \implies a \notin H_0, H_1, \dots$.

(4) Jede Menge $H_m \in \mathrm{P}$ für alle $m \in \mathbb{N}$.

Wie bereits angesprochen, werden wir in der Konstruktion Wörter der Länge $n \in H_m$ in das Orakel E einsetzen, sodass E die gewünschten Eigenschaften hat. Wörter anderer Länge werden nicht in das Orakel eingesetzt. Wir definieren folgende Zeugensprachen, welche vom Inhalt der Ebenen abhängig sind. Sei hierfür $m \in \mathbb{N}$ und $w \in \Sigma^* \cup \Sigma^\omega$ beliebig:

$$
\begin{aligned}
A_m^w &\stackrel{\text{df}}{=} \{0^n \mid n \in H_m, \text{es existiert } x \in \Sigma^n \text{ mit } x \in w \text{ und } x \text{ endet mit } 0\} \\
B_m^w &\stackrel{\text{df}}{=} \{0^n \mid n \in H_m, \text{es existiert } x \in \Sigma^n \text{ mit } x \in w \text{ und } x \text{ endet mit } 1\} \\
C_m^w &\stackrel{\text{df}}{=} \{0^n \mid n \in H_m, \text{es existiert } x \in \Sigma^n \text{ mit } x \in w\}
\end{aligned}
$$

Sind die Ebenen der Höhe H_m in w auf geeignete Weise gefüllt, lässt sich leicht sehen, dass diese entsprechenden Zeugensprachen in DisjUP^w bzw. UP^w fallen:

Behauptung 5.6. *Sei $w \in \Sigma^* \cup \Sigma^\omega$ ein beliebiges Orakel.*

(1) Wenn $|w \cap \Sigma^n| \le 1$ für alle $n \in H_m$, dann $(A_m^w, B_m^w) \in \mathrm{DisjUP}^w$.

(2) Wenn $|w \cap \Sigma^n| \le 1$ für alle $n \in H_m$, dann $C_m^w \in \mathrm{UP}^w$.

Idee und Vorschau der Konstruktion

Die Konstruktion verläuft im Wesentlichen so, dass simultan und ineinander verflochten die drei Eigenschaften von Satz 5.2 erarbeitet werden:

1. Erarbeitung von (1), bzw. Q: Für alle $j \in \mathbb{N}$ versucht die Konstruktion die Ebenen so einzurichten, dass N_j nicht total ist. Falls dies nicht möglich ist, dann wird N_j inhärent total akzeptieren. Diese Eigenschaft können wir dann algorithmisch ausnutzen, um eine relevante Portion von E zu bestimmen, und, zusammen mit der Voraussetzung $\mathrm{P} = \mathrm{PSPACE}$ so einen akzeptierenden Rechenweg für $N_j(x)$ bestimmen.
2. Erarbeitung von (2), bzw. DisjNP: Für alle $a \neq b$ versucht die Konstruktion die Ebenen so einzurichten, dass N_a, N_b beide eine Eingabe x akzeptieren, womit $x \in L(N_a) \cap L(N_b)$ und damit $(L(N_a), L(N_b)) \notin \mathrm{DisjNP}$. Falls dies nicht möglich ist, ist (N_a, N_b) inhärent ein disjunktes NP-Paar. In diesem Fall fixieren wir ein m, stellen sicher, dass (A_m, B_m) ein disjunktes UP-Paar ist, und diagonalisieren so gegen jeden PTM-Transduktor F_r, sodass F_r die Reduktion $(A_m, B_m) \le_{\mathrm{m}}^{\mathrm{pp}} (L(N_a), L(N_b))$ nicht realisiert. Dies wird folgendermaßen erreicht: (i) Für alle $n \in H_m$ fügen wir höchstens ein Wort der Länge n in E ein (und somit $(A_m, B_m) \in \mathrm{DisjUP}$), und (ii) für jedes r gibt es ein $n \in H_m$ so, dass $0^n \in A_m$, aber $N_a(F_r(0^n))$ ablehnt (oder analog $0^n \in B_m$, aber $N_b(F_r(0^n))$ ablehnt).
3. Erarbeitung von (3), bzw. UP: Für alle a versucht die Konstruktion die Ebenen so einzurichten, dass N_a eine Eingabe x auf zwei Rechenwegen akzeptiert, womit $L(N_a) \notin \mathrm{UP}$. Falls dies nicht möglich ist, $L(N_a)$ inhärent eine UP-Sprache. In diesem Fall fixieren wir ein m, stellen sicher, dass C_m eine Sprache in UP ist, und diagonalisieren gegen jeden PTM-Transduktor F_r, sodass F_r die Reduktion $C_m \le_{\mathrm{m}}^{\mathrm{pp}} L(N_a)$

nicht realisiert. Dies wird folgendermaßen erreicht: (i) Für alle $n \in H_m$ fügen wir höchstens ein Wort der Länge n in E ein (und somit $C_m \in \mathrm{DisjUP}$), und (ii) für jedes r gibt es ein $n \in H_m$ so, dass $0^n \in C_m$ genau dann wenn $N_a(F_r(0^n))$ ablehnt.

Wir weisen diesen Arbeitsschritten folgende Symbole zu, welche die einzelnen Tasks repräsentieren sollen: $\tau_j^1, \tau_{a,b}^2, \tau_{a,b,r}^2, \tau_a^3, \tau_{a,r}^3$. Symbol τ_j^1 repräsentiert den (versuchsweisen) Ausschluss der Totalität von N_j. Symbol $\tau_{a,b}^2$ repräsentiert analog den Ausschluss der Disjunktheit von $L(N_a), L(N_b)$, Symbol $\tau_{a,b,r}^2$ dann die Diagonalisierung dieses Paares gegen den Transduktor F_r. Analog für UP und $\tau_a^3, \tau_{a,r}^3$.

Intuitiv gesprochen verläuft die Konstruktion nun wie folgt: wir starten mit dem leeren partiell definierten Orakel ε, und wollen dieses nun schrittweise erweitern. In jeder Erweiterung von w_s zu $w_{s+1} \sqsupsetneq w_s$ nehmen wir uns einen noch nicht bearbeiteten Task τ, und werden die Erweiterung w_{s+1} so wählen, dass τ relativ zu dieser Erweiterung erfüllt wird. Wir passen dabei auf, dass auch in zukünftigen Erweiterungen $w_{s+2}, w_{s+3}, \dots$ die Bearbeitung von τ erhalten bleibt. Führen wir das nun unendlich oft durch, enden wir im Limit mit einem ω-langen Orakel, welcher alle unsere obigen Tasks abgearbeitet hat.

Gültigkeit von partiellen Orakeln

Beachte, wie in der oberen Vorschau einige Tasks τ bestimmte Voraussetzungen an das finale Orakel stellen. Kann zum Beispiel im Fall von Task τ_a^3 nicht erwirkt werden, dass die NPTM N_a auf mehr als einem Rechenweg akzeptiert, so muss in der Konstruktion dafür gesorgt werden, dass auch die Zeugensprache C_m in UP liegt, indem selbst in den Erweiterungen nach Bearbeiten von Task τ_a^3 für alle $n \in H_m$ höchstens ein Wort der Länge n in E eingesetzt wird.

Solche „globalen“ Voraussetzungen behandelt das Framework von Dose und Glaßer durch den Begriff der *Gültigkeit* von partiellen Orakeln. Dies ist die zweite wesentliche Zutat. Hierbei werden alle solche Eigenschaften, die im Verlauf der Konstruktion aufrecht erhalten müssen, in einer partiellen Funktion t zusammengefasst. Wenn ein Orakel $w \in \Sigma^*$ dieses Bündel t an Eigenschaften erfüllt, dann nennen wir dies t-gültig. Gültigkeit ist also eine binäre Relation zwischen den partiellen Orakeln aus Σ^* und partiellen Funktionen.

Für unsere Zwecke definieren wir uns folgende Funktionenklasse $\mathcal{T}$. Die Klasse $\mathcal{T}$ ist eine Teilmenge der partiellen Funktionen, und es gilt $t \in \mathcal{T}$ genau dann wenn

- der Definitionsbereich nur aus Tasks $\tau_j^1, \tau_{a,b}^2, \tau_a^3$ besteht,
- die Bildmenge nur aus natürlichen Zahlen besteht, und
- t injektiv auf den positiven Funktionswerten ist, also $t(\tau) = t(\tau') > 0 \implies \tau = \tau'$.

Die intendierte Interpretation ist nun wie folgt: wenn t so gewählt ist dass $t(\tau_a^3) = m > 0$, dann soll damit notiert werden dass Task τ_a^3 die Ebenen der Menge H_m für die Zeugensprache C_m reserviert hat, also in jede diese Ebenen der Wortlänge n nur höchstens ein Wort der Länge n eingesetzt werden darf. Analog für $\tau_{a,b}^2$. Die Inkeitivität sichert, dass die Tasks paarweise disjunkte Ebenenmengen zugewiesen bekommen.

Wir setzen nun für diese Konstruktion folgenden Begriff von Gültigkeit fest. Ein Orakel $w \in \Sigma^*$ ist t-gültig wenn $t \in \mathcal{T}$ und Folgendes gilt:

V1 Wenn $x < |w|$ und $|x| \notin \mathrm{img}(e)$, dann gilt $x \notin w$.

(Bedeutung: Orakel w enthält keine Wörter der Länge $\neq e(\cdot)$.)

V2 Für alle i gilt $|w \cap \Sigma^{e(i)}| \leq 2$.

(Bedeutung: Orakel w ist dünn auf den Ebenen der Länge $e(\cdot)$.)

V3 Wenn $t(\tau_j^1) = 0$, dann existiert ein z sodass $N_j^w(z)$ definitiv ablehnt.

($L(N_j) \neq \Sigma^*$ relativ zum finalen Orakel.)

V4 Wenn $t(\tau_{a,b}^2) = 0$, dann existiert ein z sodass $M_a^w(z)$ und $M_b^w(z)$ definitiv akzeptieren.

(Bedeutung: wenn $t(\tau_{a,b}^2) = 0$, dann $L(M_a) \cap L(M_b) \neq \emptyset$ relativ zum finalen Orakel.)

V5 Wenn $0 < t(\tau_{a,b}^2) = m$, dann gilt für alle $n \in H_m$ dass $|\Sigma^n \cap w| \leq 1$.

(Bedeutung: wenn $0 < t(\tau_{a,b}^2) = m$, dann $(A_m, B_m) \in$ DisjNP relativ zum finalen Orakel.)

V6 Wenn $t(\tau_a^3) = 0$, dann existiert ein z sodass $M_a^w(z)$ definitiv auf zwei Rechenwegen akzeptiert.

(Bedeutung: wenn $t(\tau_a^3) = 0$, dann $L(M_a) \notin$ UP relativ zum finalen Orakel.)

V7 Wenn $0 < t(\tau_a^3) = m$, dann gilt für alle $n \in H_m$ dass $|\Sigma^n \cap w| \leq 1$.

(Bedeutung: wenn $0 < t(\tau_a^3) = m$, dann $C_m \in$ UP relativ zum finalen Orakel.)

Induktive Definition des Orakels

Sei T eine abzählbare Aufzählung der oben genannten Tasks, sodass $\tau_{a,b,r}^2$ immer nach $\tau_{a,b}^2$ kommt, sowie $\tau_{a,r}^3$ immer nach τ_a^3 kommt. Die Orakelkonstruktion erfolgt, wie bereits skizziert, nun in abzählbar unendlich vielen *Stufen*. In jeder Stufe bearbeiten wir den kleinsten Task τ in der durch T festgelegten Reihenfolge. Anschließend entfernen wir τ aus T, und entfernen möglicherweise zusätzlich höhere Tasks aus T. In der nächsten Stufe fahren wir mit dem nächsten Task fort, die noch nicht aus T entfernt wurde. (In jeder Stufe existiert immer mindestens ein Task, die noch nicht entfernt wurde, da wir in keiner Stufe alle Tasks aus T entfernen werden.)

Eine Stufe $s \in \mathbb{N}$ identifizieren wir hierbei mit einem Orakel $w_s \in \Sigma^*$, einer Gültigkeits-Funktion $t_s \in \mathcal{T}$, und einem Task (bzw. mehreren Tasks) welcher in dieser Stufe bearbeitet wurde. Wir werden die einzelnen $w_0, w_1, w_2, \dots$ und $t_0, t_1, t_2, \dots$ so definieren, dass

$$w_0 \sqsubsetneq w_1 \sqsubsetneq w_2 \sqsubsetneq \cdots, \quad \text{und} \quad t_0 \subseteq t_1 \subseteq t_2 \subseteq \cdots,$$

heißt insbesondere, dass t_j eine Fortsetzung von t_i ist, wenn immer $j \geq i$. Außerdem werden wir sichern, dass für jedes $s \in \mathbb{N}$ das partielle Orakel w_s ein t_s-gültiges Orakel ist. Es ist klar, dass jeder Task $\tau \in T$ letztlich in irgendeiner Stufe s bearbeitet wird.

Nun zur Definition von w_s, t_s: wir werden die einzelnen Stufen induktiv definieren. Als Basisklausel setzen wir $w_0 = \varepsilon$ und $t_0 = \emptyset$. Für unsere induktive Klausel sei $s > 0$. Die Definition für w_s, t_s ergibt sich nun aus der bereits definierten Funktion $t_{s-1} \in \mathcal{T}$, und dem bereits definierten t_{s-1}-gültigen Orakel w_{s-1}, sowie dem zu bearbeitenden kleinsten verbleibenden Task τ in T. Zur Erinnerung: dieser wird unmittelbar nach der Bearbeitung aus T entfernt. In der Bearbeitung wird das Orakel strikt verlängert. Es gibt nun fünf Fälle, je nach dem welche Form der bearbeitete Task τ hat.

Task τ_j^1: Setze $t' \stackrel{\text{df}}{=} t_{s-1} \cup \{\tau_j^1 \mapsto 0\}$. Existiert ein t'-gültiges Orakel $v \sqsupsetneq w_{s-1}$, dann setze $t_s \stackrel{\text{df}}{=} t'$ und $w_s \stackrel{\text{df}}{=} v$.

Ansonsten setze $t_s \stackrel{\text{df}}{=} t_{s-1}$ und setze $w_s \stackrel{\text{df}}{=} w_{s-1}0$. Nach Behauptung 5.7 ist w_s dann t_s-gültig.

(Bedeutung: wenn das Orakel w_s so eingerichtet werden kann, dass N_j nicht mehr total arbeitet, dann erweitere genau so, vgl. V2.)

Task $\tau^2_{a,b}$: Setze $t' \stackrel{\mathrm{df}}{=} t_{s-1} \cup \{\tau^2_{a,b} \mapsto 0\}$. Existiert ein t'-gültiges Orakel $v \sqsupsetneq w_{s-1}$, dann setze $t_s \stackrel{\mathrm{df}}{=} t'$ und $w_s \stackrel{\mathrm{df}}{=} v$. Entferne außerdem alle Tasks der Form $\tau^2_{a,b,r}$ von T.

Ansonsten wähle ein hinreichend großes $m \notin \mathrm{img}(t_{s-1})$ sodass w_s kein Wort der Länge $\geq \min(H_m)$ definiert. Setze $t_s \stackrel{\mathrm{df}}{=} t_{s-1} \cup \{\tau^2_{a,b} \mapsto m\}$; damit ist w_{s-1} auch t_s-gültig. Setze $w_s \stackrel{\mathrm{df}}{=} w_{s-1}0$. Nach Behauptung 5.7 ist w_s dann t_s-gültig.

(Bedeutung: wenn das Orakel w_s so eingerichtet werden kann, dass N_a, N_b nicht mehr disjunkt akzeptieren, dann erweitere genau so, vgl. V4. Ansonsten, falls das nicht möglich ist, wähle ein geeignetes frisches m und stelle ab dieser Stufe sicher, dass $(A_m, B_m) \in \mathrm{DisjUP}$.)

Task $\tau^2_{a,b,r}$: Wir wissen dass $t_{s-1}(\tau^2_{a,b}) = m > 0$. Setze $t_s = t_{s-1}$ und wähle ein t_s-gültiges Orakel $w_s \sqsupsetneq w_{s-1}$ sodass bezüglich einem $n \in \mathbb{N}$ eine der folgenden Aussagen gilt:

- $0^n \in A^v_m$ für alle $v \sqsupseteq w_s$ und $M_a(F_r(0^n))$ lehnt relativ zu w_s definitiv ab.
- $0^n \in B^v_m$ für alle $v \sqsupseteq w_s$ und $M_b(F_r(0^n))$ lehnt relativ zu w_s definitiv ab.

Das ist möglich nach Behauptung 5.8.

(Bedeutung: erweitere zu w_s, sodass F_r nicht die Reduktion $(A_m, B_m) \leq^{\mathrm{pp}}_{\mathrm{m}} (L(N_a), L(N_b))$ realisiert.)

Task τ^3_a: Setze $t' \stackrel{\mathrm{df}}{=} t_{s-1} \cup \{\tau^3_a \mapsto 0\}$. Existiert ein t'-gültiges Orakel $v \sqsupsetneq w_{s-1}$, dann setze $t_s \stackrel{\mathrm{df}}{=} t'$ und $w_s \stackrel{\mathrm{df}}{=} v$. Entferne außerdem alle Tasks der Form $\tau^3_{a,r}$ von T.

Ansonsten wähle ein hinreichend großes $m \notin \mathrm{img}(t_{s-1})$ sodass w_s kein Wort der Länge $\geq \min(H_m)$ definiert. Setze $t_s \stackrel{\mathrm{df}}{=} t_{s-1} \cup \{\tau^3_a \mapsto m\}$; damit ist w_{s-1} auch t_s-gültig. Setze $w_s \stackrel{\mathrm{df}}{=} w_{s-1}0$. Nach Behauptung 5.7 ist w_s dann t_s-gültig.

(Bedeutung: wenn das Orakel w_s so eingerichtet werden kann, dass N_a auf mehr als einem Rechenweg akzeptiert, dann erweitere genau so, vgl. V6. Ansonsten, falls das nicht möglich ist, wähle ein geeignetes frisches m und stelle ab dieser Stufe sicher, dass $(C_m) \in \mathrm{UP}$.)

Task $\tau^3_{a,r}$: Wir wissen dass $t_{s-1}(\tau^3_a) = m > 0$. Setze $t_s \stackrel{\mathrm{df}}{=} t_{s-1}$ und wähle ein t_s-gültiges Orakel $w_s \sqsupsetneq w_{s-1}$ sodass bezüglich einem $n \in \mathbb{N}$ eine der folgenden Aussagen gilt:

- $0^n \in C^v_m$ für alle $v \sqsupseteq w_s$ und $M_a(F_r(0^n))$ lehnt relativ zu w_s definitiv ab.
- $0^n \notin C^v_m$ für alle $v \sqsupseteq w_s$ und $M_a(F_r(0^n))$ akzeptiert relativ zu w_s definitiv.

Das ist möglich nach Behauptung 5.11.

(Bedeutung: erweitere zu w_s, sodass F_r nicht die Reduktion $(C_m) \leq^{\mathrm{p}}_{\mathrm{m}} (L(N_a))$ realisiert.)

Beobachte, dass t_s immer als Element von $\mathcal{T}$ definiert ist. Beachte auch den „Existenzquantor“ in Task τ^1_j (bzw. ähnlich bei $\tau^2_{a,b}, \tau^3_a$). Dieser „testet“ in der Konstruktion, ob überhaupt eine gültige Erweiterung des Orakels existiert, welche die Totalität der NPTM N_j definitiv ausschließt. Falls ja, wird eine solche gültige Erweiterung für das Orakel gewählt. Hierzu zwei Bemerkungen: Erstens handelt es sich hierbei insbesondere um ein nicht-konstruktives Argument. Falls eine solche Erweiterung existieren sollte, dann können wir diese auch auswählen (z.B. indem die lexikographisch kleinste gewählt wird), ohne dass diese in expliziter Weise angegeben wird. Damit ist unser endgültiges Orakel E auch im Übrigen nicht entscheidbar im Sinne der Berechenbarkeitstheorie. Zweitens, falls eine solche gültige Erweiterung nicht existiert, dann ist die NPTM N_j sogar „besonders total“, in dem Sinn dass N_j sogar unter *jeder* gültigen Erweiterung total ist. Diese Konstruktionstaktik ist die dritte Zutat des Verfahrens von Dose und Glaßer.

Mit den oben genannten Tasks ist die die Definition der Stufe s abgeschlossen, und somit die beiden Folgen $\{w_s\}_{s<\omega}$ $\{t_s\}_{s<\omega}$. Wir definieren nun das finale Orakel E als $\bigcup_{s\in\mathbb{N}} w_s = \bigcup_{s\in\mathbb{N}}\{i \mid w_s[i] = 1, i < |w_s|\}$, heißt die Vereinigung über alle partiellen Orakel, oder äquivalent, der Grenzwert $\lim_{n\to\omega} w_s = E \in \Sigma^\omega$ der Folge von finiten Wörtern $w_0 \subsetneq w_1 \subsetneq \cdots$.

Wir müssen nun zwei Aussagen zeigen: erstens, dass diese Konstruktion tatsächlich möglich ist, indem wir die in der Definition angekündigten Lemmata angeben und beweisen. Zweitens müssen wir zeigen, dass die Konstruktion das Gewünschte leistet, also dass E die behaupteten Eigenschaften des Satzes 5.1 erfüllt. Beides werden wir nun im folgenden Abschnitt zeigen.

5.4 Korrektheit

Existenz

Wir zeigen im Folgenden zunächst, dass die oben definierte induktive Definition tatsächlich wohldefiniert ist, in dem Sinne dass jeder Task umgesetzt werden kann und die induktive Definition nicht „abbricht“.

Zunächst beweisen wir, dass sich ein t-gültiges Orakel w immer um ein Bit verlängern lassen kann, ohne Gültigkeit zu verletzen.

Lemma 5.7. *Sei $t \in \mathcal{T}$ und w ein t-gültiges Orakel, und sei $z = |w|$. (Denke z als das kleinste Wort, für welche die Zuordnung zum Orakel noch nicht endgültig feststeht.)*

Dann ist $w0$ auch t-gültig. (Das heißt wir verletzen nicht die Gültigkeit, wenn wir das partielle Orakel so setzen, dass z nicht enthalten ist.)

Beweis. Sei $z = |w|$. Angenommen, $w0$ ist nicht t-gültig, dann muss eine der Bedingungen V1–V7 verletzt sein.

Angenommen V3 ist verletzt, weil $N_j^{w0}(x)$ nicht definit ablehnt und gleichzeitig $t(\tau_j^1) = 0$. Da nach Voraussetzung w aber t-gültig ist, wird $N_j^w(x)$ definitiv ablehnen. Nach Beobachtung 5.3(3) wissen wir aber, dass dann auch $N_j^{w0}(x)$ definitiv ablehnen wird. Widerspruch. Also kann V3 nicht verletzt sein. Mit analoger Argumentation sehen wir auch, dass V4 und V6 nicht verletzt sein können.

Angenommen V1 ist verletzt, weil für $x < |w0|, |x| \notin \mathrm{img}(e)$ die Zugehörigkeit $x \in w$ gilt. Dann kann x nicht $< |w|$ sein, denn V1 gilt hier nach t-Gültigkeit von w. Also muss $x = z$. Nun gilt aber nach Definition $z \notin w$; Widerspruch zur Wahl von x. Also kann V1 nicht verletzt sein.

Es kann also nur V2, V5, oder V7 verletzt sein. Angenommen V5 ist verletzt, weil für $n \in H_m$ gilt $|w0 \cap \Sigma^n| > 1$. Nach Definition wissen wir, dass $w0$ und w auf allen Wörtern übereinstimmen. Das bedeutet dass auch $|w \cap \Sigma^n| > 1$. Das widerspricht der t-Gültigkeit von w. Die Bedingung V5 kann also nicht verletzt sein. Auf analoge Weise wie lässt sich auch zeigen, dass V2 und V7 nicht verletzt sein können.

Insgesamt kann also keine Bedingung V1–V7 verletzt sein; $w0$ ist t-gültig wie gewünscht. □

Damit haben wir die ersten beiden Tasks aus der induktiven Definition schon gesichert. Nun zeigen wir, dass die Bearbeitung von $\tau^2_{a,b,r}$ möglich ist.

Lemma 5.8. *Die Bearbeitung eines Tasks $\tau^2_{a,b,r}$ ist möglich: gilt $t_s = t_{s-1}, t_s(\tau^2_{a,b}) = m > 0$, dann lässt sich w_s so zu t_s-gültigem $w \sqsupsetneq w_s$ erweitern, sodass eine der folgenden Aussagen gilt:*
(1) $0^n \in A^v_m$ für alle $v \sqsupseteq w$ und $M_a(F_r(0^n))$ lehnt relativ zu w definitiv ab.
(2) $0^n \in B^v_m$ für alle $v \sqsupseteq w$ und $M_b(F_r(0^n))$ lehnt relativ zu w definitiv ab.

Skizze. Widerspruchsbeweis. Erweitere w_{s-1} so weit zu u, dass genau alle Wörter der Länge $< n = e(i) \in H_m$ definiert sind, wobei das i hinreichend groß gewählt wird. Sei für jedes $X \subseteq \Sigma^n$ das Orakel $u(X) \sqsupsetneq w_{s-1}$ jenes Orakel was entsteht, wenn die Ebene $e(i)$ mit genau den Wörtern aus X gefüllt wird, heißt $u(X)$ und X stimmen auf Σ^n überein. Beobachte, dass für $|X| \leq 1$ das Orakel $u(X)$ auch t_s-gültig ist.

Nach Annahme gilt
- für $\alpha \in \Sigma^{n-1}0$ gilt $0^n \in A^{u(\{\alpha\})}_m$ und daher akzeptiert $M_a(F_r(0^n))$ relativ zu $u(\{\alpha\})$.
- für $\beta \in \Sigma^{n-1}1$ gilt $0^n \in B^{u(\{\alpha\})}_m$ und daher akzeptiert $M_b(F_r(0^n))$ relativ zu $u(\{\beta\})$.

Kombinatorische Standardmethoden zeigen dann, dass relativ zu $u(\{\alpha, \beta\})$ mit geeignetem $\alpha \in \Sigma^{n-1}0, \beta \in \Sigma^{n-1}1$ sowohl $M_a(F_r(0^n))$ also auch $M_b(F_r(0^n))$ relativ zu $u(\{\alpha, \beta\})$ akzeptieren. Damit wäre aber auch $u(\{\alpha, \beta\})$ ein geeignetes Orakel in der Bearbeitung von Task $\tau^2_{a,b}$ und wir hätten $t_s(\tau^2_{a,b}) = 0$. □

Beweis. Wir fixieren die Werte von a, b und r im gesamten Beweis dieses Satzes.

Sei $\hat{s} < s$ die Stufe, die $\tau^2_{a,b}$ behandelt hat. Eine solche Stufe existiert, da andernfalls $t_s(\tau^2_{a,b})$ undefiniert wäre. Wir haben $m = t_{\hat{s}}(\tau^2_{a,b}) = t_s(\tau^2_{a,b}) > 0$; fixiere auch m für den Rest des Beweises.

Wir nehmen an, dass für alle t_s-gültigen $w \sqsupsetneq w_{s-1}$ weder (1) noch (2) zutrifft. Daraus werden wir einen Widerspruch ableiten, indem wir ein geeignetes Orakel $u' \sqsupsetneq w_{\hat{s}-1}$ konstruieren, das bezüglich $t' \stackrel{\text{df}}{=} t_{\hat{s}-1} \cup \{\tau^2_{a,b} \mapsto 0\}$ gültig ist. (Gemeint ist: relativ zu u' wird M_a und M_b eine Eingabe definitiv akzeptieren.) Dann folgt nach Definition, dass u' eine mögliche t'-gültige Erweiterung von $w_{\hat{s}-1}$ in Stufe $\hat{s}$ ist, daher hätte die Bearbeitung von $\tau^2_{a,b}$ eben $t_{\hat{s}} = t'$ gesetzt, damit auch $t_s(\tau^2_{a,b}) = t'(\tau^2_{a,b}) = 0$, was der Voraussetzung widerspricht.

Sei

$$\gamma(n) \stackrel{\text{df}}{=} \max(p_a(p_r(n)) + p_r(n), p_b(p_r(n)) + p_r(n))$$

das Polynom, welches die Laufzeit von $M_a \circ F_r$ und $M_b \circ F_r$ bezüglich der Eingabelänge n relativ zu einem beliebigen Orakel beschränkt. Daraus folgt: wenn ein partielles Orakel u' für alle Wörter der Länge $\leq \gamma(n)$ definiert ist, dann ist auch $M^{u'}_a(F^{u'}_r(x)), M^{u'}_b(F^{u'}_r(x))$ für alle Eingaben $x \in \Sigma^n$ definit. Sei $n \in \mathbb{N}$ eine hinreichend große Zahl sodass $n \in H_m$, und w_{s-1} keine Wörter der Länge $\geq n$ definiert, und

$$2^n > \gamma(n), \quad 2^{n-1} > 2\gamma(n). \tag{5.1}$$

Die erste Ungleichung von (5.1) stellt sicher, dass keine Ebene $a, n < a \leq \gamma(n)$ für irgendeine Zeugensprache reserviert ist, das heißt, $a \notin H_0, H_1, \ldots$ (vgl. Beobachtung 5.5(3)). Die zweite Ungleichung stellt sicher, dass es genügend Wörter der Länge n gibt, damit bestimmte kombinatorische Argumente funktionieren.

Für den restlichen Beweis fixieren wir zusätzlich n. Wir definieren nun $u \sqsupseteq w_{s-1}$ als ein t_s-gültiges partielles Orakel, das für genau alle Wörter der Länge $< n$ definiert ist.

Ein solches Orakel existiert nach Lemma 5.7, indem man w_{s-1} bitweise erweitert, so dass es t_s-gültig (bzw. identisch t_{s-1}-gültig) bleibt.

Für unseren Beweis betrachten wir nicht alle t_s-gültigen w, sondern vielmehr eine ausreichende Teilmenge davon. Für $X \subseteq \Sigma^n, |X| \leq 2$ definieren wir ein partielles Orakel $u(X) \supsetneq u$ als

$$u(X)[x] \stackrel{\mathrm{df}}{=} \begin{cases} u[x] & \text{falls } |x| < n \\ 1 & \text{falls } |x| = n, x \in X \\ 0 & \text{falls } |x| = n, x \notin X \\ 0 & \text{falls } n < |x| \leq \gamma(n) \\ \perp & \text{sonst,} \end{cases}$$

das für alle Wörter bis zur Länge $\leq \gamma(n)$ definiert ist, wobei $u(X) \cap \Sigma^n = X$. Das heißt, $u(X)$ und X stimmen in Σ^n überein.

Im Wesentlichen entsteht also $u(X)$ aus u, indem die Ebene n mit X gefüllt wird, und dann die höheren Ebenen leer gelassen werden. Es ist leicht zu sehen, dass für $|X| \leq 1$ das Orakel $u(X)$ sogar t_s-gültig ist, indem iterativ u erweitert wird.

Behauptung 5.9. *Sei $|X| \leq 1$. Dann ist $u(X)$ ein t_s-gültiges Orakel.*

Beweis. Angenommen, es ist kein t_s-gültiges Orakel. Dann ist eine der Bedingungen V1–V7 verletzt. Wir zeigen, dass diese Annahme zu einem Widerspruch führt. Klar ist, dass V3, V4, V6 nicht verletzt sein können, denn jede definite Berechnung relativ zu u ist auch eine definite Berechnung relativ zu $u(X)$ nach Beobachtung 5.3(1).

Angenommen, V1 ist verletzt. Dann muss ein x mit $|x| \leq \gamma(n), |x| \neq e(\cdot)$ existieren, sodass $x \in u(X)$. Klar ist, dass $|x| \geq n$ sein muss, da ansonsten schon u nicht t_s-gültig wäre; das wäre ein Widerspruch zur Konstruktion von u. Außerdem ist $n \neq |x|$, da ja $n \in \mathrm{img}(e)$. Also $n < |x| \leq \gamma(n)$. Dann wissen wir aber nach Definition von $u(X)$ dass $x \notin u(X)$; Widerspruch zur Wahl von x.

Angenommen, V5 ist verletzt. Dann existiert ein τ' mit $t_s(\tau') = m' > 0$ und $n' \in H_{m'}$ und $|\Sigma^{n'} \cap u(X)| > 1$. Wieder klar ist, dass $n \leq n' \leq \gamma(n) < 2^n$ sein muss, ansonsten wäre schon u nicht mehr t_s-gültig. Dann aber muss auch $n = n'$, denn ansonsten wäre $n' \notin H_{m'}$, nach Beobachtung 5.5(3). Jetzt haben wir aber $|\Sigma^{n'} \cap u(X)| = |\Sigma^n \cap u(X)| = |X| \leq 1$; das ist ein Widerspruch zur Annahme $|\Sigma^{n'} \cap u(X)| > 1$. Also kann V5 nicht verletzt sein.

Auf analoge Weise ist ersichtlich, dass auch V2 und V7 nicht verletzt sein können. Damit ist keine Bedingung verletzt, $u(X)$ muss also t_s-gültig sein. □

Wir haben angenommen, dass für t_s-gültige Orakel nicht (1) oder (2) gilt. Angewendet auf $u(X), |X| \leq 1$ bedeutet das

- Für jedes $\alpha \in \Sigma^{n-1}0$ gilt $0^n \in A_m^{u(\{\alpha\})}$ und daher akzeptiert $M_a(F_r(0^n))$ definitiv relativ zu $u(\{\alpha\})$ auf einem Rechenweg mit Menge Q_α an Orakelfragen.
- Für jedes $\beta \in \Sigma^{n-1}1$ gilt $0^n \in B_m^{u(\{\alpha\})}$ und daher akzeptiert $M_b(F_r(0^n))$ definitiv relativ zu $u(\{\beta\})$ auf einem Rechenweg mit Menge Q_β an Orakelfragen.

Wir wollen nun ein Orakel u' konstruieren, sodass sowohl M_a als auch M_b akzeptieren. Hierfür wollen wir die beiden jeweiligen akzeptierenden Rechenwege fixieren. Wir stellen das sicher, indem wir u' so wählen, dass u' mit $u(\{\alpha\})$ auf Q_α übereinstimmt, dann wird auch $M_a(F_r(0^n))$ relativ zu u' akzeptieren. Symmetrisch stellen wir das auch für M_b bzw. β sicher, sodass auch $M_a(F_r(0^n))$ relativ zu u' akzeptieren wird.

Hierzu müssen wir $\alpha \in \Sigma^{n-1}0$ und $\beta \in \Sigma^{n-1}1$ finden, die sich nicht gegenseitig „stören“. Ein Wort $\alpha \in \Sigma^{n-1}0$ *stört* $\beta \in \Sigma^{n-1}1$ falls $\alpha \in Q_\beta$, symmetrisch *stört* ein Wort $\beta \in \Sigma^{n-1}0$ ein Wort $\alpha \in \Sigma^{n-1}1$ falls $\beta \in Q_\alpha$.

Es existieren $\alpha \in \Sigma^{n-1}0$ und $\beta \in \Sigma^{n-1}1$ die sich nicht gegenseitig stören. Wir zeigen das über einen bipartiten „Störgraph“ G. Setze

$$A = \Sigma^{n-1}0, \quad B = \Sigma^{n-1}1$$

als die zwei Knotenmengen von G. Setze nun

$$E = \{(\alpha,\beta) \mid \alpha \in \Sigma^{n-1}0, \beta \in \Sigma^{n-1}1, \alpha \in Q_\beta\} \cup \{(\beta,\alpha) \mid \alpha \in \Sigma^{n-1}0, \beta \in \Sigma^{n-1}1, \beta \in Q_\alpha\}$$

als Kantenmenge. Beachte, dass der Ausgangsgrad Δ von G höchstens $\gamma(n)$ sein kann, denn ein Rechenweg auf $M_a(F_r(0^n))$ (bzw. $M_b(\cdots)$) ist auf $\leq \gamma(n)$ viele Schritte beschränkt, heißt auch $|Q_\alpha|, |Q_\beta| \leq \gamma(n)$.

Nach Ungleichung (5.1) wissen wir nun, dass $|A|, |B| > 2\Delta$. Mittels Lemma 5.4 existieren also $\alpha \in A = \Sigma^{n-1}0$, $\beta \in B = \Sigma^{n-1}1$ sodass weder $(a,b) \in E$ noch $(b,a) \in E$. In anderen Worten, es gilt $\alpha \notin Q_\beta$ und $\beta \notin Q_\alpha$.

Betrachte nun $u' \stackrel{\text{df}}{=} u(\{\alpha,\beta\})$. Wir zeigen nun zwei Dinge: erstens, dass $M_a(F_r(0^n))$ und $M_b(F_r(0^n))$ definitiv relativ zu $u(\{\alpha,\beta\})$ akzeptieren. Zweitens anschließend, dass $u(\{\alpha,\beta\})$ auch t'-gültig ist.

Für die erste Aussage beobachten wir, dass $u(\{\alpha\})$ und $u(\{\alpha,\beta\})$ auf Q_α übereinstimmen. Angenommen sie stimmen nicht überein, dann müssen sie nach Konstruktion auf β nicht übereinstimmen. Dann wäre aber $\beta \in Q_\alpha$, was ein Widerspruch zur Wahl von β ist. Symmetrisch sehen wir, dass $u(\{\beta\})$ und $u(\{\alpha,\beta\})$ auf Q_β ist. Also akzeptieren $M_a(F_r(0^n))$ und $M_b(F_r(0^n))$ definitiv relativ zu $u(\{\alpha,\beta\})$.

Nun zur zweiten Aussage. Zunächst zeigen wir $t_{\hat{s}-1}$-Gültigkeit.

Behauptung 5.10. *Das Orakel $u' = u(\{\alpha,\beta\})$ ist $t_{\hat{s}-1}$-gültig.*

Beweis. Angenommen, es ist kein $t_{\hat{s}-1}$-gültiges Orakel. Dann ist eine der Bedingungen V1–V7 verletzt. Wir zeigen, dass diese Annahme zu einem Widerspruch führt. Wieder klar ist, dass V3, V4, V6 nicht verletzt sein können, denn jede definite Berechnung relativ zu $t_{\hat{s}-1}$-gültigem u ist auch eine definite Berechnung relativ zu u' nach Beobachtung 5.3(1). Auch V1 kann nicht verletzt sein; das folgt aus dem gleichen Argument wie im Beweis von Behauptung 5.9.

Angenommen, V5 ist verletzt. Dann existiert ein τ' mit $t_{\hat{s}-1}(\tau') = m' > 0$ und $n' \in H_{m'}$ und $|\Sigma^{n'} \cap u(X)| > 1$. Wieder folgt $n = n'$. Insbesondere folgt daraus auch $m = m'$, denn n ist nur in H_m enthalten. Gleichzeitig haben wir $\tau' \neq \tau^2_{a,b}$, da $\tau^2_{a,b} \notin \mathrm{dom}(t_{\hat{s}-1})$. Damit gilt aber $t_s(\tau') = t_{\hat{s}-1}(\tau') = m' = m = t_s(\tau^2_{a,b})$; Widerspruch zur Injektivität von $t_s \in \mathcal{T}$ auf dem Support. Also kann V5 nicht verletzt sein. Auf analoge Weise ist ersichtlich, dass auch V7 nicht verletzt sein kann.

Angenommen, V2 ist verletzt. Dann existiert ein $n' \in \mathrm{img}(e)$ mit $|u' \cap \Sigma^{n'}| > 2$. Wieder folgt $n = n'$. Jetzt haben wir aber $|\Sigma^{n'} \cap u'| = |\Sigma^n \cap u(\{\alpha,\beta\})| = |\{\alpha,\beta\}| = 2$; das ist ein Widerspruch zur Annahme $|\Sigma^{n'} \cap u'| > 2$. Also kann V2 nicht verletzt sein.

Damit ist keine Bedingung verletzt, u' muss also $t_{\hat{s}-1}$-gültig sein. $\square$

Wir erinnern uns daran dass $t' = t_{\hat{s}-1} \cup \{\tau^2_{a,b} \mapsto 0\}$. Nun ist es leicht zu sehen, dass $u' \stackrel{\text{df}}{=} u(\{\alpha,\beta\})$ auch t'-gültig ist. Das Orakel u' ist nach voriger Behauptung $t_{\hat{s}-1}$-gültig, und bei der Erweiterung von $t_{\hat{s}-1}$ zu t' kommt nur $\tau^2_{a,b} \mapsto 0$ hinzu. Das bedeutet, dass wir nur noch die Bedingung V4 bezüglich a, b verifizieren müssen. Sei nun $y = F_r(0^n)$

relativ zu u'. Nach voriger Aussage oben wissen wir aber, dass $M_a(y)$ und $M_b(y)$ relativ zu u' beide definitiv akzeptieren, wie von V4 verlangt.

Da nun also $u' \sqsupsetneq u \sqsupsetneq w_{\hat{s}-1}$ auch t'-gültig ist, sind wir fertig und erreichen einen Widerspruch, wie anfangs argumentiert: während der Bearbeitung von Task $\tau^2_{a,b}$ in Stufe $\hat{s}$ war das Orakel u' eine mögliche t'-gültig Erweiterung von $w_{\hat{s}-1}$, denn es ist t'-gültig und $u' \sqsupsetneq w_{\hat{s}-1}$. Also wäre nach Definition des Tasks $\tau^2_{a,b}$ dann $t_{\hat{s}} = t'$ gesetzt worden. Damit wäre dann auch $t_s(\tau^2_{a,b}) = t'(\tau^2_{a,b}) = 0$, was der Voraussetzung dieses Lemma 5.8 widerspricht. □

Ohne größeren Veränderungen lässt sich auf fast gleiche Weise auch zeigen, dass die Tasks $\tau^3_{a,r}$ bearbeitet werden können.

Lemma 5.11. *Die Bearbeitung eines Tasks $\tau^3_{a,r}$ ist möglich: gilt $t_s = t_{s-1}, t_s(\tau^3_a) = m > 0$, dann lässt sich w_s so zu t_s-gültigem $w \sqsupsetneq w_{s-1}$ erweitern, dass eine der folgenden Fälle eintritt:*

- *$0^n \in C^v_m$ für alle $v \sqsupseteq w$ und $M_a(F_r(0^n))$ lehnt relativ zu w definitiv ab.*
- *$0^n \notin C^v_m$ für alle $v \sqsupseteq w$ und $M_a(F_r(0^n))$ akzeptiert relativ zu w definitiv.*

Beweis. Wir verfahren wie im Beweis von Lemma 5.8. Um wieder einen Widerspruch abzuleiten, nehmen wir an, dass für alle t_s-gültigen $w \sqsupsetneq w_{s-1}$ weder (1) noch (2) zutrifft. Definiere wieder identisch u und $u(X)$. Wieder gilt nach Behauptung 5.9 dass $u(X)$ immer t_s-gültig ist wenn $|X| \leq 1$. Wir haben also:

- Die Berechnung $M_a(F_r(0^n))$ lehnt relativ zu $u(\emptyset)$ definitiv ab.
- Für $\xi \in \Sigma^n$ gilt $0^n \in C^{u(\{\xi\})}_m$ und daher akzeptiert $M_a(F_r(0^n))$ definitiv relativ zu $u(\{\xi\})$ auf einem Rechenweg mit Menge Q_ξ an Orakelfragen.

Beachte, dass $\xi \in Q_\xi$ ist, andernfalls stimmen $u(\emptyset)$ und $u(\{\xi\})$ auf Q_ξ überein, damit folgt mit Beobachtung 5.3(3), dass $M_a(F_r(0^n))$ relativ zu $u(\emptyset)$ akzeptiert, was der Annahme widerspricht.

Wir fahren nun fort wie in Lemma 5.8. Seien $\alpha \in \Sigma^n, \beta \in \Sigma^n$ zwei unterschiedliche Wörter, die sich nicht gegenseitig stören. Das heißt, $\alpha \notin Q_\beta, \beta \notin Q_\alpha$. Diese zwei Wörter existieren nach dem gleichen kombinatorischen Argument wie bei Lemma 5.8.

Setze nun $u(\{\alpha, \beta\})$. Wir zeigen nun, dass $M_a(F^r(0^n))$ auf zwei unterschiedlichen Rechenwegen akzeptieren. Wieder haben wir, dass $u(\{\alpha\})$ und $u(\{\alpha, \beta\})$ auf Q_α übereinstimmen, und symmetrisch $u(\{\beta\})$ und $u(\{\alpha, \beta\})$ auf Q_β übereinstimmen. Mittels Beobachtung 5.3(3) sehen wir also, dass es zwei Rechenwege von $M_a(F_r(0^n))$ relativ zu $u(\{\alpha, \beta\})$ gibt, welche definitiv akzeptieren: einer mit Orakelfragen Q_α, und einer mit Q_β. Diese zwei Rechenwege sind tatsächlich unterschiedlich: einerseits gilt aus obiger Feststellung dass $\alpha \in Q_\alpha$, aber nach Wahl von β ist $\alpha \notin Q_\beta$. Auf einem Rechenweg wird also α erfragt, auf dem anderen nicht; also sind die zwei Rechenwege unterschiedlich.

Wie im Beweis von Lemma 5.8 können wir jetzt $t' \overset{\text{df}}{=} t_{\hat{s}-1} \cup \{\tau^3_a \mapsto 0\}$ setzen. Wieder gilt nach Behauptung 5.10 dass $u' \overset{\text{df}}{=} u(\{\alpha, \beta\})$ auch $t_{\hat{s}-1}$-gültig ist. Dann lässt sich auch leicht sehen, dass u' auch t'-gültig ist. Es kommt V6 bezüglich a hinzu, aber wir haben ja eben gesehen, dass M_a auf zwei unterschiedlichen Rechenwegen definitiv akzeptiert.

Da nun also $u' \sqsupsetneq u \sqsupsetneq w_{\hat{s}-1}$ auch t'-gültig ist, sind wir fertig und erreichen einen Widerspruch: während der Bearbeitung von Task $\tau^2_{a,b}$ in Stufe $\hat{s}$ war das Orakel u' eine mögliche t'-gültige Erweiterung von $w_{\hat{s}-1}$. Also wäre nach Definition des Tasks $\tau^2_{a,b}$ dann $t_{\hat{s}} = t'$ gesetzt worden. Damit wäre dann auch $t_s(\tau^2_{a,b}) = t'(\tau^2_{a,b}) = 0$, was der Voraussetzung dieses Lemma 5.11 widerspricht. □

Damit ist die Konstruktion möglich.

Eigenschaften des konstruierten Orakels

Nun müssen wir noch verifizieren, dass E tatsächlich alle gewünschten Eigenschaften erfüllt, die in Satz 5.2 behauptet wurden. Zur Erinnerung: wir haben $E = \bigcup_{s \in \mathbb{N}} w_s$. Beachte dass $w_s \sqsubsetneq E$ für alle w_s. Sei außerdem $t_\omega \overset{\text{df}}{=} \bigcup_{s \in \mathbb{N}} t_s$. Beachte, dass nach Definition von $t_0, t_1, \ldots$ auch t_ω eine (totale) Funktion ist. Beachte außerdem, dass $|w_0| < |w_1| < |w_2| < \ldots$ eine nach oben unbeschränkte Folge ist. Wir starten mit einer Beobachtung über den groben Aufbau von E, der sich sofort aus der Ebenen-Konstruktion und unserem Gültigkeitsbegriff ergibt.

Behauptung 5.12. (1) *E enthält keine Wörter der Länge $\neq e(\cdot)$.*

(2) *Es gilt $|E \cup \Sigma^{e(i)}| \leq 2$ für alle i.*

(3) *Wenn $t_\omega(\tau_j^1) = 0$, dann gilt $L(N_j^E) \neq \Sigma^*$.*

(4) *Wenn $t_\omega(\tau_{a,b}^2) = m > 0$, dann gilt $|E \cap \Sigma^n| \leq 1$ für alle $n \in H_m$. Insbesondere gilt damit $(A_m^E, B_m^E) \in \mathrm{DisjUP}^E$.*

(5) *Wenn $t_\omega(\tau_a^3) = m > 0$, dann gilt $|E \cap \Sigma^n| \leq 1$ für alle $n \in H_m$. Insbesondere gilt damit $(C_m^E) \in \mathrm{UP}^E$.*

Beweis. Zu (1): Sei x ein Wort mit $|x| \neq e(\cdot)$. Wir zeigen dass $x \notin E$. Wähle ein s sodass $x < |w_s|$, heißt w_s wird von x definiert. Nun ist w_s auch t_s-gültig, und mit V1 gilt insbesondere $x \notin w_s$. Da nun $w_s \sqsubsetneq E$, gilt auch $x \notin E$ und wir sind fertig.

Zu (2): Beweis läuft analog wie bei (1), unter Berufung auf V2.

Zu (3): Nach Voraussetzung muss es ein s geben, für das $t_s(\tau_j^1) = 0$. Insbesondere ist w_s auch t_s-gültig, und mit V3 existiert ein z sodass $N_j^{w_s}(z)$ definitiv ablehnt. Da $w_s \sqsubseteq E$, folgt mit Beobachtung 5.3(1), dass $N_j^E(z)$ auch definitiv ablehnen wird.

Zu (4): Angenommen, es existiert ein $n \in H_m$ für das $|E \cap \Sigma^n| > 1$. Nach Voraussetzung existiert eine Stufe $\hat{s}$, in der $\tau_{a,b}^2$ bearbeitet wurde. Wähle nun ein $s > \hat{s}$ sodass w_s alle Wörter der Länge $\leq n$ definiert. Dieses w_s ist insbesondere t_s-gültig.

Nun gilt $t_s(\tau_{a,b}^2) = t_\omega(\tau_{a,b}^2) = m > 0$, und mit V5 gilt insbesondere $|w_s \cap \Sigma^n| \leq 1$. Da nun w_s und E auf Σ^n übereinstimmen, haben wir auch $|E \cap \Sigma^n| \leq 1$. Das widerspricht der Annahme.

Also gilt $|E \cap \Sigma^n| \leq 1$ für alle $n \in H_m$. Dann folgt aus Beobachtung 5.6(1) auch schon sofort, dass $(A_m^E, B_m^E) \in \mathrm{DisjUP}$.

Zu (5): Beweis läuft analog wie bei (4). □

Über die Definition von Task $\tau_{a,b}^2$ bzw. $\tau_{a,r}^3$ ist nun in Verbindung mit der vorigen Beobachtung auch ersichtlich, dass für DisjNP bzw. UP keine vollständigen Elemente existieren können.

Behauptung 5.13. (1) *Kein Paar aus DisjNP^E ist $\leq_m^{pp}$-hart für DisjUP^E.*

(2) *Keine Menge aus UP^E ist $\leq_m^{p}$-vollständig für UP^E.*

Beweis. Wir zeigen hier nur (1), der Beweis für Aussage (2) folgt analog.

Angenommen es existiert ein solches $\leq_m^{pp}$-hartes Paar $(U_1, U_2) \in \mathrm{DisjNP}^E$. Dann existieren über die Wahl der Standardenumeration auch zwei NPTM N_a, N_b sodass $L(N_a^E) = U_1, L(N_b^E) = U_2$. Betrachte den Task $\tau_{a,b}^2$; dieser wurde in der Konstruktion von E in Stufe $\hat{s}$ bearbeitet. Wir erinnern uns, dass $w_{\hat{s}} \sqsubsetneq E$ ein $t_{\hat{s}}$-gültiges Orakel ist.

Wenn $t_{\hat{s}}(\tau^2_{a,b}) = 0$ ist, dann besagt V4, dass eine Eingabe $x \in \Sigma^*$ existiert für die sowohl $M_a(x)$ als auch $M_b(x)$ definitiv relativ zu $w_{\hat{s}}$ akzeptieren. Da $w_{\hat{s}} \sqsubseteq E$ ist, akzeptieren $M_a(x)$ und $M_b(x)$ relativ zu E, nach Beobachtung 5.3(1). Das bedeutet aber $x \in U_1 \cap U_2$; Widerspruch zur Wahl von U_1, U_2 als disjunkt.

Daher können wir annehmen, dass $m \stackrel{\text{df}}{=} t_{\hat{s}}(\tau^2_{a,b}) > 0$ ist. Insbesondere ist daher $t_\omega(\tau^2_{a,b}) > 0$ und nach voriger Behauptung 5.12(4) gilt dann auch $(A^E_m, B^E_m) \in \mathrm{DisjUP}^E$. Also gilt nach Annahme auch $(A^E_m, B^E_m) \leq^{\mathrm{pp}}_{\mathrm{m}} (U_1, U_2)$ relativ zu E.

Also muss ein r existieren, sodass F^E_r diese Reduktion realisiert. Betrachte den Task $\tau^2_{i,j,r}$, welcher in einem bestimmten Schritt s behandelt wird. Nach Definition dieses Task existiert ohne Beschränkung der Allgemeinheit ein $n \in \mathbb{N}$ sodass $0^n \in A^E_m$, und $M_a(F_r(0^n))$ definitiv relativ zu w_s ablehnt. (Der andere Fall in dem M_b ablehnt läuft symmetrisch.)

Mit Beobachtung 5.3(1) erhalten wir dann, dass $M_a(F_r(0^n))$ relativ zu E ablehnt. Also auch $F^E_r(0^n) \notin U_1$. Dies widerspricht der Annahme, dass F^E_r die Reduktion realisiert, denn wir haben ja $0^n \in A^E_m$. □

Damit erfüllt E schon die Aussagen (1) und (2) von Satz 5.1. Die letzte Aussage (3) ist aufwändiger zu beweisen. Folgende Behauptung sagt aus, dass totale NPTM N_j relativ zu E „besonders total" sind. Nimmt man (ab hinreichender Länge) Wörter der Länge $= e(\cdot)$ aus E weg, und erhält $T \subseteq E$, bleibt N_j relativ zu T immer noch total. Das ergibt auch Sinn, denn in der Bearbeitung von Task τ^1_j wurden sämtlichen geeigneten Erweiterungen ausprobiert, welche die Totalität von N_j zerstören würden. Da keine solche Erweiterung existiert, wird auch T nicht die Totalität von N_j zerstören können.

Behauptung 5.14. *Sei N_j eine totale NPTM, d.h. $L(N^E_j) = \Sigma^*$. Es existiert eine Länge n mit folgender Eigenschaft: falls $T \subseteq E$ mit E auf Wörtern der Länge $\leq n$ übereinstimmt, dann $L(N^T_j) = \Sigma^*$.*

Beweis. Sei s die Stufe bei der τ^1_j bearbeitet wurde, und setze $n = |w_s|$. Wir zeigen nun, dass dieses n die behauptete Eigenschaft erfüllt. Angenommen, dies gilt nicht, dann existiert ein $T \subseteq E$ dass mit E auf Wörtern der Länge $\leq n$ übereinstimmt, aber für ein Wort z lehnt $N^T_j(z)$ ab.

Sei $m \in \mathbb{N}$ so gewählt dass $m > n, p_j(|x|)$, und definiere das folgende partielle Orakel v, das genau für alle Wörter der Länge $\leq m$ definiert ist:

$$v[x] \stackrel{\text{df}}{=} \begin{cases} T[x] & \text{falls } |x| \leq m \\ \bot & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, dass $N^v_j(z)$ definitiv ablehnt, stimmen v und T ja auf allen Wörtern der Länge $\leq p_j(|x|)$ überein. Außerdem ist klar, dass $v \sqsupsetneq w_s \sqsupsetneq w_{s-1}$.

Sei $t' \stackrel{\text{df}}{=} t_{s-1} \cup \{\tau^1_j \mapsto 0\}$ Wir zeigen unten, dass v auch t_{s-1}-gültig ist. Damit ist dann klar, dass v auch t'-gültig ist: es kommt nur V3 bezüglich N_j hinzu, aber wir haben ja bereits gesehen dass $N^v_j(z)$ definitiv ablehnt.

Insbesondere ist dann während der Bearbeitung von Task τ^1_j in Stufe s das Orakel v eine mögliche t'-gültige Erweiterung von w_{s-1}, denn es ist t'-gültig und $v \sqsupsetneq w_{s-1}$. Also wäre nach Definition des Tasks τ^1_j dann $t_s = t'$ gesetzt worden, und wir hätten $t_s(\tau^1_j) = 0$. Dann wäre aber auch $t_\omega(\tau^1_j) = 0$, und nach voriger Behauptung 5.12(3) damit $L(N^E_j) \neq \Sigma^*$. Widerspruch zur Wahl von N_j.

Es bleibt zu zeigen dass v auch t_{s-1}-gültig ist. Angenommen, v ist nicht t_{s-1}-gültig, dann muss eine Bedingung V1–V7 verletzt sein. Zur Erinnerung: $v \sqsupsetneq w_{s-1}$ und w_{s-1} ist

t_{s-1}-gültig. Wieder klar ist, dass V3, V4, V6 nicht verletzt sein können, denn jede definite Berechnung relativ zu w_{s-1} ist auch eine definite Berechnung relativ zu v nach Beobachtung 5.3(1).

Angenommen V1 ist verletzt. Dann existiert ein $x < |v|$ und $|x| \neq e(\cdot)$ und $x \in v$. Nun stimmt v und T aber insbesondere auf x überein, heißt wir haben

$$x \in v \implies x \in T \implies x \in E,$$

das widerspricht aber der Beobachtung 5.12(1) dass $x \notin E$. Widerspruch zur Annahme; V1 kann nicht verletzt sein.

Angenommen, V5 ist verletzt. Dann existiert ein τ' mit $t_{s-1}(\tau') = m' > 0$ und $n' \in H_{m'}$ und $|\Sigma^{n'} \cap v| > 1$. Insbesondere gilt $t_\omega(\tau') = m' > 0$. Außerdem muss $n' \leq m$ sein, da ansonsten $\Sigma^{n'}$ nicht durch v definiert ist (und damit $v \cap \Sigma^{n'} = \emptyset$). Insbesondere stimmen damit v und T auf $\Sigma^{n'}$ überein. Damit gilt

$$|\Sigma^{n'} \cap v| = |\Sigma^{n'} \cap T| \leq |\Sigma^{n'} \cap E| \leq 1,$$

wobei die erste Ungleichung aus $T \subseteq E$ folgt, und die zweite Ungleichung aus voriger Behauptung 5.12(4) bzw. (5) folgt. Das widerspricht der Annahme $|\Sigma^{n'} \cap v| > 1$, also kann V5 nicht verletzt sein. Auf analoge Weise lässt sich zeigen, dass V2 und V7 nicht verletzt sein können.

Damit ist keine Bedingung verletzt, v muss also t_{s-1}-gültig sein. □

Nun können wir für die Hypothese Q argumentieren. Hierfür müssen wir, gegeben eine totale NPTM N_j^E, effizient einen akzeptierenden Rechenweg von $N_j^E(x)$ bestimmen. Zur Erinnerung: wir haben in der Konstruktion angenommen, dass P $=$ PSPACE. Das Ausrechnen eines akzeptierenden Rechenwegs ist daher offensichtlich zumindest relativ zu $\emptyset$ (anstelle E) möglich, da unter unserer Annahme insbesondere P $=$ NP gilt.

Hätten wir nun die Möglichkeit, effizient E entscheiden zu können, dann wäre auch $\mathrm{P}^E = \mathrm{NP}^E$ und wir könnten einen akzeptierenden Rechenweg von $N_j^E(x)$ einfach ausrechnen. Nun können wir zwar E nicht effizient entscheiden, aber wir können mittels N_j und Behauptung 5.14 zumindest iterativ eine relevante finite Approximation $T \subseteq E$ von E bestimmen. Das soll kurz skizziert werden: Gegeben eine vorläufige finite Approximation $T \subseteq E$, können wir einen akzeptierenden Rechenweg α von $N_j^T(x)$ ausrechnen. Dieser existiert insbesondere nach Behauptung 5.14. Nun können wir die Orakelfragen auf α mit dem „echten“ Orakel E abgleichen, und so überprüfen, ob unserer Approximation Wörter fehlen, und diese ggf. T hinzufügen. Nachdem E dünn ist, enden wir spätestens nach polynomiell vielen Iterationen bei einer Approximation T, die auf allen relevanten Wörtern mit E übereinstimmt.

Behauptung 5.15. *Sei N_j eine totale NPTM, d.h. $L(N_j^E) = \Sigma^*$. Dann existiert eine Funktion $g \in \mathrm{FP}^E$ sodass $g(x)$ einen akzeptierenden Rechenweg von $M_j^E(x)$ ausgibt. Damit gilt nach Definition die Hypothese* Q *relativ zu E.*

Beweis. Sei n hinreichend groß, sodass diese vorige Behauptung 5.14 erfüllt ist. Damit gilt

$$\begin{aligned} &T \subseteq E, T \text{ und } E \text{ stimmen auf } \Sigma^{\leq n} \text{ überein} \\ &\implies L(N_j^T) = \Sigma^*. \end{aligned} \tag{5.2}$$

Wir betrachten nun folgenden formalen Algorithmus relativ zu E auf Eingabe x:

```
1 T ← E ∩ Σ^{≤n} (Konstante, muss nicht berechnet werden)
2 wiederhole (Invariante: T ⊆ E)
3     Sei α ein akzeptierender Rechenweg auf N_j^T(x) und Q die Menge an
      Orakelfragen
4     wenn existiert eine Frage q ∈ Q für die q ∈ E und q ∉ T dann
5         T ← E ∪ {q}
6     sonst (für alle q ∈ Q gilt: q ∉ E ∨ q ∈ T)
7         Gebe α aus
```

Korrektheit: Beobachte zunächst die Invariante dass $T \subseteq E$, und dass T und E auf $\Sigma^{\leq n}$ übereinstimmen. Damit gilt nach (5.2) auch $L(N_j^T) = \Sigma^*$ und insbesondere existiert dann auch ein akzeptierender Rechenweg auf $N_j^T(x)$. Damit ist Z. 3 wohldefiniert. Terminiert nun der Algorithmus mit einem Rechenweg α in Z. 7, dann stimmt T mit E auf Q überein: Falls $q \in T$, dann auch $q \in E$ (nach Invariante). Falls $q \notin T$, dann $q \notin E$ (nach Negation der If-Bedingung). Also wird auch $N_j^E(x)$ mit Rechenweg α akzeptieren, nach Beobachtung 5.3(3).

Laufzeit: Wir zeigen dass der Algorithmus in polynomiell beschränkter deterministischer Zeit (abhängig von $|x|$) relativ zu E arbeitet. Wir wissen, dass für jede Orakelfrage $q \in Q$ gilt, dass $|q| \leq p_j(|x|)$. Zusammen mit oben genannten Invariante gilt $T \subseteq \{w \in E \mid \exists i. |w| = e(i) \leq p_j(|x|)\}$. Nun ist aber E dünn. Im Speziellen gilt mit Behauptung 5.12(1) und (2) dass

$$\ell(T) \leq \sum_{\substack{i\in\mathbb{N} \\ e(i)\leq p_j(|x|)}} \ell(E \cap \Sigma^{e(i)}) = \sum_{\substack{i\in\mathbb{N} \\ e(i)\leq p_j(|x|)}} e(i) \cdot |E \cap \Sigma^{e(i)}| \leq \sum_{\substack{i\in\mathbb{N} \\ e(i)\leq p_j(|x|)}} 2e(i) \leq 2p_j(|x|)^2. \tag{5.3}$$

Das heißt, dass nach polynomiell vielen Iterationen wird kein weiteres Wort zu T hinzugefügt und der Algorithmus terminiert.

Wir zeigen nun abschließend, dass Zeile 3 in polynomiell beschränkter deterministischer Zeit berechnet werden kann. Hierfür werden wir das finite T in N_j „hineincodieren“. Gemeint ist, T in das „Programm“ N_j hineincodieren, sodass für Queries $q, |q| = e(\cdot)$ nur getestet wird ob $q \in T$ anstelle einer echten Orakelfrage. Es existiert also eine Orakel-NPTM N_j', sodass $N_j'(x)$ (ohne Orakel) äquivalent zu $N_j^T(x)$ arbeitet. Und da $\mathrm{P} = \mathrm{PSPACE}$, können wir in FP auch einen akzeptierenden Rechenweg β' von $N_j'(x)$ effizient bestimmen. Das geht dann auch in $\mathrm{FP}^E \supseteq \mathrm{FP}$. Dieser Rechenweg β' entspricht dann einem akzeptierenden Rechenweg β für $N_j^E(x)$. Also können wir auch einen akzeptierender Rechenweg von $N_j^E(x)$ in FP^E berechnen. Beachte insbesondere, dass das Hineincodieren in Polynomialzeit möglich ist, ist ja $\ell(E)$ nach Ungleichung (5.3) polynomiell in Abhängigkeit von $|x|$ beschränkt. □

Damit ist der Beweis von Satz 5.2 abgeschlossen: unter Annahme von $\mathrm{P} = \mathrm{PSPACE}$ erfüllt das Orakel E alle Eigenschaften. Relativ zu E existiert nach Behauptung 5.13(1) kein $\leq_m^{pp}$-hartes Paar in DisjNP für DisjUP, und (2) kein $\leq_m^{p}$-vollständige Menge für UP. Nach Behauptung 5.15 kann für jede totale NPTM N_j eine Funktion $g \in \mathrm{FP}$ angegeben werden, für die $g(x)$ ein akzeptierender Rechenweg für $N_j(x)$ ist. Mit den Eigenschaften der Standardenumeration ist klar, dass dass auch für *alle* totalen NPTM N gilt. (Übersetze N in äquivalente Standard-NPTM N_j, übersetze dann den akzeptierenden Rechenweg $g(x) = \alpha$ von $N_j(x)$ in einen akzeptierenden Rechenweg α' von $N(x)$ zurück.) Damit gilt relativ zu E also auch die Hypothese Q (3).

Es lässt sich einfach überprüfen, dass die vorhergegangene Konstruktion relativiert. Sei nun D ein PSPACE-vollständiges Orakel. Wie anfangs dieses Kapitels argumentiert, gilt nun mit $O \overset{\mathrm{df}}{=} E \oplus D$ auch Satz 5.1 *ohne* die Annahme $\mathrm{P} = \mathrm{PSPACE}$.

Wir schließen dieses Kapitel mit einer Zusammenfassung weiterer Eigenschaften von O, die sich unmittelbar aus relativierenden Implikationen ergeben. An dieser Stelle seien noch zwei Funktionenklassen definiert, welche von Fenner, Fortnow, Naik und Rogers (2003) im Kontext der Hypothese Q untersucht wurden. Definiere NPbV als diejenige Teilmenge der Multifunktionen $f \in \mathrm{NPMV}$ für die $\mathit{set}\text{-}f(x) \subseteq \{0,1\}$ für alle x gilt, und definiere NPkV als diejenige Teilmenge der Multifunktionen $f \in \mathrm{NPMV}$ für die $|\mathit{set}\text{-}f(x)| \leq k$ für alle x gilt.

Korollar 5.16. *Folgende Aussagen gelten relativ zum Orakel O:*

(1) $\mathrm{P} = \mathrm{UP} \cap \mathrm{coUP} = \mathrm{NP} \cap \mathrm{coNP} \subsetneq \mathrm{UP} \subsetneq \mathrm{NP}$.
(2) NP, UP, NE, NEE *sind nicht abgeschlossen unter Komplement.*
(3) P, E, EE *sind nicht abgeschlossen unter Nichtdeterminismus.*
(4) $\mathrm{UP} \not\subseteq \mathrm{coNP}$.
(5) $\mathrm{NPSV_t} \subseteq \mathrm{FP}$.
(6) $\mathrm{NP}b\mathrm{V_t} \subseteq_{\mathrm{c}} \mathrm{FP}$.
(7) $\mathrm{NP}k\mathrm{V_t} \subseteq_{\mathrm{c}} \mathrm{FP}$ *für alle* $k \geq 1$.
(8) $\mathrm{NPMV_t} \subseteq_{\mathrm{c}} \mathrm{FP}$.
(9) $\mathrm{NPMV_t} \subseteq_{\mathrm{c}} \mathrm{NPSV_t}$.
(10) $\mathrm{NPMV_t} \subseteq_{\mathrm{c}} \mathrm{TFNP}$.
(11) $\mathrm{NPMV} \not\subseteq_{\mathrm{c}} \mathrm{NPSV}$.
(12) $\mathrm{NPMV_t} \not\subseteq_{\mathrm{c}} \mathrm{TFNP}$.
(13) $\mathrm{TFNP} \subseteq_{\mathrm{c}} \mathrm{FP}$.
(14) *Hypothese* Q.
(15) *Hypothese* Q′.
(16) *Es existiert eine NP-Relation R mit $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständigem* Proj$(R)$ *für* NP *und R ist nicht $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig für* FNP.
(17) $\mathrm{NP} \cap \mathrm{coNP}$ *hat eine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge.*
(18) UP *hat keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge.*
(19) DisjNP *hat kein $\leq_{\mathrm{m}}^{\mathrm{pp}}$-vollständiges Paar.*
(20) NPSV *hat keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Multifunktion.*
(21) $\mathrm{NPMV_t}$ *hat eine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Multifunktion.*
(22) TFNP *hat eine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Multifunktion.*
(23) DisjCoNP *hat ein $\leq_{\mathrm{m}}^{\mathrm{pp}}$-vollständiges Paar.*
(24) *Für keine $\leq_{\mathrm{m}}^{\mathrm{p}}$-vollständige Menge A für* coNP *existiert ein optimales Beweissystem, heißt* $\mathrm{TAUT^N}$ *und* TAUT.
(25) *Für alle Mengen $A \in$* coNP *existiert ein* P*-optimales Beweissystem, heißt* ¬SAT.
(26) *Es existiert ein* P*-inseparables* DisjNP*-Paar.*
(27) *Jedes* DisjCoNP*-Paar ist* P*-separierbar.*
(28) *Weder* NP *noch* coNP *haben die* shrinking property *(Glaßer, Reitwießner und Selivanov, 2011).*
(29) coNP *hat die* separation property *(Glaßer, Reitwießner und Selivanov, 2011).*
(30) NP *hat nicht die* separation property.

Beweis. Es ist klar, dass sich aus Satz 5.1 unmittelbar (14), (18) und (19) ergeben. Die Aussagen (3), (5), (13), (15--17), (20--27) erschließen sich sofort mit Zuhilfenahme der in Abbildung 7 eingezeichneten Äquivalenzen und Implikationen, zusammen mit Satz 4.27.

Folgende Aussagen bleiben noch offen:

Zu (1): $\mathrm{P} = \mathrm{UP} \cap \mathrm{coUP} = \mathrm{NP} \cap \mathrm{coNP} \subsetneq \mathrm{UP}$ klar aus der Abbildung, die Echtheit der Inklusion $\mathrm{UP} \subsetneq \mathrm{NP}$ folgt daraus, dass NP eine vollständige Menge hat, UP aber nicht, nach (18).

Zu (2): Angenommen $\mathrm{UP} = \mathrm{coUP}$ dann auch $\mathrm{UP} = \mathrm{UP} \cap \mathrm{coUP} = \mathrm{P}$, was ein Widerspruch zu (1) ist. Restliche Klassen sind ersichtlich über die Abbildung.

Zu (4): Angenommen $\mathrm{UP} \subseteq \mathrm{coNP}$, dann wäre $\mathrm{UP} \subseteq \mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$, was ein Widerspruch zu (1) ist.

Zu (6): Ist nach Fenner, Fortnow, Naik und Rogers (2003, Thm. 4) äquivalent zu (14).

Zu (7): Ist nach Fenner, Fortnow, Naik und Rogers (2003, Thm. 14) äquivalent zu (6).

Zu (8): Nach Satz 4.1.

Zu (9)–(10): Aus (8) folgt insbesondere $\mathrm{NPMV_t} \subseteq_c \mathrm{FP_t}$ und $\mathrm{FP_t} \subseteq \mathrm{NPSV_t}$, TFNP.

Zu (11): Es ist leicht zu sehen, dass (relativ zu jedem Orakel) eine kanonische $\leq_m^p$-vollständige Multifunktion *f* für NPMV existiert. Angenommen $\mathrm{NPMV} \subseteq_c \mathrm{NPSV}$, dann wäre auch $f \in_c \mathrm{NPSV}$ und die Verfeinerung insbesondere auch $\leq_m^p$-vollständig für NPSV. Das widerspricht (20).

Zu (12): Angenommen $\mathrm{NPMV_t} \subseteq \mathrm{TFNP}$, dann gilt mit Beobachtung 3.4 auch $\mathrm{P} = \mathrm{NP}$; das widerspricht (1).

Zu (28): Folgt aus (19) und (2) durch Ergebnisse von Glaßer, Reitwießner und Selivanov (2011). Wenn es kein $\leq_m^{pp}$-vollständiges Paar für DisjNP gibt, dann hat NP nicht die *shrinking property*. Außerdem hat coNP die *shrinking property* genau dann wenn $\mathrm{NP} = \mathrm{coNP}$.

Zu (29): Ist nach Glaßer, Reitwießner und Selivanov (2011) äquivalent zu (6).

Zu (30): Folgt nach Glaßer, Reitwießner und Selivanov (2011) aus (4). □

6 Fazit

Zum Abschluss werden hier noch einmal die wesentlichen Ergebnisse dieser Arbeit zusammengefasst, sowie noch einmal offene Fragen gesammelt, welche sich im Verlauf der Arbeit ergeben haben. Bezüglich der Ergebnisse können wir uns an den vier Forschungsdesideraten orientieren, die in der Einleitung genannt wurden.

Suchprobleme vs. Entscheidungsprobleme

Erstens, wurde in Kapitel 3 die Beziehung zwischen NP-Suchproblemen und den entsprechenden NP-Entscheidungsproblemen erarbeitet. Hierfür orientiert sich die Arbeit an vorhandenen Definitionen von Suchproblemen, die intuitiv neben dem Berechnungsproblem „Gilt $x \in L$?“ auch nach einem „Beweis“ oder „Zertifikat“ für die Aussage „$x \in L$“ fragen. Auf dieser Basis wurde gezeigt, dass die übliche Argumentation, das Suchproblem auf das Entscheidungsproblem reduzieren zu können (*search reduces to decision*) im Allgemeinen nicht immer zutrifft. Ergebnisse aus der Literatur zeigen, dass dies unter geeigneten Bedingungen nicht für die NP-Intermediates (also jene Probleme, die nicht NP-vollständig sind, aber auch nicht in P liegen) zutrifft. Im Kontext der *Selbstreduzierbarkeit*, also der Lösung des Suchproblems gegeben einem Orakel für „kleinere“ Instanzen, liegen dagegen nur wenige Ergebnisse vor. In der Arbeit wurde insbesondere auf Ergebnisse von Harsha, Mitropolsky und Rosen (2023) hingewiesen, welche Anzeichen geben, dass das Suchproblem der Faktorisierung nicht selbstreduzierbar ist.

Wie die Suchprobleme lassen sich auch die Entscheidungsprobleme mittels Reduktionen untereinander nach ihrer Schwierigkeit vergleichen. In dieser Arbeit wurde insbesondere die Levin-Reduktion als Reduktion auf den NP-Suchproblemen produktiv gemacht. Dieser Reduktionsbegriff kann als natürliche Generalisierung der Many-one- bzw. Karp-Reduktion auf den Entscheidungsproblemen verstanden werden. Hieran anschließend wurde auf die Beziehung zwischen Levin-Vollständigkeit der Suchprobleme und der Karp-Vollständigkeit der entsprechenden Entscheidungsprobleme hingewiesen. Erstens verstärkt die Levin-Vollständigkeit die Karp-Vollständigkeit. Zweitens sehen wir, empirisch gesprochen, dass für alle *natürlichen* Probleme diese beiden Vollständigkeitsbegriffe zusammenfallen. Dennoch bleibt Folgendes unklar: *Gegeben ein beliebiges NP-Suchproblem R, für welches das entsprechende Entscheidungsproblem* $\mathrm{Proj}(R)$ *Karp-vollständig ist, ist dann auch R Levin-vollständig?*

Die Formulierung dieser Frage ist der zentrale Beitrag dieser Arbeit. Sie stellt zwei umfassende „Aufgabentypen“ (Suchen vs. Entscheiden) und Reduktionsbegriffe der Komplexitätstheorie gegenüber, und trotz der einfachen Formulierung beschäftigt sich nur wenig Forschung mit dieser Frage. Es gibt keine Hinweise, welche auf eine positive oder negative Beantwortung dieser Frage hinweisen; im Kontext dieser Arbeit wurde die negative Beantwortung als Hypothese KvL formuliert.

NP-Suchprobleme im Pudlákschen Programm

Zweitens, wurde die Hypothese Q (erstmals formuliert von Fortnow und Rogers, 1993) in den Blick genommen, welche als Aussage „alle totalen NP-Suchprobleme sind effizient lösbar“ unmittelbar in direkter Beziehung zu den NP-Suchproblemen steht. Die vorliegende Arbeit zeigt nun insbesondere auch folgende Charakterisierungen von Q:

- Es gilt Q genau dann wenn jede Karp-Reduktion zwischen Entscheidungsvarianten von zwei NP-Suchproblemen zu einer Levin-Reduktion zwischen eben jenen Suchproblemen verstärkt werden kann. Damit ist $\neg$Q auch eine notwendige Bedingung für KvL.
- Es gilt Q genau dann wenn das Standardbeweissystem eines beliebigen Levin-vollständigen NP-Suchproblems P-optimal ist. Das eliminiert damit ein Padding-Argument von Messner (2000) und verallgemeinert sein Ergebnis, dass Q äqivalent zir P-Optimalität des Standardbeweissystems *sat* ist. Über diese Charakterisierung wird auch deutlich, dass $\neg$Q auch eine notwendige Bedingung für die Pudláksche Hypothese SAT ist.

Im Kontext des dritten Forschungsdesiderat wurde das Pudláksche Programm um weitere Hypothesen verfeinert. Insbesondere über die zweite Charakterisierung von Q lässt sich die Hypothese $\neg$Q in das Pudláksche Programm einordnen. Für die Hypothese KvL war dagegen eine Einordnung schwieriger, und es konnte nur gezeigt werden, dass $\mathsf{SAT}^{\mathsf{q}}$, eine natürliche Verstärkung von SAT, hinreichend für KvL ist.

Viertens, wurde abschließend ein Orakel konstruiert, relativ zu diesem die (relativierten) Hypothesen DisjNP, UP und Q gelten. Damit wurde insbesondere Q und KvL von einigen der Pudlákschen Hypothesen getrennt. Insbesondere erhalten wir damit das Resultat, dass selbst unter Annahme von DisjNP $\wedge$ UP es nicht mit relativierenden Beweisen möglich ist, auf Q (und auch nicht auf KvL) zu schließen. Für die Hypothese Q ergibt sich ferner unter relativierbaren Beweisen eine paarweise Unabhängigkeit zu den originalen Pudlákschen Hypothesen, außer für jene Fälle, bei der eine relativierende Implikation bekannt ist:

Korollar 6.1. *Sei* $\mathsf{A} \in \{\mathsf{DisjNP}, \mathsf{UP}, \mathsf{TAUT}^{\mathsf{N}}, \mathsf{TAUT}, \mathsf{DisjCoNP}, \mathsf{TFNP}, \mathsf{SAT}\}$. *Für jede Wahl von* A *existiert je ein Orakel, relativ zu diesem* $\mathsf{A} \wedge \mathsf{Q}$, $\mathsf{A} \wedge \neg\mathsf{Q}$, $\neg\mathsf{A} \wedge \mathsf{Q}$, *bzw.* $\neg\mathsf{A} \wedge \neg\mathsf{Q}$ *gilt, außer für die Fälle* $\mathsf{DisjCoNP} \wedge \mathsf{Q}$, $\mathsf{TFNP} \wedge \mathsf{Q}$, $\mathsf{SAT} \wedge \mathsf{Q}$.

Offene Fragen

Zum Schluss sind hier noch die offenen Fragen aufgelistet, welche sich im Verlauf dieser Arbeit ergeben haben, und welche eine Basis für zukünftige Forschung bilden könnten:

1. Die zentrale offene Frage dieser Arbeit ist jene zwischen Karp- und Levin-Vollständigkeit, bzw. der Hypothese KvL (vgl. Frage 3.23). Um mehr Evidenz für die Gültigkeit von KvL zu sammeln, wären weitere hinreichende natürliche Annahmen, z.B. kryptographischer Art, wünschenswert. Das betrifft u.a. auch die Hypothese Q. Lässt sich die Äquivalenz KvL $\Leftrightarrow$ $\neg$Q zeigen, wie in Abschnitt 4.1 vermutet? Allgemein gesprochen wäre hierzu auch ein besseres Verständnis der Zusammenhänge und Beziehungen zwischen den Hypothesen KvL, Q, SAT und $\mathsf{SAT}^{\mathsf{eff}}$ erstrebenswert, sowie angrenzend nähere Untersuchungen zum Begriff der P-Quasi-Simulation; die vorliegende Arbeit hat hierzu nur den ersten Anfang gemacht.

2. Konkret auch auf das Pudláksche Programm bezogen, wären weitere Orakelkonstruktionen dienlich, die KvL von den anderen Hypothesen trennen, um so KvL besser situieren zu können. Folgende Trennungen wurden in Abschnitt 4.3 als besonders interessant erachtet: konstruiere je ein Orakel, sodass relativ zu diesem $\mathsf{KvL} \land \neg\mathsf{SAT}^{\mathsf{q}}$, $\mathsf{KvL} \land \neg\mathsf{SAT}$ bzw. $\neg\mathsf{Q} \land \neg\mathsf{KvL}$ gilt. Schon allein ein Orakel O mit KvL^{O} wäre erstrebenswert, denn damit könne man relativierende Beweise für $\neg\mathsf{KvL}$ ausschließen.
3. Konkret in Bezug auf die Hypothese Q könnte der Begriff der Levin-Paddability intensiver betrachtet werden, wie er z.B. in der Charakterisierung der Hypothese Q eingesetzt wird (Definition 4.18). Besonders interessant wäre, ob sich diese zusätzliche Bedingung in der Charakterisierung durch Satz 4.24(9) streichen lässt. In anderen Worten: gilt Q genau dann wenn für eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation R das Standardbeweissystem std_R jedes ehrliches Beweissystem P-simulieren kann? Zeige hierfür beispielsweise, dass (nicht) jede $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation auch Levin-paddable ist (vgl. Frage 4.23).
4. Angrenzend an die Karp-vs-Levin-Frage bzw. Suche-vs-Entscheiden-Frage wäre ein besseres Verständnis zur Selbstreduzierbarkeit unter den Suchproblemen erstrebenswert. Wie schon in Abschnitt 3.2 beschrieben, ist die Forschungslage hierzu noch sehr dünn, insbesondere auf den nicht-totalen NP-Suchproblemen. Lassen sich die Erkenntnisse der Selbstreduzierbarkeit von *Entscheidungsproblemen* hierzu produktiv machen? Hierzu auch angrenzend wäre die Betrachtung schwächerer Reduzierbarkeits-Begriffe auf NP-Relationen (z.B. ähnlich zu Turing- oder Truth-Table-Reduktionen), und damit verbunden eine Abschwächung der Hypothese KvL, vielversprechend.

Literatur

Adleman, Leonard und Kenneth Manders (1977). „Reducibility, Randomness, and Intractibility (Abstract)". In: *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing* (Boulder, Colorado, USA, 4. Mai 1977). New York, USA: Association for Computing Machinery, 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* (Boston, Massachusetts, USA, 22. Juni 1992). 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, S. 781–793. DOI: 10.4007/annals.2004.160.781.

Arora, Sanjeev und Boaz Barak (2009). *Computational Complexity: a modern approach*. Cambridge, Massachusetts, USA: Cambridge University Press. 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, S. 175–184. URL: http://www.mat.ucm.es/serv/revista/vol2-23/vol2-23b.pdf.

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.

Berman, Leonard und Juris Hartmanis (1977). „On Isomorphisms and Density of NP and Other Complete Sets". In: *SIAM Journal on Computing* 6.2 (Juni 1977), S. 305–322. DOI: 10.1137/0206023.

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.

Beyersdorff, Olaf und Zenon Sadowski (2011). „Do there exist complete sets for promise classes?: Do there exist complete sets for promise classes?" In: *Mathematical Logic Quarterly* 57.6 (Dez. 2011), S. 535–550. DOI: 10.1002/malq.201010021.

Blum, Manuel und Russell Impagliazzo (1987). „Generic oracles and oracle classes". In: *28th Annual Symposium on Foundations of Computer Science* (Los Angeles, California, USA, 12. Okt. 1987). Washington, District of Columbia, USA: IEEE Computer Society Press, Okt. 1987, S. 118–126. DOI: 10.1109/SFCS.1987.30.

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. URL: https://hdl.handle.net/1813/6540.

Buhrman, Harry, James Kadin und Thomas 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.

Buss, Samuel R. (1996). *Lectures on Proof Theory*. Techn. Ber. Montreal, Kanada: School of Computer Science, McGill University. URL: https://mathweb.ucsd.edu/~sbuss/ResearchWeb/Barbados95Notes/reporte.pdf. Protokoll einer Reihe von Vorlesungen gehalten am Bellairs Research Institute, Holetown, Barbados im März 1995.

Cai, Jin-Yi und Artem Govorov (2021). „The Complexity of Counting Edge Colorings for Simple Graphs“. In: *Theoretical Computer Science* 889 (8. Okt. 2021), S. 24–24.

Cai, Jin-Yi, Thomas Gundermann, Juris Hartmanis, Lane A. Hemachandra, Vivian Sewelson, Klaus Wagner und Gerd Wechsung (1988). „The Boolean Hierarchy I: Structural Properties“. In: *SIAM Journal on Computing* 17.6 (Dez. 1988), S. 1232–1252. DOI: 10.1137/0217078.

Cai, Jin-Yi, Thomas Gundermann, Juris Hartmanis, Lane A. Hemachandra, Vivian Sewelson, Klaus Wagner und Gerd Wechsung (1989). „The Boolean Hierarchy II: Applications“. In: *SIAM Journal on Computing* 18.1 (Feb. 1989), S. 95–111. DOI: 10.1137/0218007.

Cai, Jin-Yi und Lane Hemachandra (1986). „The boolean hierarchy: Hardware over NP“. In: *Structure in Complexity Theory* (Berkeley, California, USA, 2. Juni 1986). Hrsg. von Alan L. Selman. Bd. 223. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, S. 105–124. DOI: 10.1007/3-540-16486-3_93.

Cook, Stephen A. (1971). „The Complexity of Theorem-Proving Procedures“. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (Shaker Heights, Ohio, USA, 3. Mai 1971). New York, USA: Association for Computing Machinery, 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. DOI: 10.2307/2273702.

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

Dose, Titus (2020a). „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.

Dose, Titus (2020b). „Balance Problems for Integer Circuits and Separations of Relativized Conjectures onIncompleteness in Promise Classes“. Diss. 23. Juli 2020.

Dose, Titus (2020c). „Further oracles separating conjectures about incompleteness in the finite domain“. In: *Theoretical Computer Science* 847 (22. Dez. 2020), S. 76–94. DOI: 10.1016/j.tcs.2020.09.040.

Dose, Titus und Christian Glaßer (2019). *NP-Completeness, Proof Systems, and Disjoint NP-Pairs*. Techn. Ber. TR19-050. Electronic Colloquium on Computational Complexity. URL: https://eccc.weizmann.ac.il/report/2019/050/.

Egidy, Fabian, Christian Glaßer und Martin Herold (2023). „Upward Translation of Optimal and P-Optimal Proof Systems in the Boolean Hierarchy over NP“. In: *48th International Symposium on Mathematical Foundations of Computer Science* (Bordeaux, Frankreich, 28. Aug. 2023). Hrsg. von Jérôme Leroux, Sylvain Lombardy und David Peleg. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl: Leibniz-Zentrum für Informatik, Aug. 2023, 44:1–44:15. DOI: 10.4230/LIPIcs.MFCS.2023.44.

Ehrmanntraut, Anton, Fabian Egidy und Christian Glaßer (2022). „Oracle with P = NP ∩ coNP, but No Many-One Completeness in UP, DisjNP, and DisjCoNP“. In: *47th International Symposium on Mathematical Foundations of Computer Science* (Wien, 22. Aug. 2022). Hrsg. von Stefan Szeider, Robert Ganian und Alexandra Silva. Bd. 241. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl: Leibniz-Zentrum für Informatik, Aug. 2022, 45:1–45:15. DOI: 10.4230/LIPIcs.MFCS.2022.45.

Fenner, Stephen A., Lance Fortnow, Stuart A. Kurtz und Lide Li (2003). „An oracle builder's toolkit“. In: *Information and Computation* 182.2 (1. Mai 2003), S. 95–136. DOI: 10.1016/S0890-5401(03)00018-X.

Fenner, Stephen A., Lance Fortnow, Ashish V. Naik und John D. Rogers (1996). „Inverting onto functions“. In: *Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity* (Philadelphia, Pennsylvania, 24. Mai 1996). Washington, District of Columbia, USA: IEEE Computer Society Press, Mai 1996, S. 213–222. DOI: 10.1109/CCC.1996.507683.

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.

Fischer, Sophie, Lane A. Hemaspaandra und Leen Torenvliet (1995). „Witness-isomorphic reductions and the local search problem“. In: *Mathematical Foundations of Computer Science 1995* (Prag, Tschechische Republik, Aug. 1995). Hrsg. von Jiří Wiedermann und Petr Hájek. Bd. 969. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 277–287. DOI: 10.1007/3-540-60246-1_134.

Fortnow, Lance und John D. Rogers (1993). *Separability and one-way functions*. Techn. Ber. 93-14. Chicago, Illinois, USA: Department of Computer Science, University of Chicago, 30. Aug. 1993. URL: https://newtraell.cs.uchicago.edu/research/publications/techreports/TR-93-14.

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.

Glaßer, Christian, Christian Reitwießner und Victor Selivanov (2011). „The shrinking property for NP and coNP“. In: *Theoretical Computer Science* 412.8 (4. März 2011), S. 853–864. DOI: 10.1016/j.tcs.2010.11.035.

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. 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.

Goldberg, Paul W. und Christos H. Papadimitriou (2018). „Towards a unified complexity theory of total functions“. In: *Journal of Computer and System Sciences* 94 (1. Juni 2018), S. 167–192. DOI: 10.1016/j.jcss.2017.12.003.

Goldreich, Oded (2008). *Computational Complexity: a Conceptual Perspective*. Cambridge, Massachusetts, USA: Cambridge University Press. 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.

Harsha, Prahladh, Daniel Mitropolsky und Alon Rosen (2023). „Downward Self-Reducibility in TFNP“. In: *14th Innovations in Theoretical Computer Science Conference* (Cambridge, Massachusetts, USA, 10. Jan. 2023). Hrsg. von Yael Tauman Kalai. Bd. 251. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl: Leibniz-Zentrum für Informatik, 67:1–67:17. DOI: 10.4230/LIPIcs.ITCS.2023.67.

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.

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.

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

Johnson, David S., Christos H. Papadimitriou und Mihalis Yannakakis (1988). „How easy is local search?“ In: *Journal of Computer and System Sciences* 37.1 (1. Aug. 1988), S. 79–100. DOI: 10.1016/0022-0000(88)90046-3.

Karp, Richard M. (1972). „Reducibility among Combinatorial Problems“. In: *Complexity of Computer Computations* (Yorktown Heights, New York, USA, 20. März 1972). Hrsg. von Raymond E. Miller, James W. Thatcher und Jean D. Bohlinger. Boston, MA: Springer, S. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

Khaniki, Erfan (2022). „New Relations and Separations of Conjectures about Incompleteness in the Finite Domain“. In: *The Journal of Symbolic Logic* 87.3 (Sep. 2022), S. 912–937. DOI: 10.1017/jsl.2021.99.

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* (New Deli, Indien, 13. Dez. 2000). Hrsg. von Sanjiv Kapoor und Sanjiva Prasad. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 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.

Koucký, Michal (2023). „Automata and Formal Languages: Shall we let them go?“ In: *Bulletin of EATCS* 140.2 (25. Mai 2023). URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/759.

Kozen, Dexter (1997). *Automata and Computability*. New York, USA: Springer. ISBN: 978-0-387-94907-9.

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, S. 115–116. URL: https://www.mathnet.ru/ppi914. Eine Übersetzung erscheint bei Trakhtenbrot (1984, S. 399–400).

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. 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. URL: https://citeseerx.ist.psu.edu/pdf/bec3958d845653cfa73493258d3b550a17e8defd. Archivierte Fassung des Originals.

Meyer, Albert R. und Michael S. Paterson (1979). *Whith what frequency are apparently intractable problems difficult?* Techn. Ber. 126. Cambridge, Massachusetts, USA: Laboratory for Computer Science, MIT. URL: https://hdl.handle.net/1721.1/148954.

Papadimitriou, Christos H. (1994). *Computational Complexity*. Reading, Massachusetts, USA: Addison-Wesley. ISBN: 978-0-201-53082-7.

Post, Emil L. (1944). „Recursively enumerable sets of positive integers and their decision problems“. In: *Bulletin of the American Mathematical Society* 50.5, S. 284–316. DOI: 10.1090/S0002-9904-1944-08111-1.

Pudlák, Pavel (2013). *Logical Foundations of Mathematics and Computational Complexity*. Heidelberg: Springer International Publishing. ISBN: 978-3-319-00118-0.

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

Rackoff, Charles (1982). „Relativized Questions Involving Probabilistic Algorithms“. In: *Journal of the ACM* 29.1 (Jan. 1982), S. 261–268. DOI: 10.1145/322290.322306.

Razborov, Alexander A. (1994). „On provably disjoint NP-pairs“. In: *BRICS Report Series* 1.36 (30. Nov. 1994). DOI: 10.7146/brics.v1i36.21607.

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.

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

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

Trakhtenbrot, Boris 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, S. 189–201. DOI: 10.1016/0304-3975(79)90044-6.

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

Welsh, Dominic (1993). *Complexity: Knots, Colourings and Countings*. Cambridge, Massachusetts, USA: Cambridge University Press. ISBN: 978-0-521-45740-8.

Erklärung

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig verfasst zu haben, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, 20. Februar 2024

. . . .
Anton Ehrmanntraut