Komplexität von Suchproblemen und Beweissystemen

Anton Ehrmanntraut

5. Dezember 2023

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 an, 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 eine 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 z.B. 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 Lösungsmöglichkeiten 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, welche dieser Aufgaben effizient durchgeführt werden können, und somit als umsetzbar für Computer angesehen werden können.

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. In anderen Worten: wird die Eingabe, z.B. der Graph bei Beispiel (3), doppelt so groß, dann braucht dieser effiziente Algorithmus c-mal so lange.

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 eine 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 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 sogar 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 Sequenzkalüls.) Dann stand die Komplexitätstheorie der späten 1950er nah an der Automatentheorie und der Theorie der formalen Sprachen, als da 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 ressourcenfokussierte Komplexitätstheorie fort, die seit den 1960ern den Ressourcenverbrauch von Algorithmen als zentrale 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. Das 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 als 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 Suchprobleme 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 Festhaltung an Entscheidungsproblemen an, und will sich in vier Forschungsdesiderata 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 der

- die Lösung – falls sie existieren sollte – höchstens polynomiell länger als das 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 genanntne Suchprobleme bilden NP-Suchprobleme. Hierbei werden die „negativen Antworten" als „ex. 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 lange 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.

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.

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 (gerade 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 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 in der 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 für das Suchproblem (1) der Faktorisierung zu. 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 u. a. (2003) können die Vermutung „alle totalen NP-Suchprobleme sind effizient lösbar" in verschiedensten äquivalenten Formulierungen charakterisieren, so zum Beispiel als Invertierbarkeit von surjektiven Funktionen, oder als das effiziente Lösen vom Suchproblem (7) unter Angabe einer nichtdeterministische Turing-Maschine, die `SAT` erkennt. Fenner u. a. fassen diese jeweils äquivalenten Charakterisierungen unter der Hypothese `Q` zusammen. Für diese Arbeit werden wir hier folgende Formulierung von `Q` anwenden:

Vermutung 1.1 (Q, Fenner u. a., 2003)**.** *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.*

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 zu „Beweissystemen" im intuitiven Sinn. Wir gehen das bei 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. Damit 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 ein Beweissystem für die Tautologien, aber wie bereits angesprochen keins mit *polynomiell langen* Beweisen. Zumindest für die Resolution bildet damit (8) kein NP-Suchproblem. Existiert ein 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, i.e. $f(w) = \varphi$.

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

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

Hat ein aussagenlogisches Beweissystem f für jede 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 obere Frage, ob (8) ein NP-Suchproblem ist, lässt sich äquivalent charakterisieren mit der Frage, 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–Reckow-Programm* (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, sowie auch Beweissysteme allgemein für beliebige Mengen (und nicht nur Tautologien) betrachtet.

Die Existenz von optimalen Beweissystemen bzw. p-optimalen (d.i. optimal in de 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

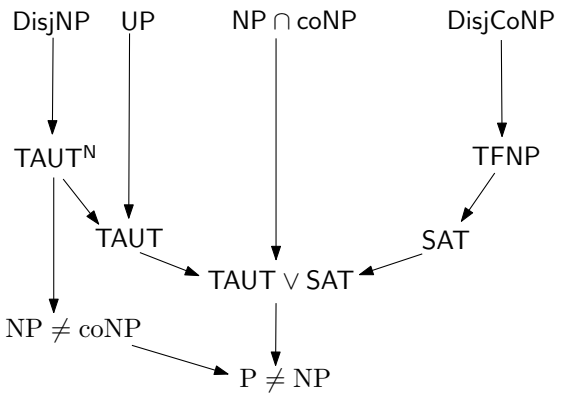


Abbildung 1: Implikationen zwischen Pudláks Hypothesen (2017). Beachte dass diese Implikationen relativieren.

p-optimalen bzw. optimalen Beweissystemen, Arithmetik mit polynomiell beschränkten Quantoren („*bounded arithmetic*“) und der Existenz von vollständigen Elementen sogenannter *Promise-Klassen*. Promise-Klassen sind solche Komplexitätsklassen, die durch speziell operierende Turing-Maschinen mit speziellen Eigenschaften erkannt werden können, wobei diese Eigenschaften üblicherweise über (Nicht)determinismus und polynomiell Laufzeit hinaus gehen. Für die Klasse UP $\subseteq$ 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 nichtdeterministischne Rechenweg akzeptieren darf.

Razborov (1994) zeigt hierbei als erstes eine Verbindung zwischen der Promise-Klasse DisjNP und der Existenz von optimalen aussagenlogischen Beweissystemen (für TAUT). Viele weitere Beziehungen zur Existenz vollständiger Elemente der Promise-Klassen UP, NP $\cap$ 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 $\leq^\text{p}_\text{m}$-vollst. Menge für NP mit p-opt. Beweissystem
TAUT	:	es ex. keine $\leq^\text{p}_\text{m}$-vollst. Menge für coNP mit p-opt. Beweissystem
TAUT^N	:	es ex. keine $\leq^\text{p}_\text{m}$-vollst. Menge für coNP mit opt. Beweissystem
NP $\cap$ coNP	:	es ex. keine $\leq^\text{p}_\text{m}$-vollst. Menge für UP
UP	:	es ex. keine $\leq^\text{p}_\text{m}$-vollst. Menge für NP $\cap$ coNP
DisjNP	:	es ex. kein $\leq^{\text{pp}}_\text{m}$-vollst. disjunktes NP-Paar für DisjNP
DisjCoNP	:	es ex. kein $\leq^{\text{pp}}_\text{m}$-vollst. disjunktes coNP-Paar für DisjCoNP

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

Es muss hervorgehoben werden, dass Pudlák die Hypothese TAUT nicht in dieser Form formuliert hat, sondern als Aussage über die Nicht-Existenz eines p-optimalen Beweissystems speziell für TAUT, genau wie anfangs des Abschnitts gefragt wurde. (In seiner Notation die Hypothese CON.) Die beiden Charakterisierungen sind aber äquivalent; Gesagtes gilt analog auch für SAT (vgl. Abschnitt 2.5).

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 Sepziellen nach natürlichen plausiblen stärkeren Hypothesen (die bspw. 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 *Pudlákssches 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 gehen wir noch auf den „Orakel“-Teil des Pudlákschen Programms ein.

Orakel und Relativierungen

Orakel und Orakel-Turing-Maschinen ist ein Begriff aus der Berechenbarkeitstheorie um die relative Schwierigkeit von algorithmischen Entscheidungsproblemen zu untersuchen, die über *berechenbar vs. unberechenbar* hinaus gehen. 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. 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 entscheiden 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 dieser 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 (selbst mit Orakel-Zugriff auf A) kann L entscheiden.

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 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 dieser 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 weitere Orakel konstruieren, um Hypothesen (unter relativierbaren Beweisen) zu trennen.

Beitrag und Überblick

Der Aufbau der Arbeit und die einzelnen Beiträge seien 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. Wir werden diese außerdem mit Funktionenklassen gegenüberstellen und einen Reduktions- und Vollständigkeits-Begriff über die sogenannte Levin-Reduzierbarkeit auf NP-Suchproblemen definieren.

Der Rest dieses Kapitels hat dabei den Charakter eines Überblickswerks bzw. eines Surveys. Zum einen wird die Beziehung zwischen NP-Suchproblemen und NP-Entscheidungsproblemen erläutert,

was auch das *search-reduces-to-decision*-Argument umfasst. Es werden Ergebnisse zusammen getragen, wann dieses Argument zutrifft, und wann es insbesondere nicht zutrifft. Zum anderen werden Arbeiten vorgestellt und eingeordnet, welche gemeinsame Eigenschaften und Strukturen von NP-Suchproblemen untersuchen. Besonders interessant sind hierbei die vollständigen NP-Suchprobleme, die sich ähnlich zu den sonst üblichen NP-vollständigen Mengen/Entscheidungsproblemen verhalten. Hierbei lässt sich – analog zur Berman–Hartmanis-Isomorphievermutung – zu vielen NP-vollständigen Suchproblemen eine gemeinsame Struktur bzw. Isomorphie erkennen.

In Kapitel 4 werden wir Hypothesen zu NP-Suchproblemen und die Aussage Q in das Pudlákssche Programm einordnen. Erstens untersuchen wir, ob die (Levin-)Vollständigkeit eines NP-Suchproblems mit der (Karp-)Vollständigkeit des entsprechenden NP-Suchproblems übereinstimmt. Insbesondere wird plausibilisiert, dass diese Übereinstimmung nicht gilt. Zweitens werden wir (nicht-relativierbare) Charakterisierungen von Q durch Fenner u. a. (2003) und Messner (2000) verallgemeinern und relativieren, womit wir als Nebeneffekt auch präzise zu (relativierbaren) Implikationen zwischen Q und den Pudlákschen Hypothesen kommen. Drittens zum Abschluss des Kapitels, wieder in einer Form eines Surveys, wird das Pudlákssche Programm durch Hinzufügen weiterer Hypothesen erweitert und verfeinert, die in der Literatur diskutiert werden. Hinzu erarbeiten wir eine Übersicht über die Implikationen zwischen diesen Hypothesen, bzw. Orakeln welche diese Implikationen unter relativierbaren Beweisen trennen. Im Wesentlichen vergrößern wir Abbildung 1 zu Abbildung 6.

In Kapitel 5 werden zwei Orakel konstruiert, die TODO: Dinge machen.

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

2 Grundlagen

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

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

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*. Beachte, dass eine totale bijektive Funktion $A \to B$ einen Mengenisomorphismus zwischen A und B bildet. Binäre Relationen über Wörtern aus Σ^*, welche nicht unbedingt Funktionen sind, verstehen wir manchmal auch aus historischen Gründen als (partielle) Multifunktionen, dem Begriff der „*partial multivalued function*“ nachempfunden.

Für eine binäre Relation $R \subseteq \Sigma^* \times \Sigma^*$ schreiben wir $\text{Proj}(R)$ für die Menge $\{x \mid (x, y) \in R\}$. Für ein Wort $x \in \Sigma^*$ 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. $\textit{set-}Q(n) \stackrel{\text{df}}{=} \{0, 1, \ldots, n\}$ schreiben um die Relation $Q = \{(a, b) \mid b \leq a\}$ zu definieren. Falls f eine Funktion bzw. funktional ist, meinen wir mit $f(x)$ wie üblich das *Bildelement* der Funktion f meinen, und nicht die *Bildmenge*.

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

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

alle $x \in \mathrm{dom}(f)$. Die Funktion f nennen wir *ehrlich* wenn ein Polynom q existiert sodass $q(|f(x)|) \geq |x|$ für alle $x \in \mathrm{dom}(f)$.

Beachte dass Funktionen nur spezielle Relationen sind. Wenn also f eine Funktion ist, meinen wir mit „$f \in \mathrm{P}$“ dass *der Graph* von f in Polynomialzeit entschieden werden kann („Gegeben Tupel (x, y), gilt $f(x) = y$?“). Das ist eine schwächere Aussage als „$f \in \mathrm{FP}$“ die wie in üblicher Interpretation besagen soll, dass aus x das Bild $f(x)$ in Polynomialzeit berechnet werden kann.

Im Folgenden definieren wir noch den Begriff der *Verfeinerung*. Seien F, G zwei Multifunktionen. Wir nennen G eine *Verfeinerung* von F wenn $\mathrm{Proj}(F) = \mathrm{Proj}(G)$ und *set-*$G(x) \subseteq$ *set-*$F(x)$ für alle $x \in \mathrm{Proj}(F)$ (bzw. äquivalent $\in \mathrm{Proj}(G)$). Ist F eine Multifunktion, und $\mathcal{G}$ eine Klasse von Multifunktion, schreiben wir $F \in_{\mathrm{c}} \mathcal{G}$ wenn $\mathcal{G}$ eine Verfeinerung $G \in \mathcal{G}$ von F enthält. Für zwei Klassen $\mathcal{F}, \mathcal{G}$ von Multifunktionen schreiben wir $\mathcal{F} \subseteq_{\mathrm{c}} \mathcal{G}$ falls für jede Multifunktion $F \in \mathcal{F}$ auch $F \in_{\mathrm{c}} \mathcal{G}$ 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 invertierbar 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 $\preceq_{\mathrm{lex}}$ auf den endlichen Wörtern. Diese Übereinstimmung ist nach den Eigenschaften des Ordnungsisomorphismus auch kompatibel mit der Identifikation von Wörtern mit Zahlen. Beachte dass der Längenoperator $|\cdot|$ ordnungserhaltend ist: wenn $a \leq b$ (oder eben äquivalent $a \preceq_{\mathrm{lex}} b$) für zwei Wörter $a, b \in \Sigma^*$ dann ist auch $|a| \leq |b|$, bzw. ist das Wort a höchstens so lang wie das Wort b. Mit den Ausdrücken 0^n und 1^n meinen wir immer die zwei Wörter $000\cdots$ und $111\cdots$ aus Σ^n.

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

2.2 Maschinenmodell

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

Wir betrachten zunächst deterministische TM. Sei M eine deterministische TM, und x eine Eingabe. Dann induziert eine Berechnung $M(x)$ einen Rechenweg α, der in einem ausgezeichnetem Zustand

q terminiert. Wir sagen dann auch, dass α *der Rechenweg von Berechnung* $M(x)$ ist. Wenn der terminierende Zustand q dieses Rechenwegs α ein akzeptierender Zustand ist, dann sagen wir auch dass $M(x)$ *mit Ausgabe* y *akzeptiert* oder kurz $M(x)$ *akzeptiert* wobei y jenes Wort ist, welches auf dem Ausgabeband steht. Wenn ansonsten der terminierende Zustand q kein akzeptierender Zustand ist, dann sagen wir dass $M(x)$ *(ohne Ausgabe) ablehnt*.

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

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

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

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

- Die von N *entschiede Sprache* ist die Menge

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

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

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

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.

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

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

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

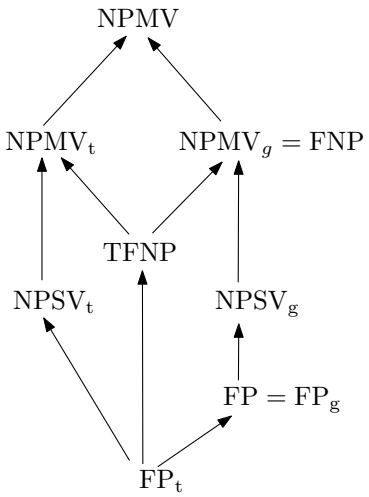


Abbildung 2: Inklusionen zwischen den in dieser Arbeit definierten Funktionenklassen. Eine Pfeil von $\mathcal{F}$ nach $\mathcal{G}$ sagt aus dass $\mathcal{G} \subseteq \mathcal{F}$.

2.3 Komplexitätsklassen

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

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

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

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

Die nichtdeterministischen Varianten $\mathrm{NE}, \mathrm{NEE}$ und Komplementklassen $\mathrm{coNE}, \mathrm{coNEE}$ sind analog definiert.

Die Funktionenklassen $\mathrm{FP}, \mathrm{NPMV}, \mathrm{NPSV}$ ist analog definiert (Selman, 1994):

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

Wir definieren $\mathrm{NPMV}_{\mathrm{t}}$ als die Teilmenge von NPMV der Multifunktionen, die linkstotal sind. Analog $\mathrm{NPSV}_{\mathrm{t}}$. Die Inklusionen zwischen den einzelnen Funktionenklassen ist in Abbildung 2 skizziert. Ist für eine Funktion $f \in \mathrm{FP}$ auch $f^{-1} \in_{\mathrm{c}} \mathrm{FP}$, also eine (funktionale) Verfeinerung g von f^{-1} in FP, dann sagen wir auch, dass f *p-invertierbar* ist. Beachte, dass die Ehrlichkeit von f eine notwendige Bedingung für die p-Invertierbarkeit von f ist.

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

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

Analog können wir die Klasse DisjCoNP aller disjunkten coNP-Paare definieren. ◁

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

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 $A \subseteq P$ und $B \subseteq \overline{P}$.

Reduktionen

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

- $A \leq^{\mathrm{p}}_{\mathrm{T}} B$ wenn A mittels einer Orakel-PTM mit Orakel B entschieden werden kann, bzw. $A \in \mathrm{FP}^B$ (Turing- bzw. Cook-Reduzierbarkeit; siehe unten für einen präzisen Orakel-Begriff).
- $A \leq^{\mathrm{p}}_{\mathrm{m}} B$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (Many-one- bzw. Karp-Reduzierbarkeit).
- $A \leq^{\mathrm{p}}_{1} B$ wenn eine injektive Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$ (One-one-Reduzierbarkeit).
- $A \leq^{\mathrm{p}}_{1,\mathrm{i}} B$ wenn eine injektive und p-invertierbare Funktion $f \in \mathrm{FP}$ existiert mit $x \in A \iff f(x) \in B$.

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

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

Auf den Paaren aus DisjNP und DisjCoNP hat sich folgender Begriff von Reduzierbarkeit herausgebildet.[3] Seien $(A, B), (C, D)$ zwei disjunkte NP-Paare (bzw. zwei disjunkte coNP-Paare):

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

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

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

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

Sei $\mathcal{C}$ eine Komplexitätsklasse und $\preceq$ eine der obigen Reduktionsordnungen. Wie üblich nennen wir nun eine Sprache A *$\preceq$-hart für $\mathcal{C}$* wenn A eine obere Schranke für $\mathcal{C}$ geordnet über $\preceq$ ist (d.h. $B \preceq A$ für alle $B \in \mathcal{C}$). Wir nennen A *$\preceq$-vollständig für $\mathcal{C}$* wenn $A \in \mathcal{C}$ ein größtes Element von $\mathcal{C}$ geordnet über $\preceq$ ist (d.h. $B \preceq A$ für alle $B \in \mathcal{C}$ und $A \in \mathcal{C}$). Auf Grundlage der Existenz universeller effizienter Turing-Maschinen können für die Klassen P und NP jeweils eine kanonische $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge angegeben werden. Für NP ist diese

Definition 2.2.

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

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

Beweis. Die Zugehörigkeit $\mathtt{KAN} \in \mathrm{NP}$ folgt unmittelbar aus der Existenz universeller nichtdeterministischer Turing-Maschinen mit polynomiellen Overhead.

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

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

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

Polynomialzeit-Isomorphie

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

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

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

- $A \equiv^{\mathrm{p}} B \overset{\mathrm{df}}{\iff}$ es existiert eine bijektive und p-invertierbare Funktion $f \in \mathrm{FP}$ 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 $\equiv^{\mathrm{p}}_{1,\mathrm{i}}$-äquivalente Sprachen dann p-isomorph sind, wenn die jeweiligen Reduktionsfunktionen verlängernd sind.

Satz 2.4 (Berman und Hartmanis, 1977)**.** *Gilt $A \leq^{\mathrm{p}}_{1,\mathrm{i}} B$ via f und $B \leq^{\mathrm{p}}_{1,\mathrm{i}} A$ via g, und f und g sind verlängernd, dann gilt $A \equiv^{\mathrm{p}} B$.*

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

Definition 2.5. Eine Sprache $A \neq \emptyset$ heißt (Berman–Hartmanis-)*paddable* genau dann wenn eine injektive und p-invertierbare Funktion $g \in \mathrm{FP}$ 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. ◁

Es gilt:

Satz 2.6 (Berman und Hartmanis, 1977, Thm. 5, 7)**.** *(1) Ist A paddable so gibt es für jedes B mit $B \leq^{\mathrm{p}}_{\mathrm{m}} A$ eine injektive p-invertierbare verlängernde Funktion die $B \leq^{\mathrm{p}}_{1,\mathrm{i}}$ realisiert.*

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

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

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

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

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

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

2.4 Orakel und Relativierungen

Wie in der Einleitung schon angesprochen, ist die Idee hinter Orakel-Berechnungen 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 $q_?, q_{\text{yes}}, y_{\text{no}}$.

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 den Zustand q_{no}.

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

Wir können nun die relativierten Komplexitätsklassen $\text{P}^O, \text{NP}^O, \text{FP}^O, \text{NPMV}^O, \ldots$ relativ zu einem gegebenen Orakel O definieren, wobei in der jeweiligen Definition die TM mit OTM ersetzt werden, die Zugriff auf das Orakel O haben. Diese Relativierung überträgt sich auch auf unsere weiteren Definitionen, wie z.B. Reduktion und Vollständigkeit. Wir schreiben z.B. $A \leq_{\text{m}}^{\text{p},O} B$ wenn eine Funktion $f \in \text{FP}^O$ existiert mit $x \in A \leftrightarrow f(x) \in B$. Die kanonische NP-vollständige Menge $\texttt{KAN}$ kann ebenso zu $\texttt{KAN}^O$ relativiert werden. Zur Vollständigkeit:

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

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

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

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

2.5 Beweissysteme

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

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

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

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

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

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

Dieses Beweissystem hat kurze Beweise.

Cook und Reckhow machen dagegen die Beobachtung im Fall von der Menge TAUT der aussagenlogischen Tautologien die Beobachtung, dass TAUT genau dann ein Beweissystem mit kurzen Beweisen hat, wenn NP = coNP. Diese Einsicht motivierte das sogenannte *Cook–Reckhow-Programm*: man nähert sich der Frage NP ≠ 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 ist 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 äquivalent auch $g \leq^{\mathrm{p}}_{\mathrm{m}} f$ schreiben um zu sagen, dass f das Beweissystem g p-simuliert

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 2.11. 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 $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit von f für die Teilmenge der Funktionen aus FP mit Bildmenge L. ◁

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

Jedes Beweissystem mit kurzen Beweisen ist auch optimal.

Beobachtung 2.12. *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 Polynom q sodass μ jedem $x \in L$ einem 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 Polyom 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 die Voraussetzung auf NEE = coNEE ab.)

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

Beobachtung 2.13. *(1) Ist $A \in \mathrm{P}$, dann existiert ein p-optimales Beweissystem für A mit kurzen Beweisen.*

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

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

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

Diese Funktion ist definitiv ein Beweissystem für A. Sie ist in FP, ist ja der Test „$x \in A$“ in Polynomi-

alzeit möglich. Klar ist auch dass h kurze Beweise hat. Dieses Beweissystem ist p-optimal, denn wenn g ein weiteres Beweissystem für A ist, und wenn w ein g-Beweis für x ist, dann ist $h(g(w)) = h(x) = x$, also $g(w)$ ein h-Beweis für x, wie gewünscht.

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

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

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

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

Beobachtung 2.14. *Sei $A \in \text{NP}$ 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 2.12. 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. $\leq^{\text{p}}_{\text{m}}$ nach unten überträgt:

Lemma 2.15 (Messner, 2000, Thm. 3.2)**.** *Hat A ein (p-)optimales Beweissystem und $B \leq^{\text{p}}_{\text{m}} A$, dann hat auch B ein (p-)optimales Beweissystem.*

Beweis. Sei f die Reduktionsfunktion, die $B \leq^{\text{p}}_{\text{m}} A$ realisiert, und sei h ein p-optimales Beweissystem für A. Definiere TODO: Skizze

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

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

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

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

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

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

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

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

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

Korollar 2.16. *Folgende Aussagen sind äquivalent:*

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

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

Analoge Äquivalenzen gelten für coNP*.*

Das erklärt auch die Form, in der wir in der Einleitung die Hypothese $\mathsf{SAT}, \mathsf{TAUT}$ gewählt haben. Ursprünglich sagte die Hypothese TAUT aus, dass kein p-optimales aussagenlogische Beweissystem existiert, also kein Beweissystem für die coNP-vollständige Menge $\mathtt{TAUT}$ existiert. Mit vorigem 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 $\mathtt{TAUT}$ 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 $\mathtt{TAUT}$.

Die hier gewählte Charakterisierung hat vor allem den Vorteil, dass $\mathsf{SAT}, \mathsf{TAUT}, \mathsf{TAUT}^{\mathsf{N}}$ auf natürliche Weise relativieren. So relativiert beispielsweise SAT auf Orakel O zur Aussage „kein $\leq_{\mathrm{m}}^{\mathrm{p},O}$-vollständiges $L \in \mathrm{NP}^O$ 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 vergleichen können, und welche Ähnlichkeiten zwischen den NP-Suchproblemen erkennbar sind. 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-Suchprobleme 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 Vollständigkeits-Begriff.

Abschließen wird in Abschnitt 3.4 noch der Forschungsstand zur gemeinsamen Struktur von vollständigen NP-Suchproblemen erläutert: die üblichen vollständigen NP-Suchprobleme teilen sich neben dieser Interreduzierbarkeit noch weitere Eigenschaften (z.B. Gleichmächtigkeit der Lösungsmengen, bekannt unter dem Begriff *parsimonious*), die hier erläutert und verglichen werden.

3.1 Definition von Suchproblemen

Wir geben hier noch einmal die Definition von Suchproblemen wieder, welche schon in der Einleitung erarbeitet wurde. Als Suchprobleme verstehen wir das algorithmische Problem, gegeben eine Probleminstanz x, eine entsprechende positive Lösungsinstanz y zu berechnen, oder negativ abzulehnen. Hier noch einmal 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 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 zweistellige 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) p-balaciert ist, d.h. es existiert ein Polynom q, sodass

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

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

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

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

Suchproblem zur Relation R:

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

Zur Erinnerung:

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

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

Entscheidungsproblem zur Relation R:

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

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

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

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

- $\texttt{rPERFECTMATCHING} \stackrel{\text{df}}{=} \{(G, M) \mid G$ ist ein Graph, M ein größtes Matching auf $G\}$. Das korrespondiert zu Suchproblem (3) aus der Einleitung.
- $\texttt{rSAT} \stackrel{\text{df}}{=} \{(\varphi, w) \mid \varphi$ ist eine aussagenlogische Formel, w erfüllende Belegung für $\varphi\}$. Das korrespondiert zu Suchproblem (7) aus der Einleitung.
- $\texttt{rVC} \stackrel{\text{df}}{=} \{((G, k), C) \mid G$ ist ein Graph, C eine Knotenüberdeckung, und $|C| \leq k\}$.
- $\texttt{rHAMCYCLE} \stackrel{\text{df}}{=} \{(G, P) \mid G$ ist ein Graph, P ein Zyklus der jeden Knoten genau einmal berührt$\}$.
- $\texttt{rANOTHERHAMCYCLE} \stackrel{\text{df}}{=} \{((G, P), P') \mid G$ ist ein Graph, P, P' je ein Zyklus der jeden Knoten genau einmal berührt, $P \neq P'\}$.
- $\texttt{rFACTORIZATION} \stackrel{\text{df}}{=} \{(n, (p_1, p_2, \ldots, p_k)) \mid n \in \mathbb{N}, n > 1$, alle p_i Primzahlen ungleich 2 oder n, und $n \stackrel{\text{df}}{=} p_1 \cdots p_k\}$. Das korrespondiert zu Suchproblem (1) aus der Einleitung.
- $\texttt{rFACTOR} \stackrel{\text{df}}{=} \{(n, p) \mid n \in \mathbb{N}$, $n > 1$ ist nicht prim, und p ist ein nichttrivialer Faktor von $n\}$.
- $\texttt{rSMALLFACTOR} \stackrel{\text{df}}{=} \{((n, a), p) \mid n \in \mathbb{N}$, $n > 1$ ist nicht prim, und p ist ein nichttrivialer Faktor von n und $p \leq a\}$.
- $\texttt{rGI} \stackrel{\text{df}}{=} \{((G, H), \sigma) \mid G, H$ 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.

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

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 $f \in \text{NPMV}$, für

die (der Graph) f in P liegt. Es lässt sich leicht sehen, dass die hier definierte Klasse FNP identisch zu Selman definierten Klasse $\mathrm{NPMV_g}$ ist, solange man Multifunktionen mit binären Relationen identifiziert.

Mit dieser Perspektivierung ist auch einfach zu definieren, was mit „Suchproblem lösen“ gemeint ist. Wir machen hierbei Gebrauch von Verfeinerungen (von Multifunktionen). Wir sagen, dass das Suchproblem zur NP-Relation R *in Polynomialzeit lösbar ist*, wenn $R \in_{\mathrm{c}} \mathrm{FP}$. 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 $y \in \textit{set-}R(x)$ gilt, bzw. in anderen Worten, eine Lösung y für x. 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.

Damit haben wir für die intuitive Aussage argumentiert, 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 $R \in_{\mathrm{c}} \mathrm{FP}$, dann gilt* $\mathrm{Proj}(R) \in \mathrm{P}$.

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

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

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

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

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

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

Definition 3.3 (TFNP)**.** Die Klasse TFNP ist die Teilmenge von $\texttt{FNP}$ 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 $\texttt{rFACTORIZATION}'$ und $\texttt{rFACTOR}'$. Für Megiddo und Papadimitriou befinden sich in $\texttt{TFNP}$ 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 $\mathrm{NPMV_t}$; es macht sich hier die gleiche Unterscheidung wie bei FNP vs. NPMV auf: 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 Teilmengen von $\mathrm{NPMV_t}$ ist, außer P = NP:

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

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

$$\textit{set-}f(x) = \begin{cases} \{1,2\} & \text{falls } x \in \texttt{SAT}, \\ \{1\} & \text{sonst.} \end{cases}$$

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

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

Wie in der Einleitung angesprochen kann sich die Aussage „alle Suchprobleme in TFNP lassen sich effizient lösen“ äquivalent als Hypothese Q schreiben.

Beobachtung 3.5 (Fenner u. a., 2003)**.** *Folgende Aussagen sind äquivalent:*

(1) Aussage Q*: für jede NPTM N mit $L(N) = \Sigma^*$ 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)⇒(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)⇒(1): Sei N eine NPTM mit $L(N) = \Sigma^*$. Klar ist, dass die Relation $R \stackrel{\mathrm{df}}{=} \{(x, \alpha) \mid N(x) \text{ akz. auf Rechenweg } \alpha\}$ 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 Unterklassen von TFNP hinzu. Jede dieser Unterklassen verinnerlicht 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 ist der identische wie auf den TFNP-Problemen (Megiddo und Papadimitriou, 1991).

3.2 Suchprobleme vs. Entscheidungsprobleme

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

In diesem Abschnitt werden 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.

Wir wollen zunächst auf die Beziehungen zwischen NP-Relationen und NP-Sprachen hinweisen. Tatsächlich haben wir bereits gesehen, dass wir über die Projektion jedem NP-Suchproblem bzw. NP-Relation ein korrespondierendes Entscheidungsproblem aus NP zuordnen kontnen. Tatsächlich lässt sich diese Zuordnung auch umkehren: zu jeder Sprache bzw. Entscheidungsproblem $L \in \mathrm{NP}$ existiert (mind.) eine NP-Relation R mit $L = \mathrm{Proj}(R)$, und zu jeder NP-Relation bzw. NP-Suchproblem R ist $\mathrm{Proj}(R) \in \mathrm{NP}$. Das ist die übliche „Zertifikats-Charakterisierung" von NP aus den Lehrbüchern.

Beobachtung 3.6 (Zertifikats-Definition von NP)**.** $\mathrm{NP} = \{\mathrm{Proj}(R) \mid R \textit{ ist eine NP-Relation}\}$.

Beweis. Wir müssen nur noch die Inklusion von links nach rechts zeigen. Sei hierfür $L \in \mathrm{NP}$ eine Sprache und N eine NPTM die L entscheidet, wobei die Laufzeit durch das Polynom p beschränkt ist. Definiere nun die Relation

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

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

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

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

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

Beweis. Die Richtung von links nach rechts ist klar, sind ja Suchprobleme schwieriger als Entscheidungsprobleme (Beobachtung 3.2 mit Beobachtung 3.6.)

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

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

Es ist leicht zu sehen dass $A \in \mathrm{NP}$. Also gilt nach Annahme aus $A \in \mathrm{P}$. Nun kann gegeben eine

Instanz x iterativ ein Präfix eines Zertifikats verlängert werden:

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

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

Search reduces to decision

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

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

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

Relativ zum Orakel $\mathrm{Proj}(R)$ gilt $\mathrm{P} = \mathrm{NP}$, ist ja $\mathrm{Proj}(R)$ vollständig für NP. Damit gilt mit vorigem Lemma 3.7 auch $\mathrm{FNP} \subseteq_{\mathrm{c}} \mathrm{FP}$ relativ zu $\mathrm{Proj}(R)$. Da $R \in \mathrm{FNP}$, gilt also auch $R \in_{\mathrm{c}} \mathrm{FP}$ relativ zu $\mathrm{Proj}(R)$. □

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

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

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

und

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

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

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

Durch Orakelfragen an $\mathrm{Proj}(\texttt{rSMALLFACTOR})$ kann dann mit binärer Suche ein solcher Faktor auch gefunden werden. In anderen Worten, $\texttt{rSMALLFACTOR} \in_{\mathrm{c}} \mathrm{FP}^{\mathrm{Proj}(\texttt{rSMALLFACTOR})}$.

Es lässt sich im Übrigen zeigen, dass $\mathrm{Proj}(\texttt{rSMALLFACTOR}) \in \mathrm{UP} \cap \mathrm{coUP}$. Die Projektion ist einerseits in UP: gegeben (n, a), n nicht prim, rate zunächst eine Primfaktorzerlegung von n mit aufsteigenden Primfaktoren. Akzeptiere genau dann wenn ein Faktor $p \leq a$ in dieser Zerlegung erhalten ist. Der Fundamentalsatz der Arithmetik sichert, dass die (aufsteigend geordnete) Primfaktorzerlegung eindeutig ist, heißt der oben skizzierte nichtdeterministische Polynomialzeit-Algorithmus akzeptiert auf höchstens einem Rechenweg. Auf ähnliche Weise lässt sich zeigen dass die Projektion in coUP liegt. Damit ergibt sich auch der auf S. 21 angegebene Stand, dass ein Kollaps von $\mathrm{UP} \cap \mathrm{coUP}$ mit P zur Folge hätte, dass $\texttt{rSMALLFACTOR} \in_{\mathrm{c}} \mathrm{FP}$.

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:

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

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

Für andere Suchprobleme existieren aber nichttriviale Möglichkeiten das Suchproblem auf das (natürliche) Entscheidungsproblem zu reduzieren, auch wenn das Entscheidungsproblem nicht in der Form einer Bisektion/Präfixsuche ist. Hierbei wird die spezifische Struktur des Problems ausgenutzt. Ein Beispiel ist $\texttt{rSAT}$: Gegeben Formel φ, teste mittels dem Orakel, ob $\varphi[x_1/0] \in \texttt{SAT}$ oder $\varphi[x_1/1] \in \texttt{SAT}$. 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 $\varphi[x_1/0] \in \texttt{SAT}$ 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 $\varphi[x_1/1] \in \texttt{SAT}$ ist analog.) Es gilt daher $\texttt{rSAT} \in \text{FP}^{\text{Proj}(\texttt{rSAT})}$. (Beachte aber, dass $\text{Proj}(\texttt{rSAT}) = \texttt{SAT}$ schon NP-vollständig ist. Damit folgt $\texttt{rSAT} \in \text{FP}^{\texttt{SAT}}$ schon aus Korollar 3.8.)

Ein weiteres nichttriviales Beispiel wäre die NP-Relation $\texttt{rGI}$. Zur Erinnerung: dieses Suchproblem sucht nach einem Graphisomorphismus zwischen zwei gegebenen Graphen. Deren Projektion ist mutmaßlich nicht NP-vollständig. Gleichzeitig gilt $\texttt{rGI} \in_{\text{c}} \text{FP}^{\text{Proj}(\texttt{rGI})}$: 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 für $\texttt{rLINDIV}$ möglich ist, scheint in der Literatur nicht untersucht.

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

Satz 3.9 (Impagliazzo und Sudan, 1991; Borodin und Demers, 1976, Thm. 5)**.** *Angenommen* $\text{E} \neq \text{NE}$ *oder* $\text{P} \neq \text{NP} \cap \text{coNP}$*. Dann existiert eine NP-Relation* R *mit* $\text{Proj}(R) \in \text{NP} - \text{P}$ *für die* $R \notin_{\text{c}} \text{FP}^{\text{Proj}(R)}$ *gilt.*

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

Satz 3.10 (Bellare und Goldwasser, 1994, Thm. 1.1; Impagliazzo und Sudan, 1991)**.** *Angenommen* $\text{EE} \neq \text{NEE}$ *oder* $\text{NE} \neq \text{coNE}$*. Dann existiert eine Menge* $L \in \text{NP} - \text{P}$ *sodass* $R \notin_{\text{c}} \text{FP}^L$ *für jede NP-Relation* R *für* L*, d.h. für die* $\text{Proj}(R) = L$ *gilt.*

Beachte dass zu keiner dieser Relationen R aus den beiden vorigen Sätzen die Projektion $\text{Proj}(R)$ eine NP-Intermediate ist; $\text{Proj}(R)$ kann nicht $\leq^{\text{p}}_{\text{m}}$-vollständig sein, denn das wäre ein Widerspruch zu Korollar 3.8.

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

Definition 3.11. (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 für A robuste OTM M existiert, für die $\text{time}^A_M(x)$ polynomiell in abh. von $|x|$ wächst, i.e. M^A ist eine POTM (zumindest mit dem Orakel A angeschlossen). ◁

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.

Satz 3.12 (Balcázar, 1989)**.** *Sei* $A \in \text{NP}$*. Folgende Aussagen sind äquivalent:*

(1) A *ist selbsthelfend.*

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

Selbstreduzierbarkeit in TFNP

Für *totale* Suchprobleme (i.e. aus TFNP) kann nicht sinnvoll gefragt werden, ob hier das Suchproblem auf das Entscheidungsproblem reduziert werden kann, ist ja für $R \in \text{TFNP}$ 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 können wir folgenden Begriff von Reduzierbarkeit definieren: Betrachte hierbei folgende Variante eine POTM-Transduktors relativ zu $R \in \text{TFNP}$: Dieser Transduktor ist wie ein üblicher PTM-Transduktor, hat zusätzlich aber Zugriff auf ein *funktionales* Orakel, in dem Sinne dass er Orakelfragen der Form „*gib mir ein Zertifikat y für x'*“ stellen kann. Das Orakel antwortet dann mit einem solchen 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: Wäre die TFNP-Relation $\texttt{rFACTOR}' \in \text{TFNP}$ (i.e., suche einen nichttrivialen Faktor für n, oder gebe „prim“ aus) nach unten selbstreduzierbar, dann würde das bedeuten dass ein Faktor von n effizient gefunden werden kann, wenn wir nach Faktoren von Zahlen $\leq n/2$ fragen dürfen. Welche TFNP-Probleme nach unten selbstreduzierbar sind, ist erstaunlich wenig untersucht, und eine Beforschung in dieser präzisen Formulierung wurde wohl erst durch Harsha, Mitropolsky und Rosen (2023) angetreten. Sie zeigen die Selbstreduzierbarkeit nach unten für folgendes TFNP-Problem „Iterate with source“, welches als ein kanonischer Repräsentant für die Unterklasse PLS (zur Erinnerung: *polynomial local search*) gilt.

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

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

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

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

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

Es ist offen ob $\texttt{rFACTOR}' \overset{?}{\in} \texttt{PLS}$ und zumindest unplausibel, weil unklar ist wie Faktorisierung als lokales Suchproblem repräsentiert werden kann. Tatsächlich zeigen die Autorinnen sogar die stärkere Konsequenz $\texttt{rFACTOR}' \overset{?}{\in} \text{UEOPL}$, was auch noch $\texttt{rFACTOR}' \in \text{PPAD}$ zur Folge hätte. Auch die Frage ob $\texttt{rFACTOR}' \overset{?}{\in} \text{PPAD}$ ist offen, und wurde breit untersucht. Eine positive Antwort wäre zumindest sehr überraschend (vgl. Harsha, Mitropolsky und Rosen, 2023, 67:15; siehe ebd. auch für eine Def. von UEOPL, PPAD). Insgesamt ist die Forschung bezüglich Selbstreduzierbarkeit nach unten für TFNP-Probleme (und allgemeiner auch für FNP-Probleme) noch sehr klein, und 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 Konzeptionalisierung 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 „kürzer“ über eine beliebige polynomiell wohlfundierte und längenbeschränkte Halbordnung auf den Wörtern verallgemeinern.

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 Redu-

zierbarkeit zwischen NP-Relationen herausgebildet (vgl. Papadimitriou, 1994, S. 229; Goldreich, 2008, S. 61; Arora und Barak, 2009, S. 50):

Definition 3.14 (Levin-Reduzierbarkeit)**.** Seien Q, R zwei NP-Relationen. Wir sagen dass Q *sich 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) $x \in \mathrm{Proj}(Q) \iff f(x) \in \mathrm{Proj}(R)$,

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

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

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

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

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

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

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.

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. *Wenn $R \leq_{\mathrm{L}}^{\mathrm{p}} Q$ dann gilt* $\mathrm{Proj}(R) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{Proj}(Q)$.

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. Intuitiv formt die Levin-Reduktion $\leq_{\mathrm{L}}^{\mathrm{p}}$ auf den Suchproblemen das Analog der Many-one-Reduktion $\leq_{\mathrm{m}}^{\mathrm{p}}$ auf den Entscheidungsproblemen.

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 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:

TODO: Skizze

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

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

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

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

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

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

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

i.e, $q(x)$ ist ein Zertifikat für x, wie gewünscht. Hierbei folgt die dritte Implikation nach Definition 3.14(2). □

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

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

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

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

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

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

Satz 3.18. *Die kanonische NP-Relation* $\mathtt{rKAN}$ *ist* $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$*-vollständig für* FNP*.*

Beweis. Es ist leicht zu sehen dass $\mathtt{rKAN} \in \mathrm{P}$, da „$\alpha$ ist Rechenweg für $N(x)$“ in Polynomialzeit (abh. von $|\alpha|, |N|, |x|, n$) verifiziert werden kann. Für die p-Balanciertheit von $\mathtt{rKAN}$ können wir voraussetzen, dass jeder Schritt eines Rechenwegs α als Konfiguration codiert ist, also Inhalt der Bänder, Kopfpositionen und Zustände. Damit ist jede Konfigureation als ein polynomiell langes Wort abhängig von $|N|$, $|x|$, n beschreibbar, also auch α als Liste von $\leq n$ vielen Konfigurationen nur polynomiell länger als $|N|$, $|x|$, n. Damit $\mathtt{rKAN} \in \mathrm{FNP}$.

Sei R eine beliebige NP-Relation mit Zertifikatsschranke r, i.e. $(x, y) \in R \implies |y| \leq r(|x|)$. Sei M die PTM welche R entscheidet, mit Laufzeitschranke p. Sei N eine NPTM welche auf Eingabe x zunächst ein Zertifikat $y, |y| \leq r(|x|)$ rät, und dann testet ob $M(x, y)$ akzeptiert. Die Laufzeit von N ist beschränkt auf $p(|(x, y)|) \in O(p(r(|x|)))$ (hier nutzen wir die effiziente Listencodierung von 2.1 aus). Sei daher q ein Polynom, welches die Laufzeit von N beschränkt.

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

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

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

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

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

Lemma 3.19. *Sei* R *eine* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständige NP-Relation für* FNP*. Es gelten folgende Aussagen:*

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

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

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

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

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

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

Satz 3.20 (Satz von Cook und Levin)**.** *Die NP-Relation* $\mathtt{rSAT}$ *ist* $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$*-vollständig für* FNP*.*

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

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

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

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

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

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

Und damit ist $f(x) \stackrel{\mathrm{df}}{=} C_{M,n,x}$, $f \in \mathrm{FP}$ eine Reduktionsfunktion von R nach $\texttt{rCSAT}$:

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

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

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

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

Nach Goldreich (2008, S. 104) sind die folgenden NP-Relationen jedenfalls definitiv $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{i}}$-vollständig: $\texttt{rSAT}$, $\texttt{rSETCOVER}$, $\texttt{rVC}$, $\texttt{rCLIQUE}$, $\texttt{r3COLORABILITY}$. Die von Papadimitriou (1994, S. 193–198) angegebene Reduktion $\texttt{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} \texttt{HAMCYCLE}$ lässt sich leicht zu $\texttt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} \texttt{rHAMCYCLE}$ erweitern, womit $\texttt{rHAMCYCLE}$ auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Ebenso ist $\texttt{rANOTHERHAMCYCLE}$ auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig (1994, S. 232). Damit gilt mit Lemma 3.19(3) auch die in Abschnitt 3.1 angegebene Äquivalenz NP = P genau dann wenn $\texttt{rSAT}$, $\texttt{rVC}$, …$\in_{\mathrm{c}}$ FP.

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

Frage 3.21. *Wenn* $\mathrm{Proj}(R)$ *eine* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständige Menge für* NP *ist, ist dann auch* R *eine* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständige NP-Relation für* FNP*?*

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

Vermutung 3.22 (Karp-vs-Levin-Vermutung; KvL)**.** *Es existiert eine NP-Relation* R *sodass* $\mathrm{Proj}(R)$ $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständig für* NP *ist, aber* R *ist nicht* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* FNP*.*

Obwohl diese Frage erstaunlich auf natürlich scheint, und zwei umfassende Reduktionsbegriffe der Komplexitätstheorie in Beziehung setzen versucht, gibt es erstaunlicherweise kaum Forschung welche sich dieser Hypothese annähert. Ein Beweis von KvL ist jedenfalls mindestens so schwer wie die P-NP-Frage, denn KvL $\Rightarrow$ P $\neq$ NP. In Abschnitt 4.1 werden wir diese Hypothese und dessen Beziehung zu anderen Hypothesen erarbeiten. Dort werden dann auch Argumente geliefert, die für diese negative Antwort sprechen.

Die oben genannte Frage lässt sich auf natürliche Weise abschwächen, indem man von der konkreten NP-Relation abstrahiert: Wenn L eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge für NP ist, existiert dann zumindest *eine* NP-Relation R_L für L (i.e., $\mathrm{Proj}(R) = L$) sodass R_L $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist? In anderen Worten, existiert ein hinreichend 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 3.23 (Buhrman, Kadin und Thierauf, 1998)**.** *Für jede Menge $L \in \mathrm{NP}$ die p-isomorph zu* SAT *ist, existiert eine NP-Relation R_L sodass $\mathrm{Proj}(R_L) = L$ und R_L auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist.*

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

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

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 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

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

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

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

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

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

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

denn jede potentielle Reduktion würde Definition 3.14(1) verletzen, i.e. die Many-one-Reduktion auf den jeweiligen Projektionen.

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

Vermutung 3.24 (TFNP)**.** *Es existiert keine NP-Relation $R \in \mathrm{TFNP}$ die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig für* TFNP *ist.*

Auch hier ist ein Beweis für diese Vermutung mindestens so schwer wie ein Beweis für NP ≠ coNP. Die Beziehung dieser Vermutung mit weiteren Vermutungen betreffend Promise-Probelemen wird in Kapitel 4 erarbeitet.

3.4 Zur gemeinsamen Struktur von vollständigen Suchproblemen

Feinere Reduktionsbegriffe

Aus den Erfahrungen in den Entdeckungen von NP-vollständigen Mengen bzw. der Entwicklung der hierfür notwendigen Reduktionen wurde intuitiv deutlich, dass die NP-vollständigen Mengen sich viele nicht-offensichtliche Eigenschaften teilen, die über „Equi-Lösbarkeit“ ($A \in \mathrm{P}$ genau dann wenn $B \in \mathrm{P}$ für $\leq^{\mathrm{p}}_{\mathrm{T}}$-vollständige Mengen A, B) hinaus geht. Das beginnt schon bei der üblichen Many-one-Reduktion, die eine Verstärkung der Turing-Reduktion darstellt. Hier wird auf die fundamental

ähnliche Struktur der $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigen Mengen A, B hingewiesen: eine Instanz x von A kann als ein „äquivalenter“ Fall $f(x)$ von B repräsentiert werden.

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

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

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

Definition 3.25 (Sparsame Reduktionen)**.** Seien Q, R NP-Relationen. Wir sagen dass sich Q auf R (in Polynomialzeit) *sparsam* (*„parsimonious“*) reduzieren lässt, bzw. $Q \leq^{\mathrm{p}}_{\mathrm{pars}} R$ wenn eine Funktion $f \in \mathrm{FP}$ existiert mit

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

◁

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

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

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 $\texttt{rSAT} \leq^{\mathrm{p}}_{\mathrm{pars}} \texttt{rKAN}$ und $\texttt{rKAN} \leq^{\mathrm{p}}_{\mathrm{pars}} \texttt{rSAT}$ angeben lassen können. Damit kann das Zählproblem zur rSAT-Instanz x als rKAN-Instanz $f(x)$ repräsentiert werden und umgekehrt – die beiden Zählprobleme sind relativ zum jeweils anderem gleich schwer. Auf eine weitere Präsentation der Komplexitätstheorie des Zählens muss hier 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.

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

Fischer, Hemaspaandra und Torenvliet (1995) gingen noch einen Schritt weiter und definieren *zertifikats-isomorphe* Reduktionen („witness-isomorphic reduction“) zwischen zwei NP-Relationen. Hier erhält die Reduktionsfunktion $A \to B$ auf den Instanzen nicht nur die *Anzahl* der Zertifikate, sondern es werden zusätzlich die Zertifikate für $x \in A$ mit den Zertifikaten für $f(x) \in B$ in eine effiziente in Polynomialzeit berechenbare Eins-zu-Eins-Korrespondenz gesetzt. Dieses Vorgehen ist intendiert als eine Generalisierung der p-Isomorphie (Berman und Hartmanis, 1977) nicht nur auf Instanzen sondern auch auf Zertifikaten.

Definition 3.26 (Zertifikats-Isomorphie; Fischer, Hemaspaandra und Torenvliet, 1995)**.** Seien Q, R NP-Relationen. Wir sagen dass sich Q auf R (in Polynomialzeit) *zertifikats-isomorph reduzieren lässt*, bzw. $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$, wenn Funktion $f : \Sigma^* \to \Sigma^*, f, f^{-1} \in \mathrm{FP}$ und Funktion $g : \Sigma^* \times \Sigma^* \to \Sigma^* \times \Sigma^*, g, g^{-1} \in \mathrm{FP}$, und

(1) $(x, y) \in Q \implies \exists z.\, g(x, y) = (f(x), z) \wedge (f(x), z) \in R$ (Vorwärts-Translation von Zertifikaten),

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

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

Beachte dass aus $f^{-1}, g^{-1} \in \mathrm{FP}$ insbesondere die Injektivität von f und g folgt. Die Punkte (1)–(2) in der Definition der Zertifikats-Isomorphie können alternativ auch folgendermaßen äquivalent formuliert werden: für jedes x gilt

$$g(Q \cap (\{x\} \times \Sigma^*)) = R \cap (\{f(x)\} \times \Sigma^*). \tag{3.2}$$

Damit ist auch leicht zu sehen, dass für alle $x \in \mathrm{Proj}(Q)$ die Funktion $g_x(y) = g(x, y)$ eine totale bijektive Abbildung zwischen den Zertifikaten für x und den Zertifikaten für $f(x)$ ist. Damit gilt insbesondere

$$|\mathit{set}\text{-}Q(x)| = |Q \cap (\{x\} \times \Sigma^*)| = |g(Q \cap (\{x\} \times \Sigma^*))| = |R \cap (\{f(x)\} \times \Sigma^*)| = |\mathit{set}\text{-}R(f(x))|$$

und wir sehen dass f eine sparsame Reduktion realisiert (und damit auch eine Many-one-Reduktion).

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

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

Lemma 3.27. *Seien Q, R zwei NP-Relationen. Falls $Q \leq^{\mathrm{p}}_{\mathrm{wi}} R$, dann ist $Q \leq^{\mathrm{p}}_{\mathrm{pars}} R$ und $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$ und $Q \leq^{\mathrm{p}}_{\mathrm{m}} R$.*

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

Wir müssen für Levin-Reduzierbarkeit nur noch zeigen, dass wir Zertifikate z für $f(x)$ wieder zu Zertifikaten für x rückübersetzen können. Das ist wegen der p-Invertierbarkeit von g möglich: Mit Definition 3.26(2) gilt

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

Nachdem $g^{-1} \in \mathrm{FP}$, lässt sich leicht eine entsprechende Translationsfunktion für die Levin-Reduktion angeben. □

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

Universelle Relationen

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

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

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

Definition 3.28 (Strenge NP-Relation)**.** Sei R eine NP-Relation mit Zertifikatsschranke q. Wir nennen R *streng* wenn für R gilt, dass $(x, y) \in R \implies |y| = q(|x|) > 0$. In anderen Worten, jedes Zertifikat y für x ist nicht ε und hat genau die Länge $q(|x|)$. $\lhd$

Dies sollte keine Einschränkung darstellen: zu jeder natürlichen NP-Relation R kann eine strenge NP-Relation R' angegeben werden, welche nur die Zertifikate mit Leerzeichen padded. Damit gilt $\mathrm{Proj}(R) = \mathrm{Proj}(R')$ und $R \equiv^{\mathrm{p}}_{\mathrm{L}}$.

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

Definition 3.29 (Universelle Relation; Agrawal und Biswas, 1992a)**.** Sei R eine strenge NP-Relation mit Zertifikatsschranke q. Die Relation R ist *universell* wenn alle drei folgenden Eigenschaften erfüllt sind:

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

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

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

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

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

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

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

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

Wir illustrieren diese Definition anhand von `rSAT`. Wir nehmen dafür an, dass `rSAT` als strenge NP-Relation gegeben ist. Die Formeln φ seien so codiert, dass in φ nur die Variablen $x_0, \dots, x_{|\varphi|-1}$ vorkommen. Zertifikate für φ sind dann Strings w der Form $\Sigma^{|\varphi|}$ sodass
(1) φ durch die Variablenbelegung $x_0 = w[0], x_1 = w[1], \dots$ erfüllt wird, und
(2) $w[j] = 0$ wenn x_j nicht in φ vorkommt.

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

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

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

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

Definition 3.30 (Projektive Levin-Reduktion)**.** Seien Q und R zwei strenge NP-Relationen. Die NP-Relation Q lässt sich über eine *projektive Levin-Reduktion* auf R reduzieren, wenn eine Funktion $f \in \mathrm{FP}$ existiert, welche die folgenden Bedingungen erfüllen:

(1) $f(x) = (z, \alpha)$ wobei $x, z \in \Sigma^*$ und $\alpha \in \mathbb{N}_{>0}^{q(|x|)}$ ist eine Sequenz von Indizes der Länge $q(|x|)$.

(2) $\{y[\alpha] \mid y \in \mathit{set}\text{-}R(z)\} = \mathit{set}\text{-}Q(x)$. ◁

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

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

Satz 3.31 (Agrawal und Biswas, 1992a)**.** *Sei R eine strenge NP-Relation. Folgende Aussagen sind äquivalent:*

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

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

Diese Äquivalenz gilt nur im unrelativierten Fall.

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

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

Korollar 3.32. *Wenn R eine universelle NP-Relation ist, dann ist R $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständig für* FNP*. Diese Aussage gilt nur im unrelativierten Fall.*

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

Die Abbildung 3 fasst noch einmal die relative Stärke der Vollständigkeits-Begriffe zusammen. Die eingezeichneten Trennungen werden im nächsten Abschnitt erläutert.

Lemma 3.33. *Es gelten die in Abbildung 3 eingezeichneten Inklusionen.*

Beweis. Mit Lemma 3.27 bleiben nur noch drei nichttriviale Implikationen offen:

1. Falls R universell ist und die beteiligten Funktionen *join* und *clp* injektiv und p-invertierbar sind, dann ist schon aus Satz 3.31 klar, dass R auch projektiv Levin-vollständig ist. Es existiert also für NP-Relation Q eine Funktion $f \in \mathrm{FP}$, und es gilt für eine Q-Instanz x dass $f(x) = (z, \alpha)$. Hier ist z die R-Instanz, auf die reduziert wird. Aus dem Beweis des Satzes von Agrawal und Biswas (1992a) geht hervor, dass sich dieses z aus der kombinierten Anwendung von *join* und *cpl* entsteht, also lässt sich auch aus z wieder aufgrund p-Invertierbarkeit die Instanz x zurückgewinnen. Es lässt sich leicht sehen, dass sich so eine $\leq_{\mathrm{L,1,i}}^{\mathrm{p}}$-Reduktion von Q nach R konstruieren lassen kann.

2. Falls R universell ist und die beteiligte Funktion *join* injektiv und p-invertierbar ist, dann ist $\mathrm{Proj}(R)$ auch paddable, und damit p-isomorph zu `SAT` (Agrawal und Biswas, 1992a, Thm. 8.2). Das lässt sich leicht nachvollziehen: durch durch an-*join*-en von Dummy-Instanzen an Instanz x lassen sich beliebige Werte in x hineincodieren, und durch die p-Invertierbarkeit wieder extrahieren. Konkret, sei $z_0 \notin \mathrm{Proj}(R)$ und $z_1 \in \mathrm{Proj}(R)$, dann definiere

$$h(x, y) \stackrel{\mathrm{df}}{=} \mathit{join}(x, z_{y[0]}, z_{y[1]}, \ldots, z_{y[|y|-1]}).$$

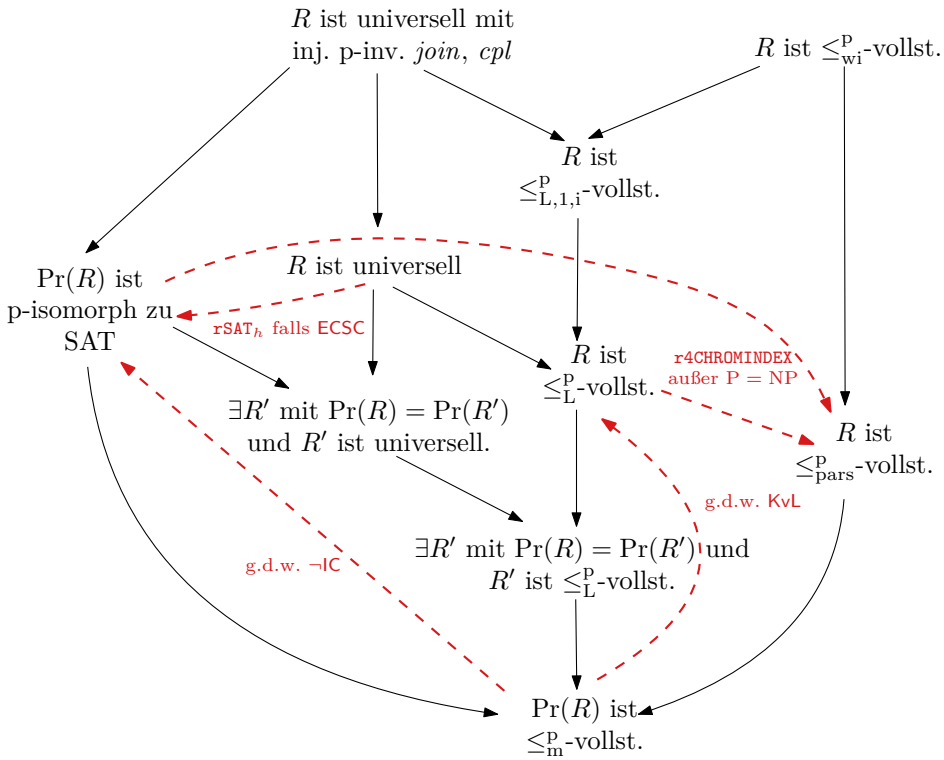


Abbildung 3: Implikationen zwischen den Vollständigkeits-Begriffen, wobei R eine beliebige aber feste NP-Relation ist. Ein unterbrochener Pfeile von A nach B sagt aus, dass ein Gegenbeispiel Q für die Implikation A $\Rightarrow$ B existiert, also eine NP-Relation Q die A erfüllt und gleichzeitig ¬B erfüllt.

Mit der p-Invertierbarkeit von *join* ist leicht zu sehen, dass h eine Padding-Funktion für $\mathrm{Proj}(R)$ ist.

3. Sei L eine Menge, die p-isomorph zu $\mathtt{SAT}$ ist. Dann existiert auch eine NP-Relation R' sodass $\mathrm{Proj}(R') = L$ und R' ist universell. Diese Aussage ist eine einfache Generalisierung von Beobachtung 3.23 (vgl. Agrawal und Biswas, 1992a, Prop. 8.5). □

Trennungen

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

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

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. 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

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

sogar universell ist. Wir haben also

Beobachtung 3.34. *Angenommen* ECSC *dann existiert eine NP-Relation R die universell ist, aber* $\mathrm{Proj}(R)$ *ist nicht p-isomorph zu* $\mathtt{SAT}$.

Nun werden wir uns auf die sparsamen Reduktionen konzentrieren. Zu einem Graphen G mit Knotenmenge $\{0, 1, \ldots, n-1\}$ können wir einen *Schnitt* als einen String $w \in \Sigma^n$ schreiben, wobei $V_0 \stackrel{\mathrm{df}}{=} \{i \mid i < n, w[i] = 0\}$ und $V_1 \stackrel{\mathrm{df}}{=} \{i \mid i < n, w[i] = 0\}$ 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

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

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

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

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

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

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

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

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

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

Zu einem Graphen G 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 Partitionierungen* 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

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

Satz 3.35 (Cai und Govorov, 2021 nach Edward und Welsh[4])**.** *Die NP-Relation* r4CHROMINDEX *ist nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig, außer* P = NP*.*

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

- Unter den Graphen mit $\chi'(G) = 4$ ist $K_{1,k}$ der einzige Graph mit eindeutiger 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. Wir skizzieren hier den Beweis für den Fall $\chi'(G) = 3$. Sei 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

4. 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 dieses Preprint 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 diese Interpretation der Isomorphievermutung (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

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 $x_1, \ldots, x_n$ vorkommen. Wir werden nun in Polynomialzeit entscheiden ob $\varphi \in$ SAT. Definiere eine zweite SAT-Formel

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

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

Wir haben nun

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

und damit SAT $\in$ P. □

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

Beobachtung 3.36. *Die NP-Relation* r4CHROMINDEX *ist* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig, und* Proj(r4CHROMINDEX) *ist p-isomorph zu* SAT*. Sie ist insbesondere nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig außer* P = NP*.*

Gleichzeitig ist nicht klar, ob sich dieses Ergebnis zur Universalität von r4CHROMINDEX verstärken kann. (Das würde Universalität von $\leq^{\mathrm{p}}_{\mathrm{pars}}$-Vollständigkeit trennen.) Weder ist ist klar, wie sich ein *building block* angeben kann, noch wie (für Aussage (2) von Satz 3.31) sich eine projektive Levin-Reduktion von rSAT auf r4CHROMINDEX angeben kann. Die wesentliche Schwierigkeit liegt darin, die *projektive* Natur der projektiven Levin-Reduktion umzusetzen: aus den Färbungen bzw. Zertifikaten kann nicht Bit für Bit eine Lösung herausgelesen werden, wie sie die Definition 3.29 (bzw. äquivalent Definition 3.30) verlangt.

Dies sei im Folgenden am etwas einfacherem Fall der 3-Kantenfärbbarkeit (die auch $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig ist) illustriert; die wesentliche Idee überträgt sich auch auf k-Kantenfärbbarkeit, $k \geq 3$. Holyer (1981) zeigt die $\leq^{\mathrm{p}}_{\mathrm{m}}$-Vollständigkeit der 3-Kantenfärbbarkeit, indem von 3SAT in CNF darauf reduziert wird. Das ist, gegeben eine 3CNFSAT-Formel φ in konjunktiver Normalform wird ein 3-regulärer Graph G konstruiert der 3-färbbar ist genau dann wenn φ erfüllbar ist. Wie üblich ist G aus einzelnen Gadgets zusammengesetzt welche spezielle (aussagenlogische) Aufgaben übernehmen Die „Verdrahtung“ der einzelnen Gadgets erfolgt hierbei je über ein *Paar von zwei Kanten.* In einer 3-Kantenfärbung repräsentiert diese Paar den Wert „wahr“ wenn die zwei Kanten die gleiche Farbe haben, und „falsch“ wenn die zwei Kanten unterschiedliche Farben haben. Ein Gadget zum Invertieren eines Bits konstruiert Holyer z.B. wie in Abbildung 4(a), wobei die Paare (a, b) und (c, d) die Bits übertragen. Aufbauend darauf lassen sich dann größere Gadgets konstruieren, welche Variablen bzw. Klauseln darstellen. Abbildung 4(b) zeigt z.B. ein Gadget für die Variablenbelegung, bei der jeder Output den gleichen Wahrheitswert (entweder alle „falsch“ oder alle „wahr“) hat.

Das zentrale Problem ist nun, dass sich selbst unter einer geeigneten Codierung der Färbungen in den Zertifikaten w nicht mit einem Bit aus dem Zertifikat w der von der Färbung „zugewiesene“ Wahrheitswert einer Variable ausgelesen werden kann. Mit einer flexibleren allgemeinen Levin-Reduktion lässt sich dies aber umsetzen (i.e. lese an zwei Stellen in w die zugewiesene Farbe von zwei Kanten aus und vergleiche die Farben). Ob r4CHROMINDEX universell im Sinne von Definition 3.29 ist, bzw. äquivalent vollständig bezüglich projektiven Levin-Reduktionen ist, sei hier offen gelassen und als Frage formuliert:

Frage 3.37. *Ist* r4CHROMINDEX *(bzw. eine geeignete natürliche Variante) universell?*

Die Frage lässt sich – im Hinblick auf die Separationen der Vollständigkeits-Begriffe – auch folgendermaßen verallgemeinern:

Frage 3.38. *Angenommen* $\mathrm{P} \neq \mathrm{NP}$*. Existiert dann eine natürliche NP-Relation R die universell ist, aber nicht* $\leq^{\mathrm{p}}_{\mathrm{pars}}$*-vollständig ist?*

Je ein Argument spricht für bzw. gegen eine positive Beantwortung von Frage 3.37. Einerseits das oben schon skizzierte Argument, dass sich Färbbarkeiten offenbar nicht gut mit der projektiven Levin-Reduzierbarkeit verträgt. Es sei darauf hingewiesen, dass Agrawal und Biswas (1992a) in ihrer

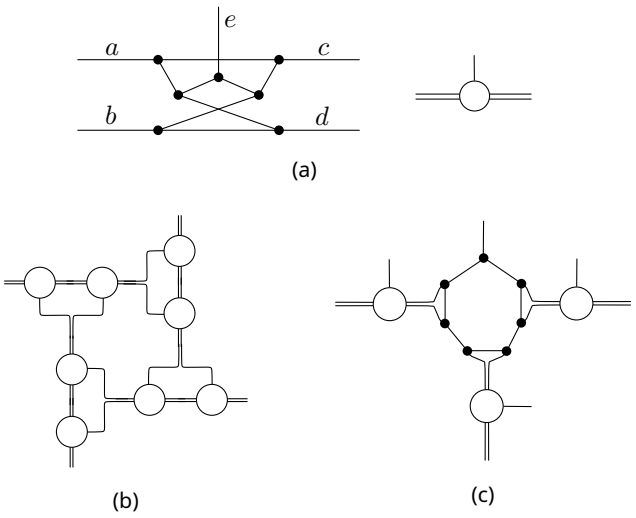


Abbildung 4: Die von Holyer (1981) verwendeten Gadgets um die NP-Vollständigkeit der 3-Kantenfärbbarkeit in 3-regulären Graphen zu zeigen.
(a) Das Gadget zum Invertieren. Beachte dass in einer gültigen Färbung die Kanten a und b die gleiche Farbe haben („wahr“) genau dann wenn c und d ungleiche Farben haben („falsch“). Außerdem haben entweder a, b, e oder c, d, e alle drei unterschiedliche Farben. Das Symbol rechts ist die schematische Darstellung dieses Gadgets in den Abbildungen (b) und (c).
(b) Gadget für je eine Variable. Beachte dass in einer gültigen Färbung alle Outputs entweder „wahr“ oder „falsch“ sind.
(c) Gadget für eine Klausel. In einer gültigen Färbung ist mindestens einer der drei Inputs „wahr“.

Arbeit zwar exemplarisch die Universalität vieler Suchprobleme aus verschiedensten kombinatorischen Bereichen gezeigt haben, Färbungsprobleme wurden hierbei aber nicht betrachtet. Auch offen ist, inwiefern sich Universalität generalisieren lässt, indem das projektive „Auslesen“ von Werten aus den Zertifikaten abgeschwächt wird, z.B. über eine Art polynomialzeit-berechenbares Schema.

Andererseits ist es für Färbungsprobleme nicht *prinzipiell* unmöglich, Universalität zu zeigen. Beispielsweise ist das Problem `r3COL` der 3-Färbbarkeit eines Graphen durchaus als universelle NP-Relation darstellbar, wenn wieder (wie schon bei `rMAXCUT'` oder `r4CHROMINDEX`) nach der unter Permutationen der Farben lexikographisch kleinsten Färbung (sortiert anhand der Knoten) gesucht wird. Ein Lehrbuch-Beweis der $\leq_{\mathrm{m}}^{\mathrm{p}}$-Vollständigkeit über `3CNFSAT` $\leq_{\mathrm{m}}^{\mathrm{p}}$ `3COL` startet üblicherweise mit einem „Palette“-Gadget aus drei Knoten $v_{\mathrm{false}}, v_{\mathrm{true}} v_{\mathrm{base}}$, vgl. Abbildung 5. In einer gültigen Färbung entspricht die Farbe von v_{true} dann der Farbe „wahr“. Nummeriert man in einer `3COL`-Instanz G_φ für `3CNFSAT`-Instanz φ die Knoten so um, dass $v_{\mathrm{false}}, v_{\mathrm{true}} v_{\mathrm{base}}$ die ersten Knoten von G_φ sind, dann hat immer v_{false} die Farbe 1 und v_{true} die Farbe 2 (ansonsten existiert eine Permutation der Farben sodass die Färbung lexikographisch kleiner ist.) Eine projektive Levin-Reduktion kann also nun in einem Bit auslesen ob beliebiger Knoten v die Farbe „wahr“ hat, denn diese ist immer 2. Eine entsprechende Codierung des Zertifikats wäre z.B. von der Form $h(c_1)h(c_2)\cdots$ wobei $c_i \in \{1, 2, 3\}$ die Farbe des i-ten Knotens ist, und h eine one-hot-Codierung umsetzt, i.e. $h(1) = 100, h(2) = 010, h(3) = 001$. Es lässt sich dann leicht eine projektive Levin-Reduktion von `r3CNFSAT` auf `3COL` angeben, und nach Satz 3.31 universell.

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

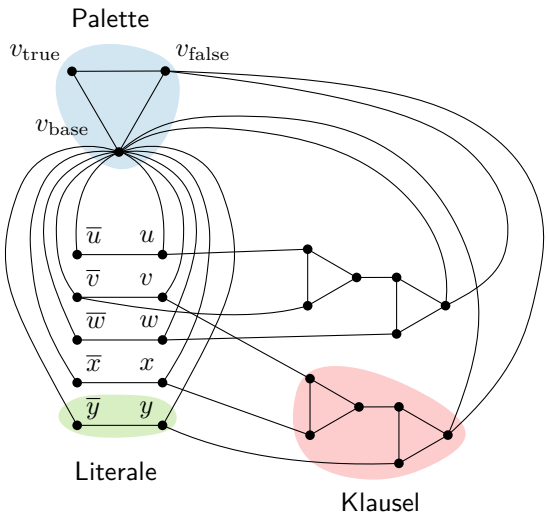


Abbildung 5: Reduktion der 3CNFSAT-Instanz $\varphi = (u \vee \overline{v} \vee \overline{w}) \wedge (v \vee x \vee y)$. Eine Dreifärbung dieses Graphen entspricht einer erfüllenden Belegung von φ und umgekehrt. Beachte dass zu jeder Variable genau ein Knoten zu einem entsprechendem Literal (positiv bzw. negativ) die gleiche Farbe wie v_{true} hat, und der Knoten zum anderen Literal die gleiche Farbe wie v_{false} hat. Es ist leicht zu sehen, dass jedes Klausel-Gadget genau dann dreifärbbar ist, wenn mindestens eine der drei angeschlossenen Literale die gleiche Farbe wie v_{true} hat.

4 Suchprobleme und die Hypothese Q im Kontext des Pudlákschen Programms

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

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

Im Kapitel werden wir uns grob folgenden drei Desiderata 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 Pudlákschen Programm einzuordnen.

Zweitens, in Abschnitt 4.2, verallgemeinern wir Charakterisierungen Q, die sich insbesondere auf Suchprobleme und deren assoziierte Beweissysteme beziehen. Insbesondere zeigen wir für eine große Klasse von vollständigen NP-Suchproblemen R (nämlich jene die Levin-paddable sind) dass das zu R assoziierte *Standardbeweissystem* ((x, y) mit $R(x, y)$ ist ein Beweis für x) p-optimal ist, genau dann wenn Q. Damit wird die p-Optimalität des entsprechenden Standardbeweissystems zu einer Invariante, die entweder für *alle* Levin-paddable NP-Suchprobleme zutrifft, oder für *keins*.

Drittens ergänzen wir im gesamten Verlauf dieses Kapitels das Pudlákschen Programm um weitere Hypothesen, sodass Abbildung 1 der Beziehungen zwischen den Pudlákschen Hypothesen vergrößert wird. Damit erreichen wir den Stand, der in Abbildung 6 dargestellt wird.

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

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

(1) Hypothese Q*.*

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

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

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

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

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

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

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

Satz 4.2. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*.*

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

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

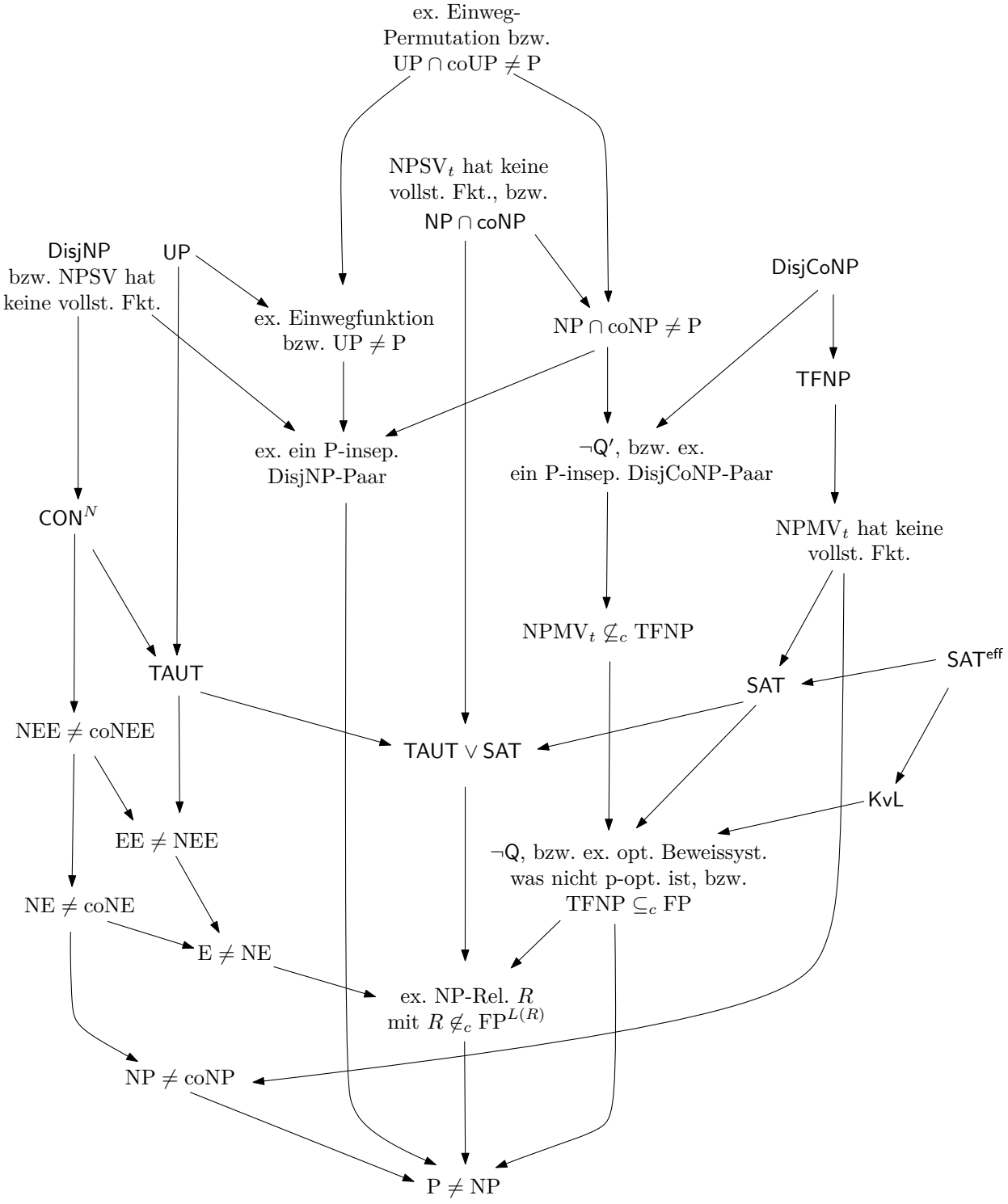


Abbildung 6: Bekannte (relativierenden) Implikationen zwischen den betrachteten Hypothesen und weiteren Aussagen. Satz 4.24 gibt Belegstellen für jede dieser Implikationen an.

(3) (Messner, 2000) Das Standardbeweissystem sat

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

für rSAT *ist p-optimal.*

Dieser Satz relativiert nicht.

In anderen Worten sagt Aussage (2) aus, dass es modulo Umcodieren nur einen einzigen SAT-Solver gibt, und insbesondere alle SAT-Solver äquivalent zum trivialen Solver ist, welcher nur alle möglichen Belegungen ausprobiert. Die Aussage (3) macht eine analoge Aussage über Beweissysteme: egal wie komplex ein Beweissystem h für SAT ist, wir können immer einen h-Beweis für φ in eine erfüllende Belegung für φ (quasi ein trivialer Beweis für $\varphi \in$ SAT) transformieren. Damit ist auch leicht zu sehen, dass $\mathsf{Q} \Rightarrow \neg\mathsf{SAT}$, 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 u. a. und Messner zu einer *relativierbaren* Variante verallgemeinert werden. Mit dieser Verallgemeinerung ist es dann auch für uns möglich, Q formal in das Pudlák-sche Programm (u.a. durch $\mathsf{Q} \Rightarrow \neg\mathsf{SAT}$) einzuordnen. Hierfür führen wir jetzt schon den Begriff eines Standardbeweissystems formal ein.

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

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

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

Beobachtung 4.4. *Für jede NP-Relation R ist das Standardbeweissystem std_R für* $\mathrm{Proj}(R)$ *ehrlich, optimal, und hat kurze Beweise.*

Beweis. Aus der p-Balanciertheit folgt sofort dass std_R kurze Beweise hat. Nach Beobachtung 2.12 damit auch optimal. Insbesondere hat std_R *nur* polynomiell längere Beweise, also ist std_R ehrlich. □

4.1 Karp-Vollständigkeit vs. Levin-Vollständigkeit

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

Frage 3.21. *Wenn* $\mathrm{Proj}(R)$ *eine* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständige Menge für* NP *ist, ist dann auch R eine* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständige NP-Relation für* FNP*?*

Vermutung 3.22 (Karp-vs-Levin-Vermutung; KvL)**.** *Es existiert eine NP-Relation R sodass* $\mathrm{Proj}(R)$ $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständig für* NP *ist, aber R ist nicht* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig für* FNP*.*

Ich möchte argumentieren, dass die obige Frage bzw. Vermutung eng mit der Hypothese Q zusammenhängt. Im Speziellen werden wir sehen, dass die Hypothese Q so charakterisiert werden kann, dass sie einer Verstärkung der Vermutung KvL entspricht.[5]

Satz 4.5. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*, bzw.* $\mathrm{TFNP} \subseteq_{\mathrm{c}} \mathrm{FP}$*.*

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

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

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

5. Fenner u. a. (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.

Definiere folgende Relation R mit

$$\textit{set-}R(w) = \begin{cases} \{y \mid y \in \Sigma^{\leq q(|x|)}, (x,y) \in A\} & \text{falls } w = (x,y'),\ (f(x),y') \in B \\ \{\varepsilon\} & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, dass R eine totale NP-Relation ist. Nach (1) existiert nun eine (totale) Verfeinerung $g \in \mathrm{FP}$ von R.
Damit lässt die Levin-Reduktion von A auf B angeben: wähle f als Reduktionsfunktion, und sei die Funktion g von oben die Translationsfunktion. Dann gilt

$$\begin{aligned} &(f(x),y') \in B \implies (x,y') \in \mathrm{Proj}(R) \\ &\implies ((x,y'),g(x,y')) \in R \\ &\implies (x,g(x,y')) \in A \text{ nach Def. von } R \end{aligned}$$

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

(2) $\implies$ (1): Sei A eine totale NP-Relation Definere 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 $\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(B)$ über die Identitätsfunktion. Nach Annahme (2) lässt sich nun diese Reduktion zu einer Levin-Reduktion $A \leq^{\mathrm{p}}_{\mathrm{L}} B$ verstärken, mit Reduktionsfunktion $f \in \mathrm{FP}$ und Translationsfunktion $g \in \mathrm{FP}$. Für alle x gilt nun $(f(x),\varepsilon) \in B$ nach Definition, nach Levin-Reduktion also auch $(x, g(x,\varepsilon)) \in A$. Definieren wir nun $h(x) \stackrel{\mathrm{df}}{=} g(x,\varepsilon)$, dann ist $(x,h(x)) \in A$ für alle x, also $h \in \mathrm{FP}$ eine Verfeinerung von A, also $A \in_{\mathrm{c}} \mathrm{FP}$, wie gewünscht. □

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.

Satz 4.6. KvL $\implies$ ¬Q.

Beweis. Wir zeigen die Kontraposition, und starten mit der Voraussetzung Q. Wir wollen nun ¬KvL zeigen. Sei hierfür R eine beliebige NP-Relation sodass $\mathrm{Proj}(R)$ $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständig ist. Damit gilt also schon für alle weiteren NP-Relationen A, dass $\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}} \mathrm{Proj}(R)$. Nach Satz 4.21 gilt also auch die Aussage 4.21(6), und damit $A \leq^{\mathrm{p}}_{\mathrm{L}} R$. Also ist R auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig, wie gewünscht und wir haben ¬KvL gezeigt. □

Was sind natürlich 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.

Besonders interessant erscheint aber die Beziehung zur Hypothese ¬Q, also genau die Umkehrung von Satz 4.6. Zumindest in der obigen Charakterisierung von Satz scheint ¬Q schwächer, denn mit Satz würde das bedeuten, dass ein *beliebiges* Paar A, B von NP-Relationen existiert mit $\mathrm{Proj}(A) \leq^{\mathrm{p}}_{\mathrm{m}}$ $\mathrm{Proj}(B)$, aber $A \not\leq^{\mathrm{p}}_{\mathrm{L}} B$. Weder A noch B müssen eine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Projektion haben, was KvL ja verlangt.

Paradoxerweise scheint die Charakterisierung von ¬Q durch Fenner u. a. in Satz 4.2(2) dienlicher: Betrachten wir hierbei exemplarisch den Fall Relationen für SAT. Ich vermute, dass ¬Q ⇒ KvL; um das zu plausibilisieren möchte ich zeigen, dass ¬Q ∧ ¬KvL unwahrscheinlich ist.

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

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

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

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

Unter der Annahme ¬KvL existiert nun eine Levin-Reduktion $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} R_N$ mit Reduktions- bzw. Translationsfunktion f, g. Das ist zunächst kein Widerspruch, denn es könnte ja $f \neq \mathrm{id}$. Gleichzeitig

wäre die Existenz einer solchen Reduktion überraschend. Wir hätten nach Definition

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

Einerseits ist es also nicht möglich, aus dem Rechenweg w effizient eine akzeptierende Belegung für $f(\varphi)$ zu bestimmen, obwohl w bezeugt dass $f(\varphi)$ erfüllbar ist. (Ersetze in (4.1) ψ mit $f(\varphi)$.) Andererseits reicht der „Beweis“ w aber aus, um (zusammen mit der Information φ) effizient wieder eine erfüllende Belegung für φ zu berechnen. Das *plausibilisiert* zwar einen Widerspruch, bzw. dass $\neg\mathsf{Q} \wedge \neg\mathsf{KvL}$ wahrscheinlich falsch ist, ist aber natürlich kein solcher. Die Umkehrung von Satz 4.6 bleibt offen.

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

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

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

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

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

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

In anderen Worten, falls $h' \leq^{p}_{\mathrm{eff}} h$, dann kann h zwar nicht *jeden* h'-Beweis w für $x \in L$ in einen h-Beweis für (das gleiche) x effizient umrechnen, es kann aber zumindest alle *relevanten* h'-Beweise effizient umrechnen, nämlich für jedes $x \in L$ die h'-Beweise für $f(x)$ in h-Beweise für x. Anstelle „h p-simuliert effektv h'“ ließe sich äquivalent auch $h^{-1} \leq^{\mathrm{p}}_{\mathrm{L}} h'^{-1}$ schreiben. Beachte, dass die Relation h^{-1} nur Lösungen mit ihren Beweisen reliert. Klar ist: p-Simulation impliziert effektive p-Simulation impliziert Simulation unter Beweissystemen.

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

Vermutung 4.8 (KvL formuliert unter Beweissystemen)**.** *Es existiert eine NP-Relation Q mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem* $\mathrm{Proj}(Q)$*, wobei std_Q nicht alle anderen optimalen Beweissysteme für* $\mathrm{Proj}(Q)$ *effektiv p-simulieren kann.*

Dass die Formulierung der Vermutungen 3.22 und 4.8 äquivalent sind, zeigt folgende Beobachtung:

Beobachtung 4.9. *Folgende Aussagen sind äquivalent:*

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

(2) Für jede NP-Relation Q mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem $\mathrm{Proj}(Q)$ *kann std_Q jedes optimale Beweissystem h für* $\mathrm{Proj}(Q)$ *effektiv p-simulieren. (Das ist die Negation von der Vermutung 4.8.)*

Beweis. (1)⇒(2): Sei R eine NP-Relation mit $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständigem $\mathrm{Proj}(R)$. Wir zeigen, dass std_R jedes andere optimale Beweissystem h effektiv p-simulieren kann. Nachdem h optimal ist, hat es auch kurze Beweise (Beob. 2.14): für jedes $x \in \mathrm{Proj}(R)$ existiert ein h-Beweis w mit $|w| \leq q(|x|)$ für geeignetes Polynom q. Definiere

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

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

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

Nach Definition gilt also

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

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

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

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

Definiere

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

Wir zeigen, dass h ein Beweissystem für $\mathrm{Proj}(Q)$ ist. Es ist offenbar dass $h \in \mathrm{FP}$. Die Funktion h ist korrekt: wenn $h(x,y) = x$ dann ist $f(x) \in \mathrm{Proj}(R)$ und nach Eigenschaft von f auch $x \in \mathrm{Proj}(Q)$. Die Funktion h ist vollständig: Sei $x \in \mathrm{Proj}(Q)$. Dann ist schon $f(x) \in \mathrm{Proj}(R)$ und es gibt ein y mit $(f(x),y) \in R$. Also ist (x,y) ein h-Beweis für x.
Außerdem ist klar, dass h kurze Beweise hat, damit ist h auch optimal (Beob. 2.14). Damit gilt nach (2) nun, dass $h \leq^{\mathrm{p}}_{\mathrm{eff}} std_Q$. Also existieren Funktionen $f', g' \in \mathrm{FP}$ sodass

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

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

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

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

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

Vermutung 4.10 ($\mathsf{SAT}^{\mathsf{eff}}$)**.** *Keine $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge $L \in \mathrm{NP}$ hat ein optimales Beweissystem h, welches alle anderen optimalen Beweissysteme für L effektiv p-simulieren kann.*

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

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

2. Zu (2): Wieder klar aus Kontraposition. Unter ¬KvL folgt mit der Formulierung aus Vermutung 4.8 dass für jede NP-Relation Q, $\mathrm{Proj}(Q)$ vollständig, das (optimale) Standardbeweissystem std_Q alle optimalen Beweissysteme für $\mathrm{Proj}(Q)$ effektiv p-simulieren kann. Das gilt dann insbesondere auch für die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation `rKAN`, also hat die $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge `KAN` *ein* Beweissystem, welches alle optimalen Beweissysteme effektiv p-simulieren kann. □

Wir haben also je eine notwendige (¬Q) und eine hinreichende Hypothese ($\mathsf{SAT}^{\mathsf{eff}}$) für KvL. Nichtsdestotrotz bleiben noch viele Fragen offen, die wir hier aus Platzgründen nicht weiter verfolgen werden. Wir konnten zwar KvL als Aussage über Beweissysteme formulieren, aber sind auch andere Charakterisierungen (z.B. ähnlich wie bei Q) möglich? Gibt es natürliche (z.B. kryptographische) Annahmen die hinreichend für KvL sind? Wie ist die Beziehung zu den anderen Pudlákschen Hypothesen? Wie verhält sich insbesondere SAT zu $\mathsf{SAT}^{\mathsf{eff}}$?

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

4.2 Hypothese Q und Suchprobleme

Wie im Einstieg des Kapitels angesprochen, geben Fenner u. a. (2003) bzw. Messner (2000) äquivalente Charakterisierungen der Hypothese Q an, welche sich im Wesentlichen auf auf der $\leq_{\mathrm{L}}^{\mathrm{p}}$-Vollständigkeit von rSAT aufbauen (Satz 4.2). Wir wiederholen hier noch einmal die Aussage, aber mit einer etwas abstrakteren Notation. Ganz ähnlich wie NP-Relationen ein Standardbeweissystem induzieren, können wir auch das Standardbeweissystem bezüglich einer NPTM definieren:

Definition 4.12 (Standardbeweissystem von NPTM)**.** Sei N eine NTM. Wir definieren bezüglich N das das *Standardbeweissystem* std_N für $L(N)$ wie folgt:

$$std_N(w) \stackrel{\mathrm{df}}{=} \begin{cases} x & \text{wenn } w = (x, \alpha) \text{ und } N(x) \text{ akzeptiert auf RW } \alpha, \\ \bot & \text{sonst.} \end{cases}$$ ◁

Ähnlich wie bei Standardbeweissysteme für NP-Relationen ist std_N für jede nichtdeterministische *Polynomialzeit*-TM N ehrlich, optimal, und hat kurze Beweise.

Satz 4.2. *Folgende Aussagen sind äquivalent:*

(1) Hypothese Q*.*

(2) Für jede NPTM N mit $L(N) =$ SAT *gilt $std_N \leq_{\mathrm{m}}^{\mathrm{p}}$ sat. Es existiert also eine Funktion $h \in$ FP sodass*

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

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

Dieser Satz relativiert nicht.

Diese beiden Charakterisierungen wollen wir im Folgenden verallgemeinern und 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.

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

Eine dieser stärkeren Eigenschaften von rSAT, welche Fenner u. a. in ihren Beweisen gebrauchten, ist die $\leq_{1,\mathrm{i}}^{\mathrm{p}}$-Vollständigkeit von rSAT. Wir schwächen im Folgenden diese Voraussetzunt ab, und verlangen nur, dass eine NP-Relation unter ehrlichen Reduktionen vollständig ist. Dies gilt insbesondere für rSAT. Für welche $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständigen NP-Relationen das noch zutrifft, werden wir unten betrachten.

Mit dieser ehrlichen Levin-Vollständigkeit lässt sich nun die Charakterisierung von Fenner u. a., Thm. 2 generalisieren:

Lemma 4.13. *Sei R eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation, mit der zusätzlichen Eigenschaft dass für die jeweilige entsprechende Problem-Reduktionsfunktion $f\colon Q \to R$ für $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ immer gilt, dass f ehrlich ist. Folgende Aussagen sind äquivalent:*

(1) Aussage Q*.*

(2) Für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ lassen sich akzeptierende Rechenwege von N in Zertifikate umrechnen: es gilt $std_N \leq_{\mathrm{m}}^{\mathrm{p}} std_R$, bzw. existiert eine Funktion $h \in$ FP sodass

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

Beweis. (1)⇒(2): Nachdem Q gilt, gilt auch TFNP $\subseteq_{\mathrm{c}}$ FP nach Beobachtung 3.5. Sei nun R eine beliebige NP-Relation mit Zertifikatsschranke q, und sei N eine beliebige NPTM mit $L(N) = \mathrm{Proj}(R)$. Definiere nun folgende Relation Q mittels:

$$\textit{set-}Q(x, \alpha) \stackrel{\mathrm{df}}{=} \begin{cases} \{y \mid y \in \Sigma^{\leq q(|x|)}, (x, y) \in R\} & \text{falls } N(x) \text{ auf RW } \alpha \text{ akzeptiert,} \\ \{\varepsilon\} & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, dass Q eine totale NP-Relation ist. Insbesondere im ersten Fall gilt $x \in L(N) = \mathrm{Proj}(R)$, also existiert auch mindestens ein $y \in \textit{set-}R(x)$.

Nach Annahme gilt also $Q \in_c \mathrm{FP}$, sei also $h \in \mathrm{FP}$ eine Verfeinerung von Q. Nun gilt

$$\begin{aligned}
&\mathit{std}_N(x, \alpha) = x \\
&\implies N(x) \text{ akz. mit Rechenweg } \alpha \\
&\implies \mathit{set\text{-}Q}(x, \alpha) = \mathit{set\text{-}R}(x) \\
&\implies h(x, \alpha) \in \mathit{set\text{-}R}(x) \implies (x, h(x, \alpha)) \in R, \\
&\implies \mathit{std}_R(x, h(x, \alpha)) = x.
\end{aligned}$$

wie gewünscht.

(2)⇒(1): Sei $Q \in \mathrm{TFNP}$. Sei ferner R eine $\leq_{\mathrm{L}}^{\mathrm{p}}$-vollständige NP-Relation unter ehrlichen Problem-Reduktionsfunktionen (z.B `rKAN`), und Zertifikatsschranke p. Da R ja vollständig ist, gilt $Q \leq_{\mathrm{L}}^{\mathrm{p}} R$ via $f, g \in \mathrm{FP}$ und (nach Voraussetzung) ist f ehrlich; es existiert ein Polynom q sodass $q(|f(x)|) \geq |x|$.
Definiere nun die folgende NPTM $N'(w)$:

1 Rate nichtdeterministisch $x \in \Sigma^{\leq q(|w|)}$
2 **wenn** $f(x) = w$ **dann** akzeptiere
(Ab hier kann man x wegwerfen)
3 Rate nichtdeterministisch $y \in \Sigma^{\leq p(|w|)}$
4 Akzeptiere genau dann wenn $(w, y) \in R$.

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

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

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

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

Beobachte wie für $N'(f(x))$ immer ein trivialer akzeptierender Rechenweg α_x existiert: nämlich jener, welcher in Z. 1 das Urbild x rät. Beobachte dass die Umformung $x \mapsto \alpha_x$ in Polynomialzeit möglich ist.
Um nun (1) zu zeigen müssen wir aus $x \in \Sigma^*$ effizient einen akzeptierenden Rechenweg für N bestimmen. Wir haben

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

Damit ist $r \in \mathrm{FP}$, $r(x) \stackrel{\mathrm{df}}{=} g(h(f(x), \alpha_x))$, eine Verfeinerung von Q und $Q \in_c \mathrm{FP}$, wie gewünscht. □

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

Definition 4.14 (Levin-Paddability)**.** Eine NP-Relation R ist *Levin-paddable* wenn Funktionen $\mathit{pad} \in \mathrm{FP}$ und $\mathit{padsol} \in \mathrm{FP}$ existieren, sowie ein Polynom r sodass

(1) $x \in \mathrm{Proj}(R) \iff \mathit{pad}(x, 1^n) \in \mathrm{Proj}(R)$,
(2) $(\mathit{pad}(x, 1^n), y) \in R \implies (x, \mathit{padsol}(x, 1^n, y)) \in R$,
(3) $r(|\mathit{pad}(x, 1^n)|) \geq n$. (Funktion pad ist ehrlich bzgl. der zweiten Komponente.) ◁

Beachte dass wir im Gegensatz zur Berman–Hartmanis-Paddability keine Invertierbarkeit der Padding-Funktion verlangen. Später werden wir sehen, welche NP-Relationen alle diese Eigenschaft der Levin-Paddability erfüllen. Festhalten können wir aber, dass `rSAT` Levin-paddable ist. Das ist

einfach zu sehen: padde Formeln φ auf, indem z.B. Disjunktionen neue Variablen hinzugefügt werden, i.e.

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

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

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

Beobachtung 4.15. *Die kanonische Levin-vollständige NP-Relation* `rKAN` *ist Levin-paddable.*

Skizze. Padde Instanzen $(N, x, 1^n)$ zu $(N', x, 1^n)$ auf, wobei die NTM N' aus N hervorgeht, indem Zustände hinzugefügt werden die über die Transitionsrelation von N nicht erreichbar sind. Ein akzeptierender Rechenweg auf $N'(x)$ ist dann genau ein akzeptierender Rechenweg auf $N(x)$. □

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

Lemma 4.16. *Sei R eine NP-Relation die Levin-paddable ist. Folgende Aussagen sind äquivalent:*

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

(2) Für alle NTM N (ohne Laufzeitbeschränkung) mit $L(N) = \mathrm{Proj}(R)$ gilt $std_N \leq^{\mathrm{p}}_{\mathrm{m}} std_R$.

(3) Für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ gilt $std_N \leq^{\mathrm{p}}_{\mathrm{m}} std_R$.

Beweis. (1)⇒(2): Klar.

(2)⇒(3): Klar.

(3)⇒(1): Angenommen (3) gilt. Seien *pad*, *padsol* die entsprechenden Funktionen, welche die Levin-Paddability von R realisieren. Das Polynom r sei so gewählt dass $r(|pad(x, 1^n)|) \geq n$ (vgl. 4.14(3)).
Wir wollen nun zeigen, dass std_R auch p-optimal ist. Sei hierfür f ein beliebiges Beweissystem für $\mathrm{Proj}(R)$. Wir zeigen nun, dass $f \leq^{\mathrm{p}}_{\mathrm{m}} std_R$. Seien *pad*, *padsol* die entsprechenden Padding-Funktionen von R. Definiere nun

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

Es ist leicht zu sehen, dass f' ein Beweissystem für $\mathrm{Proj}(R)$ ist. Außerdem ist f' ehrlich es ist ehrlich für Eingaben $0z$, denn das Standardbeweissystem std_R ist ehrlich nach Beobachtung 4.4. Es ist ehrlich für Eingaben $w = 1z$, denn

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

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

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

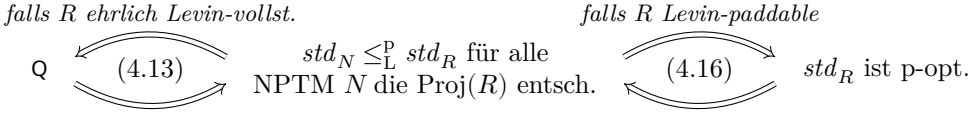
Jetzt können wir zeigen, dass std_R das Beweissystem f p-simuliert: sei z ein f-Beweis für x, d.h. $f(z) = x$. Wir wissen, dass $f'(1z) = pad(x, 1^{|1z|}) = x'$. Daher können wir aus z einen Rechenweg α_z konstruieren, sodass $N_{f'}(x')$ akzeptiert, nämlich jener der den f'-Beweis $1z$ rät. Die Abbildung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

Nun gilt

$$\begin{aligned}
&N_{f'}(x') \text{ akz. mit } \alpha_z \implies (x', \underbrace{h(x', \alpha_z)}_{y'}) \in R \text{ nach (4.3)} \\
&\implies (pad(x, 1^{|1z|}), y') \in R \text{ mit } y' = h(x', \alpha_z) \text{ und obiger Def. von } x' \\
&\implies (x, \underbrace{padsol(x, 1^{|1z|}, y')}_{y}) \in R \text{ nach Def. 4.14(2)} \\
&\implies std_R(x, y) = x \text{ mit } y = padsol(x, 1^{|1z|}, y')
\end{aligned}$$

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

Wir fassen kurz den aktuellen Stand zusammen. TODO: Irgendwie möglich von std_N wegzukommen und stattdessen über optimale ps zu sprechen? Sei R eine NP-Relation. Wir haben nun folgendes Bild:



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

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

Lemma 4.17. *Die in Lemma 4.13 und 4.16 genannten Voraussetzungen an die NP-Relation R werden von allen solchen R erfüllt, die $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig sind und Levin-paddable sind.*

Beweis. Es ist sofort klar, dass R die Voraussetzungen von Lemma 4.16 erfüllt. Es bleibt nur zu zeigen, dass für jede NP-Relation Q eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-Reduktion angegeben werden kann, bei dem die Problem-Reduktionsfunktion ehrlich ist. Wir nutzen hierbei aus, dass R eine Levin-paddable Relation ist.

Nachdem R vollständig ist, gilt $Q \leq^{\mathrm{p}}_{\mathrm{L}} R$; sei $f, g \in \mathrm{FP}$ 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, 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.

Definiere

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

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

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

und nach Definition der Translationsfunktion g gilt dann

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

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

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

Beobachtung 4.18. *(1) Gilt* $\texttt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}} R$*, und ist die zugehörige Reduktionsfunktion* f *ehrlich, dann ist* R *Levin-paddable (und* $\leq^{\mathrm{p}}_{\mathrm{L}}$*-vollständig).*

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

Damit können wir schon als Ergebnis festhalten, dass jede $\leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}}$-vollständige Relation R die in Lemma 4.16 und 4.13 genannten Voraussetzungen an die NP-Relation R erfüllt. Das sind nach Goldreich (2008) unrelativierten Fall u.a. `rSAT`, `rSETCOVER`, `rVERTEXCOVER`, `rCLIQUE`, `r3COLORABILITY`.

Beweis zu Beobachtung 4.18. Aussage (2) folgt unmittelbar aus (1): Wir haben $\texttt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L},1,\mathrm{inv}} R$ und damit ist die entsprechende Reduktionsfunktion f p-invertierbar, und damit ehrlich.

Für (1) nutzen wir die Levin-Paddability von `rKAN` aus: übersetze Instanz x von R nach `rKAN`, padde dort hoch, und 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 $\texttt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}} R$ bezeugen, und seinen analog f', g' jene Funktionen, welche $R \leq^{\mathrm{p}}_{\mathrm{L}} \texttt{rKAN}$ bezeugen. Erstere existieren nach Voraussetzung, zweitere existieren weil $\texttt{rKAN} \leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig ist. Nach Voraussetzung ist f ehrlich. Und nach Beobachtung 4.15 existieren für `rKAN` Padding-Funktionen $pad_{\texttt{rKAN}}$, $padsol_{\texttt{rKAN}}$. Sei q ein entsprechendes Polynom mit $q(|pad_{\texttt{rKAN}}(x, 1^n)|) \geq n$, $q(|f(x)|) \geq |x|$.

Definiere nun

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

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

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

Ferner gilt

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

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

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

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

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

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

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

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

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. □

Anstelle der Betrachtung, wie die *Reduktionen* zwischen den einzelnen NP-Relationen aufgebaut sind, können wir auch strukturelle Eigenschaften von NP-Relationen ausnutzen, um Paddability zu zeigen. Hierbei macht die Definition von *Universalität* durch Agrawal und Biswas (1992a) aus dem Abschnitt 3.4 einen produktiven Beitrag. Ist eine NP-Relation *joinable*, dann können wir auch zu einer

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

Beobachtung 4.19. *Jede strenge NP-Relation $R \neq \emptyset$ die* joinable *ist, ist auch Levin-paddable.*

Vor dem Beweis können wir mit dieser Aussage festhalten, dass jede universelle Relation R die in Lemma 4.16 und 4.13 genannten Voraussetzungen an die NP-Relation R erfüllt. Das sind nach Agrawal und Biswas (1992a) u.a. $\texttt{rSAT}, \texttt{rHAM}, \texttt{rINDSET}, \texttt{rKNAPSACK}, \texttt{rMAXCUT}$.

Beweis zu Beobachtung 4.19. Sei R eine NP-Relation, mit zugehörigem Polynom q, welches die Zertifikatsgröße spezifiziert. Zur Erinnerung, nachdem R streng ist, gilt $(x, y) \in R \Rightarrow |y| = q(|x|) > 0$. Ferner haben wir eine Instanz $z \in \text{Proj}(R)$. Damit existiert also auch ein w mit $(z, w) \in R$ und $q(|z|) = |w| > 0$.

Wir zeigen zunächst, wie wir für beliebige Instanz x und $n \in \mathbb{N}$ auf eine Instanz x' hochpadden, in dem Sinne dass $q(|x'|) \geq n$. Nach Voraussetzungen ist die Relation R auch *joinable*, das heißt wir haben eine Funktion $\mathit{join} \in \text{FP}$. Sei

$$(x', \delta) \stackrel{\text{df}}{=} \mathit{join}(x, \underbrace{z, z, \ldots, z}_{n \text{ mal}}).$$

Intuitiv muss nun δ (und damit x') lang sein, da nun aus all den n vielen Instanzen z wieder das jeweilige Zertifikat aus jedem Zertifikat für x' extrahiert werden muss. Nach Definition 3.29 gilt

$$q(|x'|) \geq |\delta| = q(|x|) + \underbrace{q(|z|) + \cdots + q(|z|)}_{n \text{ mal}} \geq n \cdot q(|z|) \geq n.$$

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

$$\mathit{pad}(x, 1^n) \stackrel{\text{df}}{=} x' \quad \text{wobei } (x', \delta) = \mathit{join}(x, \underbrace{z, z, \ldots, z}_{n \text{ mal}}).$$

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

Wir zeigen jetzt, dass die Zugehörigkeit zu $\text{Proj}(R)$ erhalten bleibt. Zur Erinnerung, wir haben nach Eigenschaften der *join*-Funktion

$$\{y'[\delta] \mid y' \in \mathit{set}\text{-}R(x')\} = \{y y_1 y_2 \cdots y_n \mid y \in \mathit{set}\text{-}R(x), y_1, y_2, \ldots, y_n \in \mathit{set}\text{-}R(z)\}. \tag{4.4}$$

Gilt $x \notin \text{Proj}(R)$, dann ist die rechte Menge in (4.4) leer, also auch die linke Menge und damit $x' = \mathit{pad}(x, 1^n) \notin \text{Proj}(R)$. Falls anders herum $x \in \text{Proj}(R)$, dann ist die rechte Menge nicht leer, existiert ja ein Zertifikat y für x und je ein $y_i = w$ für jedes z. Also ist auch die linke Menge nicht leer, damit $\mathit{pad}(x, 1^n) \in \text{Proj}(R)$.

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

$$\mathit{padsol}(x, 1^n, y') \stackrel{\text{df}}{=} y'[\delta][0, 1, \ldots, q(|x|) - 1] \quad \text{wobei } (\cdot, \delta) = \mathit{join}(x, \underbrace{z, z, \ldots, z}_{n \text{ mal}}).$$

Wir verifizieren: Sei $(\mathit{pad}(x, 1^n), y') \in R$, dann ist nach (4.4) $y'[\delta] = y y_1 y_2 \cdots$ wobei $(x, y) \in R$. Nachdem R streng ist, gilt insbesondere $y \in \Sigma^{q(|x|)}$ und wir haben

$$\mathit{padsol}(x, 1^n, y') = y'[\delta][0, 1, \ldots, q(|x|) - 1] = (y y_1 y_2 \cdots)[0, 1, \ldots, q(|x|) - 1] = y$$

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

Es bleibt die Frage offen, ob Levin-Paddability für *alle* vollständigen NP-Relationen zutrifft. Unter Annahmen einer geeigneten Einwegfunktion ist dies nicht der Fall. Die Argumentation verläuft hier ähnlich zur *Encrypted Complete Set Conjecture*. Wir setzen hier eine stärkere *secure one-way function* (Grollmann und Selman, 1988) f voraus, die selbst mithilfe funktionaler Orakel-Queries nur auf einer dünnen Menge p-invertierbar ist. Präzise meinen wir damit folgendes: sei A ein beliebiger Polynomialzeit-Algorithms, der auf Eingabe w versucht, das Urbild $f^{-1}(w)$ zu berechnen. Zusätzlich darf A das Urbild $f^{-1}(w')$ von einem Wort $w' \neq w$ erfragen. Selbst dann wird A nur auf einer dünnen Menge $W \subseteq \Sigma^*$ das korrekte Urbild aller $w \in W$ bestimmen können. (Vgl. die Ähnlichkeit zur Selbstreduzierbarkeit aus Abschnitt 3.2.) Die Existenz einer solchen Einwegfunktion erscheint aus kryptographischer Perspektive naheliegend.

Betrachte nun, analog zur Encrypted Complete Set Conjecture, die NP-Relation

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

Es ist leicht zu sehen dass $\mathtt{rSAT} \leq^{\mathrm{p}}_{\mathrm{L}} Q$ und damit ist Q auch $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständig. Gleichzeitig kann dann Q nicht Levin-paddable sein. Denn angenommen, Q ist Levin-paddable, dann lässt sich f mit einem funktionalen Orakel-Query *zumindest auf den Werten* $f(\mathtt{SAT})$ p-invertieren: gegeben $w \in f(\mathtt{SAT})$, berechne erst eine zweite Instanz $w' = pad(w, 1^n) \in \mathrm{Proj}(Q)$ mit hinreichend langem n sodass $w' \neq w$. Frage dann an das Orakel und erhalte $(x', z') \in set\text{-}Q(w')$. Dann gilt $(x, z) = padsol(w, 1^n, (x', z'))$ mit $f(x) = w$, i.e. x ist das gesuchte Urbild von w. Wir können also auf der Bildmenge $f(\mathtt{SAT})$ die Einwegfunktion f invertieren. Diese Menge ist tatsächlich nicht dünn: unabhängig der gewählten Codierung von $\mathtt{SAT}$ folgt aus der Existenz der Einwegfunktion f schon $\mathrm{P} \neq \mathrm{NP}$, und damit ist insbesondere die $\leq^{\mathrm{p}}_{\mathrm{m}}$-vollständige Menge $f(\mathtt{SAT})$ eine nicht-dünne Menge nach dem Satz von Mahaney (1982). Das widerspräche nun den Eigenschaften von f, also ist Q nicht Levin-paddable. (Diese Argumentation relativiert, wenn anstelle $\mathtt{rSAT}$ eine relativierbare vollständige Menge gewählt wird, die z.B $\mathtt{rKAN}$.)

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

Frage 4.20. *Ist jede $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige NP-Relation R auch Levin-paddable? Existiert ggf. ein Gegenbeispiel in einer geeigneten relativierten Umgebung?*

Unabhängig von dieser Frage können wir nun aber abschließend die vorigen Ergebnisse zur Beziehung zwischen Suchproblemen und der Hypothese $\mathtt{Q}$ in folgendem Satz zusammenfassen. Beachte dass diese Charakterisierungen relativieren. Die Äquivalenz zu Aussage (8) ist hierbei eine einfache relativierbare Generalisierung von Beweisen durch Messner (2000, Thm. 5.3).

Satz 4.21 (Äquivalente Formulierungen der Hypothese $\mathtt{Q}$)**.** *Folgende Aussagen sind äquivalent:*

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

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

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

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

(5) Jede surjektive 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:

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

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

(9) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable NP-Relation R sodass für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ auch $std_R \leq^{\mathrm{p}}_{\mathrm{m}} std_R$ gilt.

(10) Es existiert eine $\leq^{\mathrm{p}}_{\mathrm{L}}$-vollständige Levin-paddable 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 u. a. (2003, Thm. 2).

2. (1) $\iff$ (3): nach Beobachtung 3.5.

3. (1) $\iff$ (7): nach Lemma 6.

4. (1) $\iff$ (9) $\iff$ (10): nach Lemma 4.13 und 4.16.

5. (2) $\implies$ (8): Die Richtung von rechts nach links ist klar. Sei für die andere Richtung h ein optimales Beweissystem für eine Menge L. Wir wollen zeigen, dass h auch p-optimal ist. Sei dafür g ein weiteres Beweissystem für L. Nach Voraussetzung 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':

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

Es lässt sich leicht zeigen, dass $f' \in$ NPMV, über einen geeigneten NPTM-Transduktor. Es ist sogar $f' \in \mathrm{NPMV}_t$, denn für jedes w mindestens $\pi(w) \in \mathit{set}\text{-}f'(w)$.
Nach (2) gilt also $f' \in \mathrm{NPMV}_t \subseteq_\mathrm{c}$ FP, also existiert eine Funktion $f'' \in$ 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

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

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

6. (8) $\Longrightarrow$ (10): klar, denn rKAN ist $\leq^\mathrm{p}_\mathrm{L}$-vollständig, ist Levin-paddable, und das Standardbeweissystem $\mathit{std}_{\mathtt{rKAN}}$ ist (wie jedes Standardbeweissystem einer NP-Relation) optimal. Zusammen mit (7) ist es also auch p-optimal. □

Analysiert man die Beweise bezüglicher der Äquvalenz von Aussage Q zu (9) und (10) können wir sogar feststellen, dass die Wahl der Relation R beliebig ist. Wir können daher Q über universell quantifizierte Varianten von (9) und (10) charakterisieren.

Satz 4.22. *Entweder gelten die Aussagen (1), (9), (10) oder die Aussagen (1'), (9'), (10'):*

(1) Q.

(9) *Für alle $\leq^\mathrm{p}_\mathrm{L}$-vollständigen Levin-paddable NP-Relationen R, alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ gilt $\mathit{std}_N \leq^\mathrm{p}_\mathrm{m} \mathit{std}_R$.*

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

(1') ¬Q.

(9') *Es existiert keine $\leq^\mathrm{p}_\mathrm{L}$-vollständige Levin-paddable NP-Relation R, sodass für alle NPTM N mit $L(N) = \mathrm{Proj}(R)$ auch $\mathit{std}_N \leq^\mathrm{p}_\mathrm{m} \mathit{std}_R$ gilt.*

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

Beachte dass (9') nicht die negierte Version von (9) ist, für (10) gilt dies analog.

4.3 Bekannte Implikationen und Orakel, offene Trennungen

Im letzten Abschnitt dieses Kapitels werden wir nun die in Abbildung 6 abgebildeten Implikationen und Äquivalenzen nachweisen. Damit werden insbesondere auch die Hypothesen Q und KvL in das Pudlákske Programm eingeordnet. Zum Schluss wird noch angegeben, welche der Hypothesen im (vergrößertem) Pudlákschen Programm durch ein Orakel separiert sind, und welche Separierungen noch offen sind.

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

Vermutung 4.23 (Q′, Fenner u. a., 2003)**.** *Für jede NPTM N mit $L(N) = \Sigma^*$ existiert eine Funktion $g \in$ FP sodass für alle x das Bild $g(x) \in \{0, 1\}$ das erste Bit eines akzeptierenden Rechenwegs von $N(x)$ ist.*

Jetzt können wir auch die in Abbildung 6 abgebildeten Implikationen und Äquivalenzen nachweisen.

Satz 4.24. *Es gelten die in Abbildung 6 abgebildeten Implikationen und Äquivalenzen.*

Beweis. Es gelten die notierten Äquivalenzen:
1. ¬Q $\Leftrightarrow \exists$ optimales Beweissystem was nicht p-optimal ist $\Leftrightarrow$ TFNP $\not\subseteq_\mathrm{c}$ FP, nach Satz 4.21.
2. ¬Q′ $\Leftrightarrow \exists$ P-inseparierbares DisjCoNP-Paar, nach Fortnow und Rogers (1993, Lemma 2.12, vgl. Appendix).
3. NP ∩ coNP ≠ P $\Leftrightarrow$ $\mathrm{NPSV}_t \not\subseteq$ FP, nach Fenner u. a. (2003, Prop. 1).
4. UP ≠ P $\Leftrightarrow \exists$ Einwegfunktionen, nach Grollmann und Selman (1988, Thm. 10).
5. UP ∩ coUP ≠ P $\Leftrightarrow \exists$ Einwegpermutationen, nach Homan und Thakur (2003).
6. NP ∩ coNP $\Leftrightarrow$ NPSV_t hat keine vollständige Funktion, nach Beyersdorff, Köbler und Messner (2009, Prop. 3).
7. DisjNP $\Leftrightarrow$ NPSV hat keine vollständige Funktion, nach Glaßer, Selman und Sengupta (2005, Thm. 9).

Es gelten die eingezeichneten Implikationen:
1. $\mathsf{DisjNP} \Rightarrow \mathsf{TAUT}^{\mathsf{N}}$ nach Köbler, Messner und Torán (2003, Cor. 6.1).
2. $\mathsf{UP} \Rightarrow \mathsf{TAUT}$ nach Köbler, Messner und Torán (2003, Cor. 4.1).
3. $\mathsf{TAUT}^{\mathsf{N}} \Rightarrow \mathrm{NEE} \neq \mathrm{coNEE}$ nach Köbler, Messner und Torán (2003, Cor. 7.1).
4. $\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P} \Rightarrow \neg\mathsf{Q}' \Rightarrow \mathrm{NPMV_t} \not\subseteq_{\mathrm{c}} \mathrm{TFNP} \Rightarrow \neg\mathsf{Q}$ nach Fenner u. a. (2003, Prop. 9, Thm. 6).
5. $\mathrm{E} \neq \mathrm{NE} \Rightarrow \exists$ NP-Relation die nicht auf Entscheidung reduzierbar ist, nach Impagliazzo und Sudan (1991).
6. $\mathrm{UP} \neq \mathrm{P} \Rightarrow \exists$ P-inseparierbares DisjNP-Paar, nach Grollmann und Selman (1988, Thm. 5).
7. $\mathsf{NP} \cap \mathsf{coNP} \Rightarrow \mathsf{TAUT} \vee \mathsf{SAT}$ nach Köbler, Messner und Torán (2003, Cor. 5.1).
8. $\mathrm{NPMV_t}$ hat keine vollständige Funktion $\Rightarrow \mathsf{SAT}$ 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. $\mathrm{NPMV_t}$ hat keine vollständige Funktion $\Rightarrow \mathrm{NP} \neq \mathrm{coNP}$ nach Satz 4.25.
10. $\mathsf{SAT}^{\mathsf{eff}} \Rightarrow \mathsf{SAT}, \mathsf{SAT}^{\mathsf{eff}} \Rightarrow \mathsf{KvL}$, nach Satz 4.11.
11. $\mathsf{KvL} \Rightarrow \neg\mathsf{Q}$, nach Satz 4.6.
12. $\neg\mathsf{Q} \Rightarrow \exists$ NP-Relation die nicht auf Entscheidung reduzierbar ist, denn unter $\neg\mathsf{Q}$ gilt mit Satz 4.21 auch die Negation von 4.21(1), also eine NPTM N mit $L(N) = \Sigma^*$ wobei keine Funktion $g \in \mathrm{FP}$ existiert, welche für alle x durch $g(x)$ einen akzeptierenden Rechenweg von $N(x)$ bestimmt. Definiere die NP-Relation R_N mit $(x, \alpha) \in R_N$ genau dann wenn $N(x)$ mit Rechenweg α existiert. Nun gilt nach Vorigem auch $R \notin_{\mathrm{c}} \mathrm{FP} = \mathrm{FP}^{\Sigma^*} = \mathrm{FP}^{L(R)}$.
13. $\mathsf{DisjCoNP} \Rightarrow \mathsf{TFNP} \Rightarrow \mathrm{NPMV_t}$ hat keine vollständig Funktion, nach Pudlák (2017, Prop. 5.6, 5.10).
14. $\mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P} \Rightarrow \exists$ P-inseparierbares DisjNP-Paar, denn wenn alle DisjNP-Paare p-separierbar, 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. $\mathsf{DisjNP} \Rightarrow \exists$ P-inseparierbares DisjNP-Paar; ist klar, denn wenn alle DisjNP-Paare p-separierbar wären, dann wären auch alle Paare trivialerweise $\leq^{\mathrm{pp}}_{\mathrm{m}}$-vollständig.
16. $\mathsf{DisjCoNP} \Rightarrow \exists$ P-inseparierbares DisjCoNP-Paar; ist aus selben Gründen klar.
17. $\mathsf{TAUT}^{\mathsf{N}} \Rightarrow \mathsf{TAUT}$ klar, weil aus p-Optimalität auch Optimalität folgt.
18. $\mathsf{SAT} \Rightarrow \neg\mathsf{Q}$ klar: wen Q, dann ist nach Satz 4.21 jedes optimale Beweissystem auch p-optimal. Dann gilt auch $\neg\mathsf{SAT}$: jede Menge $L \in \mathrm{NP}$ hat ein optimales Beweissystem h (Beobachtung 2.13) und das ist nach Voraussetzung p-optimal.
19. $\mathsf{UP} \Rightarrow \mathrm{UP} \neq \mathrm{P}$ klar.
20. $\mathsf{NP} \cap \mathsf{coNP} \Rightarrow \mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P}$ klar.
21. $\exists$ P-inseparierbares DisjNP-Paar $\Rightarrow \mathrm{P} \neq \mathrm{NP}$ klar.
22. $\mathrm{UP} \cap \mathrm{coUP} \Rightarrow \mathrm{UP} \neq \mathrm{P}$, $\mathrm{UP} \cap \mathrm{coUP} \Rightarrow \mathrm{NP} \cap \mathrm{coNP} \neq \mathrm{P}$ klar.
23. $\mathrm{NEE} \neq \mathrm{coNEE} \Rightarrow \mathrm{NE} \neq \mathrm{coNE} \Rightarrow \mathrm{NP} \neq \mathrm{coNP} \Rightarrow \mathrm{P} \neq \mathrm{NP}$ klar.
24. $\mathrm{NEE} \neq \mathrm{coNEE} \Rightarrow \mathrm{EE} \neq \mathrm{NEE} \Rightarrow \mathrm{E} \neq \mathrm{NE}$ klar. □

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

Satz 4.25. *Wenn* $\mathrm{NP} = \mathrm{coNP}$ *dann existiert eine* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-vollständige Multifunktion* f *für* $\mathrm{NPMV_t}$.

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

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

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

Betrachte nun die Multifunktion f, die durch folgenden nichtdeterministischen Transduktor

$T'(i, x, 1^n)$ berechnet wird:

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

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

- Entweder ist $\mathit{set}\text{-}T_i(x) \neq \emptyset$, dann existiert für jedes $y \in \mathit{set}\text{-}T_i(x)$ ein akzeptierender Rechenweg α der Länge $\leq n$ auf T_i der y ausgibt, und damit wird auch f dieses y in Z. 5 ausgeben. Gleichzeitig ist damit $(i, x, 1^n) \notin U$ und Z. 7 niemals erreicht. Es gilt also $\mathit{set}\text{-}f(i, x, 1^n) = \mathit{set}\text{-}T_i(x)$.
- Oder es gilt $\mathit{set}\text{-}T_i(x) = \emptyset$. Dann wird jeder Rechenweg der Länge $\leq n$ von $T_i(x)$ ablehnen, und f definitiv nicht in Z. 5 akzeptieren. Andererseits gilt dann $(i, x, 1^n) \in U$ und Z. 7 wird auf mindestens einem Rechenweg von f erreicht. Es gilt also $\mathit{set}\text{-}f(i, x, 1^n) = \{\varepsilon\}$.

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 i sodass der nichtdeterministische Transduktor T_i diese Multifunktion g in berechnet, und dabei terminiert $T_i(x)$ in $\leq |x|^i + i$ vielen Schritten.

Nun gilt nach obiger Beobachtung schon dass

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

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

TODO: Orakel angeben

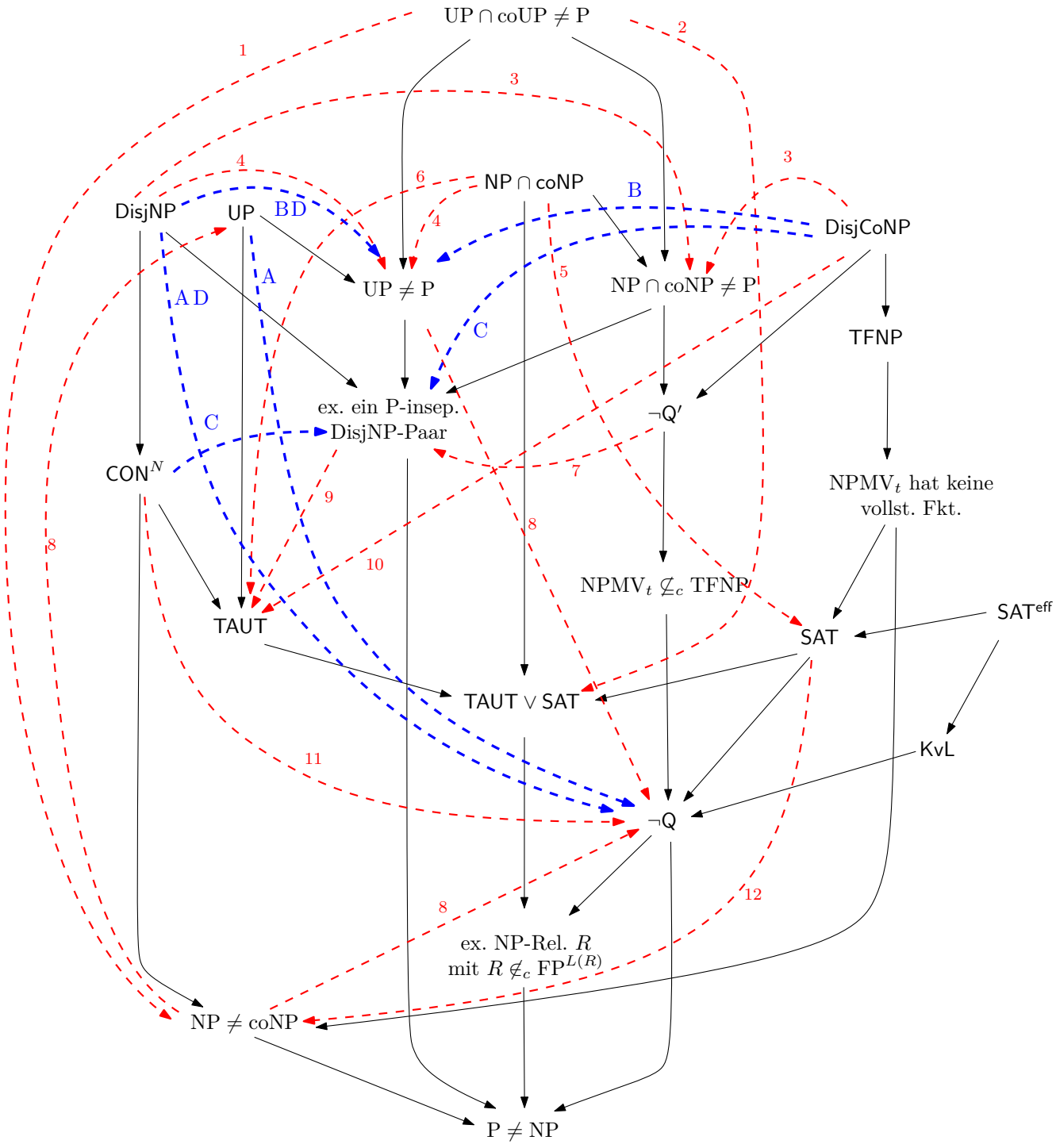

UP ∩ coUP ≠ P
NP ∩ coNP
DisjNP
UP
DisjCoNP
UP ≠ P
NP ∩ coNP ≠ P
TFNP
ex. ein P-insep. DisjNP-Paar
¬Q′
CON^N
NPMV_t hat keine vollst. Fkt.
TAUT
NPMV_t ⊈_c TFNP
SAT
SAT^eff
TAUT ∨ SAT
KvL
¬Q
ex. NP-Rel. R mit R ∉_c FP^{L(R)}
NP ≠ coNP
P ≠ NP
A
B
C
D
1
2
3
4
5
6
7
8
9
10
11
12

Abbildung 7

5 Orakel

6 Diskussion und Fazit

Literatur

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

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

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

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

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

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

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

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

Berman, L. und J. 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.

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

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

Buss, Samuel R. 1996. *Lectures on Proof Theory*. Techn. Ber. Montreal, Kanada: School of Computer Science, McGill University, 1996. 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.

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

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

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

Dose, Titus. 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*. 050. 2019. URL: https://eccc.weizmann.ac.il/report/2019/050/.

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 (MFCS 2022)*. Hrsg. von Stefan Szeider, Robert Ganian und Alexandra Silva. Bd. 241. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl: Schloss Dagstuhl, Leibniz-Zentrum für Informatik, Aug. 2022, 45:1–45:15. DOI: 10.4230/LIPIcs.MFCS.2022.45.

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*. Hrsg. von Jiří Wiedermann und Petr Hájek. Bd. 969. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, S. 277–287. DOI: 10.1007/3-540-60246-1_134.

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.

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

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

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: Cambridge University Press, 2008. 606 S. ISBN: 978-0-521-88473-0.

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

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

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.

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

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, Russel 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. ISSN: 0022-0000. DOI: 10.1016/0022-0000(88)90046-3.

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

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*. Hrsg. von Sanjiv Kapoor und Sanjiva Prasad. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, S. 361–372. DOI: 10.1007/3-540-44450-5_29.

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

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 (besucht am 27. 11. 2023).

Kozen, Dexter. 1997. *Automata and computability*. New York: Springer, 1997. 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 (1973), S. 115–116. URL: https://www.mathnet.ru/ppi914. Eine Übersetzung erscheint bei Trakhtenbrot (1984, S. 399–400).

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

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

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

Messner, Jochen. 2000. „On the simulation order of proof systems“. Diss. Universität Ulm, 2000. URL: https://citeseerx.ist.psu.edu/pdf/bec3958d845653cfa73493258d3b550a17e8defd. 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, 1979. URL: https://hdl.handle.net/1721.1/148954.

Papadimitriou, Christos H. 1994. *Computational complexity*. Reading, Massachusets, USA: Addison-Wesley, 1994. 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 (1944), 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, 2013. ISBN: 978-3-319-00118-0.

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

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, 1975. URL: https://hdl.handle.net/1813/6975.

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

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

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

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

Welsh, Dominic. 1993. *Complexity: Knots, Colourings and Countings*. Cambridge: Cambridge University Press, 1993. ISBN: 978-0-521-45740-8.