Math. Systems Theory 27, 257–273 (1994)

Mathematical Systems Theory

On the Complexity of Graph Reconstruction*

Dieter Kratsch[1] and Lane A. Hemaspaandra[2]

[1] Fakultät für Mathematik und Informatik,
Friedrich-Schiller-Universität,
07740 Jena, Germany

[2] Department of Computer Science,
University of Rochester,
Rochester, NY 14627, USA

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

1. GRAPH ISOMORPHISM $\leq_m^l$ LEGITIMATE DECK, and
2. GRAPH ISOMORPHISM $\equiv_{iso}^l$ 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 for graphs of bounded degree, partial k-trees for any fixed k, and graphs of bounded genus, in particular for planar graphs.

1. Introduction

Harary's survey of the reconstruction conjecture recounts the origins of the problem [Ha3]:

---

* The work of L. A. Hemaspaandra was supported in part by the National Science Foundation under Grants CCR-8809174/CCR-8996198, CCR-8967604, and NSF-INT-9116781/JSPS-ENGR-207. Work done in part while visiting Friedrich-Schiller-Universität and Universität Ulm.

> The author first heard of this fascinating problem when Kelly [Ke] proved the theorem for trees in 1957. This result was obtained in Kelly's doctoral dissertation which was written under Ulam, who published [Ul] 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 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 [GH], [Ha3], [BH2], [NW]. 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* [Ke]. Since that time, many graph classes have been shown to be reconstructible. Among such classes are: disconnected graphs [Ha1], regular graphs [NW], separable graphs with no pendant vertex [Bon], maximal outerplanar graphs [Ma2], outerplanar graphs [Gi1], unicyclic graphs [Ma1], (nontrivial) Cartesian product graphs [Do], squares of trees [Gu], unit interval graphs, and threshold graphs [Ri] (see [Ha1], [GM], [Ch], [CKS], [MW], [La1], and [La2] for further discussion of reconstructibility of graph classes).

In 1964 Harary [Ha1] 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 [NW]).

In this paper we are concerned with *complexity-theoretic* aspects of reconstruction. This line of inquiry springs from a question of Nash-Williams [NW]: 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 (pairwise nonisomorphic) 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: $\{(G_1, G_2) | G_1$ 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.

GI is known to belong to the class coAM; that is, there is an interactive proof protocol for graph nonisomorphism ([GMW] plus the IP = AM result of [GS]; a simple alternate proof that GI belongs to coAM can be found in [Sc2]). Since the second level of Schöning's low hierarchy [Sc1] contains NP $\cap$ coAM, it follows that graph isomorphism is low [Sc2]. Consequently, the polynomial-time hierarchy would collapse if GI were in the high hierarchy [Sc2], and in particular if GI were NP-complete [GS], [BHZ], [GMW]. 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 graphs of bounded genus [Mi], [FM], graphs of bounded degree [Lu], graphs with bounded eigenvalue multiplicity [BGM], partial k-trees for fixed k [Bod1], and some special classes of perfect graphs, e.g., directed path graphs [Di], permutation, series-parallel, and grid graphs (see [Jo1]).

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, bipartite, and line graphs (see [Jo1]).

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. $\delta(G) = \min\{deg_G(v) | v \in V(G)\}$ denotes the minimum degree of G. For graphs G and H, $G \cup H$ 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 $E(G) - \{\{v, x\} | \{v, x\} \in E(G)\}$; 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, $V = \{v_1, v_2, \ldots, v_n\}$ we call the multiset $\langle G - v_1, G - v_2, \ldots, G - v_n\rangle$ the deck[2] of G, denoted $deck(G)$. We call $G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$, a preimage of the deck $\langle G_1, G_2, \ldots, G_n\rangle$ if there is a permutation $\pi \in S_n$ such that $G - v_i$ is isomorphic to $G_{\pi(i)}$ for every $i \in \{1, 2, \ldots, n\}$ [KST2]. If G is a preimage of the deck $\langle G_1, G_2, \ldots, G_n\rangle$, we say $deck(G) = \langle G_1, G_2, \ldots, G_n\rangle$.

---

[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.).

[2] We say *the* deck, as at this point we are viewing each graph as an abstract object, as opposed to being a particular representation of such an object. The following definition of preimage addresses the issue of obscuring representation.

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 a preimage G of $\langle G_1, G_2, \dots, G_n \rangle$ exist?

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.

Preimage Counting (abbreviated PCou)
Instance: Collection of graphs $G_1, G_2, \dots, G_n$.
Task: Count the number of (nonisomorphic) preimages of the deck $\langle G_1, G_2, \dots, G_n \rangle$.

Clearly, these problems are connected to the reconstruction conjecture. Note that if the reconstruction conjecture holds, then for every $\langle G_1, G_2, \dots, G_n \rangle$ it holds that $\mathrm{PCou}(\langle G_1, G_2, \dots, G_n \rangle) = 1$ if and only if $\langle G_1, G_2, \dots, G_n \rangle \in \mathrm{LD}$.

For standard graph-theoretic notions not defined here we refer to [Ha2], [Go] and [Be]. For definitions of graph classes we refer to [Go] and [Jo1]. 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 [HU], [WW], [Jo2], and for definitions of complexity theoretic notions such as NP and NP-completeness [GJ], reductions [LLS], [LL], and isomorphism [Yo]. In particular, we use $\leq_m^p$ to denote polynomial-time many–one reductions, $\leq_m^l$ to denote logspace many–one reductions, $\equiv_m^p$ to denote polynomial-time many–one equivalence, and $\equiv_{iso}^l$ 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. The main idea of the proof is that "'multiset of n graphs on $n-1$ vertices'-isomorphism" many–one polynomial-time reduces to GI.

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 $(G; \langle G_1, \ldots, G_n \rangle) \in \text{DC}$ iff $(G', G'') \in \text{GI}$. 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 (Figure 1). 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:

$$V(G') = \bigcup_{i=1}^{n} V(G_i) \cup \{c'\} \cup \{u'_1, u'_2, \ldots, u'_n\},$$

$$V(G'') = \bigcup_{i=1}^{n} V(H_i) \cup \{c''\} \cup \{u''_1, u''_2, \ldots, u''_n\},$$

$$E(G') = \bigcup_{i=1}^{n} E(G_i) \cup \bigcup_{i=1}^{n} \{\{u'_i, v\} \mid v \in V(G_i)\} \cup \bigcup_{i=1}^{n} \{\{c', u'_i\}\},$$

$$E(G'') = \bigcup_{i=1}^{n} E(H_i) \cup \bigcup_{i=1}^{n} \{\{u''_i, v\} \mid v \in V(H_i)\} \cup \bigcup_{i=1}^{n} \{\{c'', u''_i\}\}.$$

Clearly, the transformation is polynomial. Thus we must show

$$(G; \langle G_1, G_2, \ldots, G_n \rangle) \in \text{DC} \Leftrightarrow (G', G'') \in \text{GI}.$$

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 c' and c'' are the only vertices of degree n with n neighbors of degree n in G' and G'', respectively, we have $I(c') = c''$. Since $\{u'_1, u'_2, \ldots, u'_n\}$ and $\{u''_1, u''_2, \ldots, u''_n\}$ are, respectively, the only neighbors of c' and c'' 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 $(G; \langle G_1, G_2, \ldots, G_n \rangle) \in \text{DC}$. □

The original version of this paper showed that GI $\leq_m^p$ DC, if the graph reconstruction conjecture holds. Using the technique of our proof of Theorem 4.1, Köbler *et al.* have, via the following proof, removed the assumption regarding the graph reconstruction conjecture [KST1].

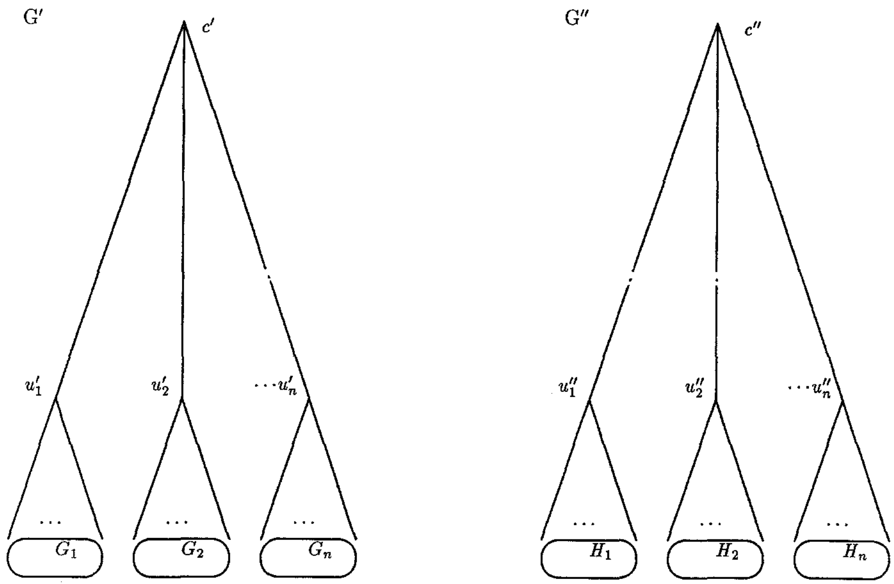

Fig. 1. The graphs G' and G''.

Lemma 3.2 [KST1]. GI $\leq_m^p$ DC.

Proof. Without loss of generality, we may restrict graph isomorphism to connected graphs, since GI restricted to connected graphs is isomorphism-complete. Let (G, H) be an input for GI. We use the following notation: $G + i$ is the graph G with i additional isolated vertices. We show that $(G, H) \in \mathrm{GI}$ if and only if $(G + 1; \langle H \rangle \cup \langle G_i + 1 \mid G_i \in deck(G) \rangle) \in \mathrm{DC}$.

Suppose G and H are isomorphic. Then

$$deck(G + 1) = \langle H \rangle \cup \langle G_i + 1 \mid G_i \in deck(G) \rangle.$$

Suppose $(G + 1; \langle H \rangle \cup \langle G_i + 1 \mid G_i \in deck(G) \rangle) \in \mathrm{DC}$. Since G is connected, there is exactly one card in the deck of $G + 1$ with a connected graph, namely, G. H is the only connected graph in the given deck, hence G and H are isomorphic. Since $(G + 1; \langle H \rangle \cup \langle G_i + 1 \mid G_i \in deck(G) \rangle)$ can be computed in polynomial time, the lemma follows. □

As a consequence of the above lemmas:

Theorem 3.3. DC $\equiv_m^p$ GI.

4. The Complexity of Legitimate Deck

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

Theorem 4.1. GI $\leq_m^p$ LD.

Proof. 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. Again, $G + i$ is the graph G with i additional isolated vertices. We show that $(G, H) \in \mathrm{GI}$ iff $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \mid G_i \in deck(G)\rangle$ is a legitimate deck.

Suppose G and H are isomorphic. Then a preimage of $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \mid G_i \in deck(G)\rangle$ is $G + 2$ and the deck is legitimate. Suppose $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \mid G_i \in deck(G)\rangle \in \mathrm{LD}$. Let a preimage be the graph G'. Then we have

$$
\begin{aligned}
|E(G')| &= \frac{1}{|V(G')| - 2} \cdot \sum_{G_i' \in deck(G')} |E(G_i')| \\
&= \frac{1}{|V(G)|} \cdot \left(|E(G)| + |E(H)| + \sum_{i=1}^{n} |E(G_i)| \right) \\
&= \frac{1}{n} \cdot \left(2 \cdot |E(G)| + \sum_{i=1}^{n} |E(G_i)| \right) \\
&= \frac{1}{n} \cdot (2 \cdot |E(G)| + (n-2) \cdot |E(G)|) \\
&= |E(G)|.
\end{aligned}
$$

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 $K + 2$ for some graph K. Then K has no isolated vertex. Therefore, exactly two cards in the deck of $G' = K + 2$ 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 $\langle G + 1\rangle \cup \langle H + 1\rangle \cup \langle G_i + 2 \mid G_i \in deck(G)\rangle$ 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.

Finally, we mention an exciting recent result of Köbler *et al.* They have shown that PCou is almost in the function class GapP [FFK]: there is a polynomial-time function f and a GapP function g such that g maps from decks to $f(n)$ times the number of (nonisomorphic) preimages of the given deck [KST2]. Hence:

Theorem 4.3 [KST2]. *If the Reconstruction Conjecture holds, then* LD *is in the class* LWPP [FFK] *and therefore is low for* $C_=P$ [Wa], [Si] *and* PP [Gi2], [Si] (*that is,* $C_=P^{LD} = C_=P$ *and* $PP^{LD} = PP$).

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 $\leq_m^l$ GI.
2. GI $\leq_m^l$ DC.
3. GI $\leq_m^l$ LD.
4. DC $\leq_m^l$ LD.

We wish to strengthen parts 1 and 2 above, by proving that DC and GI are logspace isomorphic. This will show that DC and GI are essentially the same problem under different naming schemes.

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

Lemma 5.2 [Ha4]. *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 [Ha4]. 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 [Ha4]. *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 a one-to-one logspace reduction f' of A to B exists such that:

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

Theorem 5.5 [Ha4]. *Let the set A be logspace reducible to B and let 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 DC fulfill the conditions of Theorem 5.5. For the case of polynomial-time padding functions, the paddability of GI has already been proven by Booth [Boo]; however, for completeness, and to cover the logspace case explicitly, we includes proofs.

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 $\hat{G}_y$ (Figure 2) as follows (keeping in mind that $n = |V(G)|$):

$$V(\hat{G}_y) = \{c_i | 1 \le i \le n + 5\} \cup \{d_i | 1 \le i \le r + 1\} \cup \{e_i | y_i = 1,\ 1 \le i \le r\},$$

$$E(\hat{G}_y) = \{\{c_i, c_j\} | 1 \le i < j \le n + 5\} \cup \{\{c_1, d_1\}\} \cup \{\{d_i, d_{i+1}\} | 1 \le i \le r\} \cup \{\{d_i, e_i\} | y_i = 1,\ 1 \le i \le r\}.$$

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

$$S_{\mathrm{DC}}((G; \langle G_1, G_2, \ldots, G_n\rangle), y) = (G \cup \hat{G}_y; \langle G_1 \cup \hat{G}_y\rangle \cup \langle G_2 \cup \hat{G}_y\rangle \cup \cdots \cup \langle G_n \cup \hat{G}_y\rangle \cup \langle G \cup (\hat{G}_y - v) | v \in V(\hat{G}_y)\rangle).$$

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

1. Check that H has a connected component, say F_2, with $|V(F_1)| < |V(F_2)|$, where $V(F_1)$ is $V(H) - V(F_2)$.
2. Check that F_2 has the shape of $(\hat{F}_1)_y$ (the notation here is analogous to that of the definition of $\hat{G}_y$) for some y, and determine the bits of y by scanning the "caterpillar" added to $K_{|V(F_1)|+5}$.

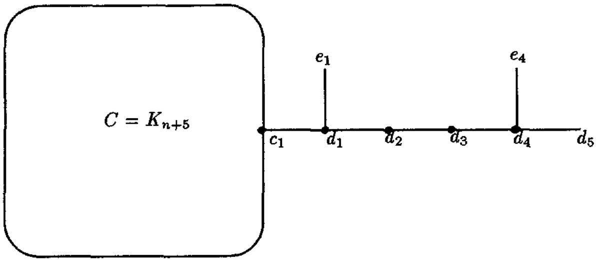

Fig. 2. The graph $\hat{G}_y$, where $y = 1001$.

(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 $\hat{G}_y$, S_{DC} and D_{DC} are logspace computable and satisfy Lemma 5.2. $\square$

The following two lemmas are logspace analogs of the polynomial-time padding result of Booth [Boo], and are included for completeness.

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

Proof. The construction is similar to that of Lemma 5.6:

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

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

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. $\square$

Theorem 5.9. GI *is logspace isomorphic to* DC.

Proof. Follows immediately from Theorem 5.5 and Lemmas 5.6–5.8. $\square$

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 polynomial-time computable (or logspace computable) (S, D) functions would seem to have first actually to 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. Using the terminology that is now standard [BH1], [MY] (but differing from the usage above of "padding function"), LD is a potentially "unpaddable" set—a set A such that $A \times \{0, 1, 2, \ldots\}$ is not polynomial-time isomorphic to A. Indeed, the only known example of a GI-hard NP set that lacks obvious (S, D) functions is a certain interesting but artifical NP-complete set constructed by Joseph and Young [JY]. 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 limit 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 $\mathcal{G}$. For any graph class $\mathcal{G}$, we designate by $\mathrm{PCou}_{\mathcal{G}}$ the variant of PCou where we are interested in counting only the number of (pairwise nonisomorphic) preimages of the deck that happen to be in $\mathcal{G}$. When we say that LD, PCon, and $\mathrm{PCou}_{\mathcal{G}}$ *are solvable in polynomial time when restricted to* $\mathcal{G}$, we mean that there is a polynomial-time algorithm that

(A) will compute correctly (that is, will compute correctly the solution respectively to LD, PCon, or $\mathrm{PCou}_{\mathcal{G}}$) whenever the input graphs $G_1, G_2, \ldots, G_n$ are in $\mathcal{G}$, and furthermore
(B) if not all the input graphs are in $\mathcal{G}$, will print "not all the input graphs are in the class $\mathcal{G}$."

Theorem 6.1. LEGITIMATE DECK, PREIMAGE CONSTRUCTION, *and* PREIMAGE COUNTING$_{\mathcal{G}}$ *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 (polynomial-time recognizable class)* $\mathcal{G}$.
3. *The minimum degree of each graph* $G \in \mathcal{G}$ *is bounded by a constant, k, not depending on* $|V(G)|$; *i.e.*, $\max\{\delta(G) \mid G \in \mathcal{G}\} \leq k$.[3]

Proof. The polynomial-time algorithm is as follows. For input $\langle G_1, G_2, \ldots, G_n \rangle$ we first check to make sure that all the graphs in the deck belong to $\mathcal{G}$; if not, immediately halt and print "not all the graphs in this deck are in class $\mathcal{G}$." Otherwise, all of the graphs in the deck are in $\mathcal{G}$, so each, by condition 3, has minimum degree at most k, and thus if there is any preimage of this deck, it has minimum degree at most $k + 1$. Thus, let us hypothetically assume $\langle G_1, G_2, \ldots, G_n \rangle \in \mathrm{LD}$ and let G be a (hypothetical) preimage of $\langle G_1, \ldots, G_n \rangle$ such that, at least for one card $G_i = G - v_i$, it holds that $deg_G(v_i) \leq k + 1$ (note that for each card in a legitimate deck we can easily compute the degree of the vertex that card is missing, and thus we can easily choose such a card G_i). If no card is missing a vertex of degree at most $k + 1$, then we halt and output "this deck has no preimage in $\mathcal{G}$, and, indeed, has no preimage out of $\mathcal{G}$ either." Henceforward, assume that we have found a card, G_i, that is missing a vertex of degree at most $k + 1$. Hence, we get G from G_i by adding exactly $deg_G(v_i)$ edges from v_i to vertices of G_i. Thus, there are at most $O(n^{k+1})$ possibilities. For each preimage candidate G we have to solve "$(G; \langle G_1, \ldots, G_n \rangle) \in \mathrm{DC}$?" Let $v_1, v_2, \ldots, v_n$ be a labeling of $V(G)$. Then first we check whether $(\forall i\colon 1 \leq i \leq n)\ [G - v_i \in \mathcal{G}]$; by condition 1, this is possible in polynomial time.

[3] Recall that $\delta(G)$ denotes the minimum degree of G.

If for no preimage candidate G does it hold that $(\forall i: 1 \leq i \leq n)\ [G - v_i \in \mathscr{G}]$, then we argue that the purported deck is not legal. Why? We already argued that any preimage (whether in $\mathscr{G}$ or not) must have minimum degree at most $k + 1$, and thus will be one of our preimage candidates. However, we already (if we got to this point) know that all the elements of the deck are in $\mathscr{G}$. Thus, if no preimage candidate has all its point-deleted subgraphs in $\mathscr{G}$, then the deck has no preimage. So halt and print "the purported deck that was input does not have a preimage within the class $\mathscr{G}$, and, indeed, does not have a preimage out of $\mathscr{G}$ either."

Otherwise, for each candidate preimage G for which we found that all its one-vertex-deleted subgraphs were in $\mathscr{G}$, do the following. For every pair $i, j \in \{1, 2, \ldots, n\}$ we check whether $(G - v_i, G_j) \in \mathrm{GI}$. By condition 2, this can be done in polynomial time (note that, by condition 1, the "graph isomorphism on $\mathscr{G}$" algorithm can easily first check to make sure that its inputs *are* in $\mathscr{G}$). We construct a bipartite graph with bipartition $\{G - v_i | 1 \leq i \leq n\}$ and $\{G_j | 1 \leq j \leq n\}$. There is an edge between $G - v_i$ and G_j iff $(G - v_i, G_j) \in \mathrm{GI}$. 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 be checked in time $O(|V|^{1/2} \cdot |E|)$ [HK], [MV]. Thus checking $\langle G; G_1, \ldots, G_n \rangle \in \mathrm{DC}$ for one candidate G can be done in polynomial time. Hence, checking all $O(n^{k+1})$ possible candidates G can be done in polynomial time. Thus, LD and PCon can be solved in polynomial time when restricted to $\mathscr{G}$.

Among all the preimage candidates that passed muster, check which are in $\mathscr{G}$, and, among those, use—in the obvious fashion—the polynomial-time graph isomorphism testing (on $\mathscr{G}$) algorithm to determine how many distinct (i.e., pairwise nonisomorphic) preimages the input deck has in $\mathscr{G}$. Thus $\mathrm{PCou}_{\mathscr{G}}$ can be solved in polynomial time when restricted to $\mathscr{G}$. □

As pointed out by an anonymous referee, adding more edges to a graph never removes a perfect matching, and thus the above proof in fact shows that, for any polynomial-time recognizable minimum-degree-bounded class of graphs $\mathscr{G}$, the legitimate deck problem for $\mathscr{G}$ reduces via a polynomial-time positive-truth-table reduction to GI. (We also note that in appropriate cases, and with appropriate assumptions about the "failure" mode of the algorithms discussed in the theorem, the assumption that the class is recognizable in polynomial time can be in part relaxed.)

Corollary 6.2. LEGITIMATE DECK, PREIMAGE CONSTRUCTION, *and* PREIMAGE COUNTING$_{\mathscr{G}}$ *are solvable in polynomial time when restricted to graphs of bounded degree, to partial k-trees for any fixed k, to graphs of bounded genus, and thus, in particular, to planar graphs.*

Proof. These classes fulfill conditions 1 and 2 of Theorem 6.1 (see [Bod1], [Jo1], and [Lu]). It is well known that any planar graph has a vertex of degree at most 5 and that graphs of bounded genus have a bounded minimum degree also (see p. 216 of [GT]).

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, condition 3 is fulfilled. $\square$

Note that partial k-trees for fixed k (see Definition 2.1) form a quite large class of graphs, containing, e.g., graphs of bounded bandwidth, graphs of bounded cutwidth, Halin graphs, and series-parallel graphs (see [Bod2]). They correspond to graphs with bounded treewidth, defined by Robertson and Seymour [RS].

In fact, for the classes in Corollary 6.2, an even stronger result holds. When we say that PCou *is solvable in polynomial time when restricted to* $\mathscr{G}$, we mean the same thing we meant when we said $\mathrm{PCou}_{\mathscr{G}}$ is solvable in polynomial time when restricted to $\mathscr{G}$, except that now we expect the count of *all* distinct (i.e., pairwise nonisomorphic) preimages of the input deck (and not merely the count of those in $\mathscr{G}$).

Note that the classes of the corollary have a special property.

Lemma 6.3.

1. *For every graph G and integer k: if every graph in the deck of G is of bounded degree, with bound k, then the graph G is of bounded degree, with bound* $k + 1$.
2. *For every graph G and integer k: if every graph in the deck of G is a partial k-tree, then the graph G is a partial* $(k + 1)$*-tree.*
3. *There is a constant* $c(k)$*, depending only on k, such that, for every graph G and integer k: if every graph in the deck of G is of genus at most k, then the genus of G is at most* $c(k)$.

Proof. Statement 1 follows from the simple observation that the degree of a vertex in the one-vertex-deleted subgraph is either equal to the degree in the preimage or exactly one less.

Statement 2 follows immediately from a well-known fact about treewidth: if $G - v$ is a partial k-tree, then G is a partial k-tree or a partial $(k + 1)$-tree.

The interesting part is the proof of statement 3, regarding graphs of bounded genus. If G is disconnected, the statement is clearly true, since the genus of G is the maximum of the genus of all components of G. Let $\delta(G)$ be as defined in Section 2. Let $\delta\delta(k)$ be defined as

$$\max\{\delta(H) \mid H \text{ is a connected graph of orientable genus at most } k\}.$$

Similarly, let $\overline{\delta\delta}(k)$ be defined as

$$\max\{\delta(H) \mid H \text{ is a connected graph of nonorientable genus at most } k\}.$$

We are going to show that there are bounds depending only on k.

We use the following well-known fact (e.g., mentioned on p. 21 of [WB]): If $G = (V, E)$ is a connected graph embeddable in S_k, an orientable surface of genus k, then $|E| \leq 3 \cdot (|V| + 2k - 2)$. If $G = (V, E)$ is a connected graph embeddable in

N_h, a nonorientable surface of genus h, then $|E| \leq 3 \cdot (|V| + h - 2)$. From this we get: If G has orientable genus k, then

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

If G has nonorientable genus h, then

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

Hence, there are bounds linear in k for both $\delta\delta(k)$ and $\overline{\delta\delta}(k)$.

Now let G be a connected graph such that all graphs of its deck have orientable genus at most k. Consequently, the graph G has a vertex of degree at most $\delta\delta(k) + 1$. Therefore, there is a graph $G - v$ of the deck such that G can be constructed from $G - v$ by adding v and at most $\delta\delta(k) + 1$ edges. $G - v$ has oriented genus at most k, hence G has genus at most $k + \delta\delta(k) + 1$, so let this define $c(k)$. The proof for the nonoriented case is analogous. □

From this lemma it is not hard to see that we obtain the following, as the preimage "creeps up only a constant number of notches," and thus remains in a class for which we have polynomial-time recognition and graph isomorphism algorithms.

Corollary 6.4. PREIMAGE COUNTING *is solvable in polynomial time when restricted to graphs of bounded degree, to partial k-trees for any fixed k, to graphs of bounded genus, and thus, in particular, to planar graphs.*

7. 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 [Sc1]. Furthermore, is it possible that LD $\leq_m^p$ GI or at least that LD Turing reduces to GI or that LD strong-nondeterministically [Se], [Lo] reduces to GI?

We noted that LD is not obviously paddable, and used this as evidence that LD is isomorphic to neither GI nor SATISFIABILITY. Can it be proven that LD *is* paddable? GI and SATISFIABILITY are known to be disjunctively self-reducible. LD is not obviously disjunctively self-reducible, which might also be taken as evidence that it is isomorphic to neither GI nor SATISFIABILITY. Can it in fact be shown that LD *is* disjunctively self-reducible?

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 [NW]. Furthermore, polynomial-time algorithms for LD, PCon, and PCou, when restricted to $\mathscr{G}$, do not imply anything about the reconstructibility of $\mathscr{G}$. In fact, this question is still open for planar graphs, graphs of bounded genus, and partial k-trees for fixed k (even $k = 2$, i.e., series-parallel graphs).

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 Hans Bodlaender, William Gasarch, Johannes Köbler, Detlef Seese, and Gerd Wechsung for enjoyable discussions, to two anonymous referees for many helpful suggestions, and especially to Uwe Schöning and Gerd Wechsung for hosting visits and workshops at which much of this research was performed. We thank Johannes Köbler, Uwe Schöning, and Thomas Thierauf for generously allowing us to include their result as our Lemma 3.2. We are grateful to B. McKay for pointing out to us two important references.

References

[Be] C. Berge. *Graphs.* North-Holland, Amsterdam, 1985.

[BGM] L. Babai, D. Grigoryev, and D. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 310–324, 1982.

[BH1] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6(2):305–322, 1977.

[BH2] J. Bondy and R. Hemminger. Graph reconstruction—a survey. *J. Graph Theory*, 1:227–268. 1977.

[BHZ] R. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inform. Process. Lett.*, 25:127–132, 1987.

[Bod1] H. Bodlaender. Polynomial Algorithms for Chromatic Index and Graph Isomorphism on Partial k-trees. *J. Algorithms*, 11:631–643, 1990.

[Bod2] H. Bodlaender. Dynamic programming of graphs with bounded treewidth. *Proceedings of the 15th International Colloquium on Automata, Languages, and Programming*, pages 105–119. Lecture Notes in Computer Science, Vol. 317. Springer-Verlag, Berlin, 1988.

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

[Boo] K. Booth. Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems. *SIAM J. Comput.*, 7(3):273–279, 1978.

[Ch] P. Chinn. A graph with ρ points and enough distinct $(\rho - 2)$-order subgraphs is reconstructible. In M. Capobianco, J. B. Frechen, and M. Krolik, editors, *Recent Trends in Graph Theory*, pages 71–73. Lecture Notes in Mathematics, Vol. 186. Springer-Verlag, Berlin, 1971.

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

[Di] P. Dietz. Intersection Graph Algorithms. Ph.D. thesis. Computer Science Department, Cornell University, Ithaca, NY, 1984.

[Do] W. Doerfler. Some results on the reconstruction of graphs. *Infinite and Finite Sets*, pages 361–383. Colloquia Mathematica Societatis Janos Bolayi, Vol. 10. North-Holland, Amsterdam, 1975.

[FFK] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Proceedings of the 6th Structure in Complexity Theory Conference*, pages 30–42. IEEE Computer Society Press, New York, 1991.

[FM] I. Filotti and J. Mayer. A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 236–243, April 1980.

[GH] D. Greenwell and R. Hemminger. Reconstructing graphs. In G. Chartrand and S. F. Kapoor, editors, *The Many Facets of Graph Theory*, pages 91–114. Lecture Notes in Mathematics, Vol. 110. Springer-Verlag, Berlin, 1969.

[Gi1] W. Giles. The reconstruction of outerplanar graphs. *J. Combin. Theory Ser. B*, 16:215–226, 1974.

[Gi2] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977.

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

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

[GMW] O. Goldreich, S. Micali, and A. Widgerson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 174–187, April 1986.

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

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

[GT] J. Gross and T. Tucker. *Topological Graph Theory.* Wiley, New York, 1987.

[Gu] S. Gupta. Reconstruction conjecture for square of a tree. In K. M. Koh and H. P. Yap, editors, *Graph Theory*, pages 268–278. Lecture Notes in Mathematics, Vol. 1073. Springer-Verlag, Berlin, 1984.

[Ha1] F. Harary. On the reconstruction of a graph from a collection of subgraphs. In M. Fiedler, editor, *Theory of Graphs and Its Applications*, pages 47–52. Prague, 1964.

[Ha2] F. Harary. *Graph Theory.* Addison-Wesley, Reading, MA, 1969.

[Ha3] F. Harary. A survey of the reconstruction conjecture. In R. A. Bari and F. Harary, editors, *Graphs and Combinatorics*, pages 18–28. Lecture Notes in Mathematics, Vol. 406. Springer-Verlag, Berlin, 1974.

[Ha4] J. Hartmanis. On log-tape isomorphisms of complete sets. *Theoret. Comput. Sci.*, 7:273–286, 1978.

[HK] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[HU] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.

[Jo1] D. Johnson. The NP-completeness column: an ongoing guide. *J. Algorithms*, 6:434–451, 1985.

[Jo2] D. Johnson. A catalog of complexity classes. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Chapter 2, pages 67–161. MIT Press/Elsevier, Cambridge, MA/Amsterdam, 1990.

[JY] 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.

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

[KST1] K. Köbler, U. Schöning, and T. Thierauf. Personal communication, 1991.

[KST2] J. Köbler, U. Schöning, and J. Torán. Graph isomorphism is low for PP. *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, pages 401–411. Lecture Notes in Computer Science, Vol. 577. Springer-Verlag, Berlin, 1992.

[La1] J. Lauri. Proof of Harary's conjecture on reconstruction of trees. *Discrete Math.*, 43:79–90, 1983.

[La2] J. Lauri. Graph reconstruction—some techniques and new problems. *Ars Combin.*, 24(B):35–61, 1987.

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

[LLS] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoret. Comput. Sci.*, 1(2):103–124, 1975.

[Li] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theoret. Comput. Sci.*, 21:1–25, 1982.

[Lu] E. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. System Sci.*, 25:42–65, 1982.

[Ma1] B. Manvel. Reconstruction of unicyclic graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 103–107. Academic Press, New York, 1969.

[Ma2] B. Manvel. Reconstruction of maximal outerplanar graphs. *Discrete Math.*, 2:269–278, 1972.

[Mi] G. Miller. Isomorphism testing for graphs of bounded genus. *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 225–235, 1980.

[MV] S. Micali and V. Vazirani. An $O(V^{1/2}E)$ algorithm for finding maximum matching in general graphs. *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, pages 17–27, New York, 1980.

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

[MY] S. Mahaney and P. Young. Reductions among polynomial isomorphism types. *Theoret. Comput. Sci.*, 39:207–224, 1985.

[NW] C. St. J. A. Nash-Williams. The reconstruction problem. In L. W. Beineke and R. J. Wilson, editors, *Selected Topics in Graph Theory*, pages 205–236. Academic Press, New York, 1978.

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

[RS] N. Robertson and P. D. Seymour. Graph minors—a survey. In I. Anderson, editor, *Surveys in Combinatorics 1985: Invited Papers for the Tenth British Combinatorial Conference*, pages 153–171. Cambridge University Press, Cambridge, 1985.

[Sc1] U. Schöning. A low and high hierarchy in NP. *J. Comput. System Sci.*, 27:14–28, 1983.

[Sc2] U. Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. System Sci.*, 37:312–323, 1988.

[Se] A. Selman. Polynomial time enumeration reducibility. *SIAM J. Comput.*, 7(4):440–457, 1978.

[Si] J. Simon. On Some Central Problems in Computational Complexity. Ph.D. thesis, Cornell University, Ithaca, NY, January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.

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

[Wa] K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Inform.*, 23:325–356, 1986.

[WB] A. White and L. Beineke. Topological graph theory. In L. W. Beineke and R. J. Wilson, editors, *Selected Topics in Graph Theory*, pages 15–49. Academic Press, New York, 1978.

[WW] K. Wagner and G. Wechsung. *Computational Complexity.* Mathematics and Its Applications. Reidel, Dordrecht, 1986.

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

Received March 24, 1992 and in final form August 11, 1992.

Note added in proof. Recently, we learned that the result stated here as Theorem 4.1 has appeared in an earlier paper [A. Mansfield, The relationship between the computational complexities of the legitimate deck and the isomorphism problems, *Quart. J. Oxford Ser.* (2), 33:345–347, 1982], which thus deserves credit for that result. We also note that A. Mansfield [cited above] and F. Harary, M. Plantholt, and R. Statman [The graph isomorphism problem is polynomially equivalent to the legitimate deck problem for regular graphs, *Caribbean J. Math.*, 1:15–23, 1982] have shown that for the case of regular graphs LD and GI are polynomially equivalent. However, notice that this does not imply our more general result (Theorem 5.9). In fact, Theorem 5.9 resolves, in the affirmative and in much stronger form, the conjecture of F. Harary *et al.* [cited above] that LD on general graphs is polynomially equivalent to LD on regular graphs and to GI on regular graphs.