The "Berman-Hartmanis" Conjecture, NP-isomorphism, padding

Zampetakis Konstantinos

MPLA

November 20, 2014

Overview

Polynomial-time isomorphism

Definition(Polynomial-time isomorphism)

We say that tow languages $K, L \subset \Sigma^*$ are polynomialy isomorphic if there is a function $h : \Sigma^* \to \Sigma^*$, such that:

1. h is one-to-one and onto
2. For each $x \in \Sigma^*$, $x \in K \Leftrightarrow h(x) \in L$
3. Both, h and h^{-1}, are polynomial-time computable.

Polynomial-time isomorphism vs Polynomial-time reduction

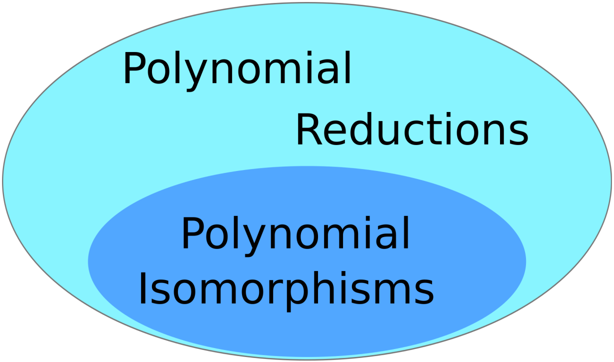

Figure : A polynomial-time isomorphism is also a polynomial-time reduction

Polynomial-time isomorphism vs Polynomial-time reduction

- But which polynomial-time reductions are polynomial-time isomorphisms?

Polynomial-time isomorphism vs Polynomial-time reduction

- But which polynomial-time reductions are polynomial-time isomorphisms?
- **Most of them are not!**

Polynomial-time isomorphism vs Polynomial-time reduction

- But which polynomial-time reductions are polynomial-time isomorphisms?
- **Most of them are not!**
- But the reduction between CLIQUE to INDEPENTENT SET is.

From Polynomial-time reduction to Polynomial-time isomorphism ?

Question: Can we turn, systematicly, a reduction to an isomorphism?

From Polynomial-time reduction to Polynomial-time isomorphism ?

Question: Can we turn, systematicly, a reduction to an isomorphism?

Answer: Padding!

Padding

Definition(Padding)

Let $L \subset \Sigma^*$ be a language. We say tha a function $pad : (\Sigma^*)^2 \rightarrow \Sigma^*$ is a padding function for L if it holds that:

1. Is computable in logarithmic space (or polynomial time)
2. For any $x, y \in \Sigma^*$, $pad(x, y) \in L \Leftrightarrow x \in L$
3. For any $x, y \in \Sigma^*$, $|pad(x, y)| > |x| + |y|$.
4. There exist a logarithmic space (or polynomial time) algorithm which, given $pad(x, y)$ recovers y

Padding Example 1 SAT

Input formula:

$$x = (x_1 \vee \neg x_3 \vee x_2) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_2)$$

Word y:

$$y = 0101$$

Padding Example 1 SAT

Padding Result:

$$pad(x, y) =$$

$$(x_1 \vee \neg x_3 \vee x_2) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_2) \wedge (x_5) \wedge (x_5) \wedge (x_5) \wedge (\neg x_6) \wedge (x_7) \wedge (\neg x_8) \wedge (x_9)$$

Padding Example 2 CLIQUE

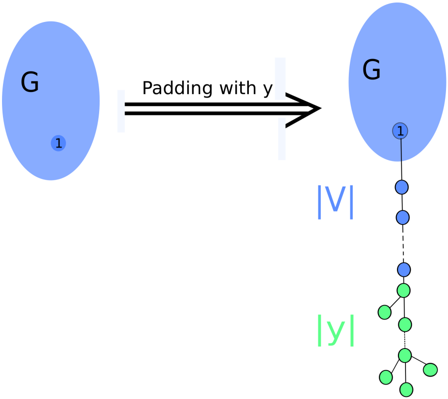

Figure : Padding $x = (G, K)$ with y

Padding

Lemma

Supose that R is a reduction from the language K to language L, and that *pad* is a padding function for L. Then the function mapping $x \in \Sigma^*$ to $pad(R(x), x)$ is length-increasing, one-to-one reduction. Also, there is a logarithmic space (polynomial time) algorithm R^{-1} which, given $pad(R(x), x)$ recovers x.

Padding

Proof.

The fact that $pad(R(x), x)$ is a reduction and is length-increasing follows easily from the properties 1,2 and 3 of padding functions,respectively. The last property gives us that $pad(R(x), x)$ recovers x in logarithmic space (polynomial time). $\square$

Padding

Theorem

Suppose that $L, K \subset \Sigma^*$, *and* $R : K \to L$, $S : L \to K$ *are reductions. Suppose further that these reductions are one-to-one, length-increasing, and logarithmic space (polynomial time) invertible. Then* K *and* L *are polynomially isomorphic.*

Padding

Proof.

Let the S-chain of x is defined as:

$$(x, S^{-1}(X), R^{-1}(S^{-1}(X)), S^{-1}(R^{-1}(S^{-1}(X))), ...),$$

It's finite, since S^{-1}, R^{-1} are length-decreasing. We define$h : \Sigma^* \to \Sigma^*$ as

- $h(x) = S^{-1}(x)$, if the S-chain stops on S
- $h(x) = R(x)$, if the S-chain stops on R

Then if $h(x) = h(y)$ we have
$h(x) = S^{-l}(X) = R(y) = h(y), y = R^{-1}(S^{-1}(X))$, contradiction.
For onto, similarly we define :

- $h^{-1}(x) = S(x)$, if the R-chain stops on S
- $h^{-1}(x) = R^{-1}(x)$, if the R-chain stops on R

The other properties are trivial. $\square$

Berman-Hartmanis Conjecture(1977)

Berman-Hartmanis Conjecture (Isomorphism Conjecture)

All NP -complete languages are pairwise polynomial-time isomorphic (P - isomorphic) to each other.

Berman-Hartmanis Conjecture(1977)

Berman-Hartmanis Conjecture (Isomorphism Conjecture)

All NP -complete languages are pairwise polynomial-time isomorphic (P - isomorphic) to each other.

In their paper, Berman and Hartmanis showed that all the then-known NP -complete problems were pairwise P -isomorphic, by finding a padding for each one of them.

Berman-Hartmanis Conjecture(1977)

Berman-Hartmanis Conjecture (Isomorphism Conjecture)

All NP -complete languages are pairwise polynomial-time isomorphic (P - isomorphic) to each other.

In their paper, Berman and Hartmanis showed that all the then-known NP -complete problems were pairwise P -isomorphic, by finding a padding for each one of them.

Remark

If Berman-Hartmanis Conjecture holds $\Rightarrow P \neq NP$

Sparse Languages

Definition

A set A is called *sparse*, if there is exist a polynomial p such that

$$|\{x \in A : |x| \leq n, \text{where } n \in \mathbb{N}\}| \leq p(n)$$

Sparse Languages

Definition

A set A is called *sparse*, if there is exist a polynomial p such that

$$|\{x \in A : |x| \leq n, \text{where } n \in \mathbb{N}\}| \leq p(n)$$

Remarks:

1. The SAT is not sparse, since there are constants $\epsilon > 0$ and $\delta > 0$ such that at least $\epsilon 2^{\delta n}$ strings of length at most n belong to SAT.

Sparse Languages

Definition

A set A is called *sparse*, if there is exist a polynomial p such that

$$|\{x \in A : |x| \leq n, \text{where } n \in \mathbb{N}\}| \leq p(n)$$

Remarks:

1. The SAT is not sparse, since there are constants $\epsilon > 0$ and $\delta > 0$ such that at least $\epsilon 2^{\delta n}$ strings of length at most n belong to SAT.
2. No sparse language can be P-isomorphic to SAT.

Mahaney's Theorem

We can show something stronger than that:

Mahaney's Theorem

We can show something stronger than that:

Theorem (Mahaney's Theorem)

If $P \neq NP$, then no NP-complete language can be sparse.