Survey of Disjoint NP-pairs and Relations to Propositional Proof Systems

Christian Glaßer[1], Alan L. Selman[2⋆], and Liyu Zhang[2]

[1] Theoretische Informatik, Universität Würzburg,
Am Hubland, 97074 Würzburg, Germany
glasser@informatik.uni-wuerzburg.de

[2] Department of Computer Science and Engineering,
University at Buffalo, Buffalo, NY 14260
{selman,lzhang7}@cse.buffalo.edu

Abstract. We survey recent results on disjoint NP-pairs. In particular, we survey the relationship of disjoint NP-pairs to the theory of proof systems for propositional calculus.

1 Introduction

A *disjoint* NP-*pair* is a pair (A, B) of nonempty, disjoint sets A and B such that both A and B belong to the complexity class NP.[3] We let DisjNP denote the collection of all disjoint NP-pairs. A *separator* of a disjoint NP-pair (A, B) is a set S such that $A \subseteq S$ and $B \subseteq \overline{S}$ (Figure 1). A fundamental question is whether (A, B) has a separator belonging to P. In this case the pair is P-*separable*; otherwise, it is P-*inseparable.*

To state this fundamental question differently, we want to know whether there is an efficient algorithm whose set of yes-instances includes the set A and whose set of no-instances includes the set B. The algorithm behaves arbitrarily on instances in the complement of $A \cup B$. That is, a disjoint NP-pair is a promise problem. To learn about promise problems we refer to Goldreich's survey paper [8] in this volume. The second author first became interested in promise problems, and specifically, in disjoint NP-pairs, in 1982 while working with Shimon Even and Yacov Yacobi. At that time they formulated the problem of cracking a public-key cryptosystem as a promise problem and observed that secure public-key cryptosystems do not exist unless P-*inseparable* pairs exist [4].

Disjoint NP-pairs also relate naturally to the theory of proof systems for propositional calculus [21, 20] and that is the connection we will explore here.

2 Preliminaries

The notations $\leq_m^p$ and $\leq_T^p$ denote polynomial-time-bounded many-one and Turing reducibility, respectively. Thus, we write $A {\leq_m^p} B$ if there is a function f computable in polynomial time, such that for all instances x, $x \in A \Leftrightarrow f(x) \in B$ and

[⋆] Research partially supported by NSF grant CCR-0307077.

[3] Nonemptyness ensures $A \not\subseteq B$ and $B \not\subseteq A$, which simplifies several proofs.

O. Goldreich et al. (Eds.): Shimon Even Festschrift, LNCS 3895, pp. 241–253, 2006.

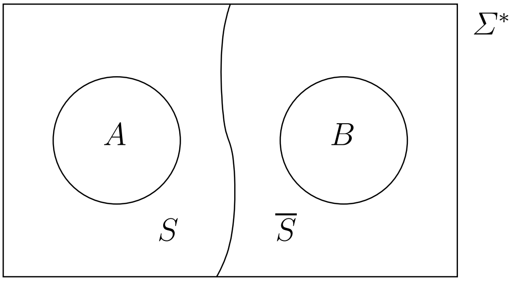

Fig. 1. An NP-pair (A, B) that is separated by S.

we write $A \leq_T^p B$ if there is an oracle Turing machine M such that $A = L(M, B)$ is the language accepted by M using B as the oracle.

We let PF denote the class of all polynomial-time-computable functions. A function f is *honest* if there is a polynomial q such that for every $y \in range(f)$ there exists $x \in dom(f)$ such that $f(x) = y$ and $|x| \leq q(|y|)$. If $f \in$ PF is an honest function and $A \in$ NP, then $f(A) \in$ NP.

To review the definition of standard exponential-time complexity classes,

$$\text{E} = \bigcup\{\text{DTIME}(k^n) \mid k \geq 1\}$$

and

$$\text{NE} = \bigcup\{\text{NTIME}(k^n) \mid k \geq 1\}.$$

For any complexity class $\mathcal{C}$, co$\mathcal{C} = \{L \mid \overline{L} \in \mathcal{C}\}$.

A nondeterministic Turing machine that has at most one accepting computation for any input is called by Valiant [27] an *unambiguous* Turing machine. Let UP denote the set of languages accepted by unambiguous Turing machines in polynomial time. Obviously, P $\subseteq$ UP $\subseteq$ NP, and it is not known whether either inclusion is proper. One reason UP is an interesting complexity class is because there exists a one-to-one, honest function $f \in$ PF whose inverse f^{-1} is not computable in polynomial time if and only if P $\neq$ UP [9].

A set A is *sparse* if there is a polynomial p such that for all n, A contains at most $p(n)$ strings of length n. We let SPARSE denote the collection of all sparse sets.

3 Propositional Proof Systems

Resolution calculus is one well-known example of a propositional proof system. A propositional formula ϕ in disjunctive normal form is a tautology if and only if there exists a proof w in the resolution system showing that $\neg\phi$ is not satisfiable. (Recall that a formula ϕ is a tautology if and only if its negation $\neg\phi$ is not satisfiable.) All propositional proof systems have three properties in common:

1. **Correctness:** If there is a proof in the system, then the formula is indeed a tautology.
2. **Completeness:** Every tautology can be proved within the system.
3. **Verifiability:** The validity of a proof can be easily verified.

Cook and Reckhow [3] formalized the intuitive notion of a proof system as follows: A *propositional proof system* (proof system for short) is a total function $f : \Sigma^* \to \text{TAUT}$ such that f is onto and polynomial-time-computable. (TAUT denotes the set of tautologies.) For any tautology ϕ, if $f(w) = \phi$, then w is a proof in the system f (*f-proof*) showing that ϕ is a tautology.

That the validity of proofs is easy to verify is formalized by the requirement that f is polynomial-time-computable. Furthermore, the definition requires that only tautologies have proofs (correctness) and that every tautology has a proof (completeness).

The following function shows that the resolution calculus can be interpreted as a propositional proof system in the formal sense.

$$f(w) = \begin{cases} \phi & : \quad \text{if } w = (\phi, u) \text{ and } u \text{ is a resolution refutation of } \neg\phi \\ \phi & : \quad \text{if } w = (\phi, u) \text{ and } |u| \geq 2^{|\phi|} \text{ and } \phi \text{ is a tautology} \\ \text{true} & : \quad \text{otherwise} \end{cases}$$

Note that in this definition, both lines, the first and the second one, are necessary. The first line makes sure that tautologies in disjunctive normal form have f-proofs not much longer than the corresponding resolution proofs. The second line takes care of all the tautologies that are not in disjunctive normal form. Note that in this line, the test for tautology can be done by exhaustive search in polynomial time, since $2^{|\phi|} \leq |w|$.

A propositional proof system f is not necessarily honest; it is possible that a formula $\phi \in \text{TAUT}$ has only exponentially long proofs w, i.e., $f(w) = \phi$ and $|w| = 2^{\Omega(|\phi|)}$. A proof system f is *polynomially bounded* if the function f is honest. Cook and Reckhow demonstrated that NP = coNP if and only if there exists a polynomially-bounded proof system, and they proposed attacking the question of whether NP equals coNP by studying propositional proof systems.

Let f and f' be two proof systems. We say that f *simulates* f' if there is a polynomial p and a function $h : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$, $f(h(w)) = f'(w)$ and $|h(w)| \leq p(|w|)$. So for every f'-proof w, $h(w)$ is an f-proof of the same tautology. If f simulates f', then f-proofs are not much longer than f'-proofs. If additionally $h \in \text{PF}$, then we say that f *p-simulates* f'.

A proof system is *optimal* (resp., *p-optimal*) if it simulates (resp., p-simulates) every other proof system. The notion of simulation between proof systems is analogous to the notion of reducibility between problems. Using that analogy, optimal proof systems correspond to complete problems.

It is not known whether there exist optimal propositional proof systems, but the question is interesting, because if there is an optimal proof system f, then there is a polynomially-bounded proof system if and only if f is polynomially-bounded. The question of whether optimal propositional proof systems exist has been studied in detail. Krajíček and Pudlák [19, 13] showed that NE $=$ coNE implies the existence of optimal proof systems. Ben-David and Gringauze [1] and Köbler, Meßner, and Torán [12] obtained the same conclusion under weaker assumptions. On the other hand, Meßner and Torán [18] and Köbler, Meßner, and Torán [12] proved that existence of optimal proof systems results in the existence of $\leq^p_m$-complete sets for the promise class NP $\cap$ SPARSE. In the same paper, they showed that there exist p-optimal proof systems only if the complexity class UP has a many-one complete set. These results hold relative to all oracles. Therefore, optimal proof systems exist relative to any oracle in which NE $=$ coNE holds. Krajíček and Pudlák [13], Ben-David and Gringauze [1], and Buhrman et al. [2] constructed oracles relative to which optimal proof systems do not exist. In addition, NP $\cap$ SPARSE does not have complete sets relative to the latter oracle.

Razborov [21] related the study of propositional proof systems to disjoint NP-pairs. For every propositional proof system f, he associated a canonical disjoint NP-pair. Furthermore, he showed that if f is an optimal proof system, then the canonical pair for f is a complete disjoint NP-pair. We will explain these results in Section 5, but first, in order to define the notion of completeness for disjoint NP-pairs, it is necessary to describe reducibilities between disjoint NP-pairs.

4 Reductions Between Disjoint NP-pairs

Since disjoint pairs are simply an equivalent formulation of promise problems, disjoint pairs easily inherit the natural notions of reducibilities that exist between promise problems [4, 26, 9]. Hence, completeness and hardness notions follow naturally also. We review these here.

Definition 1 *Let (A, B) and (C, D) be disjoint pairs.*

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

The definitions tell us that for every separator of (C, D), there is a separator of (A, B) that is no more complex. In particular, if (C, D) is P-separable, then it follows immediately that (A, B) is P-separable. On the other hand, these definitions are nonuniform. Looking at $(A, B)\leq^{pp}_T(C, D)$, for example, if S_1 is a separator of (C, D), then there is an oracle Turing machine M_1 such that the set $L(M_1, S_1)$ is a separator of (A, B). However, for a different separator S_2 of (C, D), there might be a different Turing machine M_2 so that $L(M_2, S_2)$ is a separator of (A, B). This nonuniformity makes these definitions difficult to work with. Fortunately, they have the following equivalent formulations [9, 6]. Observe that the formulation for many-one reducibility simplifies enormously.

Theorem 2 (uniform reductions for pairs). *Let (A, B) and (C, D) be disjoint pairs.*

1. *(A, B) is* many-one reducible in polynomial-time *to (C, D) if and only if there exists a polynomial-time computable function f such that $f(A) \subseteq C$ and $f(B) \subseteq D$.*
2. *(A, B) is* Turing reducible in polynomial-time *to (C, D) if and only if there exists a polynomial-time oracle Turing machine M such that for every separator T of (C, D), there exists a separator S of (A, B) such that $S \leq_T^p T$ via M. That is, $S = L(M, T)$.*

Now we clearly have uniformity. The same oracle Turing machine M is used for all separators T.

The abbreviation 'pp' in $\leq_T^{pp}$, for example, stands for *polynomial-time-bounded promise reduction.* We retain the promise problem notation in order to distinguish reductions between disjoint NP-pairs from reducibilities between sets.

If $(A, B) \leq_m^{pp} (C, D)$ and $(C, D) \leq_m^{pp} (A, B)$, then we write $(A, B) \equiv_m^{pp} (C, D)$; if $(A, B) \leq_T^{pp} (C, D)$ and $(C, D) \leq_T^{pp} (A, B)$, then we write $(A, B) \equiv_T^{pp} (C, D)$. Obviously, $\equiv_m^{pp}$ and $\equiv_T^{pp}$ are equivalence relations.

Keeping with common terminology, a disjoint pair (A, B) is $\leq_m^{pp}$-complete ($\leq_T^{pp}$-complete) for the class DisjNP if $(A, B) \in$ DisjNP and for every disjoint pair $(C, D) \in$ DisjNP, $(C, D) \leq_m^{pp} (A, B)$ ($(C, D) \leq_T^{pp} (A, B)$, respectively).

Razborov raised the question of whether DisjNP contains complete pairs (i.e., complete disjoint NP-pairs). Although we are primarily interested in the question of whether there exist many-one complete pairs, let's pause for a moment to consider the question of whether there exist Turing-complete pairs. Even, Selman, and Yacobi [4] conjectured that DisjNP does not contain a disjoint pair all of whose separators are NP-hard (i.e., $\leq_T^p$-hard for NP.) The conjecture has strong consequences, for it implies that NP $\neq$ coNP, NP $\neq$ UP, and no public-key cryptosystem is NP-hard to crack [4, 9]. For example, if NP = coNP, then for every NP-complete S, the pair $(S, \overline{S})$ is in DisjNP and all of its separators are NP-hard (since S is the only separator). We conjecture that DisjNP does not contain Turing-complete pairs, but it would be difficult to prove this, because the the latter conjecture implies the former conjecture (which in turn implies NP $\neq$ coNP).

Proposition 3 *If there do not exist $\leq_T^{pp}$-complete pairs for the class* DisjNP*, then* DisjNP *does not contain a disjoint pair all of whose separators are* NP*-hard.*

Proof. Suppose there is a disjoint pair $(A, B) \in$ DisjNP such that all separators are NP-hard. We claim that (A, B) is $\leq_T^{pp}$-complete for DisjNP. Let (C, D) belong to DisjNP. Let S be an arbitrary separator of (A, B). Note that S is NP-hard and $C \in$ NP. So $C \leq_T^p S$. Since C is a separator of (C, D), this demonstrates that $(C, D) \leq_T^{pp} (A, B)$. □

Glaßer et al. [6] constructed an oracle relative to which Turing-complete pairs do not exist for DisjNP.

5 Canonical Disjoint NP-pairs

The canonical pair of a propositional proof system f [21] is the disjoint NP-pair $(\mathrm{SAT}^*, \mathrm{REF}_f)$ where

$$\begin{aligned} \mathrm{SAT}^* &= \{(x, 0^n) \,|\, x \in \mathrm{SAT} \text{ and } n \in \mathrm{N}\} \quad \text{and} \\ \mathrm{REF}_f &= \{(x, 0^n) \,|\, \neg x \in \mathrm{TAUT} \text{ and } \exists y[|y| \leq n \text{ and } f(y) = \neg x]\}. \end{aligned}$$

Informally, SAT^* is the set of satisfiable formulas (i.e., formulas whose negations are not tautologies), and REF_f is the set of easily refutable formulas (i.e., formulas whose negations have short proofs). It is straightforward to see that SAT^* and REF_f are disjoint and that they belong to NP.

The following easy to prove proposition states a strong connection between proof systems and disjoint NP-pairs.

Proposition 4 *Let f and g be propositional proof systems. If g simulates f, then* $(\mathrm{SAT}^*, \mathrm{REF}_f) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_g)$.

Proof. By assumption there exists a total function $h : \Sigma^* \rightarrow \Sigma^*$ and a polynomial p such that for all y, $g(h(y)) = f(y)$ and $|h(y)| \leq p(|y|)$. We claim that $(\mathrm{SAT}^*, \mathrm{REF}_f) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_g)$ via reduction r where $r(x, 0^n) \overset{df}{=} (x, 0^{p(n)})$. Clearly, if $(x, 0^n) \in \mathrm{SAT}^*$, then $(x, 0^{p(n)}) \in \mathrm{SAT}^*$ as well. Let $(x, 0^n) \in \mathrm{REF}_f$, i.e., $\neg x$ is a tautology and there exists y such that $|y| \leq n$ and $f(y) = \neg x$. So for $y' \overset{df}{=} h(y)$ it holds that $|y'| \leq p(n)$ and $g(y') = \neg x$ which shows $(x, 0^{p(n)}) \in \mathrm{REF}_g$. □

Razborov's result (Corollary 8 below) states that if f is an optimal proof system, then $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is a $\leq_m^{pp}$-complete NP-pair. This result is an immediate consequence of Proposition 4 and the following new result [7]. The latter states that every disjoint NP-pair is many-one equivalent to the canonical NP-pair of some propositional proof system.

Theorem 5. *For every disjoint* NP*-pair (A, B) there exists a proof system f such that* $(\mathrm{SAT}^*, \mathrm{REF}_f) \equiv_m^{pp} (A, B)$.

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 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.

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

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

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

Therefore, f is a propositional proof system.

Claim 6 $(\text{SAT}^*, \text{REF}_f) \leq_m^{pp} (A, B)$.

Choose arbitrary elements $a \in A$ and $b \in B$. The reduction function h is as follows.

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

The exhaustive search in Line 3 is possible in quadratic time in $2^{|x|} \leq n$. So $h \in \text{PF}$.

Assume $(x, 0^n) \in \text{SAT}^*$. If we reach Line 3, then we output $a \in A$. Otherwise we reach Line 5. If $g^{-1}(x)$ 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 $(x, 0^n) \in \text{REF}_f$ (in particular $\neg x \in \text{TAUT}$). So there exists y such that $|y| \leq n$ and $f(y) = \neg x$. If we reach Line 3, then we output b. Otherwise we reach Line 5 and so it holds that $|y| \leq n < 2^{|x|}$ and $\neg x$ differs from the expression true (since the expression true does not start with the symbol $\neg$). Therefore, $f(y) = \neg x$ must be due to the first case in the definition of f. It follows that $g^{-1}(x)$ exists. So we output $g^{-1}(x)$ which belongs to B (again by the first case of f's definition). This shows Claim 6.

Claim 7 $(A, B) \leq_m^{pp} (\text{SAT}^*, \text{REF}_f)$.

The reduction function is $h'(x) \overset{df}{=} (g(x), 0^{2(|x|+p(|x|))})$. If $x \in A$, then $g(x) \in \text{SAT}$ and therefore, $h'(x) \in \text{SAT}^*$. Otherwise, let $x \in B$. Let w be an accepting path of $M(x)$ and define $z \overset{df}{=} \langle x, w \rangle$. So $|w| = p(|x|)$ and $|z| = 2(|x| + p(|x|))$. By the first case of f's definition, $f(z) = \neg g(x)$. Therefore, $h'(x) \in \text{REF}_f$. This proves Claim 7 and finishes the proof of Theorem 5. □

Corollary 8 (Razborov) *If there exists an optimal propositional proof system f, then $(\text{SAT}^*, \text{REF}_f)$ is a $\leq_m^{pp}$-complete* NP*-pair.*

Proof. Suppose that f is an optimal proof system. Let (A, B) be an arbitrary disjoint NP-pair. By Theorem 5, let g be a proof system such that

$$(A, B) \equiv_m^{pp} (\text{SAT}^*, \text{REF}_g).$$

We only use $(A,B)\leq_m^{pp}(\mathrm{SAT}^*,\mathrm{REF}_g)$ and the fact that $(\mathrm{SAT}^*,\mathrm{REF}_g)\in\mathrm{DisjNP}$. Since f is optimal, f simulates g. Thus, by Proposition 4,

$$(\mathrm{SAT}^*,\mathrm{REF}_g)\leq_m^{pp}(\mathrm{SAT}^*,\mathrm{REF}_f).$$

Then, $(A,B)\leq_m^{pp}(\mathrm{SAT}^*,\mathrm{REF}_f)$, from which it follows that $(\mathrm{SAT}^*,\mathrm{REF}_f)$ is $\leq_m^{pp}$-complete for DisjNP. □

Also, we state the following corollary. It is convenient for us to define the Turing-degree of a pair $(A,B)\in\mathrm{DisjNP}$ as follows:

$$\mathbf{d}(A,B)=\{(C,D)\in\mathrm{DisjNP}\mid (A,B)\equiv_T^{pp}(C,D)\}.$$

So the Turing-degree of (A,B) is the class of pairs that are equivalent to (A,B) with respect to Turing reductions. 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 $(\{\mathbf{d}(A,B)\mid(A,B)\in\mathrm{DisjNP}\},\leq_T^{pp})$.

Corollary 9 *Disjoint* NP*-pairs and canonical pairs for proof systems have identical degree structure.*

Every disjoint NP-pair we believe to be P-inseparable is many-one equivalent to some canonical pair that is also P-inseparable. We cannot prove that P-inseparable pairs exist, but there is evidence for their existence, for example, if $\mathrm{P}\neq\mathrm{UP}$ or if $\mathrm{P}\neq\mathrm{NP}\cap\mathrm{coNP}$. On the other hand, the hypothesis that $\mathrm{P}\neq\mathrm{NP}$ does not seem to be sufficient to obtain P-inseparable disjoint NP-pairs. Homer and Selman [11] constructed an oracle relative to which $\mathrm{P}\neq\mathrm{NP}$ and all disjoint NP-pairs are P-separable.

Glaßer et al. [6] constructed an oracle O_1 relative to which optimal proof systems exist, and therefore, relative to which many-one complete disjoint NP-pairs exist. Also, they constructed an oracle O_2 relative to which many-one complete disjoint NP-pairs exist, but optimal proof systems do not exist. So relative to this oracle, the converse of Corollary 8 does not hold. Relative to O_2, there is a propositional proof system f whose canonical pair is complete, but f is not optimal. Hence, there is a propositional proof system g such that the canonical pair of g many-one reduces to the canonical pair of f, but f does not simulate g. The results of this section (Proposition 4, Theorem 5, and Corollary 9) present tight connections between disjoint NP-pairs and propositional proof systems. Nevertheless, relative to this oracle, the relationship is not as tight as one might hope for.

In light of Corollary 9, we should try to understand the degree structure of DisjNP. Glaßer, Selman, and Zhang [7] prove 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). Their result is an analogue of Ladner's result for NP [14]. The proof is based on Schöning's formulation [25] and uses techniques of Regan

[22, 23]. 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.

Observe that the premise of the following theorem is true as long as there exist P-inseparable disjoint NP-pairs.

Theorem 10. *Suppose there exist disjoint* NP*-pairs* (A, B) *and* (C, D) *such that* A, B, C, *and* D *are infinite,* $(A, B) \leq_T^{pp} (C, D)$, *and* $(C, D) \not\leq_T^{pp} (A, B)$. *Then there exist incomparable, strictly intermediate disjoint* NP*-pairs* (E, F) *and* (G, H) *between* (A, B) *and* (C, D) *such that* E, F, G, *and* H *are infinite. Precisely, the following properties hold:*

- $(A, B) \leq_m^{pp} (E, F) \leq_T^{pp} (C, D)$ *and* $(C, D) \not\leq_T^{pp} (E, F) \not\leq_T^{pp} (A, B)$*;*
- $(A, B) \leq_m^{pp} (G, H) \leq_T^{pp} (C, D)$ *and* $(C, D) \not\leq_T^{pp} (G, H) \not\leq_T^{pp} (A, B)$*;*
- $(E, F) \not\leq_T^{pp} (G, H)$ *and* $(G, H) \not\leq_T^{pp} (E, F)$*.*

Messner [16, 17] unconditionally proved the existence of propositional proof systems f and g such that f does not simulate g and g does not simulate f. Further he shows that the simulation order of propositional proof systems is dense. However, from this we cannot conclude a dense degree structure for disjoint NP-pairs. There exist infinite, strictly increasing chains of propositional proof systems (using simulation as the order relation $\leq$) such that all canonical pairs of these proofs systems belong to the same many-one degree of disjoint NP-pairs.

6 Uniform Enumerability

In this section we describe some recent results of Glaßer, Selman, and Sengupta [5] on reductions between disjoint NP-pairs. The main result is a list of equivalent statements to the assertion that there exists a many-one complete disjoint NP-pair, which, taken together, strongly suggests that the assertion does not hold.

We begin our exposition with the following definition of strongly many-one reductions, as defined by Köbler, Meßner, and Torán [12].

Definition 11 ([12]) (C, D) strongly many-one reduces to (A, B) in polynomial time, $(C, D) \leq_{sm}^{pp} (A, B)$, *if there is a polynomial-time computable function* f *such that* $f(C) \subseteq A$, $f(D) \subseteq B$, *and* $f(\overline{C \cup D}) \subseteq \overline{A \cup B}$.

Clearly, the added condition $f(\overline{C \cup D}) \subseteq \overline{A \cup B}$ states that instances violating the promise of (C, D) are mapped into instances that violate the promise of (A, B) (Figure 2). Equivalently, $f^{-1}(A) \subseteq C$ and $f^{-1}(B) \subseteq D$. Therefore, if $(C, D) \leq_{sm}^{pp} (A, B)$ via f, then $C \leq_m^p A$ via f, and $D \leq_m^p B$ via f.

Whereas Razborov proved that existence of an optimal proof system implies existence of a many-one complete disjoint NP-pair, Köbler, Meßner, and Torán proved with the same hypothesis existence of a complete disjoint NP-pair with respect to strongly many-one reductions. In particular, the result of Glaßer,

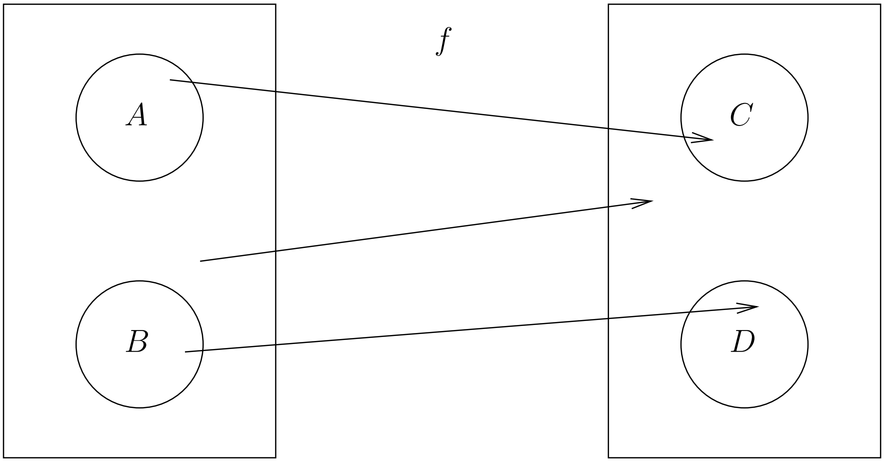

Fig. 2. A strong many-one reduction f from (A, B) to (C, D).

Selman, and Sengupta shows that these results of Razborov and Köbler, Meßner, and Torán are equivalent. That is, there exists a many-one complete disjoint NP-pair if and only if there exists a complete disjoint NP-pair with respect to strongly many-one reductions. Nevertheless, it is apparently true that the "stronger reduction" really is stronger. This is easy to see if we permit disjoint NP-pairs whose components are finite sets. However, for pairs whose components are infinite and coinfinite, strongly many-one reductions are identical to many-one reductions if and only if P = NP. We show this result now:

Theorem 12. *The following are equivalent:*

1. P $\neq$ NP.
2. *There are disjoint* NP*-pairs* (A, B) *and* (C, D) *such that* A, B, C, D, $\overline{A \cup B}$, *and* $\overline{C \cup D}$ *are infinite, and* $(A, B) \leq_m^{pp} (C, D)$ *but* $(A, B) \not\leq_{sm}^{pp} (C, D)$.

Proof. If P = NP, then given disjoint NP-pairs (A, B) and (C, D), A, B, C, and D are all in P. Given any string x, it can be determined whether $x \in A$, $x \in B$, or $x \in \overline{A \cup B}$, and x can be mapped appropriately to some fixed string in C, D, or $\overline{C \cup D}$. Therefore, $(A, B) \leq_{sm}^{pp} (C, D)$.

For the other direction, consider the clique-coloring pair (C_1, C_2) such that

$$C_1 = \{\langle G, k\rangle \mid G \text{ has a clique of size } k\}, \tag{1}$$

and

$$C_2 = \{\langle G, k\rangle \mid G \text{ has a coloring with } k-1 \text{ colors}\}. \tag{2}$$

This is a disjoint NP-pair, and is known to be P-separable [15, 20]. Let S be the separator that is in P. Note that $(C_1, C_2) \leq_m^{pp} (S, \overline{S})$ via the identity function. (Note that this reduction is also invertible.) Let

$$C = \{\langle G, 3\rangle \mid G \text{ is a cycle of odd length with at least 5 vertices}\}.$$

Let $S_1 = S - C$ and $S_2 = \overline{S} - C$. Both S_1 and S_2 are in P. Since any odd cycle with at least 5 vertices is not 2-colorable, and does not contain any clique of size 3, $C \cap C_1 = \emptyset$, and $C \cap C_2 = \emptyset$. Therefore, $(C_1, C_2) \leq_m^{pp} (S_1, S_2)$ via the identity function. Assume that $(C_1, C_2) \leq_{sm}^{pp} (S_1, S_2)$. Then $C_1 \leq_m^p S_1$, and $C_2 \leq_m^p S_2$. Hence C_1 and C_2 are in P. This is impossible, since NP $\neq$ P, and C_1 and C_2 are NP-complete. Thus, $(C_1, C_2) \not\leq_{sm}^{pp} (S_1, S_2)$. □

Next we mention smart reductions. Grollmann and Selman [9] defined smart reductions in order to analyze the conjecture of Even, Selman, and Yacobi [4] that we discussed earlier.

Definition 13 ([9]) *A* smart reduction *from (C, D) to (A, B) is a Turing reduction from (C, D) to (A, B) such that if the input belongs to $C \cup D$, then all queries belong to $A \cup B$.*

A disjoint pair $(A, B) \in$ DisjNP is *smart $\leq_T^{pp}$-complete* for DisjNP if for every (C, D) in DisjNP there is a smart reduction from (C, D) to (A, B). Note that if (A, B) is $\leq_m^{pp}$-complete for DisjNP, then (A, B) is smart $\leq_T^{pp}$-complete for DisjNP as well.

Let $\{N_i\}_i$ be an effective enumeration of nondeterministic, polynomial-time bounded Turing machines. Now we define the central concept of this section.

Definition 14 DisjNP *is* uniformly enumerable *if there is a total computable function $f : \Sigma^* \to \Sigma^* \times \Sigma^*$ such that*

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

The following theorem is a slight simplification of the main result of Glaßer, Selman, and Sengupta [5].

Theorem 15. *The following are equivalent.*

1. *There is a $\leq_m^{pp}$-complete disjoint* NP*-pair.*
2. *There is a $\leq_{sm}^{pp}$-complete disjoint* NP*-pair.*
3. *There is a smart $\leq_T^{pp}$-complete disjoint* NP*-pair.*
4. DisjNP *is uniformly enumerable.*

There is a long history of equating having complete sets with uniform enumerations. Hartmanis and Hemachandra [10], for example, proved this for the class UP, and it holds as well for NP $\cap$ co-NP and BPP. More recently, Sadowski [24] proved that there exists an optimal propositional proof system if and only if the class of all easy subsets of TAUT is uniformly enumerable.[4] It seems inconceivable that there would exist a total computable function that lists exactly the disjoint NP-pairs, and that is why we don't believe that many-one

[4] By Corollary 8, if DisjNP is not uniformly enumerable, then the class of all easy subsets of TAUT is also not uniformly enumerable.

complete disjoint NP-pairs exist, and hence, don't believe that optimal proof systems exist.

The most interesting direction of the proof is to show that if there exists a many-one complete disjoint NP-pair, then DisjNP is uniformly enumerable. We sketch this direction now:

Let (A, B) be a $\leq_m^{pp}$-complete disjoint pair. Let N_A and N_B be NP-machines that accept A and B, respectively. Let $\{f_i\}_i$ be an effective enumeration of polynomial-time computable functions. Input to the enumerator is a number encoding a triple $\langle i, j, k\rangle$. Output is a pair $\langle a, b\rangle$ to be described.

Given $\langle i, j, k\rangle$, we define nondeterministic Turing machines N_1' and N_2' as follows. On input x, N_1' computes $f_i(x) = q$ and then simulates both $N_A(q)$ and $N_B(q)$. At most one of these accepts. N_1' accepts x if $x \in L(N_j)$ and $q \in L(N_A)$. N_2' is defined similarly, except that N_2' accepts x if $x \in L(N_k)$ and $q \in L(N_B)$.

Let a and b be the indices of N_1' and N_2', respectively, and define $f(\langle i, j, k\rangle) = \langle a, b\rangle$. It is easy to see that $L(N_a)$ and $L(N_b)$ are disjoint. So for all i, j, and k, $(L(N_a), L(N_b)) \in$ DisjNP, where $f(\langle i, j, k\rangle) = \langle a, b\rangle$.

Now let (C, D) be a disjoint NP-pair. For some indices j and k, $C = L(N_j)$ and $D = L(N_k)$. Then $(C, D)\leq_m^{pp}(A, B)$ by f_i, for some i. Consider, $\langle a, b\rangle = f(\langle i, j, k\rangle)$. The remainder of the proof, which is easy, shows that $C = L(N_a)$ and $D = L(N_b)$.

References

1. S. Ben-David and A. Gringauze. On the existence of propositional proof systems and oracle-relativized propositional logic. Technical Report 5, Electronic Colloquium on Computational Complexity, 1998.
2. H. Buhrman, S. Fenner, L. Fortnow, and D. van Melkebeek. Optimal proof systems and sparse sets. In *Proceedings 17th Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *Lecture Notes in Computer Science*, pages 407–418. Springer Verlag, 2000.
3. S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.
4. S. Even, A. Selman, and J. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61:159–173, 1984.
5. C. Glaßer, A. Selman, and S. Sengupta. Reductions between disjoint NP-pairs. In *Proceedings 19th IEEE Conference on Computational Complexity*, pages 42–53. IEEE Computer Society, 2004.
6. C. Glaßer, A. Selman, S. Sengupta, and L. Zhang. Disjoint NP-pairs. *SIAM Journal on Computing*, 33(6):1369–1416, 2004.
7. C. Glaßer, A. Selman, and L. Zhang. Canonical disjoint NP-pairs of proposional proof systems. Technical Report 04-106, Electronic Colloquium on Computational Complexity, 2004.
8. O. Goldreich. On promise problems, in memory of Shimon Even (1935–2004). *This volume*, 2005.
9. J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

10. J. Hartmanis and L. A. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theoretical Computer Science*, 58:129–142, 1988.
11. S. Homer and A. Selman. Oracles for structural properties: The isomorphism problem and public-key cryptography. *Journal of Computer and System Sciences*, 44(2):287–301, 1992.
12. J. Köbler, J. Messner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184(1):71–92, 2003.
13. J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic*, 54:1063–1079, 1989.
14. R. Ladner. On the structure of polynomial-time reducibility. *Journal of the ACM*, 22:155–171, 1975.
15. L. Lovász. On the shannon capacity of graphs. *IEEE Transactions on Information Theory*, 25:1–7, 1979.
16. J. Messner. *On the Simulation Order of Proof Systems.* PhD thesis, Universität Ulm, 2000.
17. J. Messner. On the structure of the simulation order of proof systems. In *Proceedings 27rd Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 1450, pages 581–592. Springer-Verlag, 2002.
18. J. Meßner and J. Torán. Optimal proof systems for propositional logic and complete sets. In *Proceedings 15th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, pages 477–487. Springer Verlag, 1998.
19. P. Pudlák. On the length of proofs of finitistic consistency statements in first order theories. In J. B. Paris et al., editor, *Logic Colloquium '84*, pages 165–196. North-Holland Amsterdam, 1986.
20. P. Pudlák. On reducibility and symmetry of disjoint NP-pairs. In *Proceedings 26th International Symposium on Mathematical Foundations of Computer Science*, volume 2136 of *Lecture Notes in Computer Science*, pages 621–632. Springer-Verlag, Berlin, 2001.
21. A. Razborov. On provably disjoint NP-pairs. Technical Report TR94-006, Electronic Colloquium on Computational Complexity, 1994.
22. 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.
23. K. Regan. The topology of provability in complexity theory. *Journal of Computer and System Sciences*, 36:384–432, 1988.
24. Z. Sadowski. On an optimal propositional proof system and the structure of easy subsets of TAUT. *Theoretical Computer Science*, 288(1):181–193, 2002.
25. U. Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, 18:95–103, 1982.
26. A. Selman. Promise problems complete for complexity classes. *Information and Computation*, 78:87–98, 1988.
27. L. G. Valiant. Relative complexity of checking and evaluation. *Information Processing Letters*, 5:20–23, 1976.