Disjoint NP-Pairs and Propositional Proof Systems[1]

C. Glaßer,[2] A. Hughes,[3] A. L. Selman,[4] and N. Wisiol[5]




Abstract

This article surveys results on disjoint NP-pairs, propositional proof systems, function classes, and promise classes—including results that demonstrate close connections that bind these topics together. We illustrate important links between the questions of whether these classes have complete objects and whether optimal proof systems may exist.

1 Introduction

Let Σ be the alphabet $\{0,1\}$. In the following, all sets are subsets of Σ^*. Depending on the context these might be sets of natural numbers or strings that represent efficient codings of data structures. Given two nonempty, disjoint sets A and B, a *separator* of the pair (A,B) is a set S such that $A \subseteq S$ and $B \subseteq \overline{S}$. We say that S *separates* A from B. In general, we are concerned with the computational difficulty of separators. The study of separability (rather, inseparability) arose over the years in different contexts.

Separability in Descriptive Set Theory and Computability Theory. The notion of separability first appeared in descriptive set theory, where Luzin [Luz30] proved the separation theorem for analytic sets. It says that any two disjoint analytic sets are separable by a Borel set. In 1950, researchers began to study separability in the context of computability theory. Here a disjoint pair (A,B) is *computably separable* if there is a decidable separator of (A,B). Otherwise, the pair is *computably inseparable*. Kleene [Kle50] showed the existence of a disjoint pair (A,B) of computably enumerable (c.e.) sets such that (A,B) is computably inseparable. So with respect to separability we have different situations in descriptive set theory and computability theory.

Trakhtenbrot [Tra53] proved that in first-order logic, the set of finitely satisfiable formulas, the set of unsatisfiable formulas, and the set of only infinitely satisfiable formulas are pairwise computably inseparable. Shoenfield [Sho58] showed that every undecidable, computably enumerable degree contains disjoint sets A and B such that (A,B) is computably inseparable.

Smullyan [Smu58] showed for Peano arithmetic (PA), the standard theory of arithmetic, that the set of provable formulas and the set of refutable formulas are computably inseparable.

[2]University at Würzburg, Germany. glasser@informatik.uni-wuerzburg.de.
[3]University at Buffalo, New York. ahughes6@buffalo.edu.
[4]University at Buffalo, New York. selman@buffalo.edu.
[5]University at Buffalo, New York. mail@nils-wisiol.de.

Promise Problems and Disjoint NP-Pairs. Even, Selman, and Yacobi [ESY84] defined a promise problem to be a partial decision problem consisting of two predicates (unary relations) (Q, R), so that for an input word x, if the *promise* predicate Q holds, then we want to know whether the property $R(x)$ holds. Formally, a *promise problem* is a pair of predicates (Q, R). A deterministic Turing machine M *solves* (Q, R) if

$$\forall x \in \Sigma^*[Q(x) \Rightarrow [M(x) \text{ stops } \wedge [M(x) = \text{“}yes\text{”} \Leftrightarrow R(x)]]].$$

Of course, a decision problem is a promise problem with the condition that $Q(x)$ holds for all inputs x.

Promise problems are completely characterized by disjoint pairs [GS88]: Let A be the set of "yes" instances, i.e., instances x so that $Q(x) \wedge R(x)$ and let B be the set of instances x so that $Q(x) \wedge \neg R(x)$. Now a deterministic Turing machine to solve the promise problem must accept all strings in A, must reject all strings in B, and may behave arbitrarily for all other strings in Σ^*. This is the formulation most used in current studies [Gol06].

Both Even, Selman, and Yacobi [ESY84] and Grollmann and Selman [GS88] were interested in analyzing issues in public-key cryptography, and technical considerations led them to be especially interested in disjoint pairs (A, B), where A and B are in the class NP. These are called disjoint NP-pairs, and we let DisjNP denote the collection of all disjoint NP-pairs.

Separability in Computational Complexity. The interesting question now is whether there is a separator of (A, B) that belongs to P. If there is such a separator, then (A, B) is P-*separable*. Otherwise, the pair is P-*inseparable*.

We know that such pairs do exist if $\text{P} \neq \text{UP}$ or $\text{P} \neq \text{NP} \cap \text{coNP}$ (see Proposition 1). Further evidence for the existence of P-inseparable disjoint NP-pairs comes from cryptography. Even, Selman, and Yacobi [ESY84] formulated the problem of cracking a public-key cryptosystem as a promise problem, and Grollmann and Selman [GS88] then observed that secure public-key cryptosystems imply the existence of P-inseparable disjoint NP-pairs. Also, Even, Selman, and Yacobi [ESY84] formulated a conjecture that would imply that no public-key cryptosystem is NP-hard to crack. The conjecture is that every disjoint NP-pair has some separator that is not NP-hard.

Observe the analogies with the concepts from computability theory, where we replace c.e. by NP and decidable by membership in P. It is a theorem that every disjoint pair of c.e. sets has some separator whose Turing degree is strictly less than the degree $0'$ of the c.e.-complete sets [Sho60]. Similarly, we have already noted that computably-inseparable disjoint pairs of c.e. sets do exist.

Disjoint NP-pairs also relate naturally to the theory of proof systems for propositional calculus [Pud03, Raz94], and that is the connection we will primarily explore here. Before we do so, here is an interesting example of a disjoint NP-pair known as the Clique-Coloring pair.

Example 1. *Consider the following sets.*

$$\begin{aligned} CC_0 &= \{(G, k) \mid G \textit{ is a graph that has a clique of size } k\}. \\ CC_1 &= \{(G, k) \mid G \textit{ is a graph that can be colored with } k-1 \textit{ colors}\}. \end{aligned}$$

The sets are disjoint, since a clique of size k cannot be colored with $k-1$ colors. Moreover, CC_0 and CC_1 are NP*-complete, and hence (CC_0, CC_1) is a disjoint NP-pair. Surprisingly, this pair is* P*-separable [Pud03], a result that is based on deep combinatorial ideas by Lovász [Lov79] and Tardos [Tar88].*

2 Separability and Reducibility

The following proposition gives evidence for the existence of P-inseparable disjoint NP-pairs.

Proposition 1. *1. If* $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP}$*, then* DisjNP *contains* P*-inseparable pairs.*

2. If $\mathrm{P} \neq \mathrm{UP}$*, then* DisjNP *contains* P*-inseparable pairs [GS88].*

Proof. If $A \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$, then $(A, \overline{A})$ is a disjoint NP-pair. The only separator is $A \notin \mathrm{P}$.

Now assume $\mathrm{P} \neq \mathrm{UP}$. Let $L \in \mathrm{UP} - \mathrm{P}$ and let N be a nondeterministic, polynomial-time Turing machine that accepts L, that runs in (polynomial) time p, and that has at most one accepting path. Hence the following sets are disjoint sets that belong to NP.

$$A = \{(x,i) \mid \text{there exists an accepting path } w \text{ of } N \text{ on } x \text{ such that the } i\text{-th bit of } w \text{ is } 0\}.$$
$$B = \{(x,i) \mid \text{there exists an accepting path } w \text{ of } N \text{ on } x \text{ such that the } i\text{-th bit of } w \text{ is } 1\}.$$

Assume that (A, B) is separable by a separator $S \in \mathrm{P}$. Consider the algorithm that on input x computes in polynomial time the string $w = \chi_S(x,1)\,\chi_S(x,2) \cdots \chi_S(x, p(|x|))$ and that accepts if and only if w is an accepting path of N on x.

If $x \in L$, then w is the unique accepting path of N on x, and hence the algorithm accepts. If $x \notin L$, then there is no accepting path of N on x, and hence the algorithm rejects. So the algorithms decides L in polynomial time, which contradicts our assumption. □

Reducibilities between disjoint NP-pairs are as important to our study as are reductions between sets in order to properly formulate notions such as NP-completeness and NP-hardness. Also, the reduction notions we define here are identical to those of promise problems in general [ESY84, GS88]. We define polynomial-time many-one and Turing reductions.

Definition 2. Let (A, B) and (C, D) be disjoint pairs.

1. (A, B) is *many-one reducible in polynomial-time* to (C, D), $(A, B) \leq_m^{pp} (C, D)$, if for every separator T of (C, D), there exists a separator S of (A, B) such that $S \leq_m^p T$.

2. (A, B) is *Turing reducible in polynomial-time* to (C, D), $(A, B) \leq_T^{pp} (C, D)$, if for every separator T of (C, D), there exists a separator S of (A, B) such that $S \leq_T^p T$.

The definitions tell us that for every separator of (C, D), there is a separator of (A, B) that is no more complex. If (C, D) is P-separable, then so is (A, B). However, these definitions are nonuniform. Consider $(A, B) \leq_m^{pp} (C, D)$. If S_1 is a separator of (C, D), then for an appropriate polynomial-time computable function f_1, we have $f_1(S_1)$ is a separator of (A, B). However, for a different separator S_2 of (C, D), there might be a different function f_2 so that $f_2(S_2)$ is a separator of (A, B). This nonuniformity make these definitions extremely difficult to work with. Fortunately we have the following equivalent uniform definitions [GS88].

Definition 3. Let (A, B) and (C, D) be disjoint pairs.

1. (A, B) is *uniformly many-one reducible in polynomial time* to (C, D), $(A, B) \leq_{um}^{pp} (C, D)$, if there exists a polynomial-time computable function f such that for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_m^p T$ via f.

2. (A, B) is *uniformly Turing reducible in polynomial time* to (C, D), $(A, B) \leq_{uT}^{pp} (C, D)$, if there exists a polynomial-time oracle Turing machine M such that for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_T^p T$ via M.

Razborov [Raz94] provided yet another definition of many-one reductions.

Definition 4. Let (A, B) and (C, D) be disjoint pairs. (A, B) is Razborov reducible[6] to (C, D), if there exists a polynomial-time computable function λ such that $\lambda(A) \subseteq C$ and $\lambda(B) \subseteq D$.

The following result is fortuitous. A comprehensive proof is in the paper of Glaßer et al. [GSSZ04, Theorems 2.8, 2.10, 2.14].

Lemma 5. *Let (A, B) and (C, D) be disjoint pairs. Then the following assertions are equivalent.*

1. $(A, B) \leq_m^{pp} (C, D)$.
2. $(A, B) \leq_{um}^{pp} (C, D)$.
3. *There exists a polynomial-time computable function λ such that $\lambda(A) \subseteq C$ and $\lambda(B) \subseteq D$.*

Similarly, the following result shows that uniform and nonuniform Turing reductions are equivalent. This result was first proven by Grollmann and Selman [GS88].

Lemma 6. *Let (A, B) and (C, D) be disjoint pairs. Then the following two assertions are equivalent:*

1. $(A, B) \leq_T^{pp} (C, D)$.
2. $(A, B) \leq_{uT}^{pp} (C, D)$.

Therefore, we will only use the notation for many-one and Turing reducibility, but always intend the uniform version. If $(A, B) \leq_m^{pp} (C, D)$ and $(C, D) \leq_m^{pp} (A, B)$, then we write $(A, B) \equiv_m^{pp} (C, D)$. Similarly we define the relation $\equiv_T^{pp}$. Obviously, $\equiv_m^{pp}$ and $\equiv_T^{pp}$ are equivalence relations. The *degree* of a pair $(A, B) \in \mathrm{DisjNP}$ is defined as $\mathbf{d}(A, B) = \{(C, D) \in \mathrm{DisjNP} \mid (A, B) \equiv_T^{pp} (C, D)\}$, i.e., the class of NP-pairs that are Turing equivalent to (A, B). In a canonical way, Turing reductions extend from pairs to Turing-degrees: $\mathbf{d}(A, B) \leq_T^{pp} \mathbf{d}(C, D)$ if $(A, B) \leq_T^{pp} (C, D)$. The *degree structure* of disjoint NP-pairs is the structure of the partial ordering $(\{\mathbf{d}(A, B) \mid (A, B) \in \mathrm{DisjNP}\}, \leq_T^{pp})$.

Razborov raised the question of whether DisjNP contains complete disjoint NP-pairs. His interest was primarily in the question of whether many-one complete pairs exist, but because of the conjecture of Even, Selman, and Yacobi mentioned in the introduction, the question is also interesting for Turing-completeness. We will discuss this in another section. Here we state the formal definitions.

Definition 7. Let (A, B) be a disjoint NP-pair.

1. (A, B) is $\leq_m^{pp}$-hard (resp., $\leq_T^{pp}$-hard) for NP if every separator of (A, B) is $\leq_m^p$-hard (resp., $\leq_T^p$-hard) for NP.

[6] Razborov reducible is not a term commonly used in the literature. We will use it only until we prove its equivalence to many-one reducibility.

2. (A, B) is $\leq_m^{pp}$-complete (resp., $\leq_T^{pp}$-complete) if for every $(C, D) \in$ DisjNP it holds that $(C, D) \leq_m^{pp} (A, B)$ (resp., $(C, D) \leq_T^{pp} (A, B)$).

We conclude this section with a short description of two complexity classes of partial functions. We will mention these briefly in coming sections.

A nondeterministic transducer T computes a value y on an input x if there is an accepting computation of T on x for which y is the final contents of the output tape of T. Such transducers compute multivalued functions.

Definition 8 (see [BLS84, Sel94])**.** We consider the following function classes.

1. NPMV is the set of all partial, multivalued functions computed by nondeterministic polynomial-time-bounded transducers.

2. NPSV is the set of all $f \in$ NPMV that are single-valued.

3. $\text{NPSV}_{\{0,1\}}$ is the class of all 0,1-valued functions in NPSV [KM00] (functions that only take on the value 0 or 1).

Given partial multivalued functions f and g, define g to be a *refinement* of f if $\text{domain}(g) = \text{domain}(f)$ and for all $x \in \text{domain}(g)$ and all y, if $g(x) \mapsto y$, then $f(x) \mapsto y$. Let $\mathcal{F}$ and $\mathcal{G}$ be classes of partial multivalued functions. If f is a partial multivalued function, we define $f \in_c \mathcal{G}$ if $\mathcal{G}$ contains a refinement g of f, and we define $\mathcal{F} \subseteq_c \mathcal{G}$ if for every $f \in \mathcal{F}$, $f \in_c \mathcal{G}$.

Let *sat* be the multivalued function defined by $sat(x) \mapsto y$ if and only if x encodes a propositional formula and y encodes a satisfying assignment of x.

3 Propositional Proof Systems

The resolution principle [Rob65] provides a way to prove the unsatisfiability of Boolean formulas. From the correctness and completeness of the resolution principle it follows that

$$\overline{\text{SAT}} = \{\varphi \mid \text{there exists a resolution proof for } \varphi\}, \text{ and} \tag{3.1}$$

$$\text{TAUT} = \{\varphi \mid \text{there exists a resolution proof for } \neg\varphi\}. \tag{3.2}$$

Given a formula φ and a resolution proof w, we can easily check whether w is a valid resolution proof for $\neg\varphi$. So the following function is polynomial-time computable.

$$f(\langle\varphi, w\rangle) = \begin{cases} \varphi, & \text{if } w \text{ is a valid resolution proof for } \neg\varphi \\ \text{true}, & \text{otherwise.} \end{cases}$$

By (3.2), f maps from Σ^* onto TAUT. Hence we can think of the resolution proof system as a polynomial-time computable function that maps onto TAUT. Cook and Reckhow [CR79] observed that this yields an elegant definition for general propositional proof systems.

Definition 9. A *propositional proof system* (*proof system* for short) is a partial, polynomial-time-computable function f from Σ^* onto TAUT.

For any w, we say w is an *f-proof for φ* if $f(w) = \varphi$. If there is a polynomial p, such that for all φ and all f-proofs w of φ we have $|w| \leq p(|\varphi|)$, then f is called *polynomially bounded*.

Cook and Reckhow started a line of research [CR79] that studies the length of the shortest proof of a propositional tautology relative to the length of the tautology. To this end they introduced the notion of simulation.

Definition 10. Let f and g be proof systems. We say f *simulates* g if there is a function h and a polynomial p such that for all w it holds that $f(h(w)) = g(w)$ and $|h(w)| \leq p(|w|)$. We call h a *translation function*. If h is polynomial-time computable, we say f *p-simulates* g. A proof system that simulates (resp., p-simulates) every other proof system is called *optimal* (resp., *p-optimal*).

So f simulates g if for every g-proof for some φ there exists an f-proof for φ that is at most polynomially longer. An optimal proof system then has, up to a polynomial factor, the shortest proofs for tautologies.

Theorem 11 ([CR79])**.** *There exist polynomially-bounded proof systems if and only if* NP = coNP*.*

Proof. Assume NP = coNP, and let N be an NP-machine accepting TAUT $\in$ coNP. Let

$$f(\langle \varphi, w \rangle) = \begin{cases} \varphi, & \text{if } w \text{ is an accepting path of } N \text{ on input } \varphi \\ \text{true}, & \text{otherwise.} \end{cases}$$

This function is polynomial-time computable and it outputs only tautologies. Moreover, for every tautology φ there exists an accepting path w such that N on φ accepts along w. So $f(\langle \varphi, w \rangle) = \varphi$, which shows that f is onto TAUT. Hence f is a proof system, which is even polynomially-bounded, since $|w|$ is polynomially-bounded in $|\varphi|$.

To prove the converse, suppose there is a polynomially-bounded proof system f. Since TAUT is coNP-complete, it suffices to show TAUT $\in$ NP. Let N be a nondeterministic Turing machine that on input φ first guesses a polynomial-length f-proof w and then accepts if and only if $f(w) = \varphi$. This is an NP-machine that accepts TAUT. □

Corollary 12. *If* NP = coNP*, then optimal proof systems exist.*

Proof. Assume NP = coNP. By Theorem 11, there exists a polynomially-bounded proof system f. We show that f simulates every proof system g. Let

$$h(w) = \text{the lexicographically smallest } v \text{ such that } f(v) = g(w).$$

Hence $f(h(w)) = g(w)$. Since f is a polynomially-bounded proof system and g is polynomial-time computable, there are polynomials p, q such that $|h(w)| \leq q(|g(w)|) \leq p(|w|)$. So f simulates g. □

Köbler, Meßner, and Torán [KMT03] proved sufficient conditions for the existence of optimal proof systems under the weaker (but still unlikely) hypothesis that NEE $\cap$ TALLY $\subseteq$ coNEE. As they state in their paper, "the sufficient conditions show however that it would be very hard to prove that optimal proof systems do not exist, since this would imply a separation of complexity classes."

4 Disjoint NP-Pairs that Are Complete or NP-Hard

One of the most interesting open questions about disjoint NP-pairs is whether there exist complete pairs [Raz94].[7] A proof of the nonexistence implies NP $\neq$ coNP, and we will relate complete pairs to propositional proof systems [Raz94]. Also, it is an open question as to whether disjoint NP-pairs can be NP-hard [ESY84].[8] Even, Selman, and Yacobi [ESY84] conjectured the following:

- ESY-T: No disjoint NP-pair is $\leq_T^{pp}$-hard for NP.

This conjecture would imply that no public-key cryptosystem has an NP-hard cracking problem. Moreover, the conjecture has fascinating consequences for computational complexity: ESY-T implies that NP $\neq$ coNP, NP $\neq$ UP, and satisfying assignments of Boolean functions cannot be computed by single-valued NP-machines (i.e., NPMV $\not\subseteq_c$ NPSV) [ESY84, GS88, Sel94].

Recently, Hughes et al. [HPRS12] have studied variants of this conjecture, of which we mention here the following:

- ESY-*tt*: No disjoint NP-pair is $\leq_{tt}^{pp}$-hard for NP.
- ESY-m: No disjoint NP-pair is $\leq_m^{pp}$-hard for NP.

For a reduction r, the ESY-r conjecture says that for every disjoint NP-pair there exists a separator that is not r-hard for NP. Note that ESY-T implies ESY-*tt*, and ESY-*tt* implies ESY-m. Moreover, the ESY-conjectures immediately relate to the existence of complete pairs.

Theorem 13. *If* ESY-r does not hold, then there exists an r-complete disjoint NP pair.

Proof. Assume that ESY-r does not hold, then there is a pair (A, B) of disjoint sets in NP such that every separator is r-hard for NP. We claim (A, B) is r-complete for DisjNP. Let $(C, D) \in$ DisjNP, and let S be any separator for (A, B). Then since $C \in$ NP and S is r-hard for NP, we have $C \leq_r S$. This proves C, which is a separator of (C, D), reduces to any separator of (A, B). By definition 2, we have $(C, D) \leq_r^{pp} (A, B)$ and thus (A, B) is r-complete for DisjNP. □

For the ESY-*tt* conjecture we know the same consequences for computational complexity as for ESY-T.

Theorem 14 ([HPRS12])**.** *If* ESY-*tt* holds, then NP $\neq$ coNP, NP $\neq$ UP, and NPMV $\not\subseteq_c$ NPSV.

For this reason, ESY-*tt* is probably as difficult to settle as is ESY-T. We should note in this regard that NPMV $\subseteq_c$ NPSV holds only if the polynomial hierarchy collapses to $\mathrm{ZPP}^{\mathrm{NP}}$ [HNOS96] (and indeed even to $S_2^{\mathrm{NP}\cap\mathrm{coNP}}$ [CCHO05]). However, the following result settles ESY-m as well as can be expected.

Theorem 15 ([GSSZ04])**.** ESY-m holds if and only if NP $\neq$ coNP.

Proof. Assume ESY-m does not hold. Let $(A, B) \in$ DisjNP such that all separators are many-one-hard for NP. The set $\overline{B}$ is a separator of (A, B), and therefore SAT $\leq_m^p \overline{B}$. Thus $\overline{\mathrm{SAT}} \leq_m^p B$, which means that $\overline{\mathrm{SAT}} \in$ NP. It follows by the completeness of SAT that NP = coNP.

Now assume NP = coNP. The pair $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ has only separators that are many-one-hard for NP. □

[7] For c.e. sets this question has an affirmative answer: There exist many-one complete pairs of disjoint c.e. sets [Rog67, Theorem XII(c)].

[8] For c.e. sets this question has a negative answer: Every pair of disjoint c.e. sets has a separator whose Turing-degree is strictly less than the degree $0'$ of the c.e.-complete sets [Sho60].

5 Optimal Propositional Proof Systems

In this section, we present a proof of a result by Köbler, Meßner, and Torán [KMT03]. Here we show that the existence of an optimal proof system implies the existence of a complete set for NP ∩ SPARSE, the sparse sets that belong to NP. We tend to believe that there are no complete sets for NP ∩ SPARSE. Therefore, this important result by Köbler, Meßner, and Torán provides evidence that there are no optimal proof systems.

Theorem 16 ([KMT03])**.** *If optimal proof systems exist, then* NP ∩ SPARSE *has many-one complete sets.*

Proof. We define the set SP, containing descriptions of nondeterministic Turing machines that do not accept too many different strings if we restrict the runtime and the length of the input.

$$\begin{aligned} SP = \{\langle N, 0^l, 0^n\rangle \mid\ & (1)\ N \text{ is a nondeterministic Turing machine} \\ & (2) \text{ there are at most } l \text{ pairs } (x_i, y_i) \text{ such that} \\ & \quad \text{(a) all } x_i \text{ are distinct} \\ & \quad \text{(b) } |x_i| \le n, |y_i| \le l \\ & \quad \text{(c) } N \text{ accepts } x_i \text{ on path } y_i\}. \end{aligned}$$

A triple $\langle N, 0^l, 0^n\rangle$ does not belong to SP if and only if N is not the encoding of a nondeterministic Turing machine or there exist $l+1$ pairs (x_i, y_i) such that (a)-(c) holds. This shows $SP \in$ coNP.

Now we define polynomial-time computable subsets of SP. Assume N is a nondeterministic Turing machine with polynomial runtime q such that for every n, N accepts at most $q(n)$ strings of length at most n. Hence $\langle N, 0^{q(n)}, 0^n\rangle \in SP$ for all $n \ge 1$. So the set $SP_N = \{\langle N, 0^{q(n)}, 0^n\rangle \mid n \ge 1\}$ is a subset of SP. Moreover, SP_N is polynomial-time decidable: On input $\langle M, 0^l, 0^n\rangle$, the program checks whether $M = N$ and $l = q(n)$, where N and q are coded into the program.

We now define a set $S \in$ NP ∩ SPARSE and use the sets SP_N to prove the completeness of S. Let h be an optimal proof system and let γ be a many-one reduction from SP to TAUT. Define

$$\begin{aligned} S = \{\langle 0^N, 0^j, 0^l, x\rangle \mid\ & (\text{I})\ N \text{ is a nondeterministic Turing machine} \\ & (\text{II}) \text{ there exists a string } w \text{ of length at most } j \text{ such that} \\ & \quad \text{(a) } h(w) = \gamma(\langle N, 0^l, 0^{|x|}\rangle) \\ & \quad \text{(b) } N \text{ accepts } x \text{ in at most } l \text{ steps}\}. \end{aligned}$$

S belongs to NP. To see that S is sparse, we fix N, j, l, and $m = |x|$. If $\langle 0^N, 0^j, 0^l, x\rangle \in S$ where $|x| = m$, then by (II)(b), N accepts x in at most l steps. By (II)(a), $\gamma(\langle N, 0^l, 0^{|x|}\rangle) \in$ TAUT and hence $\langle N, 0^l, 0^{|x|}\rangle \in SP$. So there are at most l inputs of length at most $|x| = m$ that are accepted by N in at most l steps. Thus for fixed N, j, l, and m there are at most l triples $\langle 0^N, 0^j, 0^l, x\rangle \in S$ such that $|x| = m$. For triples of length n there is only a polynomial number of possibilities for the values N, j, l, and $m = |x|$, because of the tally encoding of N, j, and l. Hence S is sparse.

We argue that every $S' \in$ NP∩SPARSE many-one reduces to S. Let q be a polynomial such that S' contains at most $q(n)$ strings of length at most n, and S' is accepted by N in time q. Since we know that SP_N is polynomial-time decidable, we can define a polynomial-time-computable function

$$g_N(w) = \begin{cases} \gamma(z) & \text{if } w = 1z \text{ and } z \in SP_N, \\ h(z) & \text{if } w = 0z, \\ h(\varepsilon) & \text{otherwise,} \end{cases}$$

with range TAUT. So g_N is a proof system and thus is simulated by the optimal proof system h. Hence, there exists a translation function λ and a polynomial r such that for all v, we have $h(\lambda(v)) = g_N(v)$ and $|\lambda(v)| \leq r(|v|)$. Let $r'(n) = r(|1\langle N, 0^{q(n)}, 0^n\rangle|)$. We show that S' reduces to S via the polynomial-time-computable function $x \mapsto \langle 0^N, 0^{r'(|x|)}, 0^{q(|x|)}, x\rangle$.

Assume $x \in S'$. Let $z = \langle N, 0^{q(|x|)}, 0^{|x|}\rangle$. By definition, $z \in SP_N$. Thus, $g_N(1z) = \gamma(z)$, and therefore $h(\lambda(1z)) = \gamma(z)$. Define $w = \lambda(1z)$, $j = r'(|x|)$, and $l = q(|x|)$. So $|w| = |\lambda(1z)| \leq r(|1z|) = r'(|x|) = j$ and $h(w) = \gamma(z)$, which shows that w satisfies condition (II)(a). Moreover, N accepts x in at most l steps, since $x \in S'$. This shows $\langle 0^N, 0^{r'(|x|)}, 0^{q(|x|)}, x\rangle \in S$.

For $x \notin S'$ condition (II)(b) is not satisfied and hence $\langle 0^N, 0^{r'(|x|)}, 0^{q(|x|)}, x\rangle \notin S$. □

Köbler, Meßner, and Torán [KMT03] show that this proof can be generalized to other promise classes such as UP, Few, FewP, NP ∩ SPARSE, and NP ∩ coNP.

Theorem 17 ([KMT03])**.**

1. *If p-optimal proof systems exist, then* UP *has many-one complete sets.*
2. *If optimal proof systems exist, then* UP *contains sets that are complete under nonuniform many-one reducibility.*

Another remarkable consequence of the existence of optimal proof systems is the existence of complete NP-pairs, first proven by Razborov [Raz94]. Our proof requires preparation, which we provide in the next section.

6 Canonical Disjoint NP-Pairs for Proof Systems

In the introduction we mentioned the inseparability of Peano Arithmetic. The analogous question for propositional calculus is extremely difficult, since the set of provable formulas SAT and the set of refutable formulas $\overline{\text{SAT}}$ are P-separable if and only if P = NP. Razborov [Raz94] studied the variant of this question that asks for *short* proofs that refute a formula. For a propositional proof system f, he defined the *canonical pair* $(\text{SAT}^*, \text{REF}_f)$, where

$$\begin{aligned} \text{SAT}^* &= \{(\varphi, 1^m) \mid \varphi \in \text{SAT and } m \geq 0\} \text{ and} \\ \text{REF}_f &= \{(\varphi, 1^m) \mid \neg\varphi \in \text{TAUT and } \exists y, |y| \leq m, \text{ such that } f(y) = \neg\varphi\}. \end{aligned}$$

These sets are disjoint and they belong to NP. We know that there exists a propositional proof system f such that SAT^* and REF_f are P-inseparable if and only if NP contains a pair of disjoint, P-inseparable sets [GSZ07]. Moreover, the notion of canonical pairs is closely related to the notion of simulation of proof systems.

Proposition 18 ([Raz94])**.** *Let f and g be propositional proof systems. If f simulates g, then* $(\text{SAT}^*, \text{REF}_g) \leq_m^{pp} (\text{SAT}^*, \text{REF}_f)$.

Proof. Since f simulates g, there is a function h and a polynomial p such that $g(w) = f(h(w))$ and $|h(w)| \leq p(|w|)$. Define $\lambda(\varphi, 0^n) = (\varphi, 0^{p(n)})$, we show $\lambda(\text{SAT}^*) \subseteq \text{SAT}^*$ and $\lambda(\text{REF}_g) \subseteq \lambda(\text{REF}_f)$.

If $(\varphi, 0^n) \in \text{SAT}^*$, then $(\varphi, 0^{p(n)}) \in \text{SAT}^*$. If $(\varphi, 0^n) \in \text{REF}_g$, then $\neg\varphi \in \overline{\text{TAUT}}$ and there is some y such that $|y| \leq n$ and $g(y) = \neg w$. So $\neg w = g(y) = f(h(y))$ and $|h(y)| \leq p(|y|)$. Hence $(w, 0^{p(n)}) \in \text{REF}_f$. □

The next theorem shows the close relation between disjoint NP-pairs and canonical pairs.

Theorem 19 ([GSZ07])**.** *For every* $(A, B) \in \mathrm{DisjNP}$*, there exists a proof system* f *such that* $(A, B) \equiv_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$*.*

Proof. Let $\langle \cdot, \cdot \rangle$ be a polynomial-time computable, polynomial-time invertible pairing function such that $|\langle v, w \rangle| = 2|vw|$. Choose a function g that is polynomial-time computable and polynomial-time invertible such that $A \leq_m^p \mathrm{SAT}$ via g (such a g exists, since SAT is a paddable NP-complete set). Let M be an NP-machine that accepts B in time p. Define the following function f:

$$f(z) \stackrel{def}{=} \begin{cases} \neg g(x) & \text{if } z = \langle x, w \rangle, |w| = p(|x|), M(x) \text{ accepts along path } w \\ x & \text{if } z = \langle x, w \rangle, |w| \neq p(|x|), |z| \geq 2^{|x|}, x \in \mathrm{TAUT} \\ \text{true} & \text{otherwise.} \end{cases}$$

The function f is polynomial-time computable, since in the second case $|z|$ is large enough so that $x \in \mathrm{TAUT}$ can be decided by the brute-force algorithm in deterministic time $O(|z|^2)$. (Note that in the second case, the condition $|z| \geq 2^{|x|}$ is equivalent to the condition $\log |z| \geq |x|$.) In the first case of f's definition, $x \in B$ and so $g(x) \notin \mathrm{SAT}$. It follows that $f : \Sigma \to \mathrm{TAUT}$. The mapping is onto, since for every tautology y,

$$f(\langle y, 0^{2^{|y|}} \rangle) = y.$$

Therefore, f is a propositional proof system.

Claim 20. $(\mathrm{SAT}^*, \mathrm{REF}_f) \leq_m^{pp} (A, B)$*.*

Choose elements $a \in A$ and $b \in B$. Define the reduction function h as follows.

```
1  Input (y, 0^n)
2  if n ≥ 2^|y| then
3      if y ∈ SAT then output a else output b
4  endif
5  if g^{-1}(y) exists then output g^{-1}(y)
6  output a.
```

The condition in line 2 is equivalent to $\log n \geq |y|$. The exhaustive search in line 3 is possible in quadratic time in n. So h is computable in polynomial time.

Assume $(y, 0^n) \in \mathrm{SAT}^*$. If we reach line 3, then we output $a \in A$. Otherwise we reach line 5. If $g^{-1}(y)$ exists, then it belongs to A. Therefore, in either case (output in line 5 or in line 6) we output an element from A.

Assume $(y, 0^n) \in \mathrm{REF}_f$ (in particular $\neg y \in \mathrm{TAUT}$). So there exists z such that $|z| \leq n$ and $f(z) = \neg y$. If we reach line 3, then we output b. Otherwise we reach line 5 and so it holds that $|z| \leq n < 2^{|y|}$ and $\neg y$ syntactically differs from the expression true. Therefore, $f(z) = \neg y$ must be due to line 1 in the definition of f. It follows that $g^{-1}(y)$ exists and belongs to B (again by line 1 of f's definition). This shows Claim 20.

Claim 21. $(A, B) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$*.*

The reduction function is $h'(x) \stackrel{def}{=} (g(x), 0^{2(|x|+p(|x|))})$. If $x \in A$, then $g(x) \in \mathrm{SAT}$ and therefore, $h'(x) \in \mathrm{SAT}^*$. Otherwise, let $x \in B$. Let w be an accepting path of $M(x)$ and define $z \stackrel{def}{=} \langle x, w \rangle$. So $|w| = p(|x|)$ and $|z| = 2(|x| + p(|x|))$. By line 1 in f's definition, $f(z) = \neg g(x)$. Therefore, $h'(x) \in \mathrm{REF}_f$. This proves Claim 21 and finishes the proof of Theorem 19. □

Corollary 22. *Disjoint* NP*-pairs and canonical pairs for proof systems have identical degree structure.*

Regarding this degree structure, Glaßer, Selman, and Zhang [GSZ07] proved that between any two comparable and inequivalent disjoint NP-pairs (A, B) and (C, D) there exist P-inseparable, incomparable NP-pairs (E, F) and (G, H) whose degrees lie strictly between (A, B) and (C, D). This is an analogue of Ladner's result for NP [Lad75]. The proof is based on previous work by Schöning [Sch82] and Regan [Reg83, Reg88]. Thus, assuming that P-inseparable disjoint NP-pairs exist, the class DisjNP has a rich, dense, degree structure and each of these degrees contains a canonical pair.

From Proposition 18 and Theorem 19 we obtain an important connection between optimal proof systems and complete NP-pairs.

Corollary 23. *If f is an optimal proof system, then* $(\mathrm{SAT}^*, \mathrm{REF}_f)$ *is complete for* DisjNP*.*

Proof. For any $(A, B) \in \mathrm{DisjNP}$, there is a proof system g such that $(A, B) \equiv_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_g)$. Since f simulates g, we have $(A, B) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_g) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. □

It is an open question whether the converse of Corollary 23 holds: Does the existence of many-one complete disjoint NP-pairs imply the existence of optimal proof systems? Section 8 suggests that it is difficult to answer this question, since there exist oracles demonstrating that both answers are possible.

7 Uniform Enumerations

The notion of uniform enumeration—and most particularly the linking the existence of complete sets to the existence of enumerations of machines echoing the flavor of the class—was first explored in the 1980s in a flurry of papers on the (possible non)existence of complete sets [Sip82, Kow84, HI85, HH88]. In this section we discuss such a link for the classes DisjNP and NPSV.

Definition 24. Suppose $\{N_i\}_{i\geq 0}$ is an effective enumeration of nondeterministic polynomial-time machines. DisjNP is *uniformly enumerable* if there is a total computable function $f : \Sigma^* \to \Sigma^* \times \Sigma^*$ such that

1. $\forall (i, j) \in \mathrm{range}(f)[(L(N_i), L(N_j)) \in \mathrm{DisjNP}]$, and
2. $\forall (C, D) \in \mathrm{DisjNP} \exists (i, j)[(i, j) \in \mathrm{range}(f) \wedge C = L(N_i) \wedge D = L(N_j)]$.

Definition 25. Suppose $\{N_i\}_{i\geq 0}$ is an effective enumeration of nondeterministic polynomial-time transducers and h_i denotes the partial, multivalued function computed by N_i. NPSV is *uniformly enumerable* if there is a total computable function $f : \Sigma^* \to \Sigma^*$ such that

1. $\forall i \in \mathrm{range}(f)[h_i \in \mathrm{NPSV}]$, and
2. $\forall g \in \mathrm{NPSV}\ \exists i\ [i \in \mathrm{range}(f) \wedge h_i = g]$.

Definition 26. Let $f, g \in \mathrm{NPSV}$. g is *polynomial-time many-one reducible* to f ($g \leq_m^p f$) if there is a deterministic polynomial-time function h such that $f(h(x)) = g(x)$.

Definition 27. Let $f \in \mathrm{NPSV}$. f is $\leq_m^p$-complete for NPSV if for all $g \in \mathrm{NPSV}$, $g \leq_m^p f$.

The following theorem is extracted from Glaßer et al. [GSS05], where seven distinct open questions are proved to be logically equivalent.

Theorem 28 ([GSS05])**.** *The following are equivalent.*

1. *There is a* $\leq_m^{pp}$*-complete disjoint* NP*-pair.*
2. DisjNP *is uniformly enumerable.*
3. NPSV *has a* $\leq_m^p$*-complete function.*
4. NPSV *is uniformly enumerable.*

Proof. Glaßer, et al. [GSS05] have shown the equivalence of statements (1) - (3). We will give the proof for the equivalence of statements (3) and (4).

Let $\{M_i\}_{i\geq 0}$ (resp., $\{N_i\}_{i\geq 0}$) be a standard enumeration of deterministic (resp., nondeterministic) polynomial-time transducers. In the proof below we identify a transducer with the function that is computed by this transducer.

To show that statement (3) implies statement (4), assume $f \in$ NPSV is a $\leq_m^p$-complete function. Then define the enumerating function $h : \Sigma^* \to \Sigma^*$ as $h(i) = \langle f \circ M_i \rangle$. Suppose $g \in$ NPSV. Then $g \leq_m^p f$ via some M_i. So $g(x) = f(M_i(x))$ for all x and $h(i) = \langle f \circ M_i \rangle$. For all i, $h(i)$ outputs $f \circ M_i$ and $f \in$ NPSV so $h(i) \in$ NPSV. Therefore h is a uniform enumeration for NPSV.

To show that statement (4) implies statement (3), assume that f is a uniform enumeration for NPSV. Let M compute f. Then we will define the following function h,

$$h(z) = N_i(x) \qquad \text{if } z = 0^n 10^t 1x \text{ and } M(n) = i \text{ within } t \text{ steps.}$$

Given $z = 0^n 10^t 1x$, $h(z)$ uses a polynomial amount of time to compute $M(n)$ for t steps and to simulate $N_i(x)$. Since N_i computes a function in NPSV, this gives that h is also single-valued. So $h \in$ NPSV.

Suppose $g \in$ NPSV. Then there exists a nondeterministic transducer N_ℓ such that $g = N_\ell$. So there exists an n and t such that $M(n) = \ell$ in t steps. Then the function $\lambda(x) = 0^n 10^t 1x$ is a polynomial-time reduction from g to h. This gives that h is $\leq_m^p$-complete for NPSV. □

It seems improbable that items (2) and (4) hold. Therefore, it seems that DisjNP does not have $\leq_m^{pp}$-complete pairs. Hence, by Corollary 23 it seems that optimal proof systems do not exist.

8 Relativization

Several questions regarding disjoint NP-pairs and propositional proof systems remain open. For some of them we know that an answer to the question solves prominent open problems in computational complexity. For example, proving or disproving the ESY-m conjecture immediately solves the NP versus coNP problem (see Theorem 15). This is a satisfactory explanation for the difficulty of the question. For some other open questions we do not have such a connection to prominent open problems. In this situation, the construction of oracles relative to which the question is answered in each direction can give some evidence for the difficulty of the question.

8.1 Existence of Optimal Proof Systems

Krajíček and Pudlák [KP89] constructed an oracle relative to which optimal proof systems do not exist. Buhrman et al. [BFFvM00] constructed an oracle relative to which NP $\cap$ SPARSE has no many-one complete sets. Köbler, Meßner, and Torán [KMT03] show with a relativizing proof that this implies that optimal proof systems do not exist (see Theorem 16).

For an oracle relative to which optimal proof systems do exist, one can use the simple oracle by Baker, Gill, and Solovay [BGS75], relative to which P = NP. By Corollary 12, it follows that optimal proof systems exist relative to this oracle as well.

8.2 Do Complete Pairs Imply Optimal Proof Systems?

By Corollary 23, we know the following implication: If optimal proof systems exist, then DisjNP has many-one complete pairs. It is open whether the converse of this implication holds. Here the oracles O_1 and O_2 by Glaßer et al. [GSSZ04] give evidence for the difficulty of the question. Relative to both oracles complete NP-pairs exist, but relative to O_1 (resp., O_2) optimal proof systems exists (resp., do not exist).

8.3 P-Inseparable Pairs versus Optimal Proof Systems

The following is known regarding the relationship between the existence of optimal proof systems and the existence of P-inseparable NP-pairs. Rackoff [Rac82] constructs an oracle relative to which P $\neq$ NP = coNP. By Proposition 1 and Corollary 12, relative to this oracle there exist optimal proof systems and P-inseparable NP-pairs. Baker, Gill, and Solovay [BGS75] construct an oracle relative to which P = NP. By Corollary 12, relative to this oracle there exist optimal proof systems but no P-inseparable NP-pairs. Glaßer et al. [GSSZ04] construct an oracle relative to which optimal proof systems do not exist, but P-inseparable NP-pairs exist. We do not know of an oracle for the remaining possibility that optimal proof systems and P-inseparable NP-pairs do not exist. Such an oracle would tell us that the implication "if all NP-pairs are P-separable, then optimal proof systems exist"is difficult.

Another open question is inspired by Shoenfield's result [Sho58] that every undecidable, computably enumerable degree contains disjoint sets A and B such that (A, B) is computably inseparable. Is there a complexity-theoretic analog? We know that if P-inseparable NP-pairs exist, then the degree of NP-complete sets contains disjoint sets A and B that are P-inseparable. Do we find such sets in all degrees in NP $-$ P?

8.4 Separation of ESY-m, ESY-tt, and ESY-T

Clearly, ESY-T implies ESY-tt, and ESY-tt implies ESY-m. What about the converse implication? Glaßer and Wechsung [GW03] constructed an oracle relative to which UP = NP and NP $\neq$ coNP. It follows from Section 4 that relative to this oracle, ESY-T and ESY-tt do not hold, but ESY-m holds. This gives evidence for the difficulty of proving (ESY-m $\implies$ ESY-tt) or (ESY-m $\implies$ ESY-T). It is an open question whether there exists an oracle relative to which ESY-tt holds, but ESY-T does not.

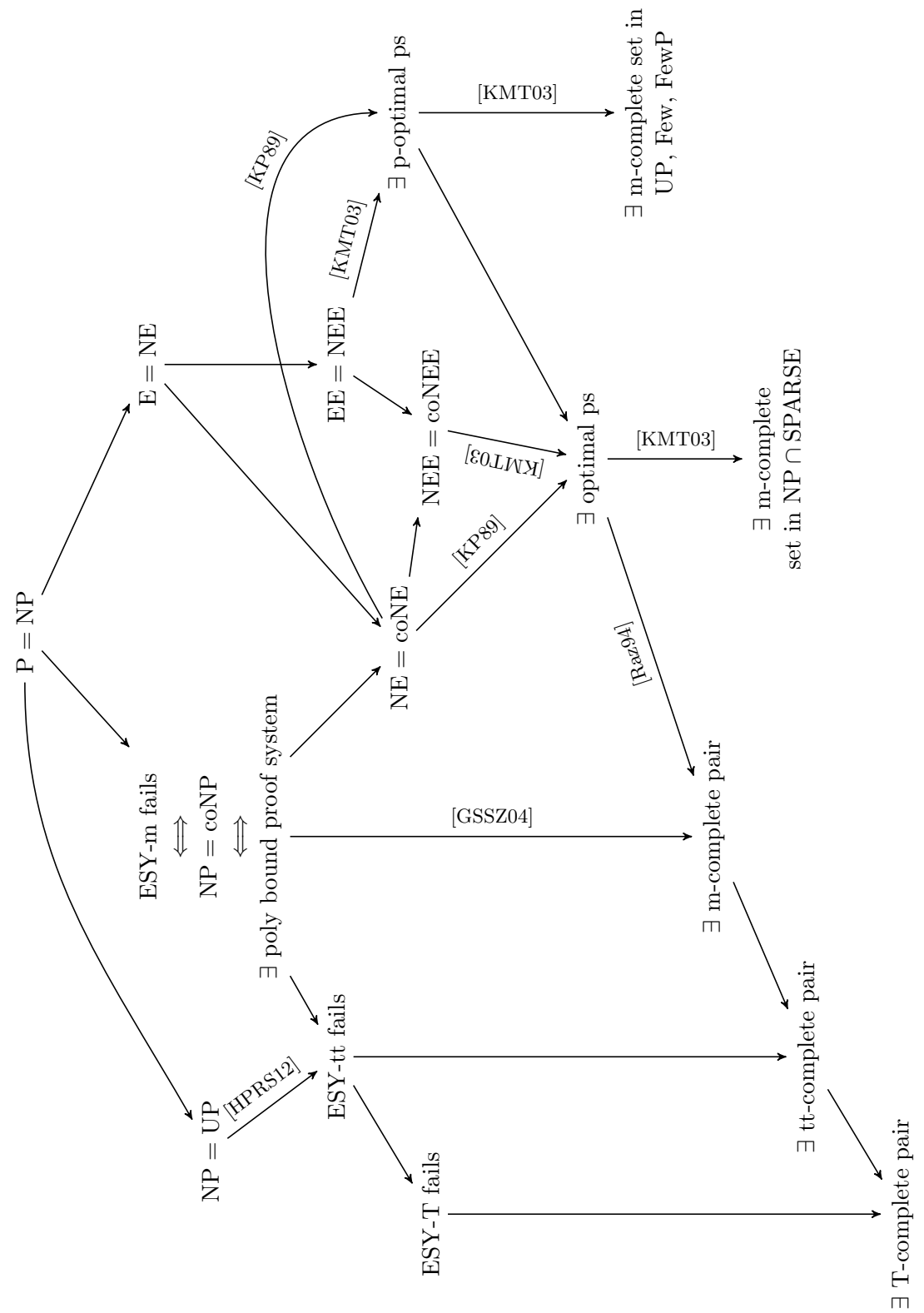

Figure 1: Known Implications for proof systems, disjoint pairs and the ESY conjecture

9 Conclusion and Open Questions

For each assertion shown in Figure 1 it is not known if the assertion is true or false. For each of the implications shown in the chart, it would be interesting to find oracles such that the converse of the implication holds or does not hold.

For the previously mentioned oracle D by Glaßer and Wechsung [GW03], it is an interesting question to see if there exist many-one complete pairs relative to D, since oracle D implies that NP $\neq$ coNP.

The most significant converse to study is the question of whether the existence of many-one complete disjoint NP-pairs implies the existence of optimal proof systems. As stated in Section 8, this question cannot be answered with a relativizable proof.

Our final question concerns random oracles. It is known that NP $\neq$ coNP holds relative to a random oracle [BG81]. So ESY-m holds relative to a random oracle. Also NP $\neq$ UP and NPMV $\not\subseteq_c$ NPSV hold relative to random oracles [NRRS98]. Nevertheless it is open whether ESY-T or ESY-tt hold relative to a random oracle.

References

[BFFvM00] H. Buhrman, S. A. Fenner, L. Fortnow, and D. van Melkebeek. Optimal proof systems and sparse sets. In Horst Reichel and Sophie Tison, editors, *STACS*, volume 1770 of *Lecture Notes in Computer Science*, pages 407–418. Springer, 2000.

[BG81] C. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq$ co-NP^A with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=NP problem. *SIAM Journal on Computing*, 4:431–442, 1975.

[BLS84] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[CCHO05] Jin-yi Cai, Venkatesan T. Chakaravarthy, Lane A. Hemaspaandra, and Mitsunori Ogihara. Competing provers yield improved Karp-Lipton collapse results. *Inf. Comput.*, 198(1):1–23, 2005.

[CR79] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.

[ESY84] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, May 1984.

[Gol06] O. Goldreich. On promise problems: A survey. In O. Goldreich, A. L. Rosenberg, and A. L. Selman, editors, *Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006.

[GS88] J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988.

[GSS05] C. Glaßer, A. L. Selman, and S. Sengupta. Reductions between disjoint NP-pairs. *Inf. Comput.*, 200(2):247–267, 2005.

[GSSZ04] C. Glaßer, A. L. Selman, S. Sengupta, and L. Zhang. Disjoint NP-pairs. *SIAM Journal on Computing*, 33(6):1369–1416, 2004.

[GSZ07] C. Glaßer, A. L. Selman, and L. Zhang. Canonical disjoint NP-pairs of propositional proof systems. *Theor. Comput. Sci.*, 370(1-3):60–73, 2007.

[GW03] C. Glaßer and G. Wechsung. Relativizing function classes. *J. UCS*, 9(1):34–50, 2003.

[HH88] J. Hartmanis and L. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theoretical Computer Science*, 58(1–3):129–142, 1988.

[HI85] J. Hartmanis and N. Immerman. On complete problems for NP $\bigcap$ coNP. In *Proceedings of the 12th International Colloquium on Automata, Languages, and Programming*, pages 250–259. Springer-Verlag *Lecture Notes in Computer Science #194*, July 1985.

[HNOS96] L. A. Hemaspaandra, A. V. Naik, M. Ogihara, and A. L. Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25:697–708, 1996.

[HPRS12] A. Hughes, A. Pavan, N. Russell, and A. L. Selman. A thirty year old conjecture about promise problems. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *International Colloquium on Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 473–484. Springer, 2012.

[Kle50] S. C. Kleene. A symmetric form of Gödel's theorem. In *Proceedings of the section of sciences*, volume 12 of *Koninklijke Nederlandse Akademie van Wetenschappen*, pages 800–802. Springer Verlag, 1950.

[KM00] J. Köbler and J. Meßner. Is the standard proof system for SAT P-optimal? In Sanjiv Kapoor and Sanjiva Prasad, editors, *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 1974 of *Lecture Notes in Computer Science*, pages 361–372. Springer, 2000.

[KMT03] J. Köbler, J. Meßner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Inf. Comput.*, 184(1):71–92, 2003.

[Kow84] W. Kowalczyk. Some connections between representability of complexity classes and the power of formal reasoning systems. In *Proceedings of the 11th Symposium on Mathematical Foundations of Computer Science*, pages 364–369. Springer-Verlag *Lecture Notes in Computer Science #176*, 1984.

[KP89] J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *J. Symb. Log.*, 54(3):1063–1079, 1989.

[Lad75] R. Ladner. On the structure of polynomial-time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[Lov79] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.

[Luz30] N. N. Luzīn. *Leçons sur les ensembles analytiques et leurs applications*. Gauthier-Villars, 1930.

[NRRS98] A. V. Naik, J. D. Rogers, J. S. Royer, and A. L. Selman. A hierarchy based on output multiplicity. *Theor. Comput. Sci.*, 207(1):131–157, 1998.

[Pud03] P. Pudlák. On reducibility and symmetry of disjoint NP-pairs. *Theoretical Computer Science*, 295:323–339, 2003.

[Rac82] C. Rackoff. Relativized questions involving probabilistic algorithms. *J. ACM*, 29(1):261–268, 1982.

[Raz94] A. A. Razborov. On provably disjoint NP-pairs. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(6), 1994.

[Reg83] K. Regan. On diagonalization methods and the structure of language classes. In *Proceedings Foundations of Computation Theory*, volume 158 of *Lecture Notes in Computer Science*, pages 368–380. Springer Verlag, 1983.

[Reg88] K. Regan. The topology of provability in complexity theory. *Journal of Computer and System Sciences*, 36:384–432, 1988.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[Rog67] H. Rogers Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.

[Sch82] U. Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, 18:95–103, 1982.

[Sel94] A. L. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*, 48(2):357 – 381, 1994.

[Sho58] J. R. Shoenfield. Degrees of formal systems. *Journal of Symbolic Logic*, 23:389–392, 1958.

[Sho60] J. R. Shoenfield. Degrees of models. *Journal of Symbolic Logic*, 25(3):233–237, 1960.

[Sip82] M. Sipser. On relativization and the existence of complete sets. In *Proceedings of the 9th International Colloquium on Automata, Languages, and Programming*, pages 523–531. Springer-Verlag *Lecture Notes in Computer Science #140*, July 1982.

[Smu58] R. M. Smullyan. Undecidability and recursive inseparability. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 4:143–147, 1958.

[Tar88] E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[Tra53] B. Trakhtenbrot. On recursive separability. *Dokl. Akad. Nauk SSSR*, 88:953–955, 1953.