Stud. Univ. Babeş-Bolyai Math. 59(2014), No. 4, 443–455

Reconstructing graphs from a deck of all distinct cards

Miklós Bartha and Amitesh S. Shuva

Abstract. The graph reconstruction conjecture is looked at from a new perspective. Given a graph G, three equivalence relations are considered on $V(G)$: card equivalence, automorphism equivalence, and the equivalence of having the same behavior. A structural characterization of card equivalence in terms of automorphism equivalence is worked out. Relative degree-sequences of subgraphs of G are introduced, and a new conjecture aiming at the reconstruction of G from the multiset of relative degree-sequences of its induced subgraphs is formulated. Finally, it is shown that graphs having a deck free from duplicate cards are reconstructible.

Mathematics Subject Classification (2010): 05C60, 18D10.

Keywords: Graph reconstruction, card equivalence, automorphism equivalence, relative degree sequences, traced monoidal categories.

1. Introduction

For a graph G and vertex $v \in V(G)$, $G - v$ is the graph obtained from G by deleting the vertex v and its incident edges. We call $G - v$ a vertex-deleted subgraph of G, or the card associated with vertex v in G. We do not distinguish between isomorphic cards, though. The multiset of cards collected from G in this way is called the deck of G, denoted $D(G)$.

Perhaps the most well-known unsolved problem of graph theory asks whether an arbitrary graph G having at least three vertices can be reconstructed in a unique way (up to isomorphism) from its deck. The likely positive answer to this question is commonly known as the Reconstruction Conjecture (R.C., for short), and it was formulated by Kelly and Ulam as early as 1942. Ever since its inception, this problem has remained a mystery. Trying to solve it is similar to conducting a criminal investigation. There is a suspect, the graph G, who leaves plenty of evidence (i.e., the deck $D(G)$) on the crime scene. Yet, no brilliant detective has been able to track

This paper was presented at the 10th Joint Conference on Mathematics and Computer Science (MaCS 2014), May 21-25, 2014, Cluj-Napoca, Romania.

down the suspect for over 70 years, and the number of works on the case is rapidly decreasing year by year. The reconstruction problem was, however, very popular in the past. According to [15], more than 300 research papers had been published on graph reconstruction between 1950 and 2004.

One of the last true champions of graph reconstruction was F. Harary. He suggested a natural analogue [7] of the R.C., which says that every graph having at least four vertices is uniquely reconstructible from the deck of its edge-deleted subgraphs. Others have come up with similar conjectures for directed graphs, cf. [14, 16], and have obtained partial results proving or disproving them. The reader is referred to [8] and [13] for two excellent surveys on graph reconstruction.

In this paper we propose an original new approach to the study of the reconstruction problem. This approach is structural, rather than combinatorial. It is deeply rooted in algebra and category theory, despite the fact that the proofs of our present results are completely elementary. The results themselves, however, have been distilled from an entirely independent study focusing on the completeness of the traced monoidal category axioms [1, 10] in different well-known mathematical structures satisfying these axioms. We shall elaborate on this study to some extent in Section 4.

2. Definitions, and some easily recoverable data

Let G be a graph having at least three vertices, fixed for the rest of the paper. As usual, $V(G)$ and $E(G)$ will denote the set of vertices and edges of G, respectively. We assume that G is *simple* in the sense that it does not contain loops or multiple edges. In general, we rely on the terminology of [12] to deal with graphs.

Two vertices $u, v \in V(G)$ are called *hypomorphic* or *card-equivalent* (c-equivalent, for short) if the card associated with u is identical with the one associated with v, i.e., $G-u \cong G-v$. (Remember that we do not distinguish between isomorphic cards.) Clearly, c-equivalence is an equivalence relation on $V(G)$. Two graphs G and H are hypomorphic if $D(G)$ and $D(H)$ are identical as multisets, that is, each card appears in $D(G)$ and $D(H)$ the same number of times. (Recall that $D(G)$ denotes the deck of G.) If G and H are hypomorphic, then a *hypomorphism* of G onto H is a bijection $\phi : V(G) \to V(H)$ such that $G - v \cong H - \phi(v)$ holds for every $v \in V(G)$. A *reconstruction* of G is a graph G' such that G and G' are hypomorphic, or, equivalently, there exists a hypomorphism of G onto G'. Using this terminology, the R.C. simply says that two graphs G and H are hypomorphic iff they are isomorphic. In other words, all reconstructions of G are isomorphic (to G, of course). Clearly, every isomorphism of G onto H is a hypomorphism, but the converse is not true, even if the R.C. holds.

Graph G is called *card-minimal* if $D(G)$ is a set, that is, each card is unique in $D(G)$. Our aim in this paper is to show that the R.C. holds true for all card-minimal graphs. (Note that any graph on two vertices has two identical cards.) One might think that this result is trivial, since there is a unique hypomorphism between any two hypomorphic card-minimal graphs G and H. While this is certainly true, we have no direct information on $E(G)$ and $E(H)$, therefore the given unique hypomorphism may not be an isomorphism. Reconstructing G from $D(G)$ is still a very complex issue for such graphs. As we shall see, any duplication of cards in $D(G)$ indicates a kind of symmetry in the internal structure of G. Consequently, the class of card-minimal graphs is really large. Our result is therefore in accordance with the observation in [6] saying that the probability that a randomly chosen graph on n vertices is not reconstructible goes to 0 as n goes to infinity.

In general, it is trivial that $|V(G)|$, the number of vertices of G, is recoverable from $D(G)$. It is still easy to see that $|E(G)|$ is also recoverable. Indeed, add up the numbers of edges appearing on the cards of $D(G)$, and observe that this sum is equal to

$$(|V(G)| - 2) \cdot |E(G)|.$$

See [13, Theorem 2.1] for the details of this simple combinatorial argument.

Once $|E(G)|$ is given, calculating the degree $d(v)$ of vertex v for card $G - v$ is straightforward:

$$d(v) = |E(G)| - |E(G - v)|.$$

Clearly, the degree of any vertex c-equivalent with v is the same as that of v. We thus have managed to recover the degree-sequence of G from $D(G)$. Recall that the *degree-sequence* of G is the sequence of degrees of G's vertices in a non-decreasing order.

A similar combinatorial argument leads to the following result, known as Kelly's Lemma [11], see also [13, Theorem 2.4].

Proposition 2.1. *For any graph Q, let $s_Q(G)$ denote the number of subgraphs of G isomorphic to Q. Then $s_Q(G) = s_Q(H)$ whenever G and H are hypomorphic and $|V(Q)| < |V(G)|$.*

Nash-Williams [13] has also shown that the so-called degree-sequence sequence of G is recoverable from $D(G)$. Essentially this means that, not only $d(v)$ can be read from the card $G - v$ as above, but also the degrees of the neighbors of v are recoverable in this way. We shall reformulate Nash-Williams' proof in Section 4 in terms of relative degree-sequences. A natural question to ask at this point is whether the degrees of the neighbors of the neighbors of v are also recoverable, and so on, moving away further and further from vertex v. This question is already a lot more difficult to answer, mainly because the desired degrees or degree-sequences are no longer c-equivalence invariant. In other words, the answer depends on the representant vertex v chosen for card $G - v$.

3. Characterizing card equivalence

The simple results discussed in Section 2 are of a strictly combinatorial nature, and they do not even touch on the structural properties of card equivalence. In this section we present a real structural characterization of c-equivalence, which is our first main result. In this characterization, card equivalence is compared to two other important equivalence relations on $V(G)$, namely automorphism equivalence and the equivalence of having the same behavior. Card equivalence will be denoted by $\sim_c$.

Definition 3.1. Two vertices $u, v \in V(G)$ are *automorphism-equivalent* (a-equivalent, for short) if there exists an automorphism of G taking u to v.

Automorphism equivalence will be denoted by $\sim_a$. It is obvious that $\sim_a$ is an equivalence relation, but its relationship to $\sim_c$ is not clear for the first sight.

Example 3.2. Let G be the graph in Fig. 1a, and consider the vertices u_1, u_2, u_3 in G. It is easy to see that $u_i \sim_c u_j$ and $u_i \sim_a u_j$ both hold for any $1 \leq i, j \leq 3$.

In general, it is clear by the definitions that $\sim_a \subseteq \sim_c$. Example 3.3 below shows, however, that $\sim_c \not\subseteq \sim_a$.

Example 3.3. Let G be the graph of Fig. 2, and consider again the vertices u_1, u_2, u_3. As it turns out, $u_1 \sim_c u_3$, but $u_1 \not\sim_a u_3$. Furthermore, G has no automorphisms other than the identity.

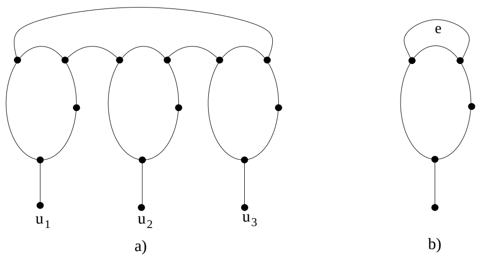

Figure 1. The graph of Example 3.2.

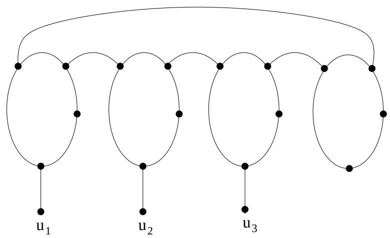

Figure 2. The graph of Example 3.3.

The reader familiar with flowchart schemes and their behaviors [1, 2, 4, 5] will notice that the graphs in Figures 1a and 2 have been inspired by appropriate flowcharts. To recover these flowcharts, make each edge bidirectional in the graphs and supply the degrees with appropriate input-output port distinctions at each vertex. The resulting flowcharts will have no entry or exit vertices, though. Also, no two lines (edges) will be joined at any input or output port. The characteristic feature of such connected "injective" flowcharts is that their proper automorphisms do not have fixed-points. The automorphisms themselves can be neatly characterized by Ésik's *commutativity axioms* [5, 3] for iteration theories. Regarding the graph G in Fig. 1a this means that G can be constructed by taking three copies of the *minimal* graph (scheme) M – shown in Fig. 1b as a multigraph – and turn the edge $e \in E(M)$ into a sequence of edges running through the three copies of $M-e$ in an appropriate way, following a cyclic permutation. This is of course a very simplistic interpretation of the otherwise truly complex commutativity axioms, but it is right to the point. On the other hand, graphs that are not scheme-like, e.g. the simple graph in Fig. 3, do have proper automorphisms with fixed-points, and the concept of minimal graph is meaningless for them.

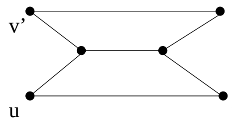

Figure 3. A non-scheme-like graph

Yet another important equivalence relation on $V(G)$ closely related to $\sim_a$ and $\sim_c$ is that of having the same behavior. The reader is referred again to [1, 4, 5] for the original definition of this concept in flowchart schemes.

Definition 3.4. The relation $\sim$ of *having the same behavior* is defined on $V(G)$ as the largest equivalence having the following two properties.

1. If $u \sim v$, then $d(u) = d(v)$.
2. If $u \sim v$ and $\{u_1, \ldots u_k\}$ ($\{v_1 \ldots v_k\}$) is the set of vertices adjacent to u (respectively, v), then the multiset of $\sim$-equivalence groups defined by the set of representants $\{u_1, \ldots u_k\}$ is the same as the one determined by $\{v_1 \ldots v_k\}$.

It is easy to see that two vertices u and v have the same behavior iff G unfolds to isomorphic infinite rooted trees starting from u and v. For example, any two vertices of a regular graph have the same behavior.

Clearly, $\sim_a \subseteq \sim$, but $\sim \not\subseteq \sim_a$. Indeed, not every two vertices of a regular graph are a-equivalent in general. On the other hand, $\sim_c$ is not comparable with $\sim$. The regular graph counterexample shows that $\sim \not\subseteq \sim_c$, and vertices u_1, u_3 in the graph of Fig. 2 demonstrate that $\sim_c \not\subseteq \sim$.

The practical importance of the equivalence $\sim$ is that it is computable in polynomial time. The algorithm to isolate the equivalence groups of $\sim$ is completely analogous to Hopcroft's [9] well-known algorithm for minimizing finite state automata. Even though $\sim_a$ is a lot more costly to compute because of the isomorphism check involved, it still helps to know that $\sim_a$ is a refinement of $\sim$.

The above comparison with the relations $\sim$ and $\sim_a$ shows that $\sim_c$ is rather inconvenient to deal with in a direct way. We need to find a characterization of $\sim_c$ that brings it in line with the much better structured equivalence $\sim_a$. The basis of this characterization is the following lemma.

Lemma 3.5. *Let u and v be two distinct vertices of G. Then $u \sim_c v$ iff there exists a sequence of vertices $x_0, x_1, \ldots, x_n$ ($n \geq 1$) in G satisfying the conditions (i) and (ii) below.*

(i) *$x_0 = v$ and $x_n = u$;*

(ii) *there exists an isomorphism ϕ of $G-u$ onto $G-v$ such that $\phi(x_i) = x_{i+1}$ for every $0 \leq i < n$.*

Proof. Notice first that the graphs $G - u$ and $G - v$ are not separated in the lemma, they both use the vertices of the common supergraph G. The lemma therefore establishes a link between two c-equivalent vertices u and v in G through a sequence of (necessarily distinct) vertices $x_1, \ldots, x_{n-1}$ in $G - u - v$. These vertices, however, need not be c-equivalent with u or each other in G. For example, in the graph of Fig. 2, if $v = u_1$ and $u = u_3$, then $n = 2$, $x_1 = u_2$, and ϕ can be derived from the automorphism of $G - \{u_1, u_2, u_3\}$ that determines a cyclic permutation of the four small cycles of G from left to right. Clearly, $u_1 \not\sim_c u_2$.

Sufficiency of condition (ii) alone for having $u \sim_c v$ is trivial. Assuming that $u \sim_c v$, choose an arbitrary isomorphism $\phi : G - u \to G - v$. Let $x_1 = \phi(v)$, $x_2 = \phi(x_1)$, and so on, until $u = x_n = \phi(x_{n-1})$ is reached. Vertex u must indeed be encountered at some point along this line, since ϕ, being an isomorphism, is an injective mapping $V(G) \setminus \{u\} \to V(G) \setminus \{v\}$. Consequently, the vertices $x_1, \ldots, x_{n-1}$ in $V(G) \setminus \{u, v\}$ will all be different until $x_n = u$ stops this necessarily finite sequence. (Mind that $x_{i+1} = \phi(x_i) \neq v$, since v is not a vertex of $G - v$.) The proof is complete. □

Theorem 3.6. *Let u and v be two distinct vertices of G. Then $u \sim_c v$ iff there exists a sequence of pairwise distinct vertices $x_0, x_1, \ldots, x_n$ $(n \geq 1)$ satisfying the following conditions.*

- (i) *$x_0 = v$ and $x_n = u$;*
- (ii) *for $X = \{x_0, x_1, \ldots, x_n\} \subseteq V(G)$ there exists an automorphism ψ of $G - X$ such that:*
  - (iia) *for every $0 \leq i < n$ and vertex $w_i \in V(G - X)$ adjacent to x_i in G (or, equivalently, in $G - u$), the vertex $w_{i+1} = \psi(w_i)$ is also in $V(G - X)$ and is adjacent to x_{i+1} in G (i.e., in $G - v$);*
  - (iib) *for every $0 \leq i < j < n$,*

$$x_i \text{ is adjacent to } x_j \text{ iff } x_{i+1} \text{ is adjacent to } x_{j+1}$$

(in G, of course).

Vertices u and v are a-equivalent iff the assignments $x_i \mapsto x_{i+1}$, $u \mapsto v$ extend the automorphism ψ in (ii) to one of G.

Proof. Intuitively, condition (iia) says that for every $0 \leq i < n$, the neighbors of x_i in $G - X$ are matched up with those of x_{i+1} in $G - X$ by the automorphism ψ. Condition (iib) settles the issue of how the vertices x_i themselves are connected in G. Notice that the question whether u is connected to v is irrelevant. Indeed, it can easily happen that $u \sim_c v$ and $u \sim_c v'$ both hold, while u is adjacent to v but not to v'. See Fig. 3.

The first statement of the theorem is in fact a simple consequence of Lemma 3.5. Regarding sufficiency, if ψ is an automorphism of $G - X$ satisfying (iia) and (iib), then it can be extended to an isomorphism ϕ of $G - u$ onto $G - v$ satisfying (ii) of Lemma 3.5. Thus, $u \sim_c v$. Conversely, if $u \sim_c v$, then the required automorphism ψ can be derived in a unique way from the isomorphism ϕ guaranteed by Lemma 3.5. Notice that the subgraph $G - X$ may turn out to be empty. The second statement of the theorem is obvious. □

At this point the reader may want to have a second look at Examples 3.2 and 3.3, and identify the underlying automorphism ψ in the graphs of Fig. 1 and Fig. 2. One important point is that, given the fact $v \sim_a u$ (and therefore $u \sim_c v$), one must not jump to the conclusion saying that $x_0 = v$ and $x_1 = u$ will do for $X = \{x_0, x_1\}$ in (ii) of Theorem 3.6, and then be taken by surprise that the desired automorphism ψ cannot be located in $G - X$. For example, in the graph G of Fig. 1, if $v = u_1$ and $u = u_2$, then $x_1 = u_3$! Consequently, $X = \{u_1, u_2, u_3\}$, and the automorphism ψ is just the one taking the three small cycles into one another following a cyclic permutation with offset 2 from left to right.

4. Relative degree-sequences

Recall from Section 2 that the degree-sequence of graph G is the sequence of degrees of its vertices in a non-decreasing order. Let Q be a subgraph of G. The degree of a vertex $v \in V(Q)$ relative to G is a pair (r, d), where d (r) is the degree of v in G (respectively, Q). We shall use the notation r^d for the pair (r, d), and say that v has relative degree r out of d. Then the *relative degree-sequence* of Q (with respect to G) is the sequence of relative degrees of its vertices in an order that is non-decreasing regarding the superscripts d and also non-decreasing in r among those degrees that have the same superscript d.

The degree-sequence of G and the relative degree-sequence of Q with respect to G will be denoted by $ds(G)$ and $rds_G(Q)$, respectively. In order to ensure that $ds(G)$ and $rds_G(Q)$ have the same length, we shall include a relative "degree" $\emptyset^d$ in $rds_G(Q)$ for each vertex $v \in V(G) \setminus V(Q)$ with degree d. The "number" $\emptyset$ is treated as 0, but the notation $\emptyset$ will distinguish between a vertex that has been deleted and one that is still present but isolated. This distinction is purely technical, however, because one can easily fill in the $\emptyset^d$ relative degrees in $rds_G(Q)$ once $ds(G)$ is known.

Example 4.1. *Consider the graph G and its subgraph Q in Fig. 4. The degree-sequence of G is* $2, 2, 3, 3$*, while the relative degree sequence of Q with respect to G is* $1^2, 1^2, 1^3, 3^3$.

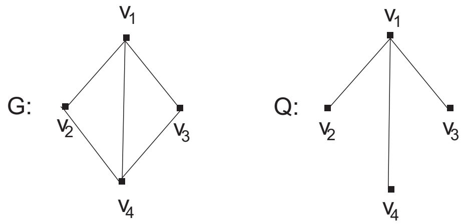

Figure 4. Graph G and its subgraph Q

The following simple combinatorial observation is equivalent to Nash-Williams' result [13, Corollary 3.5] on degree-sequence sequences.

Proposition 4.2. *For every vertex $v \in V(G)$, $rds_G(G - v)$ is recoverable from $D(G)$.*

Proof. We have seen in Section 2 that $d(v)$ and $ds(G)$ are recoverable from $D(G)$. Write the sequence $ds(G-v)$ underneath $ds(G)$ by inserting the "degree" $\emptyset$ in $ds(G-v)$ right under the position of the first occurrence of $d(v)$ in $ds(G)$. For example:

$$\begin{array}{lccccccc}
ds(G): & 2 & 2 & 2 & 3 & 3 & 4 & 4 \\
ds(G-v)): & 1 & 1 & 2 & 2 & 3 & \emptyset & 3 \\
rds_G(G-v): & 1^2 & 1^2 & 2^2 & 2^3 & 3^3 & \emptyset^4 & 3^4
\end{array}$$

Observe that the "true" degrees in $ds(G-v)$ will lag behind those in $ds(G)$, so that the difference between two degrees in aligned positions is at most 1. Therefore it is trivial to fill out the missing superscripts in $ds(G-v)$, so that the resulting sequence becomes $rds_G(G-v)$. □

Proposition 4.2 basically says that, for every card $G-v$, the degrees of the vertices adjacent to v in G are uniquely determined by $ds(G)$ and $ds(G-v)$. Indeed, these are exactly the degrees $r+1$ appearing in $rds_G(G-v)$ as r^{r+1}. Of course, we still have no information about the actual position of v's neighbors in $G-v$.

We immediately generalize Proposition 4.2 to find out the relative degree-sequence of all 2-vertex-deleted subgraphs of G. Notice that, for two distinct vertices $u,v \in V(G)$, the subgraph $G-u-v$ is no longer determined by the cards $G-u$ and $G-v$ in a unique way, since the cards themselves do not uniquely identify the vertices u,v. Moreover, the subgraph $G-u-v$, too, can be isomorphic to other subgraphs $G-u'-v'$ in which u' and v' are associated with some different cards.

Theorem 4.3. *Let u and v be two distinct vertices of G. Given the degree-sequence of the subgraph $G-u-v$, $rds_G(G-u-v)$ is uniquely determined by the data $ds(G)$, $ds(G-u)$, and $ds(G-v)$. Moreover, the question whether u and v are adjacent in G or not turns out from the data $ds(G)$, $ds(G-u-v)$, $d(u)$ and $d(v)$.*

Proof. We use the same alignment argument as in the proof of Proposition 4.2. Write the degree-sequences $ds(G)$, $ds(G-u)$, and $ds(G-v)$ under each other, inserting the $\emptyset$ symbol in the appropriate positions of $ds(G-u)$ and $ds(G-v)$. Furthermore, insert two $\emptyset$'s in $ds(G-u-v)$ aligned with the ones already inserted in $ds(G-u)$ and $ds(G-v)$. If $d(u)=d(v)=d$, then insert two consecutive $\emptyset$'s aligned with the beginning of the block marked by degree d in $ds(G)$. For example:

$$\begin{array}{lccc|cc|cccccc}
ds(G): & 2 & 2 & 2 & 3 & 3 & 4 & 4 & \dots & \dots & \dots & \dots \\
ds(G-u): & 1 & 1 & 2 & 2 & 3 & 4 & 4 & \dots & \emptyset & \dots & \dots \\
ds(G-v): & 1 & 2 & 2 & 2 & 2 & 3 & 4 & \dots & \dots & \emptyset & \dots \\
ds(G-u-v): & 0 & 1 & 1 & 2 & 2 & 3 & 4 & \dots & \emptyset & \emptyset & \dots \\
 & & & \rightarrow & \leftarrow & & & & & & & \\
rds_G(G-u-v): & 0^2 & 1^2 & 2^2 & 1^3 & 2^3 & 3^4 & 4^4 & \dots & \emptyset & \emptyset & \dots
\end{array}$$

Let $n_G(d)$ ($n_{G,Q}(r^d)$) denote the number of occurrences of d (r^d) in $ds(G)$ (respectively, $rds_G(Q)$). Assume, for simplicity, that the smallest degree in G is $d_0 \geq 2$. Then, clearly:

$$n_{G,Q}((d_0-2)^{d_0}) = n_Q(d_0-2).$$

It follows that:

$$n_{G,Q}((d_0-1)^{d_0}) = n_{G-u}(d_0-1) + n_{G-v}(d_0-1) - 2 \cdot n_Q(d_0-2), \quad \text{and}$$

$$n_{G,Q}(d_0^{d_0}) = n_G(d_0) - n_{G,Q}((d_0-2)^{d_0}) - n_{G,Q}((d_0-1)^{d_0}),$$

provided that neither of the degrees $d(u)$ and $d(v)$ equals d_0. If either or both does, then the above calculation changes in a straightforward way regarding the numbers $n_{G,Q}((d_0-1)^{d_0})$ and $n_{G,Q}(d_0^{d_0})$. One can then carry on in the same way, calculating the numbers $n_{G,Q}((d_0-1)^{d_0+1})$, $n_{G,Q}(d_0^{d_0+1})$, $n_{G,Q}((d_0+1)^{d_0+1})$, and so on. Details are left to the reader.

As to the second statement of the theorem, if

$$|E(G)| - |E(G-u-v)| = d(u) + d(v),$$

then u and v are not connected in G, otherwise they are. The numbers $|E(G)|$ and $|E(G-u-v)|$ are determined by $ds(G)$ and $ds(G-u-v)$, respectively. The proof is complete. □

Proposition 4.2 and Theorem 4.3 show that the concept of relative degree-sequence is rather fundamental in the study of graph reconstruction. To provide yet another evidence for this observation, let $Rds(G)$ denote the multiset

$$\{rds_G(Q) | Q \text{ is an induced subgraph of } G\}.$$

Thus, relative degree-sequences of subgraphs count with multiplicity in $Rds(G)$. We put forward the following conjecture, which is very closely related to the R.C..

Conjecture 4.4. *For every graph G, $Rds(G)$ identifies G up to isomorphism.*

Conjecture 4.4 is especially useful for several reasons.

1. It appears to hold for all graphs with no exceptions.
2. It provides a characterization of graph isomorphism, which has been sought for a very long time.
3. Algebraically, if $G = G_1 + G_2$, then

$$Rds(G) = Rds(G_1) \times Rds(G_2). \tag{4.1}$$

In equation 4.1 above, $\times$ stands for concatenation of sets of relative degree-sequences in the formal language sense (taking the quotient of the product by commutativity). In terms of polynomials, we can think of a relative degree r^d as a formal variable. Let X denote the set of all such variables. Then $Rds(G)$ becomes a polynomial of the variables X over the integer ring $\mathbf{Z}$, in which all coefficients are non-negative. (Treat union of multisets as addition in this polynomial.) Let $\mathbf{Z}[X]$ denote the commutative $\mathbf{Z}$-module (in fact algebra) of X-polynomials over $\mathbf{Z}$. (Mind that addition of polynomials is commutative in $\mathbf{Z}[X]$.) Our fundamental observation is that the operation $\times$ in (4.1) translates naturally into product of polynomials in the algebra $\mathbf{Z}[X]$. This product makes the algebra $\mathbf{Z}[X]$ associative and commutative, therefore a commutative ring.

Conjecture 4.4 was the starting point of the present study, and the fundamental observation in the previous paragraph served as a motivation for it. In the language of category theory this observation suggests that the traced monoidal category of graphs (flowchart schemes), in which tensor is disjoint union and trace is feedback (i.e., creating an internal edge by merging two external ones, see [1, 3]) can be embedded in a natural way into the compact closed category of free modules over the commutative algebra (ring) $\mathbf{Z}[X]$, in which tensor and trace are the standard matrix operations. There is a clear analogy in this statement with the *Int* construction, cf. [10], for the "scalar" connection between graphs and polynomials is lifted to the level of traced monoidal and compact closed categories by observing that the given translation of graphs into polynomials is compatible with the trace operation at the higher level.

Naturally enough, Conjecture 4.4 also has an "edge" version, in which $Rds(G)$ is defined as the set of relative degree-sequences of *all* subgraphs of G. This version, too, appears to hold for all graphs G with no exceptions, even for multigraphs as one would expect after the flowchart scheme analogy.

The connection between Conjecture 4.4 and the R.C. is the following. If we could compute $rds(G)$ from $D(G)$, then Conjecture 4.4 would imply the R.C.. As our main result in Section 5 shows, however, computing the whole multiset $Rds(G)$ is far too much work in order to reconstruct G. Therefore this reconstruction argument probably does not hold much water, indicating that Conjecture 4.4 is even tougher than the R.C..

On the other hand, if, given $Rds(G)$, we could isolate $Rds(G - v)$ for each vertex-deleted subgraph of G, then the R.C. would imply Conjecture 4.4 through a straightforward induction argument. Since our concern is eventually Conjecture 4.4, and the construction of the multiset of multisets

$$\{Rds(G - v) | v \in V(G)\}$$

from $Rds(G)$ looks promising, we definitely must prove the R.C. first.

5. The reconstruction of card-minimal graphs

In this section we present our second main result, which aims at the reconstruction of card-minimal graphs. Temporarily, we are going to assume a further technical condition in order to keep the reconstruction simple. Dropping this condition will be the subject of a forthcoming paper. The condition is formally defined as follows.

Definition 5.1. Graph G is *2-card reconstructible* if it is connected, and for every $u, v, x, y \in V(G)$, the isomorphism

$$G - u - v \cong G - x - y$$

implies that u, v, x, and y cannot all be distinct.

To shed some light on the intuition behind Definition 5.1, let G be card-minimal, and Q be an arbitrary graph having $|V(G)| - 2$ vertices. Consider the set C of cards in $D(G)$ in which Q is isomorphic to at least one vertex-deleted subgraph. Construct the graph G_Q which has C as its set of vertices, and any two cards $G - u$, $G - v$ are connected in G_Q iff $G - u - v \cong Q$. (Remember that G is card-minimal, therefore the definition of G_Q is correct.) Then G is 2-card reconstructible iff G_Q is either a triangle or a star graph for every 2-vertex-deleted subgraph Q of G. In other words, if $|C| > 2$, then the following two conditions are met:

1. the subgraph Q occurs $k \geq 2$ times as a vertex-deleted subgraph in some card $G - u \in C$;
2. $|C| = k + 1$ and the cards in C different from $G - u$ all have a single occurrence of Q in them, with the possible exception that $k = 2$ and all the three cards in C have two occurrences of Q in them.

See Fig. 5a for a card-minimal graph G which is, and Fig. 5b for one which is not 2-card reconstructible.

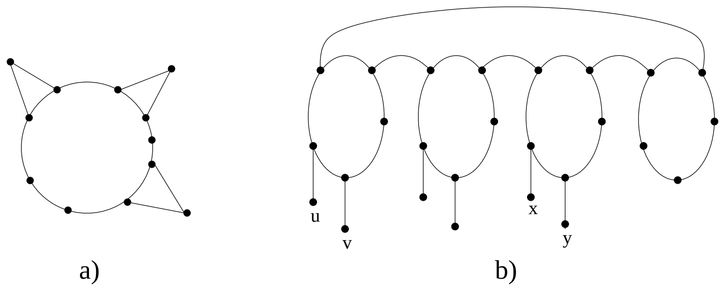

Figure 5. The 2-card reconstructibility condition

Theorem 5.2. *Every card-minimal and 2-card reconstructible graph G is reconstructible.*

Proof. Let Q be an arbitrary graph having $|V(G)| - 2$ vertices, and find the set C of cards in which Q is isomorphic to at least one vertex-deleted subgraph. If $C = \emptyset$, then drop Q as uninteresting. Otherwise C has at least two elements. If there are exactly two cards $G - u$ and $G - v$ in C, then conclude that $Q \cong G - u - v$, and use Theorem 4.3 to decide if u and v are adjacent in G or not. If C has more than two elements, then the condition of 2-card reconstructibility implies that either $|C| = 3$ and each card in C has two subgraphs isomorphic to Q, or there is exactly one card $G - u \in C$ that contains more than one subgraph isomorphic to Q. In the first case $Q \cong G - u - v$ for any pair $G - u$, $G - v$ of distinct cards in C, while in the latter $Q \cong G - u - v$ for all vertices $v \neq u$ such that $G - v \in C$. Furthermore, in this case Q is not isomorphic to any other 2-vertex-deleted subgraph of G. (In other words, $Q \ncong G - u_1 - u_2$, where $G - u_1$ and $G - u_2$ are both in C but $u_i \neq u$ for either $i = 1$ or 2.) Again, use Theorem 4.3 to find out if u is adjacent to v in G, knowing that $Q \cong G - u - v$. It is evident that the above procedure will decide for each pair of cards $G - u$, $G - v$ in $D(G)$ if the vertices u and v are adjacent in G or not. The proof is now complete. $\square$

At this point the reader might have the impression that the condition of 2-card reconstructibility is overly restrictive. In fact it is not, and a fairly simple analysis based on the combination of Proposition 2.1 and Theorem 4.3 shows that whenever

$$G - u - v \cong G - x - y$$

holds for four distinct vertices u, v, x, y, then each possible correspondence of these vertices to appropriate cards in $D(G)$ can be identified in a consistent way. This analysis is technically complicated, however, therefore we do not include it in the present introductory paper.

6. Conclusion

Motivated by an independent research on traced monoidal categories, we have presented a structural analysis of graphs with the aim of being able to reconstruct them from some partial information. The basis of the reconstruction of graph G could either be the classical multiset of G's vertex-deleted subgraphs, or the multiset of relative degree-sequences of all induced subgraphs of G.

We have introduced three equivalence relations on $V(G)$ for the better understanding of the reconstruction problem. Card equivalence is the one directly related to the reconstruction conjecture. Our examples have shown, however, that this equivalence is rather inconvenient to deal with. Automorphism equivalence and having the same behavior have been adopted from the study of flowchart schemes and their behaviors. These relations have a much more transparent structure, and both have turned out to be very closely related to card equivalence. For an evidence, we have worked out a characterization theorem for card equivalence to bring it in line with automorphism equivalence.

With respect to relative degree sequences, we have provided a generalization of an earlier observation by Nash-Williams on the degree-sequence sequence of graphs. As an application of this result we have shown that every card-minimal graph G satisfying a further simple condition is reconstructible from the deck of G. However, the condition of 2-card reconstructibility used in the proof of this result appears to be purely technical, and could be replaced by a thorough analysis of G's 2-vertex-deleted subgraphs on the basis of our characterization theorem for card equivalence.

References

[1] Bartha, M., *A finite axiomatization of flowchart schemes*, Acta Cybernetica, **8**(1987), no. 2, 203–217, `http://www.cs.mun.ca/~bartha/linked/flow.pdf`.

[2] Bartha, M., *An algebraic model of synchronous systems*, Information and Computation **97**(1992), 97–131.

[3] Bartha, M., *The monoidal structure of Turing machines*, Mathematical Structures in Computer Science, **23**(2013), no. 2, 204–246.

[4] Bloom, S.L., Ésik, Z., *Axiomatizing schemes and their behaviors*, J. Comput. System Sci., **31**(1985), 375–393.

[5] Bloom, S.L., Ésik,Z., *Iteration Theories: The Equational Logic of Iterative Processes*, Springer-Verlag, 1993.

[6] Bollobás, B., *Almost every graph has reconstruction number three*, J. Graph Theory, **14**(1990), 1–4.

[7] Harary, F., *On the reconstruction of a graph from a collection of subgraphs*, In: Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963). Publ. House Czechoslovak Acad. Sci., Prague, 1964, 47–52.

[8] Harary, F., *A survey of the reconstruction conjecture*, Lecture Notes in Mathematics, **406**(1974), 18–28.

[9] Hopcroft, J., *An n log n algorithm for minimizing states in a finite automaton*, Theory of machines and computations (Proc. Internat. Sympos., Technion, Haifa, 1971), New York, Academic Press, 1971, 189–196.

[10] Joyal, A., Street, R., Verity, D., *Traced monoidal categories*, Math. Proc. Camb. Phil. Soc., **119**(1996), 447–468.

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

[12] Lovász,L., Plummer, M.D., *Matching Theory*, North Holland, 1986.

[13] Nash-Williams, C.St.J.A., *The Reconstruction Problem*, In: L. Beineke, R.J. Wilson, eds., Selected topics in graph theory, Academic Press, 1978, 205–236.

[14] Ramachandran, S., *On a new digraph reconstruction conjecture*, Journal of Combinatorial Theory, **B31**(1981), no. 2, 143–149,

[15] Ramachandran, S., *Graph Reconstruction–Some New Developments*, AKCE J. Graphs. Combin., **1**(2004), no. 1, 51–61.

[16] Stockmeyer, P.K., *The falsity of the reconstruction conjecture for tournaments*, J. Graph Theory, **1**(1977), 19–25.

Miklós Bartha
Memorial University of Newfoundland
Department of Computer Science
St. John's, NL, A1B 3X5, Canada
e-mail: `bartha@mun.ca`

Amitesh S. Shuva
Memorial University of Newfoundland
Department of Computer Science
St. John's, NL, A1B 3X5, Canada
e-mail: `ss4545@mun.ca`