Disjoint NP-Pairs and Propositional Proof Systems

C. Glaßer, A. Hughes, A. Selman, N. Wisiol

August 2017

Outline

Motivation from Separability

Definition

Given two disjoint sets A and B, a *separator* is a set S such that $A \subseteq S$ and $B \subseteq \overline{S}$. S separates A from B.

What is the computational difficulty of separators?

Descriptive Set Theory: Any two disjoint analytic sets are separable by a Borel set. [Luzin 1930]

Computability Theory: There exist disjoint c.e. sets A and B that are computably inseparable. [Kleene 1950]

The set of provable formulas of Peano Arithmetic and the set of refutable formulas are computably inseparable. [Smullyan 1958]

Disjoint NP-Pairs and Separability

Definition

A *disjoint NP-pair* is a pair (A, B) such that $A \cap B = \emptyset$ and $A, B \in \mathrm{NP}$. We write *NP-pair* for short.

Example: $(\{0x \mid x \in \mathrm{SAT}\}, \{1x \mid x \in \mathrm{SAT}\})$

Definition

An NP-pair is *P-separable*, if it has a separator that is in P.

Are certain NP-pairs P-separable?

Do P-inseparable NP-pairs exist?
(holds in computability theory [Kleene 1950])

Example: Clique-Coloring Pair

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

The sets are disjoint, since a clique of size k cannot be colored with $k-1$ colors.

CC_0 and CC_1 are NP-complete, hence (CC_0, CC_1) is an NP-pair.

Surprisingly, this pair is P-separable [Pudlak 2003], a result based on deep combinatorial ideas [Lovász 1979, Tardos 1988].

P-Inseparable NP-Pairs

Theorem (Grollmann, Selman 1988)

1. *If $P \neq UP$, then P-inseparable NP-pairs exist.*
2. *If $P \neq NP \cap coNP$, then P-inseparable NP-pairs exist.*
3. *If secure PKCS exist, then P-inseparable NP-pairs exist.*

Reducibilities for NP-Pairs

Definition (Grollmann, Selman 1988)

Let (A, B) and (C, D) be NP-pairs.

1. (A, B) is *many-one reducible* to (C, D), $(A, B) \leq_m^p (C, D)$, 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* to (C, D), $(A, B) \leq_T^p (C, D)$, 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.

Hence the oracle access is like this:
If query $q \in C \cup D$, then oracle tells us whether $q \in C$ or $q \in D$.
If query $q \notin C \cup D$, then oracle can answer arbitrarily.

Completeness and NP-Hardness

Definition (Completeness)

An NP-pair (A, B) is $\leq_m^p$*-complete*, if for every NP-pair (C, D) it holds that $(C, D) \leq_m^p (A, B)$. Same Definition for $\leq_T^p$.

Definition (NP-Hardness)

An NP-pair (A, B) is $\leq_m^p$*-hard for NP*, if every separator of (A, B) is $\leq_m^p$-hard for NP. Same Definition for $\leq_T^p$.

Do complete NP-pairs exist?
(holds in computability theory [Rogers 1967])

Do NP-hard NP-pairs exist?
(does not hold in computability theory [Shoenfield 1958])

Do hard/complete NP-pairs exist?

Conjecture by [Even, Selman, Yacobi 1984]:

- ESY-T: No NP-pair is $\leq_T^p$-hard for NP.
- ESY-m: No NP-pair is $\leq_m^p$-hard for NP.

The conjectures hold in computability theory [Shoenfield 1958].

Theorem

1. *ESY-T $\Rightarrow$ NP $\neq$ coNP and NP $\neq$ UP [ESY 84].*
2. *If ESY-T is false, then $\leq_T^p$-complete NP-pairs exist.*
3. *If ESY-m is false, then $\leq_m^p$-complete NP-pairs exist.*

Question for complete NP-pairs is related to proof systems ...

Propositional Proof Systems

Definition (Cook, Reckow 1979)

A *propositional proof system* is a polynomial-time-computable function f from Σ^* onto TAUT. We write *proof system* for short.

If $f(w) = \varphi$, then we say w is an f-proof for the formula φ.

Polynomially Bounded Proof Systems

Definition (Cook, Reckow 1979)

A propositional proof system f is *polynomially bounded* if there is a polynomial p such that for all φ and all f-proofs w of φ, it holds that $|w| \leq p(|\varphi|)$.

Theorem (Cook, Reckow 1979)

There exists a polynomially-bounded proof system if and only if $NP = coNP$.

Polynomially Bounded Proof Systems

Proof: $\exists$ poly-bounded proof system $f \Leftarrow$ NP $=$ coNP.

If NP $=$ coNP, then there is an NP-machine N accepting TAUT.

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

$f \in$ FP and $f : \Sigma^* \to$ TAUT.

$\varphi \in$ TAUT $\Rightarrow$ $N(\varphi)$ has accepting path w $\Rightarrow$ $f(\langle \varphi, w \rangle) = \varphi$.

Hence f is onto and therefore a proof system.

f is polynomially bounded, since $|w| \leq poly(|\varphi|)$. $\square$

Polynomially Bounded Proof Systems

Proof: $\exists$ poly-bounded proof system $f \Rightarrow$ NP $=$ coNP.

NP machine N on input φ:

- guess polynomial-length f-proof w
- accept if and only if $f(w) = \varphi$

N accepts TAUT.

Hence TAUT $\in$ NP and NP $=$ coNP. $\square$

Simulation and Optimal Proof Systems

Definition (Cook, Reckow 1979)

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 $|h(w)| \leq p(|w|)$.

Definition

A proof system that simulates every other proof system is called *optimal*.

Do optimal proof systems exist?

Do Optimal Proof Systems exist?

Theorem (Cook, Reckow 1979)

If NP $=$ coNP, then optimal proof systems exist.

Proof.

By NP $=$ coNP, there is a polynomially bounded proof system f.

We show that f simulates every proof system g.

$h(w) :=$ the lexicographically smallest v such that $f(v) = g(w)$.

Hence $f(h(w)) = g(w)$.

$|h(w)| \leq poly(|g(w)|)$, since f is a poly-bounded proof system.

$poly(|g(w)|) \leq poly(w)$, since g is polynomial-time computable.

So $|h(w)| \leq poly(|w|)$, hence f simulates g. $\square$

Is there evidence for the *non-existence* of optimal proof systems?

Canonical NP-pairs of Proof Systems

Razborov's Idea:
Each (optimal) proof system induces a (complete) NP-pair.

Definition (Razborov 1994)

Let f be a proof system. The *canonical pair* $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is defined by

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

Idea: SAT^* = satisfiable formulas (which have short proofs)
REF_f = unsatisfiable formulas that have short refutations

The restriction to *short* refutations is necessary,
since $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is not an NP-pair, unless NP=coNP.

Simulation implies Reducibility of Canonical Pairs

Theorem (Razborov 1994)

Let f and g be proof systems. If f is simulated by g, then

$$(SAT^*, REF_f) \leq_m^p (SAT^*, REF_g).$$

This shows:
If g is an optimal proof system, then at least all canonical pairs are $\leq_m^p$ reducible to $(\mathrm{SAT}^*, \mathrm{REF}_g)$.

This already suffices for $\leq_m^p$-completeness ...

NP-pairs and Canonical Pairs: same Degree Structure

Theorem (Glaßer, Selman, Zhang 2007)

For every NP-pair (A, B), there exists a proof system f such that $(A, B) \equiv_m^p (SAT^*, REF_f)$.

Corollary

If f is an optimal proof system, then (SAT^*, REF_f) is a $\leq_m^p$-complete NP-pair.

Proof.

For any NP-pair (A, B) , there is a proof system g such that $(A, B) \equiv_m^p (\mathrm{SAT}^*, \mathrm{REF}_g)$. Since g is simulated by f, we have $(A, B) \leq_m^p (\mathrm{SAT}^*, \mathrm{REF}_g) \leq_m^p (\mathrm{SAT}^*, \mathrm{REF}_f)$. $\square$

NP-hard canonical pairs

Corollary

The following statements are equivalent.

1. $NP = coNP$
2. *ESY-m is false, i.e.,* $\exists$ *NP-pairs that are* $\leq_m^p$*-hard for NP*
3. $\exists$ *canonical pairs of proof systems that are* $\leq_m^p$*-hard for NP*

Single-Valued NP-Functions

Definition

A *single-valued NP-function* is a partial function f such that there exists a nondeterministic, polynomial-time Turing-transducer T such that for all x:

- If $f(x)$ is not defined, then T on x has no accepting paths.
- If $f(x) = y$, then T on x has accepting paths and each of them outputs y.

NPSV denotes the class of all single-valued NP-functions.

Reducibility and Completeness

Definition (Reducibility)

Let $f, g \in$ NPSV. We say that f *many-one reduces to* g, $f \leq_m^p g$, if there is a polynomial-time computable function h such that $g(h(x)) = f(x)$.

Definition (Completeness)

A function $g \in$ NPSV is $\leq_m^p$*-complete*, if $f \leq_m^p g$ for all $f \in$ NPSV.

Are there complete functions in NPSV?

Uniform Enumerations

Hartmanis and Hemachandra 1988:
UP has a complete set iff UP is uniformly enumerable.

Let $\{N_i\}_{i\geq 0}$ be a standard effective enumeration of nondet. poly-time Turing machines. Let $\{T_i\}_{i\geq 0}$ be a standard effective enumeration of nondet. poly-time Turing machine transducers.

Definition

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

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

Continued

For each $i \geq 0$, let $F(T_i)$ denote the partial, possibly multivalued, function computed by transducer T_i.

Definition

NPSV is *uniformly enumerable* if there is a total computable function $f : \Sigma^* \to \Sigma^*$ such that:

1. $\forall i \in \mathit{range}(f) \;\; [F(T_i) \in \mathrm{NPSV}]$
2. $\forall g \in \mathrm{NPSV} \;\; \exists i \;\; [i \in \mathit{range}(f) \text{ and } g = F(T_i)]$

Complete NP-Pairs and Complete NPSV-Functions

Theorem (Glasser, Selman, Sengupta 2005)

The following statements are equivalent.

1. *There exist $\leq_m^p$-complete NP-pairs.*
2. *NPSV has $\leq_m^p$-complete functions.*

The statements in the theorem are equivalent to:

- The class of NP-pairs is uniformly enumerable.
- NPSV is uniformly enumerable.

NP $\cap$ SPARSE and optimal proof systems

Definition

A set of words S is *sparse* if there exists a polynomial p such that S contains at most $p(n)$ words of length $\leq n$.

Do optimal proof systems exist?

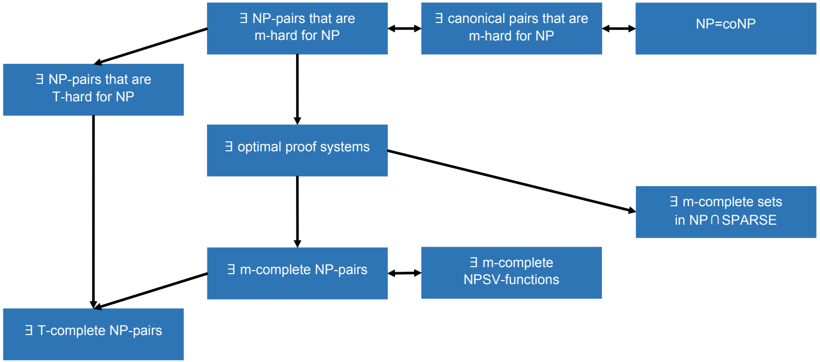
Theorem (Köbler, Meßner, Torán 2003)

If optimal proof systems exist, then NP $\cap$ SPARSE has $\leq_m^p$-complete sets.

We tend to believe that NP $\cap$ SPARSE has no complete sets.

Interpret the theorem as evidence that optimal proof systems do not exist.

Summary


∃ NP-pairs that are m-hard for NP
∃ canonical pairs that are m-hard for NP
NP=coNP
∃ NP-pairs that are T-hard for NP
∃ optimal proof systems
∃ m-complete sets in NP ∩ SPARSE
∃ m-complete NP-pairs
∃ m-complete NPSV-functions
∃ T-complete NP-pairs

More Connections

The talk skipped known connections to:

- Promise Problems
- $\leq_T^p$-complete NPSV-functions
- NP-hard NPSV-functions

Thank you!