On the isomorphism-complete problems and polynomial time isomorphism

By Gh. Grigoras

Introduction

One of the important open problems in computer science today is the computational complexity of deciding when two graphs are isomorphic. No polynomial time algorithm is known, nor is the problem known to be NP-complete. Many restrictions and generalizations of the problem have been the focus of much research during last years and many of these problems have turned out to be polynomial time equivalent to graph isomorphism ([3], [4], [6], [7], [9], [10]).

In this paper, starting from the results of Berman and Hartmanis paper on p-isomorphism [2] we give some analogous necesary and sufficient conditions for a language to be isomorphic under polynomial time mappings to graph isomorphism problem. Next we give the proof of the existence of p-isomorphism for some problems which are known to be polynomial time equivalent to graph isomorphism. We conjecture that all problems polynomial time equivalent to graph isomorphism problem are p-isomorphic.

Preliminaries

In our paper we suppose the reader is familiar with the terminology of complexity theory. In this section, we make precise some of the objects; for more details see [1], [5], [6], [8].

A language $A \subseteq \Sigma^*$ is said to be *reducible* to a language $B \subseteq \Gamma^*$ if there exists some function $f\colon \Sigma^* \to \Gamma^*$ such that $f(x) \in B$ iff $x \in A$, $\forall x \in \Sigma^*$. A is said to be *reducible* to B in *polynomial time (p-reducible)* if the function f is computed by a deterministic Turing machine M which runs in polynomial time.

A language L_0 is said to be $\mathscr{C}$-hard for some class of languages $\mathscr{C}$ if for every L in $\mathscr{C}$, L is p-reducible to L_0.

A language L_0 is complete for $\mathscr{C}$ if it is in $\mathscr{C}$ and is $\mathscr{C}$-hard.

By P (NP) we denote the class of languages accepted by deterministic (nondeterministic) Turing machines which run in polynomial time.

A language $A \subseteq \Sigma^*$ is said to be *p-isomorphic* to a language $B \subseteq \Gamma^*$ ([2]) iff there exists a bijection $f\colon \Sigma^* \to \Gamma^*$ such that f is a p-reduction of A to B and f^{-1} is a p-reduction of B to A.

Let $A \subseteq \Sigma^*$; the function $Z_A\colon \Sigma^* \to \Sigma^*$ is a *padding function* for the set A if it satisfies the following two properties

1. $Z_A(x) \in A$ iff $x \in A$, $\forall x \in \Sigma^*$;
2. Z_A is invertible (i.e. one-one).

The following theorem due to Berman and Hartmanis [2], is useful in the proof of the fact that the problems computationally equivalent with the graph isomorphism are p-isomorphic.

Theorem 1. Let $A \subseteq \Sigma^*$ and $B \subseteq \Gamma^*$ be two languages such that A is p-reducible to B and B is p-reducible to A (in other words, A and B are polynomially equivalent); furthermore let the language A have a padding function Z_A satisfying

1_A. Z_A has polynomial time complexity;

2_A. $(\forall y)[|Z_A(y)| > |y|^2+1]$;

and polynomial-time computable functions $S_A(-,-)$ and $D_A(-)$ satisfying

3_A. $(\forall x, y)[S_A(x, y) \in A$ iff $x \in A]$;

4_A. $(\forall x, y)[D_A(S_A(x, y)) = y]$.

Then B is p-isomorphic to A iff B has the polynomial-time computable functions S_B and D_B satisfying 3_B and 4_B.

Berman and Hartmanis show that all NP-complete languages known in the literature are p-isomorphic. If all NP-complete problems are p-isomorphic, then $P \neq \mathrm{NP}$.

Now, let us consider the Graph Isomorphism Problem: are given two graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ isomorphic? In other words, is there any bijection h from V_1 to V_2 for which (v, w) is an edge in E_1 if and only if $(h(v), h(w))$ is an edge in E_2?

The complexity of Graph Isomorphism Problem is unknown yet and this problem has been the focus of much research in recent years ([3], [4], [7], [9], [10]). Many of the restrictions and generalizations of the problem turn out to be polynomial time equivalent to graph isomorphism [3].

Caracterization of problems p-isomorphic to graph isomorphism

In this section we apply the theorem of Berman—Hartmanis to the Graph Isomorphism Problem.

First, let us consider an enconding scheme in which a graph $G=(V, E)$ can be described as a word over an alphabet Σ (see [6] p. 10). Let us denote by $\bar{G}$ the enconding of G, and let $\#$ be a symbol not belonging to Σ. Then, the graph isomorphism problem can be formulated as the problem of recognizing the language

$$\mathrm{GI} = \{x \mid x \in (\Sigma \cup \{\#\})^*,\ x = \bar{G}_1 \# \bar{G}_2,\ G_1 \text{ is isomorphic to } G_2\}.$$

Let us note that we consider $\&\in\Sigma$ and by the word $\bar{G}_1 \& \bar{G}_2$, where $\bar{G}_1$ and $\bar{G}_2$ are the encondings of two graphs G_1 and G_2, we mean the enconding of graph with components G_1 and G_2.

Lemma 1. The language GI has a function denoted by $S_{\mathrm{GI}}(-,-)$ with the properties

i) S_{GI} has polynomial time complexity;

ii) $(\forall x, y)[S_{\mathrm{GI}}(x, y)\in \mathrm{GI}$ iff $x\in \mathrm{GI}]$.

Proof. Let us consider the language $\Delta\subseteq\{0, 1\}^*$ defined by $y\in\Delta$ iff

1) $\exists n\in N,\quad y = y_1y_2\dots y_{n^2},\quad y_i\in\{0, 1\},\quad i = 1, 2, \dots, n^2;$

2) $\forall i,j\quad 1\leqq i, j\leqq n,\quad y_{(j-1)n+i} = y_{(i-1)m+j}.$

Note that the language Δ is decidable in polynomial time.
Now we define the function

$$S_{\mathrm{GI}}\colon (\Sigma\cup\{\#\})^*x\Delta \to (\Sigma\cup\{\#, \square, 0, 1\})^*,$$

where $\square\notin\Sigma$ is a new symbol by

$$S_{\mathrm{GI}}(x, y) = \begin{cases}\bar{G}_1\&\bar{G}\#\bar{G}\&\bar{G}_2 & \text{if}\quad x = \bar{G}_1\#\bar{G}_2,\\ x^2\square y & \text{in other cases.}\end{cases}$$

The graph G which appear in the definition of S_{GI} is constructed as follows. Let $G_1=(V_1, E_1)$, $G_2=(V_2, E_2)$, $V_1=\{v_1, v_2, \dots, v_n\}$, $V_2=\{w_1, w_2, \dots, w_m\}$ and $x=\bar{G}_1\#\bar{G}_2$. Then $G=(Z, E)$ where $Z=\{Z_1, Z_2, \dots, Z_l\}$, $l=\sqrt{|y|}$, and the edge $(Z_r, Z_s)\in E$ iff $y_{(s-1)l+r}=1$. In other words G is the graph with the adjacency matrix rows $y_{kl+1}y_{kl+2}\dots y_{(k+1)l}$, $0\leqq k\leqq l-1$.[1]

It is clear that G_1 and G_2 are isomorphic if and only if so are the graphs with encondings $\bar{G}_1$ & $\bar{G}$ and $\bar{G}$ & $\bar{G}_2$. Hence $S_{\mathrm{GI}}(x, y)\in\mathrm{GI}$ iff $x\in\mathrm{GI}$.

Furthermore it is easy to see that S_{GI} is computable in polynomial time which completes the proof of the lemma.

Lemma 2. The language GI has a function denoted by $D_{\mathrm{GI}}(-)$ with the properties

i) D_{GI} has polynomial time complexity;

ii) $(\forall x, y)$, $D_{\mathrm{GI}}(S_{\mathrm{GI}}(x, y)) = y$;

where S_{GI} is the function defined in Lemma 1.

Proof. Let us concider the function

$$D_{\mathrm{GI}}\colon(\Sigma\cup\{\#, \square, 0, 1\})^* \to \Delta\cup(\Sigma\cup\{\#, \square, 0, 1\})^*,$$

[1] For short, we say G has the adjacency matrix y.

where Δ is the language from Lemma 1, defined by

$$D_{\mathrm{GI}} = \begin{cases} y \text{ if } u=u_1 \,\&\, u_2 \# u_2 \,\&\, u_3,\ u_2 \text{ (in which \& does not occur) is the enconding} \\ \quad \text{of a graph the rows of adjacency matrix of which are } y=y_1 \dots y_r, \\ z \quad \text{if} \quad u=u_1 \Box z, \\ u \quad \text{in other cases.} \end{cases}$$

From the definition of D_{GI} it follows that, given $u \in (\Sigma \cup \{\#, \Box, 0, 1,\})^*$ the computation of $D_{\mathrm{GI}}(u)$ can be made in polynomial time depending on $|u|$.

Now, let $x \in (\Sigma \cup \{\#\})^*$ and $y \in \Delta$. If $x=\bar{G}_1 \# \bar{G}_2$ then $S_{\mathrm{GI}}(x, y)=\bar{G}_1 \,\&\, \bar{G} \# \# \bar{G} \,\&\, \bar{G}_2$ and $D_{\mathrm{GI}}(G_1 \,\&\, \bar{G} \# \bar{G} \,\&\, \bar{G}_2)=y$ (the adjacency matrix of G). If x is not of the form $\bar{G}_1 \# \bar{G}_2$, then $S_{\mathrm{GI}}(x, y)=x^2 \Box y$ and $D_{\mathrm{GI}}(x^2 \Box y)=y$.

Hence, $\forall x, y,\ D_{\mathrm{GI}}(S_{\mathrm{GI}}(x, y))=y$ and the lemma is proved.

Lemma 3. The language GI has a padding function Z_{GI} such that

i) Z_{GI} has polynomial time complexity;

ii) $\forall x \in (\Sigma \cup \{\#\})^*,\quad |Z_{\mathrm{GI}}(x)| > |x|^2+1$.

Proof: Let us define the function

$$Z_{\mathrm{GI}}\colon (\Sigma \cup \{\#\})^* \to (\Sigma \cup \{\#, \Box\})^*,$$

by

$$Z_{\mathrm{GI}}(x) = S_{\mathrm{GI}}(x, 1^{\varphi^2(|x|)}) \quad \text{for all} \quad x \in (\Sigma \cup \{\#\})^*$$

where $\varphi\colon N \to N$ is a function depending on enconding scheme. We will show that there exists this function such that condition ii) of lemma is satisfied. Let us note that Z_{GI} is a padding function. Indeed, from Lemma 1, we have $S_{\mathrm{GI}}(x, y) \in \mathrm{GI}$ iff $x \in \mathrm{GI}$, hence $Z_{\mathrm{GI}}(x) \in \mathrm{GI}$ iff $x \in \mathrm{GI}$, $\forall x \in (\Sigma \cup \{\#\})^*$. From the definition of S_{GI}, it follows that Z_{GI} is an injective function, hence Z_{GI} is invertible. It is clear that Z_{GI} has polynomial time complexity. It remains to prove that for all x,

$$|Z_{\mathrm{GI}}(x)| > |x|^2+1.$$

If $x \neq \bar{G}_1 \# \bar{G}_2$, then

$$|Z_{\mathrm{GI}}(x)| = |S_{\mathrm{GI}}(x, 1^{\varphi^2(|x|)})| = |x^2 \Box 1^{\varphi^2(|x|)}| > |x|^2+1.$$

If $x=\bar{G}_1 \# \bar{G}_2$, then

$$|Z_{\mathrm{GI}}(x)| = |S_{\mathrm{GI}}(\bar{G}_1 \# G_2, 1^{\varphi^2(|x|)})| = |\bar{G}_1 \,\&\, \bar{G} \# \bar{G} \,\&\, \bar{G}_2| =$$
$$= |\bar{G}_1 \# \bar{G}_2| + 2(|\bar{G}|+1) = |x| + 2|\bar{G}| + 2.$$

Of course, $|\bar{G}|$ depends on $|x|$ because $y=1^{\varphi^2(|x|)}$. Let $e(n)$ be the length of $\bar{G}$ where G has n vertex, and let $e(n)$ be of order $O(n^k)$, $k \geqq 1$. Then $|\bar{G}|=e(n)= =e(\varphi(|x|))=O(\varphi(|x|)^k)$. If we consider $\varphi(n)=O(n^{2/k})$ then

$$|\bar{G}| = O((O(|x|^{2/k}))^k) = O(|x|^2),$$

hence

$$|Z_{\mathrm{GI}}(x)| = |x| + 2O(|x|^2) + 2.$$

It follows that we can find a function $\varphi(n)$ such that

$$|Z_{\mathrm{GI}}(x)| > |x|^2+1.$$

Theorem 2. Let A be a language polynomial time equivalent to GI (A is p-reducible to GI and GI is p-reducible to A). Then A is p-isomorphic to GI if and only if A has two polynomial time computable functions $S_A(-,-)$ and $D_A(-)$ such that

1) $(\forall x, y)[S_A(x, y) \in A \text{ iff } x \in A]$;

2) $(\forall x, y)[D_A(S_A(x, y)) = y]$.

Proof. From Lemmas 1—3 it follows that GI satisfies the conditions of Berman—Harmanis theorem.

Problems p-isomorphic to graph isomorphism

Booth and Colbourn [3] present a comprehensive list of problems which are known to be polynomial time equivalent to graph isomorphism. Such problems are called isomorphism complete.

Now, we consider some of these problems and prove that they are p-isomorphic to graph isomorphism.

1. *Directed Graph Isomorphism.* Given two directed graphs, are they isomorphic? Miller [10] shows this problem is isomorphism complete.

2. *Oriented Graph Isomorphism.* An oriented graph [3] is a digraph in which the presence of the arc (x, y) precludes the presence of (y, x). Oriented graph isomorphism problem is isomorphism complete [3].

3. *Bipartite Graph Isomorphism.* Given two bipartite graphs, are they isomorphic? This problem is isomorphism complete [3].

4. *Semiautomata Isomorphism.* A semiautomaton is a 3-tuple $A=(I, S, f)$, where I and S are finite sets of inputs and states respectively and $f\colon S\times I \to S$ is the transition function. Two semiautomata $A_1=(I_1, S_1, f_1)$ and $A_2=(I_2, S_2, f_2)$ are isomorphic if there exist two bijections $g\colon I_1 \to I_2$ and $h\colon S_1 \to S_2$ such that the following diagram commute:

$$\begin{tikzcd}
S_1 \times I_1 \arrow[r, "f_1"] \arrow[d, "(h,g)"] & S_1 \arrow[d, "h"] \\
S_2 \times I_2 \arrow[r, "f_2"] & S_2
\end{tikzcd}$$

Semiautomata isomorphism problem is isomorphism complete ([3], [7]).

Lemma 4. Directed graph isomorphism is p-isomorphic to graph isomorphism.

Proof. Let us define the function S_{DGI} and D_{DGI} satisfying Theorem 2, where

$$\mathrm{DGI} = \{x \mid x=\bar{G}_1 \# \bar{G}_2,\ \bar{G}_1 \text{ and } \bar{G}_2 \text{ are encondings of two directed isomorphic graphs}\} \subseteq (\Sigma \cup \{\#\})^*.$$

Let us consider $\Delta=\{y \mid y\in\{0, 1\}^*,\ |y|=n^2,\ n\in N\}$. Then, for all $x\in(\Sigma\cup\{\#\})^*$, $y\in\Delta$

$$S_{\mathrm{DGI}}(x, y) = \begin{cases} \bar{G}_1 \& \bar{G} \# \bar{G} \& \bar{G}_2 & \text{if } x = \bar{G}_1 \# \bar{G}_2; \\ x \square y & \text{otherwise,} \end{cases}$$

where $\bar{G}$ is the enconding of the directed graph which has the adjacent matrix y.

Like in Lemma 2 we define D_{DGI}, by $\forall u \in (\Sigma \cup \{\Box, \#, 0, 1\})^*$

$$D_{\mathrm{DGI}}^{(u)} = \begin{cases} y & \text{if } u = u_1 \& u_2 \# u_2 \& u_3,\ u_2 \text{ is the enconding of the directed graph, the adjacent matrix of which is } y; \\ z & \text{if } u = u_1 \Box z; \\ u & \text{in other cases.} \end{cases}$$

It is obvious that S_{DGI} and D_{DGI} are polynomial time computable and

1) $(\forall x, y)\quad S_{\mathrm{DGI}}(x, y) \in \mathrm{DGI}$ iff $x \in \mathrm{DGI}$;

2) $(\forall x, y)\quad D_{\mathrm{DGI}}(S_{\mathrm{DGI}}(x, y)) = y$.

Lemma 5. Oriented graph isomorphism is p-isomorphic to graph isomorphism.

Proof. Like in Lemma 4, we construct the functions S_{OGI} and D_{OGI} satisfying Theorem 2. In this case we take

$$\Delta = \{y/y \in \{0, 1\}^*,\ |y| = n^2,\ y_{(j-1)n+i} = 1 \Rightarrow y_{(i-1)n+j} = 0\}.$$

It is clear that Δ can be recognized in polynomial time and the graph with adjacent matrix $y \in \Delta$ is an oriented graph.

The functions are defined in the manner of Lemma 4.

Lemma 6. Bipartite graph isomorphism is p-isomorphic to graph isomorphism.

Proof. Let us consider the language $\Delta \subseteq \{0, 1\}^*$ defined by

$$\Delta = \{y | y = (0^k 1^k)^k (1^k 0^k)^k,\ k \in N\}.$$

It is easy to see that Δ can be recognized in polynomial time and, the graphs with $2k$ vertices and adjacent matrix $y \in \Delta$ are bipartite graphs. Like in Lemma 4, there exist the functions S_{BGI} and D_{BGI} satisfying Theorem 2.

REMARK. The bipartite graph constructed in Lemma 6 is also a regular graph: all the vertices have the degree k. Hence the regular graph isomorphism (which is isomorphism complete [3], [10]) is p-isomorphic to graph isomorphism.

Lemma 7. Semiautomata isomorphism is p-isomorphic to graph isomorphism.

Proof. Let $A = (I, S, f)$ be a semiautomaton, $I = \{i_1, i_2, \ldots, i_n\}$, $S = \{s_1, \ldots, s_m\}$ and $f(s_k, i_j) = f_{kj} \in S$ $1 \leqq k \leqq m$, $1 \leqq j \leqq n$. We consider an enconding scheme in which A is represented by the word

$$\bar{A} = i[1] \ldots i[n] * s[1] \ldots s[m] / f_{11} f_{21} \ldots f_{m1} / f_{12} f_{22} \ldots f_{m2} / \ldots / f_{1n} f_{2n} \ldots f_{mn},$$

where

$$f_{ij} = s[l] \quad \text{if} \quad f(s_k, i_j) = s_l.$$

Now, if A_1 and A_2 are two semiautomata with the same input sets and disjoint sets of states, the semiautomaton encoded by $\bar{A}_1$ & $\bar{A}_2$ is the semiautomaton with the same inputs, the set of states is the union of states of A_1 and A_2 and the transition function is defined in natural way.

Set $\mathrm{SI}=\{\bar{A}_1 \# \bar{A}_2 | A_1$ is isomorphic to $A_2\} \subset \Gamma^*$. We define $S_{\mathrm{SI}}\colon \Gamma^* \times \Delta \to (\Gamma \cup \{\Box, 0, 1\})^*$ and $D_{\mathrm{SI}}\colon (\Gamma \cup \{\Box\})^* \to \Delta \cup (\Gamma \cup \{\Box, 0, 1\})^*$, where $\Delta \subseteqq \{0, 1\}^*$, in the following way:

Let $x=\bar{A}_1 \# \bar{A}_2 \in \mathrm{SI}$ and $y \in \Delta, y=y_1 y_2 \ldots y_l$. Consider the semiautomata $A_1'=(I_1, \Sigma, g_1)$ and $A_2'=(I_2, \Sigma, g_2)$ where I_1 and I_2 are the input sets of A_1 and A_2 respectively, $\Sigma=\{\sigma_1, \ldots, \sigma_l, \bar{\sigma}\}$ such that $\Sigma \cap S_i=\Phi$, $i=1,2$ and g_j $(j=1,2)$ are defined by $1 \leqq k \leqq l-1$,

$$g_j(\sigma_k, i_j) = \begin{cases} \sigma_{k+1} & y_k = 1, \\ \bar{\sigma} & y_k = 0, \end{cases}$$

$$g_j(\sigma_l, i_j) = \begin{cases} \sigma_1 & y_l = 1, \\ \bar{\sigma} & y_l = 0, \end{cases}$$

$$g_j(\bar{\sigma}, y) = \bar{\sigma},$$

for all $i_j \in I_j$ $(j=1,2)$. Then we define

$$S_{\mathrm{AI}}(x, y) = \begin{cases} \bar{A}_1 \& \bar{A}_1' \# \bar{A}_2' \& \bar{A}_2 & \text{if } x = \bar{A}_1 \# \bar{A}_2, \\ x \Box y & \text{otherwise} \end{cases}$$

and

$$D_{\mathrm{SI}}(u) = \begin{cases} y \text{ if } u = \bar{A}_1 \& \bar{A}_2 \# \bar{A}_3 \& \bar{A}_4 \text{ and } A_2, A_3 \text{ have} \\ \quad \text{the same states and transition functions,} \\ z \text{ if } u=x \Box z, \\ u \text{ in other cases,} \end{cases}$$

where $y \in \{0, 1\}^*$ is determined in the following way:

If $A_2=(I_2, \Sigma, f_2)$, $A_3=(I_3, \Sigma, f_3)$, $\Sigma=\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ then $y=y_1, \ldots, y_{n-1}$ where

$$y_k = 1 \quad \text{if} \quad f_2(\sigma_k, i_2) = f_3(\sigma_k, i_3) = \sigma_{k+1}, \quad \forall i_2 \in I_2, \quad i_3 \in I_3, \quad 1 \leqq n \leqq n-2;$$

$$y_{n-1} = 1 \quad \text{if} \quad f_2(\sigma_{n-1}, i_2) = f_3(\sigma_{n-1}, i_3) = \sigma_1, \quad \forall i_2 \in I_2, \quad i_3 \in I_3;$$

$$y_k = 0 \quad \text{in other cases} \quad 0 \leqq k \leqq n-1.$$

It is not hard to verify that S_{AI} and D_{SI} satisfy the conditions of Theorem 2.

Conclusions

We have given a caracterisation of p-isomorphic problems to graph isomorphism showing that graph isomorphism satisfy the conditions of Berman—Hartmanis Theorem. Next we have proved that some of the problems which are polynomial time equivalent to graph isomorphism are p-isomorphic. Are all the isomorphism complete problems p-isomorphic? Perhaps the answer of this question is useful in determining the complexity of graph isomorphism problem.

DEPT. OF APPLIED MATHEMATICS
AND COMPUTER SCIENCE
IASI UNIVERSITY
ROUMANIA

References

[1] Aho, A. V., J. F. Hopcroft, J. D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Mass., 1974.

[2] Berman, L., J. Hartmanis, On isomorphism and density of NP and other complete sets, *SIAM J. Comput.*, v. 6, 1977, pp. 305—322.

[3] Booth, K. S., C. J. Colbourn, Problems polynomially equivalent to graph isomorphism, *Tech. Rep. Toronto Univ.*, CS 77—04, Canada.

[4] Colbourn, C. J., M. J. Colbourn, Graph isomorphism and self complementary graphs, *SIGACT News*, v. 10, 1978, pp. 25—29.

[5] Cook, S. A., The complexity of theorem-proving procedures, 3rd STOC 1971, pp. 151—158.

[6] Garey, M. R., D. S. Johnson, *Computers and intractability, A guide to the theory of NP-completness*, Freeman & Comp., San Francisco, 1979.

[7] Grigoras, Gh., On the isomorphism of labeled directed graphs and finite automata, submited for publication.

[8] Karp, R. M., Reducibilities among combinatorial problems, *Complexity of Computer Computations*, ed. R. E. Miller and J. W. Thatcher, Plenum Press, 1972, pp. 85—104.

[9] Kozen, D., A clique problem equivalent to graph isomorphism, *SIGACT News*, v. 10, 1978, pp. 50—52.

[10] Miller, G. L., Graph isomorphism, General remarks, 9th STOC 1977, pp. 143—150.

(Received March 9, 1980)