Disjoint NP-Pairs and Propositional Proof Systems

Nils Wisiol

University at Buffalo

July 31st, 2014

Outline

When Decision Problems Fail

Example

Given a Hamiltonian Graph, determine whether or not it is complete.

- Try to translate this into a decision problem.
- $L = \{G \mid G \text{ is Hamiltonian and } G \text{ is complete}\}$
- L is NP-complete.
- Let's use *promise problems* instead of decision problems

When Decision Problems Fail

Example

Given a Hamiltonian Graph, determine whether or not it is complete.

- Try to translate this into a decision problem.
- $L = \{G \mid G \text{ is Hamiltonian and } G \text{ is complete}\}$
- L is NP-complete.
- Let's use *promise problems* instead of decision problems

When Decision Problems Fail

Example

Given a Hamiltonian Graph, determine whether or not it is complete.

- Try to translate this into a decision problem.
- $L = \{G \mid G \text{ is Hamiltonian and } G \text{ is complete}\}$
- L is NP-complete.
- Let's use *promise problems* instead of decision problems

When Decision Problems Fail

Example

Given a Hamiltonian Graph, determine whether or not it is complete.

- Try to translate this into a decision problem.
- $L = \{G \mid G \text{ is Hamiltonian and } G \text{ is complete}\}$
- L is NP-complete.
- Let's use *promise problems* instead of decision problems

Promise Problems

Decision Problems vs. Promise Problems

- Decision problems divide Σ^* into 2 sets: Yes- and No-Instances
- Promise problems divide Σ^* into 3 sets:
  - Yes instances
  - No instances
  - disallowed strings

Promise Problems

Decision Problems vs. Promise Problems

- Decision problems divide Σ^* into 2 sets: Yes- and No-Instances
- Promise problems divide Σ^* into 3 sets:
  - Yes instances
  - No instances
  - disallowed strings

Promise Problems

Decision Problems vs. Promise Problems

- Decision problems divide Σ^* into 2 sets: Yes- and No-Instances
- Promise problems divide Σ^* into 3 sets:
  - Yes instances
  - No instances
  - disallowed strings

Promise Problems

Decision Problems vs. Promise Problems

- Decision problems divide Σ^* into 2 sets: Yes- and No-Instances
- Promise problems divide Σ^* into 3 sets:
  - Yes instances
  - No instances
  - disallowed strings

Promise Problems

Decision Problems vs. Promise Problems

- Decision problems divide Σ^* into 2 sets: Yes- and No-Instances
- Promise problems divide Σ^* into 3 sets:
  - Yes instances
  - No instances
  - disallowed strings

Promise Problems

Definition

- A tuple (A, B) is called a promise problem, if $A \cap B = \emptyset$. The set of disallowed strings is $\overline{A \cup B}$.
- A set $S \supseteq A$ and $S \subseteq \overline{B}$ is called a separator of (A, B). The set of all separators is denoted by $\mathrm{Sep}(A, B)$.

Example

Hamiltonian graphs that are complete

- Yes: Graphs that are Hamiltonian and complete
- No: Graphs that are Hamiltonian and not complete
- disallowed strings: Graphs that are not Hamiltonian

Promise Problems

Definition

- A tuple (A, B) is called a promise problem, if $A \cap B = \emptyset$. The set of disallowed strings is $\overline{A \cup B}$.
- A set $S \supseteq A$ and $S \subseteq \overline{B}$ is called a separator of (A, B). The set of all separators is denoted by $\mathrm{Sep}(A, B)$.

Example

Hamiltonian graphs that are complete

- Yes: Graphs that are Hamiltonian and complete
- No: Graphs that are Hamiltonian and not complete
- disallowed strings: Graphs that are not Hamiltonian

Promise Problems

Definition

- A tuple (A, B) is called a promise problem, if $A \cap B = \emptyset$. The set of disallowed strings is $\overline{A \cup B}$.
- A set $S \supseteq A$ and $S \subseteq \overline{B}$ is called a separator of (A, B). The set of all separators is denoted by $\mathrm{Sep}(A, B)$.

Example

Hamiltonian graphs that are complete

- Yes: Graphs that are Hamiltonian and complete
- No: Graphs that are Hamiltonian and not complete
- disallowed strings: Graphs that are not Hamiltonian

Promise Problems

Definition

- A tuple (A, B) is called a promise problem, if $A \cap B = \emptyset$. The set of disallowed strings is $\overline{A \cup B}$.
- A set $S \supseteq A$ and $S \subseteq \overline{B}$ is called a separator of (A, B). The set of all separators is denoted by $\mathrm{Sep}(A, B)$.

Example

Hamiltonian graphs that are complete

- Yes: Graphs that are Hamiltonian and complete
- No: Graphs that are Hamiltonian and not complete
- disallowed strings: Graphs that are not Hamiltonian

Promise Problems

Definition

- A tuple (A, B) is called a promise problem, if $A \cap B = \emptyset$. The set of disallowed strings is $\overline{A \cup B}$.
- A set $S \supseteq A$ and $S \subseteq \overline{B}$ is called a separator of (A, B). The set of all separators is denoted by $\mathrm{Sep}(A, B)$.

Example

Hamiltonian graphs that are complete

- Yes: Graphs that are Hamiltonian and complete
- No: Graphs that are Hamiltonian and not complete
- disallowed strings: Graphs that are not Hamiltonian

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Reductions of Promise Problems

Definitions

- We define a promise problem (A, B) to be *r-reducible* to (C, D), if for every separator $T \in \mathrm{Sep}(C, D)$, there exists a separator $S \in \mathrm{Sep}(A, B)$ such that $S \leq_r T$.
- We denote this by $(A, B) \leq_r^{pp} (C, D)$.
- $\leq_m^{pp}$, $\leq_T^{pp}$, ...

Example

- Decision problems are promise problems. For $A \in \mathrm{NP}$: $(A, \overline{A})$
- $\mathrm{Sep}(A, \overline{A}) = \{A\}$
- $A \leq_m^p \mathrm{SAT}$
- $(A, \overline{A}) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP*-Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP-*Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP-*Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP*-Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP-*Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP-*Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

Disjoint NP-Pairs and Completeness

Definitions

- A promise problem (A, B) with $A, B \in \mathrm{NP}$ is a *Disjoint* NP*-Pair*. The set DisjNP is the set of all Disjoint NP-pairs.
- A disjoint NP-pair (A, B) is $\leq_m^{pp}$-complete, if for every $(C, D) \in \mathrm{DisjNP}$ we have $(C, D) \leq_m^{pp} (A, B)$.

Example

- Assuming $\mathrm{NP} = \mathrm{coNP}$, $(\mathrm{SAT}, \overline{\mathrm{SAT}}) \in \mathrm{DisjNP}$
- SAT is the only separator of $(\mathrm{SAT}, \overline{\mathrm{SAT}})$: $\mathrm{Sep}(\mathrm{SAT}, \overline{\mathrm{SAT}}) = \{\mathrm{SAT}\}$
- $A \in \mathrm{NP}$ is separator of $(A, B) \in \mathrm{DisjNP}$
- $(A, B) \leq_m^{pp} (\mathrm{SAT}, \overline{\mathrm{SAT}})$
- $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is $\leq_m^{pp}$-complete

ESY-Conjecture

- In 1986, Even, Selman and Yacobi introduced a conjecture about disjoint NP-pairs. [2]

ESY-Conjecture [2]

For every pair of disjoint sets in NP, there is a separator that is not T-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}} \exists_{S\in \mathrm{Sep}(A,B)} \exists_{L\in \mathrm{NP}} \quad L \not\leq_T^p S.$$

ESY-Conjecture refinements

For every pair of disjoint sets in NP, there is a separator that is not r-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}} \exists_{S\in \mathrm{Sep}(A,B)} \exists_{L\in \mathrm{NP}} \quad L \not\leq_r S.$$

- ESY-T $\implies$ ESY-tt $\implies$ ESY-m

ESY-Conjecture

- In 1986, Even, Selman and Yacobi introduced a conjecture about disjoint NP-pairs. [2]

ESY-Conjecture [2]

For every pair of disjoint sets in NP, there is a separator that is not T-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathrm{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq_T^p S.$$

ESY-Conjecture refinements

For every pair of disjoint sets in NP, there is a separator that is not r-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathrm{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq_r S.$$

- ESY-T $\implies$ ESY-tt $\implies$ ESY-m

ESY-Conjecture

- In 1986, Even, Selman and Yacobi introduced a conjecture about disjoint NP-pairs. [2]

ESY-Conjecture [2]

For every pair of disjoint sets in NP, there is a separator that is not T-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathsf{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq^p_T S.$$

ESY-Conjecture refinements

For every pair of disjoint sets in NP, there is a separator that is not r-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathsf{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq_r S.$$

- ESY-T $\implies$ ESY-tt $\implies$ ESY-m

ESY-Conjecture

- In 1986, Even, Selman and Yacobi introduced a conjecture about disjoint NP-pairs. [2]

ESY-Conjecture [2]

For every pair of disjoint sets in NP, there is a separator that is not T-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathsf{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq_T^p S.$$

ESY-Conjecture refinements

For every pair of disjoint sets in NP, there is a separator that is not r-hard for NP,

$$\forall_{(A,B)\in \mathrm{DisjNP}}\ \exists_{S\in \mathsf{Sep}(A,B)}\ \exists_{L\in \mathrm{NP}} \quad L \not\leq_r S.$$

- ESY-T $\implies$ ESY-tt $\implies$ ESY-m

Complete NP-pairs

Theorem

If ESY-r does not hold, then there exists a r-complete disjoint NP *pair.* [4]

Proof sketch.

- By negation of ESY-r: (A, B) only has r-hard separators
- $C \in \mathrm{NP}$ is a separator of $(C, D) \in \mathrm{DisjNP}$
- C r-reduces to any separator of (A, B)
- $(C, D) \leq_r^{pp} (A, B)$

$\square$

Complete NP-pairs

Theorem

If ESY-r does not hold, then there exists a r-complete disjoint NP *pair. [4]*

Proof sketch.

- By negation of ESY-r: (A, B) only has r-hard separators
- $C \in \mathrm{NP}$ is a separator of $(C, D) \in \mathrm{DisjNP}$
- C r-reduces to any separator of (A, B)
- $(C, D) \leq_r^{pp} (A, B)$

$\square$

Complete NP-pairs

Theorem

If ESY-r does not hold, then there exists a r-complete disjoint NP *pair. [4]*

Proof sketch.

- By negation of ESY-r: (A, B) only has r-hard separators
- $C \in \mathrm{NP}$ is a separator of $(C, D) \in \mathrm{DisjNP}$
- C r-reduces to any separator of (A, B)
- $(C, D) \leq_r^{pp} (A, B)$

$\square$

Complete NP-pairs

Theorem

If ESY-r does not hold, then there exists a r-complete disjoint NP pair. [4]

Proof sketch.

- By negation of ESY-r: (A, B) only has r-hard separators
- $C \in \mathrm{NP}$ is a separator of $(C, D) \in \mathrm{DisjNP}$
- C r-reduces to any separator of (A, B)
- $(C, D) \leq_r^{pp} (A, B)$

$\square$

Complete NP-pairs

Theorem

If ESY-r does not hold, then there exists a r-complete disjoint NP *pair. [4]*

Proof sketch.

- By negation of ESY-r: (A, B) only has r-hard separators
- $C \in \mathrm{NP}$ is a separator of $(C, D) \in \mathrm{DisjNP}$
- C r-reduces to any separator of (A, B)
- $(C, D) \leq_r^{pp} (A, B)$

$\square$

ESY-m Conjecture

Theorem

$\mathrm{NP} = \mathrm{coNP}$ *if and only if the ESY-m conjecture does not hold. [4]*

Proof sketch.

- "$\Leftarrow$": Let (A, B) such that all separators are many-one-hard
- $\overline{B}$ is separator of (A, B), therefore $\mathrm{SAT} \leq_m^p \overline{B}$
- $\overline{\mathrm{SAT}} \leq_m^p B$, therefore $\overline{\mathrm{SAT}} \in \mathrm{NP}$ and $\mathrm{NP} = \mathrm{coNP}$
- "$\Rightarrow$": Let $\mathrm{NP} = \mathrm{coNP}$, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a witness

$\square$

ESY-m Conjecture

Theorem

$\mathrm{NP} = \mathrm{coNP}$ *if and only if the ESY-m conjecture does not hold. [4]*

Proof sketch.

- "$\Leftarrow$": Let (A, B) such that all separators are many-one-hard
- $\overline{B}$ is separator of (A, B), therefore $\mathrm{SAT} \leq_m^p \overline{B}$
- $\overline{\mathrm{SAT}} \leq_m^p B$, therefore $\overline{\mathrm{SAT}} \in \mathrm{NP}$ and $\mathrm{NP} = \mathrm{coNP}$
- "$\Rightarrow$": Let $\mathrm{NP} = \mathrm{coNP}$, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a witness

$\square$

ESY-m Conjecture

Theorem

$\mathrm{NP} = \mathrm{coNP}$ *if and only if the ESY-m conjecture does not hold. [4]*

Proof sketch.

- "$\Leftarrow$": Let (A, B) such that all separators are many-one-hard
- $\overline{B}$ is separator of (A, B), therefore $\mathrm{SAT} \leq_m^p \overline{B}$
- $\overline{\mathrm{SAT}} \leq_m^p B$, therefore $\overline{\mathrm{SAT}} \in \mathrm{NP}$ and $\mathrm{NP} = \mathrm{coNP}$
- "$\Rightarrow$": Let $\mathrm{NP} = \mathrm{coNP}$, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a witness

$\square$

ESY-m Conjecture

Theorem

$\mathrm{NP} = \mathrm{coNP}$ *if and only if the ESY-m conjecture does not hold. [4]*

Proof sketch.

- "$\Leftarrow$": Let (A, B) such that all separators are many-one-hard
- $\overline{B}$ is separator of (A, B), therefore $\mathrm{SAT} \leq_m^p \overline{B}$
- $\overline{\mathrm{SAT}} \leq_m^p B$, therefore $\overline{\mathrm{SAT}} \in \mathrm{NP}$ and $\mathrm{NP} = \mathrm{coNP}$
- "$\Rightarrow$": Let $\mathrm{NP} = \mathrm{coNP}$, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a witness

$\square$

ESY-m Conjecture

Theorem

$\mathrm{NP} = \mathrm{coNP}$ *if and only if the ESY-m conjecture does not hold. [4]*

Proof sketch.

- "$\Leftarrow$": Let (A, B) such that all separators are many-one-hard
- $\overline{B}$ is separator of (A, B), therefore $\mathrm{SAT} \leq_m^p \overline{B}$
- $\overline{\mathrm{SAT}} \leq_m^p B$, therefore $\overline{\mathrm{SAT}} \in \mathrm{NP}$ and $\mathrm{NP} = \mathrm{coNP}$
- "$\Rightarrow$": Let $\mathrm{NP} = \mathrm{coNP}$, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a witness

$\square$

ESY-T and ESY-tt Conjectures

Theorem

If the ESY-T conjecture is true, there are no public-key crypto systems with NP*-hard cracking problems. [7]*

Theorem

If $\mathrm{NP} = \mathrm{UP}$*, then ESY-tt does not hold, that is, there exists a disjoint* NP*-pair such that all separators are truth-table-hard for* NP*. [8]*

Summary

$$
\begin{tikzcd}
\exists \text{ hard PKCS} \arrow[d] & \mathrm{NP} = \mathrm{UP} \arrow[d] & \mathrm{NP} = \mathrm{coNP} \arrow[d, leftrightarrow] \\
\begin{array}{c}\text{ESY-}T\\ \text{does not hold}\end{array} \arrow[d] & \begin{array}{c}\text{ESY-}tt\\ \text{does not hold}\end{array} \arrow[l] \arrow[d] & \begin{array}{c}\text{ESY-}m\\ \text{does not hold}\end{array} \arrow[l] \arrow[d] \\
\begin{array}{c}\exists\ T\text{-complete}\\ \mathrm{NP}\text{ pair}\end{array} & \begin{array}{c}\exists\ tt\text{-complete}\\ \mathrm{NP}\text{ pair}\end{array} \arrow[l] & \begin{array}{c}\exists\ m\text{-complete}\\ \mathrm{NP}\text{ pair}\end{array} \arrow[l]
\end{tikzcd}
$$

Figure: Summary of shown ESY conjecture results

Definition

Definition

- A polynomial-time computable function f that is onto the set of tautologies is called a *propositional proof system* or *proof system*.
- For any w, we say w is a *f-proof for x* if $f(w) = x$.
- If there is a polynomial p, such that for all x, and all f-proofs w of x, we have $|w| \leq p(|x|)$, then f is *polynomially-bounded*.

Definition

Definition

- A polynomial-time computable function f that is onto the set of tautologies is called a *propositional proof system* or *proof system*.
- For any w, we say w is a *f-proof for x* if $f(w) = x$.
- If there is a polynomial p, such that for all x, and all f-proofs w of x, we have $|w| \leq p(|x|)$, then f is *polynomially-bounded*.

Definition

Definition

- A polynomial-time computable function f that is onto the set of tautologies is called a *propositional proof system* or *proof system*.
- For any w, we say w is a *f-proof for x* if $f(w) = x$.
- If there is a polynomial p, such that for all x, and all f-proofs w of x, we have $|w| \leq p(|x|)$, then f is *polynomially-bounded*.

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

Proof.

- "$\Leftarrow$": If $\mathrm{NP} = \mathrm{coNP}$, then $\mathrm{TAUT} \in \mathrm{NP}$. Let M be non-det machine accepting TAUT.
- Given the computation path, a tautology can be computed in poly time
- "$\Rightarrow$": If f is poly-bounded, then a NP-machine can guess proofs and thus $\mathrm{TAUT} \in \mathrm{NP}$.

$\square$

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

Proof.

- "$\Leftarrow$": If $\mathrm{NP} = \mathrm{coNP}$, then $\mathrm{TAUT} \in \mathrm{NP}$. Let M be non-det machine accepting TAUT.
- Given the computation path, a tautology can be computed in poly time
- "$\Rightarrow$": If f is poly-bounded, then a NP-machine can guess proofs and thus $\mathrm{TAUT} \in \mathrm{NP}$.

$\square$

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

Proof.

- "$\Leftarrow$": If $\mathrm{NP} = \mathrm{coNP}$, then $\mathrm{TAUT} \in \mathrm{NP}$. Let M be non-det machine accepting TAUT.
- Given the computation path, a tautology can be computed in poly time
- "$\Rightarrow$": If f is poly-bounded, then a NP-machine can guess proofs and thus $\mathrm{TAUT} \in \mathrm{NP}$.

$\square$

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

Proof.

- "$\Leftarrow$": If $\mathrm{NP} = \mathrm{coNP}$, then $\mathrm{TAUT} \in \mathrm{NP}$. Let M be non-det machine accepting TAUT.
- Given the computation path, a tautology can be computed in poly time
- "$\Rightarrow$": If f is poly-bounded, then a NP-machine can guess proofs and thus $\mathrm{TAUT} \in \mathrm{NP}$.

$\square$

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

- No polynomially bounded proof system is known
- Cook and Reckhow tried to answer NP vs coNP by studying and comparing proof systems
- Notion of comparison for proof systems needed

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

- No polynomially bounded proof system is known
- Cook and Reckhow tried to answer NP vs coNP by studying and comparing proof systems
- Notion of comparison for proof systems needed

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

- No polynomially bounded proof system is known
- Cook and Reckhow tried to answer NP vs coNP by studying and comparing proof systems
- Notion of comparison for proof systems needed

Polynomially Bounded and $\mathrm{NP} = \mathrm{coNP}$

Theorem

There is a polynomially-bounded propositional proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

- No polynomially bounded proof system is known
- Cook and Reckhow tried to answer NP vs coNP by studying and comparing proof systems
- Notion of comparison for proof systems needed

Simulation of Proof Systems

Definitions

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

Simulation of Proof Systems

Definitions

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

Simulation of Proof Systems

Definitions

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

Sufficient Condition for Optimal Proof Systems

Known sufficient conditions for the existence of Optimal Proof Systems are:

- $\mathrm{NP} = \mathrm{coNP}$, because every poly-bounded proof system is optimal
- $\mathrm{NE} = \mathrm{coNE}$ by Krajíček and Pudlák [10]
- $\mathrm{NEE} = \mathrm{coNEE}$ by Meßner and Torán [11]
- Analog conditions for p-Optimality are $\mathrm{P} = \mathrm{NP}$, $\mathrm{E} = \mathrm{NE}$, $\mathrm{EE} = \mathrm{NEE}$.

Sufficient Condition for Optimal Proof Systems

Known sufficient conditions for the existence of Optimal Proof Systems are:

- $\mathrm{NP} = \mathrm{coNP}$, because every poly-bounded proof system is optimal
- $\mathrm{NE} = \mathrm{coNE}$ by Krajíček and Pudlák [10]
- $\mathrm{NEE} = \mathrm{coNEE}$ by Meßner and Torán [11]
- Analog conditions for p-Optimality are $\mathrm{P} = \mathrm{NP}$, $\mathrm{E} = \mathrm{NE}$, $\mathrm{EE} = \mathrm{NEE}$.

Sufficient Condition for Optimal Proof Systems

Known sufficient conditions for the existence of Optimal Proof Systems are:

- $\mathrm{NP} = \mathrm{coNP}$, because every poly-bounded proof system is optimal
- $\mathrm{NE} = \mathrm{coNE}$ by Krajíček and Pudlák [10]
- $\mathrm{NEE} = \mathrm{coNEE}$ by Meßner and Torán [11]
- Analog conditions for p-Optimality are $\mathrm{P} = \mathrm{NP}$, $\mathrm{E} = \mathrm{NE}$, $\mathrm{EE} = \mathrm{NEE}$.

Sufficient Condition for Optimal Proof Systems

Known sufficient conditions for the existence of Optimal Proof Systems are:

- $\mathrm{NP} = \mathrm{coNP}$, because every poly-bounded proof system is optimal
- $\mathrm{NE} = \mathrm{coNE}$ by Krajíček and Pudlák [10]
- $\mathrm{NEE} = \mathrm{coNEE}$ by Meßner and Torán [11]
- Analog conditions for p-Optimality are $\mathrm{P} = \mathrm{NP}$, $\mathrm{E} = \mathrm{NE}$, $\mathrm{EE} = \mathrm{NEE}$.

Summary of Sufficient Conditions

$$
\begin{tikzcd}
\mathrm{P} = \mathrm{NP} \arrow[r] \arrow[d] & \mathrm{E} = \mathrm{NE} \arrow[r] \arrow[d] & \mathrm{EE} = \mathrm{NEE} \arrow[r] \arrow[d] & \exists \text{ a p-optimal proof system} \arrow[d] \\
\mathrm{NP} = \mathrm{coNP} \arrow[r] \arrow[d, leftrightarrow] & \mathrm{NE} = \mathrm{coNE} \arrow[r] & \mathrm{NEE} = \mathrm{coNEE} \arrow[r] & \exists \text{ an optimal proof system} \\
\exists \text{ poly-bounded proof system} & & &
\end{tikzcd}
$$

Figure: The symmetric structure of sufficient conditions for optimal and p-optimal propositional proof systems.

Consequences of Optimal Proof Systems

Known consequences of the existence of optimal proof systems are:

- If there is an optimal proof system, then complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$ exist.
- If there is a p-optimal proof system, then UP has a complete set under non-uniform many-one reducibility. [9]

Consequences of Optimal Proof Systems

Known consequences of the existence of optimal proof systems are:

- If there is an optimal proof system, then complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$ exist.
- If there is a p-optimal proof system, then UP has a complete set under non-uniform many-one reducibility. [9]

Summary of Consequences

$$
\begin{tikzcd}
\text{$\exists$ a p-optimal proof system} \arrow[r] \arrow[d] & \text{$\exists$ m-complete set in } \mathrm{UP} \\
\text{$\exists$ an optimal proof system} \arrow[r] & \text{$\exists$ m-complete set in } \mathrm{NP} \cap \mathrm{SPARSE}
\end{tikzcd}
$$

Figure: Summary of consequences of the existence of optimal and p-optimal proof systems

Canonical Disjoint NP-Pairs

Definition

- We define the *canonical pair* $(\mathrm{SAT}^*, \mathrm{REF}_f)$ for every proof system f, where
- $\mathrm{SAT}^* = \{(\varphi, 1^m) \mid \varphi \in \mathrm{SAT} \text{ and } m \geq 0\}$,
- $\mathrm{REF}_f = \{(\varphi, 1^m) \mid \neg\varphi \in \mathrm{TAUT} \text{ and } \exists y, |y| \leq m \text{ such that } f(y) = \neg\varphi\}$.

Canonical Disjoint NP-Pairs

Definition

- We define the *canonical pair* $(\mathrm{SAT}^*, \mathrm{REF}_f)$ for every proof system f, where
- $\mathrm{SAT}^* = \{(\varphi, 1^m) \mid \varphi \in \mathrm{SAT} \text{ and } m \geq 0\}$,
- $\mathrm{REF}_f = \{(\varphi, 1^m) \mid \neg\varphi \in \mathrm{TAUT} \text{ and } \exists y, |y| \leq m$ such that $f(y) = \neg\varphi\}$.

Canonical Disjoint NP-Pairs

Definition

- We define the *canonical pair* $(\mathrm{SAT}^*, \mathrm{REF}_f)$ for every proof system f, where
- $\mathrm{SAT}^* = \{(\varphi, 1^m) \mid \varphi \in \mathrm{SAT} \text{ and } m \geq 0\}$,
- $\mathrm{REF}_f = \{(\varphi, 1^m) \mid \neg\varphi \in \mathrm{TAUT} \text{ and } \exists y, |y| \leq m \text{ such that } f(y) = \neg\varphi\}$.

Razborov's Result

Theorem

For two proof systems f and g, if f simulates g, then $(\mathrm{SAT}^*, \mathrm{REF}_g) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. [3]

Theorem

For any $(A, B) \in \mathrm{DisjNP}$, there exists a proof system f such that $(A, B) \equiv_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. [3]

Corollary

For an optimal proof system f, the pair $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is complete for DisjNP. [13]

Razborov's Result

Theorem

For two proof systems f and g, if f simulates g, then $(\mathrm{SAT}^*, \mathrm{REF}_g) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. *[3]*

Theorem

For any $(A, B) \in \mathrm{DisjNP}$, *there exists a proof system f such that* $(A, B) \equiv_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. *[3]*

Corollary

For an optimal proof system f, the pair $(\mathrm{SAT}^*, \mathrm{REF}_f)$ *is complete for* DisjNP. *[13]*

Razborov's Result

Theorem

For two proof systems f and g, if f simulates g, then $(\mathrm{SAT}^*, \mathrm{REF}_g) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. [3]

Theorem

For any $(A, B) \in \mathrm{DisjNP}$, there exists a proof system f such that $(A, B) \equiv_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$. [3]

Corollary

For an optimal proof system f, the pair $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is complete for DisjNP. [13]

Converse?

Theorem

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

Theorem

The set of all disjoint pairs and the $\leq_m^{pp}$*-relation has the same degree structure as the set of all proof systems and the simulation-relation.* *[5]*

Converse?

Theorem

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

Theorem

The set of all disjoint pairs and the $\leq_m^{pp}$*-relation has the same degree structure as the set of all proof systems and the simulation-relation.* *[5]*

Implications from Disjoint NP-Pairs and Proof Systems

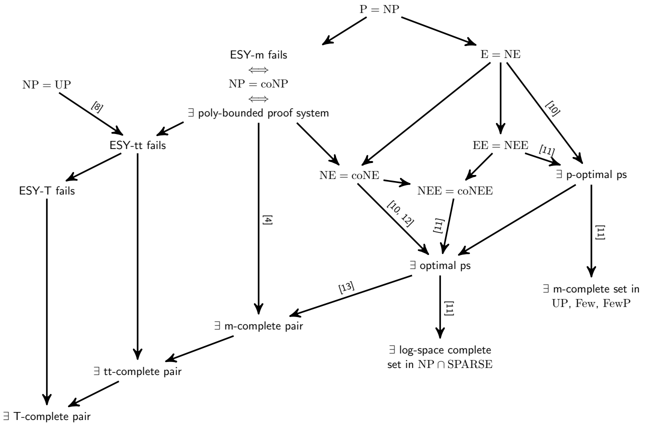

Figure: Known Implications for proof systems, disjoint pairs and the ESY conjecture

Existence of Optimal Proof Systems

Theorem

- *If there is an optimal proof system, then complete sets for* $\mathrm{NP} \cap \mathrm{SPARSE}$ *exist.*
- *If* $\mathrm{NE} = \mathrm{coNE}$*, then there is an optimal proof system.*

- Buhrman, Fenner, Fortnow and van Melkebeek [1]: Oracle such that no complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$
- Glaßer, Selman, Sengupta and Zhang [4]: Oracle such that $\mathrm{NE} = \mathrm{coNE}$

Existence of Optimal Proof Systems

Theorem

- *If there is an optimal proof system, then complete sets for* $\mathrm{NP} \cap \mathrm{SPARSE}$ *exist.*
- *If* $\mathrm{NE} = \mathrm{coNE}$, *then there is an optimal proof system.*

- Buhrman, Fenner, Fortnow and van Melkebeek [1]: Oracle such that no complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$
- Glaßer, Selman, Sengupta and Zhang [4]: Oracle such that $\mathrm{NE} = \mathrm{coNE}$

Existence of Optimal Proof Systems

Theorem

- *If there is an optimal proof system, then complete sets for* $\mathrm{NP} \cap \mathrm{SPARSE}$ *exist.*
- *If* $\mathrm{NE} = \mathrm{coNE}$*, then there is an optimal proof system.*

- Buhrman, Fenner, Fortnow and van Melkebeek [1]: Oracle such that no complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$
- Glaßer, Selman, Sengupta and Zhang [4]: Oracle such that $\mathrm{NE} = \mathrm{coNE}$

Existence of Optimal Proof Systems

Theorem

- *If there is an optimal proof system, then complete sets for* $\mathrm{NP} \cap \mathrm{SPARSE}$ *exist.*
- *If* $\mathrm{NE} = \mathrm{coNE}$*, then there is an optimal proof system.*

- Buhrman, Fenner, Fortnow and van Melkebeek [1]: Oracle such that no complete sets for $\mathrm{NP} \cap \mathrm{SPARSE}$
- Glaßer, Selman, Sengupta and Zhang [4]: Oracle such that $\mathrm{NE} = \mathrm{coNE}$

Converse of Razborov's Theorem

Theorem

- *There is an oracle such that the converse of Razborov's Theorem holds. [4]*
- *There is an oracle such that the converse of Razborov's Theorem does not hold. [4]*

Converse of Razborov's Theorem

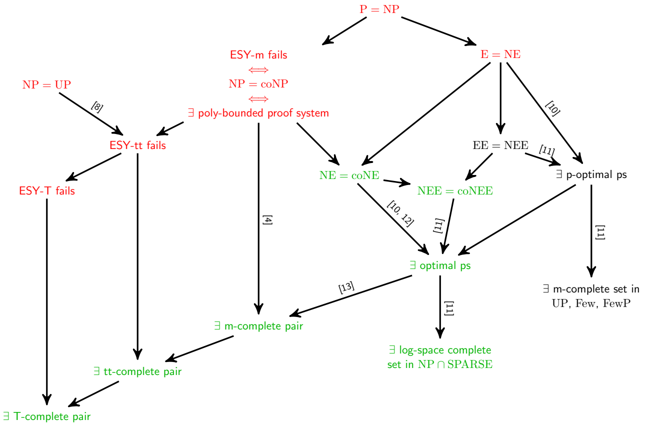

Figure: Overview of consequences oracle O_1 [4] has on the assertions about disjoint NP-pairs and proof systems.

Converse of Razborov's Theorem

Theorem

- *There is an oracle such that the converse of Razborov's Theorem holds. [4]*
- *There is an oracle such that the converse of Razborov's Theorem does not hold. [4]*

Converse of Razborov's Theorem

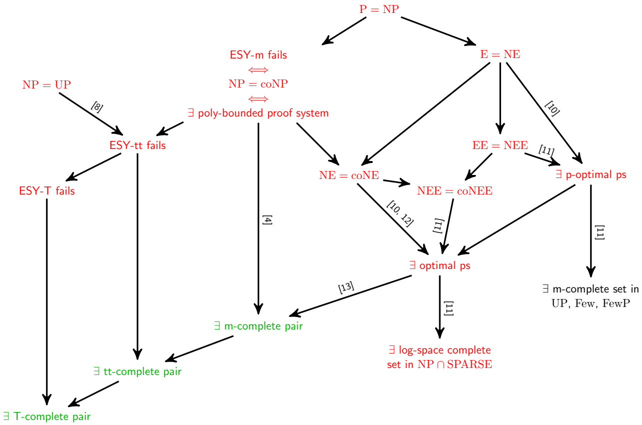

Figure: Overview of consequences oracle O_2 [4] has on the assertions about disjoint NP-pairs and proof systems.

Converse of Razborov's Theorem

Theorem

- *There is an oracle such that the converse of Razborov's Theorem holds. [4]*
- *There is an oracle such that the converse of Razborov's Theorem does not hold. [4]*

Converse of Razborov's Theorem

Theorem

- *There is an oracle such that the converse of Razborov's Theorem holds.* [4]
- *There is an oracle such that the converse of Razborov's Theorem does not hold.* [4]

Corollary

The converse of Razborov's Theorem can not be proved or disproved by relativizable techniques.

Separation of ESY-refinements

Theorem

There is an oracle such that $\mathrm{NP} = \mathrm{UP}$ *and* $\mathrm{NP} \neq \mathrm{coNP}$ *[6].*

- Relative to this oracle, we have ESY-tt does not hold; however ESY-m is true.

Separation of ESY-refinements

Theorem

There is an oracle such that $\mathrm{NP} = \mathrm{UP}$ *and* $\mathrm{NP} \neq \mathrm{coNP}$ *[6].*

- Relative to this oracle, we have ESY-tt does not hold; however ESY-m is true.

Implications from Disjoint NP-Pairs and Proof Systems

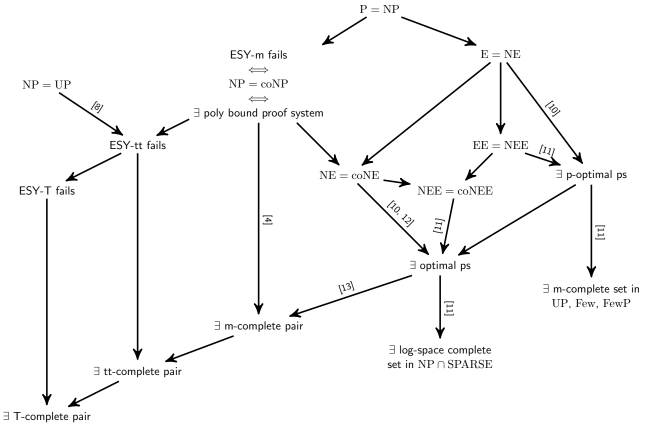

Figure: Known Implications for proof systems, disjoint pairs and the ESY conjecture

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.

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.

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

C. Glaßer, A. L. Selman, S. Sengupta, and L. Zhang.
Disjoint NP-pairs.
Electronic Colloquium on Computational Complexity (ECCC), 10(011), 2003.

C. Glaßer, A. L. Selman, and L. Zhang.
Survey of disjoint NP-pairs and relations to propositional proof systems.
In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman, editors, *Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 241–253. Springer, 2006.

Christian Glaßer and Gerd Wechsung.
Relativizing function classes.
J. UCS, 9(1):34–50, 2003.

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

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, *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 473–484. Springer, 2012.

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

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.

J. Meßner and J. Torán.
Optimal proof systems for propositional logic and complete sets.
Electronic Colloquium on Computational Complexity (ECCC), 4(26), 1997.

P. PudlÃ¡k.

On the length of proofs of finitistic consistency statements in first order theories.
Logic Colloquium, 84, 1986.

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