Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 155-162

# 3-connected Planar Graph Isomorphism is in Log-space

**Samir Datta**[1], **Nutan Limaye**[2], **Prajakta Nimbhorkar**[2]

[1] Chennai Mathematical Institute
Chennai, India

[2] The Institute of Mathematical Sciences,
Chennai, India

ABSTRACT. We consider the isomorphism and canonization problem for 3-connected planar graphs. The problem was known to be L -hard and in UL ∩ coUL [TW08]. In this paper, we give a deterministic log-space algorithm for 3-connected planar graph isomorphism and canonization. This gives an L -completeness result, thereby settling its complexity.
The algorithm uses the notion of universal exploration sequences from [Kou02] and [Rei05]. To our knowledge, this is a completely new approach to graph canonization.

## 1 Introduction

The general graph isomorphism problem is a well studied problem in computer science. Given two graphs, it deals with finding a bijection between the sets of vertices of these two graphs, such that the adjacencies are preserved. The problem is in NP , but it is not known to be complete for NP . In fact, it is known that if it is complete for NP , then the polynomial hierarchy collapses to its second level. On the other hand, no polynomial time algorithm is known. For general graph isomorphism NL and PL hardness is known [Tor00], whereas for trees, L and $NC^1$ hardness is known, depending on the encoding of the input [JT98].

In literature, many special cases of this general graph isomorphism problem have been studied. In some cases like trees [Lin92], [Bus97], or graphs with coloured vertices and bounded colour classes [Luk86], NC algorithms are known. We are interested in the case where the graphs under consideration are planar graphs. In [Wei66], Weinberg presented an $O(n^2)$ algorithm for testing isomorphism of 3-connected planar graphs. Hopcroft and Tarjan [HT74] extended this for general planar graphs, improving the time complexity to $O(n \log n)$. Hopcroft and Wong [HW74] further improved it to give a linear time algorithm. Its parallel complexity was first considered by Miller and Reif [MR91] and Ramachandran and Reif [RR90]. They gave an upper bound of $AC^1$. Verbitsky [Ver07] gave an alternative proof for the same bound. Recently Thierauf and Wagner [TW08] improved it to UL ∩ coUL for 3-connected planar graphs. They also proved that this problem is hard for L .

In this paper, we give a log-space algorithm for 3-connected planar graph isomorphism, thereby proving L -completeness. Thus the main result of our paper can be stated as follows:



FSTTCS 2008
IARCS Annual Conference on
Foundations of Software Technology and Theoretical Computer Science
http://drops.dagstuhl.de/opus/volltexte/2008/1749

**THEOREM 1.** *Given two 3-connected planar graphs $G$ and $H$, deciding whether $G$ is isomorphic to $H$ is complete for* L .

Thierauf and Wagner use shortest paths between pairs of vertices of a graph to obtain a canonical spanning tree. A systematic traversal of this tree generates a canonical form for the graph. The best known upper bound for shortest paths in planar graphs is UL ∩ coUL [TW08]. Thus the total complexity of their algorithm goes to UL ∩ coUL , despite the fact that all other steps can be done in L .

We identify that their algorithm hinges on making a systematic traversal of the graph in canonical way. Thus we bypass the step of finding shortest paths and give an orthogonal approach for finding such a traversal. We use the notion of universal exploration sequences (UXS) defined in [Kou02]. Given a graph on $n$ vertices with maximum degree $d$, a UXS is a polynomial length string over $\{0, \ldots, d-1\}$, that can be used to traverse the graph for a chosen combinatorial embedding $\rho$, starting vertex $u$ and a starting edge $e = \{u, v\}$. Reingold [Rei05] proved that such a universal sequence can be constructed in L . Using this result, we canonize a 3-connected planar graph in log-space.

In Section 2, we give some basic definitions that we use in the later sections. In Section 3, we describe our log-space algorithm. We conclude with a discussion of open problems in Section 4.

# 2 Preliminaries

In this section, we recall some basic definitions related to graphs and universal exploration sequences.

## 2.1 The Graph Isomorphism Problem

**DEFINITION 2.** *Graph isomorphism: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be isomorphic if there is a bijection $\phi : V_1 \to V_2$ such that $(u, v) \in E_1$ if and only if $(\phi(u), \phi(v)) \in E_2$.*

Let GI be the problem of finding such a bijection $\phi$ given two graphs $G_1, G_2$. Let Planar-GI be the special case of GI when the given graphs are planar. 3-connected planar graph isomorphism problem is a special case of Planar-GI when the graphs are 3-connected planar graphs. We recall the definition and properties of 3-connected planar graphs in the following section.

## 2.2 3-connected planar graphs

A graph $G = (V, E)$ is *connected* if there is a path between any two vertices in $G$. A vertex $v \in V$ is an *articulation point* if $G(V \setminus \{v\})$ is not connected. A pair of vertices $u, v \in V$ is a *separation pair* if $G(V \setminus \{u, v\})$ is not connected. A *biconnected* graph contains no articulation points. A 3-*connected* graph contains no separation pairs.

A *planar combinatorial embedding* $\rho$ for a planar graph $G$ specifies the cyclic (say, clockwise) ordering of edges around each vertex in some plane embedding of $G$. A graph $G$ with a fixed combinatorial embedding $\rho$ is called an *oriented graph* $(G, \rho)$.

In general, a planar graph can have exponentially many planar embeddings. In [Whi33], Whitney proved that 3-connected planar graphs have precisely two combinatorial embeddings. This is a special property of 3-connected planar graphs which we crucially use in our log-space algorithm.

### 2.3 Universal Exploration Sequences

Let $G = (V, E)$ be a $d$-regular graph, with given combinatorial embedding $\rho$. The edges around any vertex $u$ can be numbered $\{0, 1, \ldots, d-1\}$ according to $\rho$ arbitrarily in clockwise order. A sequence $\tau_1 \tau_2 \ldots \tau_k \in \{0, 1, \ldots, d-1\}^k$ and a starting edge $e_0 = (v_{-1}, v_0) \in E$, define a walk $v_{-1}, v_0, \ldots v_k$ as follows: For $0 \leq i \leq k$, if $(v_{i-1}, v_i)$ is the $s^{th}$ edge of $v_i$, let $e_i = (v_i, v_{i+1})$ be $(s + \tau_i)^{th}$ edge of $v_i$ modulo $d$.

**DEFINITION 3.** *Universal Exploration sequences (UXS): A sequence $\tau_1 \tau_2 \ldots \tau_l \in \{0, 1, \ldots d-1\}^l$ is a universal exploration sequence for $d$-regular graphs of size at most $n$ if for every connected $d$-regular graph on at most $n$ vertices, any numbering of its edges, and any starting edge, the walk obtained visits all the vertices of the graph. Such a sequence is called an $(n, d)$-universal exploration sequence.*

Following lemma suggests that UXS can be constructed in L [Rei05]:

**LEMMA 4.** *There exists a log-space algorithm that takes as input $(1^n, 1^d)$ and produces an $(n, d)$-universal exploration sequence.*

# 3 Log-space Algorithm for 3-connected Planar-GI

In this section, we give a log-space algorithm for 3-connected planar graph isomorphism. This, combined with the L -hardness result by [TW08] proves our main theorem:

**Theorem 1** *Given two 3-connected planar graphs G and H, deciding whether G is isomorphic to H is complete for* L .

For general planar graphs, the best known parallel algorithm runs in $AC^1$ [MR91]. Thierauf and Wagner [TW08] recently improved the bound for the case of 3-connected planar graphs to UL ∩ coUL . This case is easier due to a result by Whitney [Whi33] that every planar 3-connected graph has precisely two planar embeddings on a sphere, where one embedding is the mirror image of the other. Moreover, one can compute these embeddings in L [AM00].

### 3.1 Overview of the UL ∩ coUL algorithm of [TW08]

For a 3-connected planar graph $G$, the algorithm by Thierauf and Wagner starts by constructing a code for every edge of $G$ and for any of the two combinatorial embeddings. Of all these codes, the lexicographically smallest one is the code for $G$. The codes for two graphs are equal if and only if they are isomorphic. A code with this property is called a *canonical code* for the graph.

The main steps involved in their algorithm are as follows:

1. Construct a canonical spanning tree $T$, which depends upon the planar embedding of the graph and a fixed starting edge.
2. Traverse the tree and output a canonical list of edges.
3. Relabel the vertices of the graph according to this list to get the canonical code.

A canonical spanning tree in step 1 involves computation of shortest paths between pairs of vertices of $G$. Bourke, Tewari and Vinodchandran [BTV07] proved that planar reachability is in $\mathsf{UL} \cap \mathsf{coUL}$ . Thierauf and Wagner extend their result for computing distances in planar graphs in $\mathsf{UL} \cap \mathsf{coUL}$ . Once this spanning tree is constructed, the remaining steps can be executed in $\mathsf{L}$ .

### 3.2 Outline of our approach

Our approach bypasses the spanning tree construction step in the algorithm of [TW08] outlined above and thus eliminates distance computations. In that sense, we believe that this is a completely new approach for computing canonical codes for 3-connected planar graphs.

Our algorithm can be outlined as follows:

1. Given a 3-connected planar graph $G = (V, E)$, find a planar embedding $\rho$ of $G$.
2. Make the graph 3-regular canonically for this embedding $\rho$ to obtain an edge-coloured graph $G'$ as described in Algorithm 1.
3. Find the canon of $G'$ using Algorithm 2.

The step 1 is in log-space due to a result by Allender and Mahajan [AM00]. We prove that steps 2 and 3 can also be done in log-space. Step 3 uses the idea of UXS introduced by Koucký [Kou02]. Step 2 essentially does the preprocessing in order to make step 3 applicable.

The canonical code thus constructed is specific to the choice of the combinatorial embedding, the starting edge, and the starting vertex. Let the given 3-connected planar graphs be $G$ and $H$. For $G$, we fix an embedding, a starting edge, and a starting vertex arbitrarily and cycle through both embeddings and all choices of the starting edge and the starting vertex for $H$, comparing the codes for each of them. As there are only polynomially many choices, a log-space transducer executing this loop runs only for polynomially many steps. If the canonical codes of $G$ and $H$ match for any of the choices, we say that $G$ and $H$ are isomorphic.

### 3.3 Making the graph 3-regular

In this section, we describe the procedure to make the graph 3-regular. In Section 3.4, we use Reingold's construction for UXS [Rei05] to come up with a canonical code. As Reingold's construction [Rei05] for UXS requires the graph to have constant degree, we do this preprocessing step. In Lemma 5, we prove that two graphs are isomorphic if and only if they are isomorphic after the preprocessing step. We note that after the preprocessing step, the graph does not remain 3-connected, however, the embedding of the new graph is inherited from the given graph. Hence even the new graph has only two possible embeddings.

We describe the preprocessing steps in Algorithm 1. Note that the new graph thus obtained has $2|E|$ vertices.

**Algorithm 1** Procedure to get a 3-regular planar graph $G'$ from 3-connected planar graph $G$.

**Input:** A 3-connected planar graph $G$ with planar combinatorial embedding $\rho$.
**Output:** A 3-regular planar graph $G'$ on $2m$ vertices, with edges coloured 1 and 2 and planar combinatorial embedding $\rho'$.

**for all** $v_i \in V$ **do**
  Replace $v_i$ of by a cycle $\{v_{i1}, \ldots, v_{id_i}\}$ on $d_i$ vertices, where $d_i$ is the degree of $v_i$.
  The $d_i$ edges $\{e_{i1}, \ldots, e_{id_i}\}$ incident to $v_i$ in $G$ are now incident to $\{v_{i1}, \ldots, v_{id_i}\}$ respectively.
  Colour the cycle edges with colour 1.
  Colour $e_{i1}, \ldots, e_{id_i}$ by colour 2.
**end for**

LEMMA 5. *Given two 3-connected planar graphs $G_1, G_2$, $G_1 \cong G_2$ if and only if $G_1' \cong G_2'$ where the isomorphism between $G_1'$ and $G_2'$ respects colours of the edges.*

PROOF. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two 3-connected planar graphs with planar combinatorial embeddings $\rho_1$ and $\rho_2$ respectively. Let $\phi : V_1 \to V_2$ be an isomorphism between the oriented graphs $(G_1, \rho_1)$ and $(G_2, \rho_2)$. By isomorphism of oriented graphs we mean that the graphs are isomorphic for the fixed embeddings, in our case $\rho_1$ and $\rho_2$.

Construct $G_1'$ and $G_2'$ as described in Algorithm 1, replacing each vertex $v$ of degree $d$ by a cycle of length $d$, and colouring the new cycle edges with colour 1 and original edges with colour 2. The algorithm preserves the orientation of original edges from $G_1$ and $G_2$ and outputs the coloured oriented graphs $(G_1', \rho_1')$ and $(G_2', \rho_2')$.

Given an isomorphism $\phi$ between $(G_1, \rho_1)$ and $(G_2, \rho_2)$, we show how to derive an isomorphism $\phi'$ between $(G_1', \rho_1')$ and $(G_2', \rho_2')$. By our construction, edges around a vertex in $G_1$(respectively $G_2$) get the same combinatorial embedding around the corresponding cycle in $G_1'$ $(G_2')$. Consider an edge $\{v_i, v_j\}$ in $E_1$. Let $\phi(v_i) = u_k$ and $\phi(v_j) = u_l$. $\{u_k, u_l\} \in E_2$. Let corresponding edge in $G_1'$ be $\{v_{i_p}, v_{i_q}\}$ and that in $G_2'$ be $\{u_{k_r}, u_{k_s}\}$. Then we define a map $\phi' : V_1' \to V_2'$ which is inherited from $\phi$ such that $\phi'(v_{i_p}) = u_{k_r}$ and $\phi'(v_{j_q}) = u_{k_s}$. It is easy to see that $\phi'$ is an isomorphism for edge-coloured oriented graphs $(G_1', \rho_1')$ and $(G_2', \rho_2')$.

Now we show how to obtain an isomorphism $\phi$ between $(G_1, \rho_1)$ and $(G_2, \rho_2)$, given an isomorphism $\phi'$ between $(G_1', \rho_1')$ and $(G_2', \rho_2')$. Let $e = \{v_{i_p}, v_{i_q}\} \in E_1'$ and the corresponding edge $e' = \{\phi'(v_{i_p}), \phi'(v_{i_q})\} \in E_2'$. Let $v_{i_p}$ and $v_{i_q}$ correspond to the same vertex $v_i$ in $G_1$. Then colour of $e$ and $e'$ is 1. Thus $\phi'$ maps copies of the same vertex of $G_1$ to copies of a single vertex of $G_2$. Hence a map $\phi$ can be derived from $\phi'$ in a natural way. It is easy to see that $\phi$ is an isomorphism between oriented graphs $(G_1, \rho_1)$ and $(G_2, \rho_2)$.

### 3.4 Obtaining the canonical code

Lemma 5 from the previous section suggests that for given embeddings $\rho_1$, $\rho_2$ of $G_1$ and $G_2$, it suffices to check the 3-regular oriented graphs $(G_1', \rho_1')$ and $(G_2', \rho_2')$ for isomorphism. The Procedure $canon(G, \rho, v, e = (u, v))$ described in Algorithm 2 does this using universal exploration sequences.

**Algorithm 2** Procedure $canon(G, \rho, v, e = (u, v))$

**Input:** Edge-coloured graph $G = (V, E)$ with maximum degree 3 and combinatorial embedding $\rho$, starting vertex $v$, starting edge $e = (u, v)$.
**Output:** Canon of $G$.

1: Construct a $(n, 3)$-universal exploration sequence $U$.
2: With starting vertex $v \in V$ and edge $e = (u, v)$ incident to it, traverse $G$ according to $U$ and $\rho$ outputting the labels of the vertices.
3: Relabel the vertices according to their first occurrence in this output sequence, as in step 3 of [TW08].
4: For every $(i, j)$ in this labelling, output whether $(i, j)$ is an edge or not. If it is an edge, output its colour. This gives a canon for the graph.

We prove correctness of Algorithm 2 in the following lemma:

**LEMMA 6.** *Let* $\sigma_1 = canon(G'_1, \rho'_1, v_1, e_1 = (u_1, v_1))$ *and* $\sigma_2 = canon(G'_2, \rho'_2, v_2, e_2 = (u_2, v_2))$. *If* $\sigma_1 = \sigma_2$ *then* $G'_1 \cong G'_2$. *Further, if* $G'_1 \cong G'_2$ *then for some choice of* $\rho'_2, v_2, e_2, \sigma_1 = \sigma_2$.

PROOF. If $G'_1 \cong G'_2$, then there is a bijection $\phi : V'_1 \to V'_2$ for corresponding embeddings $\rho'_1, \rho'_2$. Let $e_1 = (u, v) \in E'_1$. Then $e_2 = (\phi(u), \phi(v)) \in E'_2$. Let $e_1$ and $e_2$ be chosen as starting edges and $v$ and $\phi(v)$ as starting vertices for traversal using UXS $U$ for $(G'_1, \rho'_1)$ and $(G'_2, \rho'_2)$ respectively. Let $T_1$ and $T_2$ be the output sequences. If a vertex $w \in V'_1$ occurs at position $l$ in $T_1$ then $\phi(w) \in V'_2$ occurs at position $l$ in $T_2$ as the oriented graphs are isomorphic, and the same UXS is used for their traversal. Thus the sequences are canonical when projected down to the first occurrences and hence $\sigma_1 = \sigma_2$.

Let $\sigma_1 = \sigma_2 = \sigma$. The labels of vertices in $\sigma$ are just a relabelling of vertices of $V'_1$ and $V'_2$. These relabellings are some permutations, say $\pi_1$ and $\pi_2$. Then $\pi_1 \cdot \pi_2^{-1} : V'_1 \to V'_2$ is a bijection.

After constructing canonical code $\sigma'$ for a graph $G'$, it remains to construct canonical code $\sigma$ for the original graph $G$. For this, we need to give a unique label to every vertex of graph $G$. It suffices to pick the minimum label among the labels of all its copies in $G'$. All copies of a vertex can be found by traversing colour 1 edges, starting from one of its copies. Thus the canonical code for graph $G$ can be constructed in log-space as follows:

For each edge $(i, j)$ of colour 2 in $\sigma'$, traverse along the edges coloured 1 starting from $i$ and find the minimum label among the vertices visited. Let it be $p$. Repeat the process for $j$. Let the minimum label among the vertices visited along edges of colour 1 be $q$. Thus the canonical labels for $i$ and $j$ are $p$ and $q$ respectively. Output the edge $(p, q)$. The sequence thus obtained contains $n$ distinct labels for vertices, each between $\{1, 2, \ldots, 2m\}$. This can further be converted into a sequence with labels for vertices between $\{1, 2, \ldots, n\}$ by finding the rank of each of the labels. This gives us $\sigma$. Correctness follows from the fact that vertices connected with edges of colour 1 are copies of the same vertex in $G$, hence they should get the same number.

Clearly, each of the above steps can be performed in $\mathsf{L}$ and hence the algorithm runs in $\mathsf{L}$ . This proves Theorem 1.

## 4 Conclusion

Our work settles the open question of the complexity of 3-connected planar graph isomorphism mentioned in [TW08] by giving a log-space algorithm. One of the most challenging questions is to settle the complexity of the general graph isomorphism problem. The other important goal is to improve upon the $\mathsf{AC}^1$ upper bound of [MR91] for planar graph isomorphism.

## 5 Acknowledgement

We thank Jacobo Torán, V. Arvind, and Meena Mahajan for helpful discussions, and the anonymous referees for helpful comments.

## References

[AM00] Eric Allender and Meena Mahajan. The complexity of planarity testing. In *STACS '00: Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science*, pages 87–98, 2000.

[BTV07] Chris Bourke, Raghunath Tewari, and N V Vinodchandran. Directed planar reachability is in unambiguous logspace. In *to appear in Proceedings of IEEE Conference on Computational Complexity CCC*, pages –, 2007.

[Bus97] Samuel R. Buss. Alogtime algorithms for tree isomorphism, comparison, and canonization. In *KGC '97: Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory*, pages 18–33, 1997.

[HT74] John Hopcroft and Robert Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[HW74] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184, 1974.

[JT98] P. McKenzie B. Jenner and J. Torán. A note on the hardness of tree isomorphism. In *COCO '98: Proceedings of the Thirteenth Annual IEEE Conference on Computational Complexity*. IEEE Computer Society, 1998.

[Kou02] Michal Koucký. Universal traversal sequences with backtracking. *J. Comput. Syst. Sci.*, 65(4):717–726, 2002.

[Lin92] Steven Lindell. A logspace algorithm for tree canonization (extended abstract). In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 400–404, 1992.

[Luk86] Eugene M. Luks. Parallel algorithms for permutation groups and graph isomorphism. In *FOCS*, pages 292–302, 1986.

[MR91] Gary L. Miller and John H. Reif. Parallel tree contraction part 2: further applications. *SIAM J. Comput.*, 20(6):1128–1147, 1991.

[Rei05] Omer Reingold. Undirected st-connectivity in log-space. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 376–385, 2005.

[RR90] Vijaya Ramachandran and John Reif. Planarity testing in parallel. Technical report, 1990.

[Tor00] J. Torán. On the hardness of graph isomorphism. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 180, 2000.

[TW08] Thomas Thierauf and Fabian Wagner. The isomorphism problem for planar 3-connected graphs is in unambiguous logspace. In *STACS*, pages 633–644, 2008.

[Ver07] Oleg Verbitsky. Planar graphs: Logical complexity and parallel isomorphism tests. In *STACS*, pages 682–693, 2007.

[Wei66] H. Weinberg. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. *Circuit Theory*, 13:142148, 1966.

[Whi33] H. Whitney. A set of topological invariants for graphs. *American Journal of Mathematics*, 55:235–321, 1933.