On the Complexity of Graph Reconstruction

Dieter Kratsch
Fakultät Mathematik
Friedrich-Schiller-Universität
O-6900 Jena, Germany

Lane A. Hemachandra[*]
Department of Computer Science
University of Rochester
Rochester, NY 14627 USA

March 11, 1991

Abstract

In the wake of the resolution of the four color conjecture, the graph reconstruction conjecture has emerged as a focal point of graph theory. This paper considers the *computational complexity* of decisions problems (DECK CHECKING and LEGITIMATE DECK), the construction problems (PREIMAGE CONSTRUCTION), and counting problems (PREIMAGE COUNTING) related to the graph reconstruction conjecture. We show that:

1. DECK CHECKING $\leq_m^\ell$ GRAPH ISOMORPHISM $\leq_m^\ell$ LEGITIMATE DECK, and
2. if the graph reconstruction conjecture holds, then GRAPH ISOMORPHISM $\equiv_{iso}^\ell$ DECK CHECKING.

Relatedly, we display the first natural GI-hard NP set lacking obvious padding functions. Finally, we show that LEGITIMATE DECK, PREIMAGE CONSTRUCTION, and PREIMAGE COUNTING are solvable in polynomial time on planar graphs, graphs with bounded genus, and partial k-trees for fixed k.

1 Introduction

Harary's survey of the reconstruction conjecture recounts the origins of the problem [19]:

> The author first heard of this fascinating problem when Kelly [23] proved the theorem for trees in 1957. This result was obtained in Kelly's doctoral dissertation which was written under Ulam, who published [37] a statement of the problem in 1960 (although it was already known to him in 1929, when he assiduously collected mathematical problems posed by his fellow graduate students and professors in Lwów, Poland.) This has led to some confusion concerning

[*]Research supported in part by the National Science Foundation under grants CCR-8809174/CCR-8996198 and CCR-8957604.

> whose name should be attached to this conjecture. The solution which I recommend heartily is to refer to this problem henceforth as the Reconstruction Conjecture.

The reconstruction conjecture states: Any graph with at least three vertices can be reconstructed from the collection of its one-vertex-deleted subgraphs. It is widely viewed as one of the most interesting and challenging open problems in graph theory, and has generated many excellent surveys [15,19,5,33]. As noted above, the first result in this field was Kelly's proof that the conjecture is true when restricted to trees; i.e., trees are *reconstructible* [23]. Since that time, many graph classes have been shown to be reconstructible. Among such classes are: disconnected graphs [17], regular graphs [33], separable graphs with no pendant vertex [4], maximal outerplanar graphs [30], outerplanar graphs [12], unicyclic graphs [29], (nontrivial) Cartesian product graphs [9], squares of trees [16], bidegreed graphs [27], unit interval graphs and threshold graphs [38] (see [17,11,7,6,31,26] for further discussion of reconstructibility of graph classes).

In 1964, Harary [17] stated a very useful formulation of the conjecture: Somebody draws on cards all one-vertex-deleted subgraphs of an unknown graph, one subgraph per card; can we reconstruct the original graph from this deck of cards, up to isomorphism? This formulation led naturally to the question of which graph parameters can be computed from the deck; among the parameters that have been considered are the number of vertices and edges, the degree sequence, and the sequence of the degrees of the neighborhood vertices (see [33]).

In the present paper, we are concerned with *complexity-theoretic* aspects of reconstruction. This line of inquiry springs from a question of Nash-Williams [33]: If we do not know that a deck is created from a graph, how difficult is it to find out whether this is the case—i.e., whether the deck is *legitimate*?

We study that question here and we also study the complexity of the following problems:

1. Given a graph and a deck, check whether that deck results from the graph.

2. Given a deck, construct a graph with this deck.

3. Given a deck, compute the number of (distinct) graphs with this deck.

We will see that there is a strong relationship between the problems above and the well-known problem GRAPH ISOMORPHISM (abbreviated GI), which is the set: $\{(G_1, G_2) \mid G_1$ is isomorphic to $G_2\}$. GI is one of the few problems known to be in NP, yet neither known to be in P nor known to be NP-complete.

Goldwasser and Sipser [13] and Schöning [35] have proven that the GI belongs to the class coAM; that is, there is an interactive proof protocol for graph nonisomorphism. Since NP∩coAM is included in the second level of the low hierarchy [34], it follows that graph isomorphism is low [35]. Consequently, if GI were NP-complete, then the polynomial-time hierarchy would collapse to its second level. However, it remains quite possible that GI is in P, even if the polynomial hierarchy is infinite.

Much effort has been devoted to determining the computational complexity of GI when restricted to special classes of graphs, and there have been some striking successes. Many graph classes have polynomial-time isomorphism algorithms, including such classes as directed path graphs [8], graphs of bounded genus, permutation, series-parallel, and grid graphs (see [21]), and partial k-trees for fixed k [3].

On the other hand, for some graph classes GI has been shown to be as difficult as for general graphs (within the flexibility of $\leq_m^p$ reductions). In such cases, the restriction is called isomorphism-complete. Among the isomorphism-complete restrictions are claw-free, perfect, and line graphs [21].

2 Preliminaries

We consider only finite, undirected, simple graphs with at least three vertices. $V(G)$ and $E(G)$ denotes the set of vertices and edges, respectively, of graph G. We use V and E when there is no ambiguity. Furthermore, we use n to denote $|V(G)|$, the cardinality of $V(G)$. The degree of $v \in V(G)$ in G, denoted $deg_G(v)$, is the number of edges incident to v. $\delta(G) = min\{deg_G(v) \mid v \in V(G)\}$ denotes the minimum degree of G. For graphs, $G \cup H$ denotes the disjoint union of the graphs.[1] The one-vertex-deleted subgraph $G - v$, with $v \in V(G)$, has vertex set $V(G) - \{v\}$ and edge set $E(G) - \{\{v, x\} \mid \{v, x\} \in E(G)\}$; we delete the vertex v and all incident edges from G to get $G - v$. $\langle\ \rangle$ indicates a multiset. For a labeling of V, $V = \{v_1, v_2, \dots, v_n\}$ we call $\langle G - v_1, G - v_2, \dots, G - v_n\rangle$ the deck of G, denoted $deck(G)$. If $deck(G) = \langle G_1, G_2, \dots, G_n\rangle$ we call G a preimage of $\langle G_1, G_2, \dots, G_n\rangle$. We are now able to define the problems we study:

Deck Checking (abbreviated DC)

instance: Graph G, collection of graphs $G_1, G_2, \dots, G_n$

question: $deck(G) = \langle G_1, G_2, \dots, G_n\rangle$?

Legitimate Deck (abbreviated LD)

instance: Collection of graphs $G_1, G_2, \dots, G_n$

question: Is that deck legitimate—i.e., does there exist a preimage G of $\langle G_1, G_2, \dots, G_n\rangle$?

Preimage Construction (abbreviated PCon)

instance: Collection of graphs $G_1, G_2, \dots, G_n$

task: Construct a preimage of the deck $\langle G_1, G_2, \dots, G_n\rangle$, or output "NO" if there is none.

[1] For sets, $\cup$ denotes set union; for multisets, $\cup$ denotes multiset union ($\langle 0, 0\rangle \cup \langle 0, 1, 2\rangle = \langle 0, 0, 0, 1, 2\rangle$, etc.).

PREIMAGE COUNTING (abbreviated PCou)

instance: Collection of graphs $G_1, G_2, \ldots, G_n$

task: Count the number of (nonisomorphic) preimages of the deck $\langle G_1, G_2, \ldots, G_n \rangle$.

Clearly, these problems are connected to the reconstruction conjecture. We have: $\langle G_1, G_2, \ldots, G_n \rangle \in$ LD iff $\langle G_1, G_2, \ldots, G_n \rangle$ produces a graph G as output for PCon. If the reconstruction conjecture holds, $\langle G_1, G_2, \ldots, G_n \rangle \in$ LD iff $\langle G_1, G_2, \ldots, G_n \rangle$ produces output "1" for PCou. For standard graph-theoretic notions not defined here we refer to [18, 14,2]. For definitions of graph classes we refer to [14,21]. We mention only the standard definition of partial k-trees.

Definition 2.1 1. (a) A complete graph on k vertices is a k-tree.

(b) If $G = (V, E)$ is a k-tree and $V' \subseteq V$ is a set of k vertices that induces a complete subgraph in G, then the graph obtained by adding a new vertex v to V together with an edge from v to every vertex in V' is also a k-tree.

(c) Only graphs that are k-trees via (possibly repeated) application of the above rules are k-trees.

2. A graph G is a partial k-tree if and only if there is some k-tree G' of which it is a subgraph—i.e., we get G by deleting vertices and edges of G'.

We refer the reader to standard texts for general complexity-theoretic background [39, 1], and for definitions of complexity theoretic notions such as NP and NP-completeness [10], reductions [25,24], and isomorphism [41]. In particular, we use $\leq_m^p$ to denote polynomial-time many-one reductions, $\leq_m^\ell$ to denote logspace many-one reductions, $\equiv_m^p$ to denote polynomial-time many-one equivalence, and $\equiv_{iso}^\ell$ to denote logspace isomorphism,

We tacitly assume that encoding details (of multisets, pairs, graphs, etc.) are handled in the standard fashion.

3 The Complexity of DECK CHECKING

First, we consider the relative complexity of GI and DC.

Lemma 3.1 DC $\leq_m^p$ GI.

Proof: Let $(G; \langle G_1, G_2, \ldots, G_n \rangle)$ be an input to DC. We construct an input (G', G'') to GI such that $(G; \langle G_1, \ldots, G_n \rangle) \in$ DC iff $(G', G'') \in$ GI. We have $|V(G_i)| = n - 1$ for every $i \in \{1, 2, \ldots, n\}$; otherwise deck $(G) \neq \langle G_1, G_2, \ldots, G_n \rangle$, so we output any nonisomorphic G' and G'' and we are done.

G' is built up from $\langle G_1, G_2, \ldots, G_n \rangle$ and G'' is built up from deck $(G) = \langle H_1, H_2, \ldots, H_n \rangle$ in a tree-like manner. We assume that the vertex sets of $G_1, G_2, \ldots, G_n$ and $H_1, H_2, \ldots, H_n$ are pairwise disjoint. Then we define:

$$
\begin{aligned}
V(G') = \bigcup_{i=1}^{n} V(G_i) \quad & \cup \quad \{c'_1, c'_2, \ldots, c'_{n-1}, c'_n\} \\
& \cup \quad \{u'_1, u'_2, \ldots, u'_n\} \\
V(G'') = \bigcup_{i=1}^{n} V(H_i) \quad & \cup \quad \{c''_1, c''_2, \ldots, c''_{n-1}, c''_n\} \\
& \cup \quad \{u''_1, u''_2, \ldots, u''_n\} \\
E(G') = \bigcup_{i=1}^{n} E(G_i) \quad & \cup \quad \bigcup_{i=1}^{n} \{\{u'_i, v\} \mid v \in V(G_i)\} \\
& \cup \quad \bigcup_{i=1}^{n} (\{\{c'_i, c'_j\} \mid i < j \leq n\} \cup \{\{c'_1, u'_i\}\}) \\
E(G'') = \bigcup_{i=1}^{n} E(H_i) \quad & \cup \quad \bigcup_{i=1}^{n} \{\{u''_i, v\} \mid v \in V(H_i)\} \\
& \cup \quad \bigcup_{i=1}^{n} (\{\{c''_i, c''_j\} \mid i < j\} \cup \{\{c''_1, u''_i\}\})
\end{aligned}
\tag{1}
$$

Clearly, the transformation is polynomial. Thus we must show: $(G; \langle G_1, G_2, \ldots, G_n \rangle) \in \text{DC} \iff (G', G'') \in \text{GI}$. Suppose $(G; \langle G_1, G_2, \ldots, G_n \rangle) \in \text{DC}$. Then $deck(G) = \langle G_1, G_2, \ldots, G_n \rangle = \langle H_1, H_2, \ldots, H_n \rangle$. Hence $(G', G'') \in \text{GI}$. Suppose $(G', G'') \in \text{GI}$ and let I be an isomorphism mapping $V(G')$ onto $V(G'')$. Since only c'_1 and c''_1 have the maximum degree $2n-1$, we have $I(c'_1) = c''_1$. Since $\{u'_1, u'_2, \ldots, u'_n\}$ and $\{u''_1, u''_2, \ldots, u''_n\}$ are, respectively, the only neighbors of c'_1 and c''_1 with degree n, we have $I(\{u'_1, u'_2, \ldots, u'_n\}) = \{u''_1, u''_2, \ldots, u''_n\}$. Therefore the graphs hung on u'_i and $I(u'_i) = u''_j$, namely G_i and H_j, must be isomorphic for every $i \in \{1, 2, \ldots, n\}$. Therefore $\langle G_1, G_2, \ldots, G_n \rangle = \langle H_1, H_2, \ldots, H_n \rangle = deck(G)$ and $(G; \langle G_1, G_2, \ldots, G_n \rangle) \in \text{DC}$. ▮

Lemma 3.2 GI $\leq^p_m$ DC, if the graph reconstruction conjecture holds.

Proof: Let (G_1, G_2) be an input for GI. Then we have $(G_1; deck(G_2))$ as input for DC. Let $(G_1, G_2) \in \text{GI}$. Then $deck(G_1) = deck(G_2)$ and therefore $(G_1; deck(G_2)) \in \text{DC}$. Let $(G_1; deck(G_2)) \in \text{DC}$. By the graph reconstruction conjecture, $deck(G_2)$ has only one preimage up to isomorphism, namely G_2. Hence $(G_1, G_2) \in \text{GI}$. ▮

As a consequence of the above lemmas:

Theorem 3.3 DC $\equiv^p_m$ GI, if the graph reconstruction conjecture holds.

4 The Complexity of LEGITIMATE DECK

This sections relates the complexity of LD to the complexity GI and that of DC.

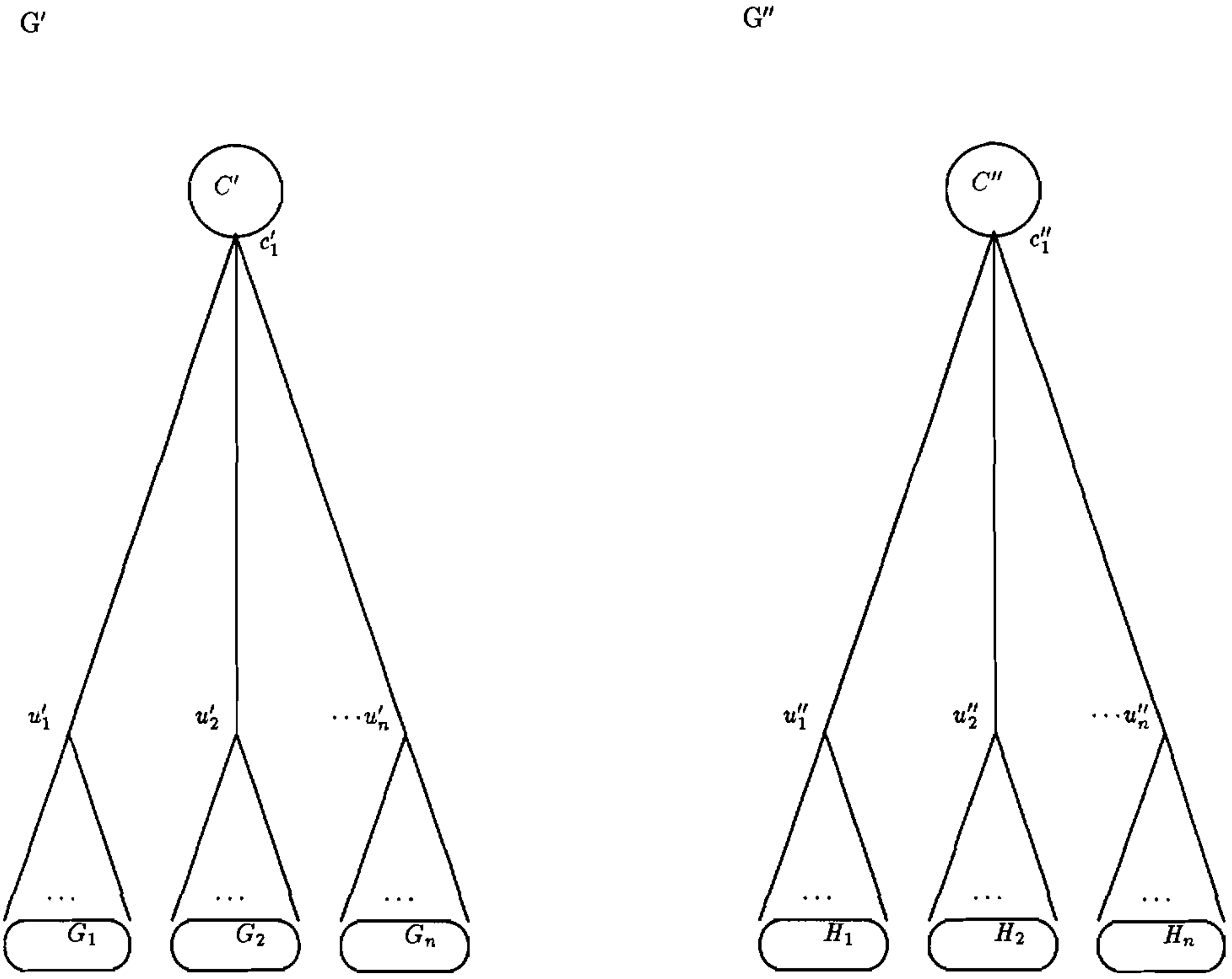

Figure 1: The Graphs G' and G''

Theorem 4.1 GI $\leq_m^p$ LD.

Proof: Without loss of generality we may restrict graph isomorphism to graphs without isolated vertices. Let (G, H) be the input to GI. Without loss of generality, let G and H be without isolated vertices, have the same number of vertices and edges, and have the same degree sequence. We use the following notation: $G + i$ is the graph G with i additional isolated vertices. We show that $(G, H) \in \text{GI}$ iff $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \,|\, G_i \in \textit{deck}(G)\rangle$ is a legitimate deck.

Suppose G and H are isomorphic. Then the preimage of $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \,|\, G_i \in \textit{deck}(G)\rangle$ is $G + 2$ and the deck is legitimate. Suppose $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \,|\, G_i \in \textit{deck}(G)\rangle \in \text{LD}$. Let the preimage be the graph G'. Then we have $|E(G')| = \frac{1}{|V(G)|} \cdot (\sum_{i=1}^{n} |E(G_i)| + |E(G)| + |E(H)|) = \frac{1}{n} \cdot (\sum_{i=1}^{n} |E(G_i)| + 2 \cdot |E(G)|)$ $= \frac{1}{n} \cdot ((n-2) \cdot |E(G)| + 2 \cdot |E(G)|) = |E(G)|$.

Since the degree of $v_i' \in V(G')$, with $G' - v_i' = G_i'$, is $|E(G')| - |E(G_i')|$, G' has exactly two isolates, namely v_1' and v_2', since $|E(G')| - |E(G)| = |E(G')| - |E(H)| = 0$. If one of the vertices v_i', $i \in \{3, \ldots, n+2\}$, with $G' - v_i' = G_{i-2} + 2$ would be an isolate then G would have an isolate, contradicting the choice of G. Suppose G' is isomorphic to $K + 2$, and K has no isolate. Then exactly two cards in the deck are isomorphic to $K + 1$. The only possible cards are those with $G + 1$ and $H + 1$. Hence G and H are isomorphic. Since the deck $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \,|\, G_i \in \textit{deck}(G)\rangle$ can be computed in polynomial time, the theorem follows. ∎

Combining Theorem 4.1 and Lemma 3.1 we have:

Corollary 4.2 DC $\leq_m^p$ LD.

5 Padding and Logspace Isomorphism

The reductions of Sections 3 and 4 are clearly computable not only in polynomial time, but indeed in logarithmic space. Thus we may strengthen the statements of Sections 3 and 4 to:

Theorem 5.1

1. DC $\leq_m^\ell$ GI.
2. If the graph reconstruction conjecture holds, then GI $\leq_m^\ell$ DC.
3. GI $\leq_m^\ell$ LD.
4. DC $\leq_m^\ell$ LD.

We wish to strengthen parts 1 and 2 above, by proving that if the graph reconstruction conjecture holds, then DC and GI are logspace isomorphic. This will show that DC and GI—under the assumption that the graph reconstruction conjecture holds—are essentially the same problem under different naming schemes.

We will first show that DC and GI have certain paddability properties. Then we will use the following results of Hartmanis to conclude that isomorphism holds.

Lemma 5.2 **([20])** *Let A be a set for which two logspace-computable functions $S_A(\, , \,)$ and $D_A(\,)$ exist such that:*

1. $(\forall x, y)[S_A(x,y) \in A$ *iff* $x \in A]$, *and*
2. $(\forall x, y)[D_A(S_A(x,y)) = y]$.

If f is any logspace reduction of C to A, the map $f'(x) = S_A(f(x), x)$ is a one-to-one logspace reduction of C to A and f'^{-1} is logspace computable.

Definition 5.3 **([20])** *Let $A \subseteq \Sigma^*$. Then $Z_A : \Sigma^* \to \Sigma^*$ is a padding function for the set A if:*

1. $Z_A(x) \in A$ *iff* $x \in A$, *and*
2. Z_A *is one-to-one.*

Lemma 5.4 **([20])** *Let f be a one-to-one logspace reduction of A to B and let f^{-1} be logspace computable. Assume that either A or B has a padding function Z_X ($X = A$ or B) that satisfies the conditions:*

1. Z_X *and* Z_X^{-1} *are logspace computable, and*
2. $(\forall y)[|Z_X(y)| > |y|^2 + 1]$.

Then there exists a one-to-one logspace reduction f' of A to B such that:

1. f'^{-1} *is logspace computable, and*
2. $(\forall y)[|f'(y)| > |y|^2]$.

Theorem 5.5 **([20])** *Let the set A be logspace reducible to B and B be logspace reducible to A; furthermore let the set A have a padding function Z_A satisfying Lemma 5.4 and functions S_A and D_A satisfying Lemma 5.2. Then B is logspace isomorphic to A iff B has functions S_B and D_B satisfying Lemma 5.2.*

We will show that GI and DC fulfill the conditions of Theorem 5.5.

Lemma 5.6 DC has functions S_{DC} and D_{DC} satisfying Lemma 5.2.

Proof: Let $(G; \langle G_1, G_2, \ldots, G_n \rangle)$ be an input to DC. We view y as a string of bits—$y = y_1 y_2 \cdots y_r$, $y_i \in \{0,1\}$—and construct a graph $\widehat{G}_y$ as follows:

$$\begin{aligned}
V(\widehat{G}_y) &= \{c_i \,|\, 1 \leq i \leq n+5\} \\
&\cup \; \{d_i \,|\, 1 \leq i \leq r+1\} \\
&\cup \; \{e_i \,|\, y_i = 1, \; 1 \leq i \leq r\} \\
E(\widehat{G}_y) &= \{\{c_i, c_j\} \,|\, 1 \leq i < j \leq n+5\} \cup \{\{c_1, d_1\}\} \\
&\cup \; \{\{d_i, d_{i+1}\} \,|\, 1 \leq i \leq r\} \\
&\cup \; \{\{d_i, e_i\} \,|\, y_i = 1, \; 1 \leq i \leq r\}
\end{aligned} \tag{2}$$

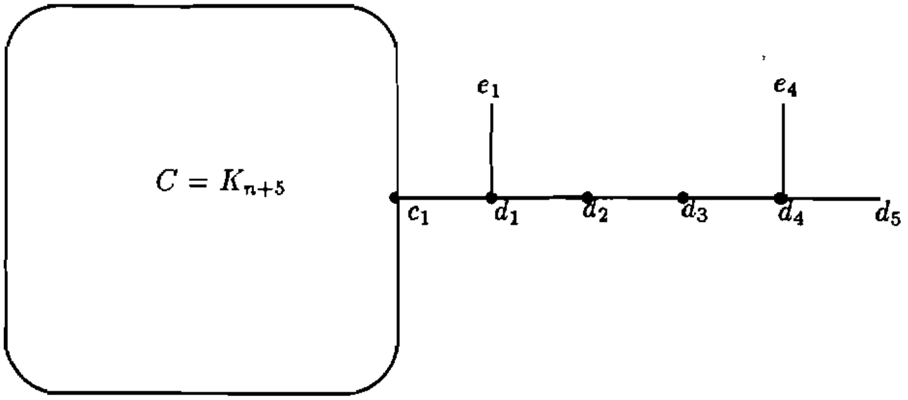

Figure 2: The Graph $\widehat{G}_y$, where $y = 1001$

We now define S_{DC} and D_{DC}:

$$S_{DC}((G;\langle G_1, G_2, \dots, G_n\rangle), y) =$$

$$(G \cup \widehat{G}_y; \langle G_1 \cup \widehat{G}_y\rangle \cup \langle G_2 \cup \widehat{G}_y\rangle \cup \dots \cup \langle G_n \cup \widehat{G}_y\rangle \cup \langle G \cup (\widehat{G}_y - v) \mid v \in V(\widehat{G}_y)\rangle).$$

(Recall that $\cup$ above means (graph) disjoint union when operating on graphs, and multiset union when operating on multisets.) For input $(H;\langle H_1, H_2, \dots, H_m\rangle)$, the function D_{DC} is computed in the following way:

1. Check that H has two connected components, say H_1 and H_2, with $|V(H_1)| < |V(H_2)|$.
2. Check that H_2 has the shape of $(\widehat{H}_1)_y$ (the notation here is analogous to that of the definition of $\widehat{G}_y$) for some y, and determine the bits of y by scanning the "caterpillar" added to $K_{|V(H_1)|+5}$.

(If some step above is unsuccessful then D_{DC} is undefined on that input.) It is not hard to see that, due to our choice of $\widehat{G}_y$, S_{DC} and D_{DC} are logspace computable and satisfy Lemma 5.2. ∎

Lemma 5.7 GI has functions S_{GI} and D_{GI} satisfying Lemma 5.2.

Proof: The construction is basically the same as in Lemma 5.6:

$$S_{GI}((G, H), y) = (G \cup \widehat{G}_y, H \cup \widehat{H}_y)$$

D_{GI} is then computed in the same way as D_{DC}. Both functions are logspace computable and satisfy Lemma 5.2. ∎

Lemma 5.8 GI has a padding function Z_{GI} satisfying Lemma 5.4.

Proof: The padding function Z_{GI} is defined as follows:

$$Z_{GI}((G,H)) = (G \cup K_{|V(G)|^3}, H \cup K_{|V(H)|^3}),$$

where K_n is a complete graph on n vertices. Z_{GI} is a padding function, since it is one-to-one, and $Z_{GI}((G,H)) \in$ GI iff $(G,H) \in$ GI. Furthermore, Z_{GI} satisfies Lemma 5.4: Z_{GI} and Z_{GI}^{-1} are logspace computable and Z_{GI} is quadratically length-increasing as required by condition (2) of Lemma 5.4. $\blacksquare$

Theorem 5.9 If the graph reconstruction conjecture holds, then GI is logspace isomorphic to DC.

Proof: Follows immediately from Theorem 5.5 and Lemmas 5.6, 5.7, and 5.8. $\blacksquare$

The question of paddability is of particular interest with respect to LD. Not only does LD lack any obvious (S,D) functions, but any straightforward attempt at providing a polynomial-time computable (or logspace computable) (S,D) functions would seem to have to actually first perform a reconstruction—namely, of the cards in the original deck.

We believe this to be the first natural GI-hard NP set that lacks obvious (S,D) functions.[2] Indeed, the only known example of a GI-hard NP set that lacks obvious (S,D)-functions is a certain interesting but artificial type of NP-complete set constructed by Joseph and Young [22]. Since GI and all standard NP-complete sets have obvious (S,D)-functions, we conjecture that LD is isomorphic to neither GI nor SATISFIABILITY.

6 Polynomial-Time Algorithms

We restrict our attention to hereditary graph classes $\mathcal{G}$—i.e., graph classes $\mathcal{G}$ with the property that when $G \in \mathcal{G}$, any induced subgraph of G belongs to $\mathcal{G}$. Let LD, PCon, and PCou *restricted to* $\mathcal{G}$ mean that the input graphs $G_1, G_2, \ldots, G_n$ and all admissible preimages of $\langle G_1, G_2, \ldots, G_n \rangle$ belong to $\mathcal{G}$.

Theorem 6.1 LEGITIMATE DECK, PREIMAGE CONSTRUCTION, and PREIMAGE COUNTING are solvable in polynomial time when restricted to any hereditary graph class $\mathcal{G}$ satisfying the following conditions:

1. there is a polynomial-time recognition algorithm for $\mathcal{G}$,
2. GI is solvable in polynomial time when restricted to $\mathcal{G}$, and
3. the minimum degree of each graph $G \in \mathcal{G}$ is bounded by a constant, k, not depending on $|V(G)|$; i.e., $\max\{\delta(G) \mid G \in \mathcal{G}\} \leq k$.

[2] Using the notation now standard (but differing from Hartmanis's earlier notation mentioned about), LD is a potentially "unpaddable" set.

Proof: The polynomial-time algorithm is as follows. For input $\langle G_1, G_2, \ldots, G_n \rangle$ we (hypothetically) assume $\langle G_1, G_2, \ldots, G_n \rangle \in$ LD and let G be a (hypothetical) preimage of $\langle G_1, \ldots, G_n \rangle$ such that, as guaranteed by (3), at least for one card $G_i = G - v_i$, it holds that $deg_G(v_i) \leq k$. Hence, we get G from G_i by adding exactly $deg_G(v_i)$ edges from v_i to vertices of G_i. There are at most $O(n^k)$ possibilities. For each preimage candidate G we have to solve "$(G; \langle G_1, \ldots, G_n \rangle) \in$ DC?" Let $v_1, v_2, \ldots, v_n$ be a labeling of $V(G)$. Then first we check whether $G \in \mathcal{G}$ and $(\forall i : 1 \leq i \leq n)[G - v_i \in \mathcal{G}]$ and $(\forall i : 1 \leq i \leq n)[G_i \in \mathcal{G}]$. By (1) and the definition of restriction for LD, PCon, and PCou, this is possible in polynomial time. Next, for every pair $i, j \in \{1, 2, \ldots, n\}$, we check whether $(G - v_i, G_j) \in$ GI. By (2), this can be done in polynomial time.

We construct a bipartite graph with bipartition $\{G - v_i \mid 1 \leq i \leq n\}$ and $\{G_j \mid 1 \leq j \leq n\}$. There is an edge between $G - v_i$ and G_j iff $(G - v_i, G_j) \in$ GI. Now $\langle G - v_1, G - v_2, \ldots, G - v_n \rangle = \langle G_1, G_2, \ldots, G_n \rangle$ iff this bipartite graph has a perfect matching. Whether a bipartite graph $G = (V, E)$ has a perfect matching can easily be checked in time $O(|V|^{\frac{1}{2}} \cdot |E|)$ [32]. Thus checking $\langle G; G_1, \ldots, G_n \rangle \in$ DC for one candidate G can be done in polynomial time. Hence, checking all $O(n^k)$ possible candidates G can be done in polynomial time. Thus, LD, PCon, and PCou can be solved in polynomial time. ∎

Corollary 6.2 Legitimate deck, preimage construction, and preimage counting are solvable in polynomial time when restricted to planar graphs, to graphs of bounded genus, and to partial k-trees for fixed k.

Proof: These classes fulfill the conditions (1) and (2) of Theorem 6.1 (see [3,21]. It is well-known that any planar graph has a vertex of degree at most 5. This can be extended to graphs of bounded genus, as [40, p. 21] mentions: If $G = (V, E)$ is a connected graph embeddable in S_k, an orientable surface of genus h, then

$$|E| \leq 3 \cdot (|V| + 2h - 2).$$

If $G = (V, E)$ is a connected graph embeddable in N_k, a non-orientable surface of genus k, then:

$$|E| \leq 3 \cdot (|V| + k - 2).$$

From this we get: If G has orientable genus h, then:

$$\delta(G) \leq 6 \cdot (|V| + 2h - 2)/|V| = 6 + \frac{12h - 12}{|V|} \leq 12h + 6.$$

If G has non-orientable genus k, then:

$$\delta(G) \leq 6 \cdot (|V| + k - 2)/|V| = 6 + \frac{6k - 12}{|V|} \leq 6k + 6.$$

Hence, any graph of bounded genus also has a bounded minimum degree; thus (3) is fulfilled.

For a partial k-tree G, let G' be a k-tree such that G is a subgraph of G'. Then the last vertex v of G added to G' during the recursive construction of G' has degree at most k in G', such that $deg_G(v) \leq k$. Hence, (3) is fulfilled. ∎

7 Conclusions and Open Problems

We have shown complexity relationships between problems on reconstruction. It is an interesting open question whether LD is NP-complete. One approach to this question would be to prove either that LD is in the low hierarchy or that LD is in the high hierarchy [34]. Furthermore, is it possible that LD $\leq_m^p$ GI or at least that LD Turing reduces to GI or that LD strong-nondeterministically [36,28] reduces to GI?

If a graph class is reconstructible this does not imply that LD, PCon, and PCou are solvable in polynomial time for that class. Nevertheless, some such proofs are constructive and yield polynomial-time algorithms, as, for example, for regular graphs [33]. Furthermore, polynomial-time algorithms for LD, PCon, and PCou, when restricted to $\mathcal{G}$, do not imply anything about the reconstructibility of $\mathcal{G}$. In fact, this question is still open for planar graphs, graphs of bounded genus, and partial k-trees.

Acknowledgments

We are grateful to Peggy Meeker for expertly converting this paper to LaTeX, to Katerina Chronaki for carefully proofreading an earlier version of this paper, to William Gasarch and Hans Bodlaender for helpful discussions, and especially to Gerd Wechsung for helpful discussions and for hosting two workshops in Jena, during which much of this research was performed.

References

[1] J.L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity.* Springer, Berlin, Vol. 1, 1988; Vol. 2, 1990.

[2] C. Berge. *Graphs.* Amsterdam, 1985.

[3] H.L. Bodlaender. Polynomial algorithms for chromatic index and graph isomorphism on partial k-trees. Technical report, Rijksuniversitet, June 1987.

[4] J.A. Bondy. On Ulam's conjecture for separable graphs. *Pacific J. Math.*, 31:281–288, 1969.

[5] J.A. Bondy and R.L. Hemminger. Graph reconstruction—a survey. *J. Graph Theory*, 1:227–268, 1977.

[6] G. Chartrand, V.H. Kronk, and S. Schuster. A technique for reconstructing disconnected graphs. *Colloq. Math.*, 27:31–34, 1973.

[7] P.Z. Chinn. A graph with ρ points and enough distinct $(\rho - 2)$-order subgraphs is reconstructible. In *Recent Trends in Graph Theory, LNM 186*, pages 71–73. Springer, Berlin, 1971.

[8] P.F. Dietz. *Intersection Graph Algorithms.* PhD thesis, Computer Science Dept., Cornell University, Ithaca, NY, 1984.

[9] W. Doerfler. Some results on the reconstruction of graphs. In *Infinite and Finite Sets*, pages 361–383. North-Holland, Amsterdam, 1975. Colloq. Math. Soc. Janos Bolayi, Vol. 10.

[10] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[11] D. Geller and B. Manvel. Reconstruction of cacti. *Canad. J. Math.*, 21:1354–1360, 1969.

[12] W.B. Giles. The reconstruction of outerplanar graphs. *J. Combinatorial Theory Ser. B*, 16:215–226, 1974.

[13] S. Goldwasser and M. Sipser. Private coints versus public coints in interactive proof systems. In *Proceedings of the 18th ACM Symposium on Theory of Computing*, pages 59–68, 1986.

[14] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[15] D.L. Greenwell and R.L. Hemminger. Reconstructing graphs. In *The Many Facets of Graph Theory*, pages 91–114. Springer, Berlin, 1969.

[16] D.K. Gupta. Reconstruction conjecture for square of a tree. In *Graph Theory, LNM 1073*, pages 268–278. Springer, Berlin, 1984.

[17] F. Harary. On the reconstruction of a graph from a collection of subgraphs. In *Theory of Graphs and its Applications*, pages 47–52. Prague, 1964.

[18] F. Harary. *Graph Theory.* Massachusetts, 1969.

[19] F. Harary. A survey of the reconstruction conjecture. In *Graphs and Combinatorics, LNM 406*, pages 18–28. Springer, Berlin, 1974.

[20] J. Hartmanis. On log-tape isomorphisms of complete sets. *Theoret. Comput. Sci.*, 7:273–286, 1978.

[21] D.S. Johnson. The NP-completeness column: An ongoing guide. *J. Algorithms*, 6:434–451, 1985.

[22] D. Joseph and P. Young. Some remarks on witness functions for non-polynomial and non-complete sets in NP. *Theoret. Comput. Sci.*, 39:225–237, 1985.

[23] P.J. Kelly. A congruence theorem for trees. *Pacific J. Math.*, 7:961–968, 1957.

[24] R. Ladner and N. Lynch. Relativization of questions about log space computability. *Mathematical Systems Theory*, 10(1):19–32, 1976.

[25] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[26] J. Lauri. Proof of Harary's conjecture on reconstruction of trees. *Discrete Math.*, 43:79–90, 1983.

[27] J. Lauri. Graph reconstruction—some techniques and new problems. *Ars Combin.*, 24(B):35–61, 1987.

[28] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science*, 21:1–25, 1982.

[29] B. Manvel. Reconstruction of unicyclic graphs. In *Proof Techniques in Graph Theory*, pages 103–107. Academic Press, New York, 1969.

[30] B. Manvel. Reconstruction of maximal outerplanar graphs. *Discrete Math.*, 2:269–278, 1972.

[31] B. Manvel and J.M. Weinstein. Nearly acyclic graphs are reconstructible. *J. Graph Theory*, 2:25–39, 1978.

[32] S. Micali and V.V. Vazirani. An $O(V^{1/2}E)$ algorithm for finding maximum matching in general graphs. In *21st Annual Symp. on Found. Comp. Sci.*, pages 17–27, New York, 1980.

[33] C. St. J.A. Nash-Williams. The reconstruction problem. In *Selected Topics in Graph Theory*, pages 205–236. Academic Press, 1978.

[34] U. Schöning. A low and high hierarchy in NP. *J. Comput. System Sci.*, 27:14–28, 1983.

[35] U. Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. System Sci.*, 37:312–323, 1988.

[36] A. Selman. Polynomial time enumeration reducibility. *SIAM Journal on Computing*, 7(4):440–457, 1978.

[37] S.M. Ulam. *A Collection of Mathematical Problems.* Interscience Publishers, New York, 1960.

[38] M. von Rimscha. Reconstructibility and perfect graphs. *Discrete Math.*, 47:79–90, 1983.

[39] K. Wagner and G. Wechsung. *Computational Complexity.* Deutscher Verlag der Wissenschaften, Berlin, 1985.

[40] A.T. White and L.W. Beineke. Topological graph theory. In *Selected Topics in Graph Theory, Vol. 1*, pages 15–49. New York, 1978.

[41] P. Young. Juris Hartmanis: Fundamental contributions to isomorphism problems. In A. Selman, editor, *Complexity Theory Retrospective*, pages 28–58. Springer-Verlag, 1990.