A COUNTEREXAMPLE TO THE RECONSTRUCTION CONJECTURE FOR LOCALLY FINITE TREES

NATHAN BOWLER, JOSHUA ERDE, PETER HEINIG, FLORIAN LEHNER, MAX PITZ

Abstract. Two graphs G and H are *hypomorphic* if there exists a bijection $\varphi\colon V(G) \to V(H)$ such that $G - v \cong H - \varphi(v)$ for each $v \in V(G)$. A graph G is *reconstructible* if $H \cong G$ for all H hypomorphic to G.

It is well known that not all infinite graphs are reconstructible. However, the Harary-Schwenk-Scott Conjecture from 1972 suggests that all locally finite trees are reconstructible.

In this paper, we construct a counterexample to the Harary-Schwenk-Scott Conjecture. Our example also answers three further questions of Nash-Williams and Halin on the reconstruction of infinite graphs.

1. Introduction

We say that two graphs G and H are *hypomorphic* if there exists a bijection φ between the vertices of G and H such that the induced subgraphs $G - v$ and $H - \varphi(v)$ are isomorphic for each vertex v of G. Any such bijection is called a *hypomorphism*. We say that a graph G is *reconstructible* if $H \cong G$ for every H hypomorphic to G. The following conjecture, attributed to Kelly and Ulam, is perhaps one of the most famous unsolved problems in the theory of graphs.

Conjecture 1.1 (The Reconstruction Conjecture)**.** *Every finite graph with at least three vertices is reconstructible.*

For an overview of results towards the Reconstruction Conjecture for finite graphs see the survey of Bondy and Hemminger [3]. Harary [8] proposed the Reconstruction Conjecture for infinite graphs, however Fisher [6] found a counterexample, which was improved to the following simpler counterexample by Fisher, Graham and Harary [7]: Consider the infinite tree G in which every vertex has countably infinite degree, and the graph H formed by taking two disjoint copies of G, which we will write as $G \oplus G$. For each vertex v of G, the induced subgraph $G - v$ is isomorphic to $G \oplus G \oplus \cdots$, a disjoint union of countably many copies of G, and similarly for each vertex w of H, the induced subgraph $H - w$ is isomorphic to $G \oplus G \oplus \cdots$ as well. Therefore, any bijection from $V(G)$ to $V(H)$ is a hypomorphism, but G and H are clearly not isomorphic. Hence, the tree G is not reconstructible.

These examples, however, contain vertices of infinite degree. Regarding locally finite graphs, Harary, Schwenk and Scott [9] showed that there exists a non-reconstructible locally finite forest. However, they conjectured that the Reconstruction Conjecture should hold for locally finite trees.

2010 *Mathematics Subject Classification.* 05C60, 05C63.

Key words and phrases. Reconstruction conjecture, reconstruction of locally finite trees, Harary-Schwenk-Scott conjecture, extension of partial isomorphisms, promise structure.

Conjecture 1.2 (The Harary-Schwenk-Scott Conjecture)**.** *Every locally finite tree is reconstructible.*

This conjecture has been verified in a number of special cases. Bondy and Hemminger [2] showed that every tree with at least two but a finite number of ends is reconstructible, and Thomassen [13] showed that this also holds for one-ended trees. Andreae [1] proved that also every tree with countably many ends is reconstructible.

A survey of Nash-Williams [11] on the subject of reconstruction problems in infinite graphs gave the following three main open problems in this area, which have remained open until now.

Problem 1.3 (Nash-Williams)**.** *Is every locally finite connected infinite graph reconstructible?*

Problem 1.4 (Nash-Williams)**.** *If two infinite trees are hypomorphic, are they also isomorphic?*

Problem 1.5 (Halin)**.** *If G and H are hypomorphic, do there exist embeddings $G \hookrightarrow H$ and $H \hookrightarrow G$?*

A positive answer to Problem 1.3 or 1.4 would verify the Harary-Schwenk-Scott Conjecture. In this paper we construct a pair of trees which are not only a counterexample to the Harary-Schwenk-Scott Conjecture, but also answer the three questions of Nash-Williams and Halin in the negative. Our counterexample will in fact have bounded maximum degree.

Theorem 1.6. *There are two hypomorphic infinite trees T and S with maximum degree three such that there is no embedding $T \hookrightarrow S$ or $S \hookrightarrow T$.*

The Reconstruction Conjecture has also been considered for general locally finite graphs. Nash-Williams [10] showed that any locally finite graph with at least three, but a finite number of ends is reconstructible, and in [12], he established the same result for two-ended graphs. The following problems, also from [11], remain open:

Problem 1.7 (Nash-Williams)**.** *Is every locally finite graph with exactly one end reconstructible?*

Problem 1.8 (Nash-Williams)**.** *Is every locally finite graph with countably many ends reconstructible?*

In a paper in preparation [4], we will extend the methods developed in the present paper to also construct counterexamples to Problems 1.7 and 1.8.

This paper is organised as follows. In the next section we will give a short, high-level overview of our counterexample to the Harary-Schwenk-Scott Conjecture. In Section 3, we will develop the technical tools necessary for our construction, and in Section 4, we will prove Theorem 1.6.

For standard graph theoretical concepts we follow the notation in [5].

2. Sketch of the construction

In this section we sketch the main ideas of the construction. For the sake of simplicity we only indicate how to ensure that the trees T and S are not isomorphic, rather than that neither embeds into the other.

Our plan is to build the trees T and S recursively, where at each step of the construction we ensure for some vertex v already chosen for T that there is a corresponding vertex w of S with $T - v \cong S - w$, or vice versa. This will ensure that by the end of the construction, the trees we have built are hypomorphic.

More precisely, at step n we will construct subtrees T_n and S_n of our eventual trees, where some of the leaves of these subtrees have been coloured in two colours, say red and blue. We will only further extend the trees from these coloured leaves, and we will extend from leaves of the same colour in the same way.

That is, the plan is that there should be two further rooted trees R and B such that T can be obtained from T_n by attaching copies of R at all red leaves and copies of B at all blue leaves, and S can be obtained from S_n in the same way. At step n, however, we do not yet know what these trees R and B will eventually be.

Nevertheless, we can ensure that the induced subgraphs, $T - v$ and $S - w$, of the vertices we have dealt with so far really will match up. More precisely, by step n we have vertices $x_1, \ldots, x_n$ of T_n and $y_1, \ldots, y_n$ of S_n for which we intend that $T - x_j$ should be isomorphic to $S - y_j$ for each j. We ensure this by arranging that for each j there is an isomorphism from $T_n - x_j$ to $S_n - y_j$ which preserves the colours of the leaves.

The T_n will be nested, and we will take T to be the union of all of them; similarly the S_n will be nested and we take S to be the union of all of them.

There is a trick to ensure that T and S do not end up being isomorphic. First we ensure, for each n, that there is no isomorphism from T_n to S_n. We also ensure that the part of T or S beyond any coloured leaf of T_n or S_n begins with a long non-branching path, longer than any such path appearing in T_n or S_n. Call the length of these long paths k_n.

Suppose now for a contradiction that there is an isomorphism from T to S. Then there must exist some large n such that the isomorphism sends some vertex t of T_n to a vertex s of S_n. However, T_n is the component of T containing t after all non-branching paths of length k_n have been removed[1], and so it must map isomorphically onto the component of S containing s after all non-branching paths of length k_n have been removed, namely onto S_n. However, there is no isomorphism from T_n onto S_n, so we have the desired contradiction.

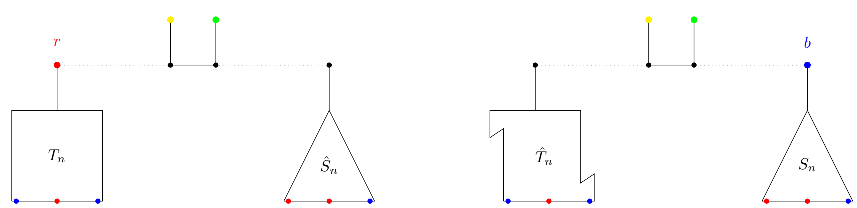


FIGURE 1. A first approximation of T_{n+1} on the left, and S_{n+1} on the right. All dotted lines are non-branching paths of length k_n.

Suppose now that we have already constructed T_n and S_n and wish to construct T_{n+1} and S_{n+1}. Suppose further that we are given a vertex v of T_n for which we wish to find a partner w in S_{n+1} so that $T - v$ and $S - w$ are isomorphic. We begin by building a tree $\hat{T}_n \ncong T_n$ which has some vertex w such that $T_n - v \cong \hat{T}_n - w$.

[1]Here and throughout this section we will omit minor technical details for brevity.

This can be done by taking the components of $T_n - v$ and arranging them suitably around the new vertex w.

We will take S_{n+1} to include S_n and $\hat{T}_n$, with the copies of red and blue leaves in $\hat{T}_n$ also coloured red and blue respectively. As indicated on the right in Figure 1, we add paths of length k_n to some blue leaf b of S_n and to some red leaf r of $\hat{T}_n$ and join these paths at their other endpoints by some edge e_n. We also join two new leaves y and g to the endvertices of e_n. We colour the leaf y yellow and the leaf g green (to avoid confusion with the red and blue leaves from step n, we take the two colours applied to the leaves in step $n+1$ to be yellow and green).

To ensure that $T_{n+1} - v \cong S_{n+1} - w$, we take T_{n+1} to include T_n together with a copy $\hat{S}_n$ of S_n, coloured appropriately and joined up in the same way, as indicated on the left in Figure 1.

The only problem up to this point is that we have not been faithful to our intention of extending in the same way at each red or blue leaf of T_n and S_n. Thus, we now copy the same subgraph appearing beyond r in Fig. 1, including its coloured leaves, onto all the other red leaves of S_n and T_n. Similarly we copy the subgraph appearing beyond the blue leaf b of S_n onto all other blue leaves of S_n and T_n.

At this point, we would have kept our promise of adding the same thing behind every red and blue leaf of T_n and S_n, and hence would have achieved $T_{n+1} - x_j \cong S_{n+1} - y_j$ for all $j \leq n$. However, by gluing the additional copies to blue and red leaves of T_n and S_n, we now have ruined the isomorphism between $T_{n+1} - v$ and $S_{n+1} - w$. In order to repair this, we also have to copy the graphs appearing beyond r and b in Fig. 1 respectively onto all red and blue leaves of $\hat{S}_n$ and $\hat{T}_n$. This repairs $T_{n+1} - v \cong S_{n+1} - w$, but again violates our initial promises. In this way, we keep adding, step by step, further copies of the graphs appearing beyond r and b in Fig. 1 respectively onto all red and blue leaves of everything we have constructed so far.

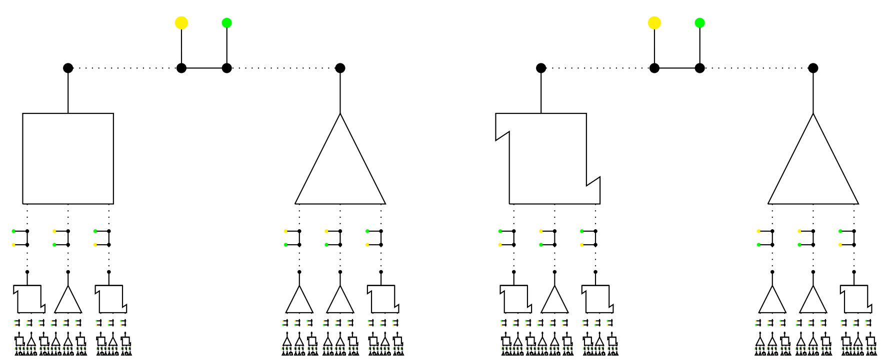


FIGURE 2. A sketch of T_{n+1} and S_{n+1} after countably many steps.

At every step we preserved the colours of leaves in all newly added copies, so we get new red leaves and blue leaves, and we continue the process of copying onto those new leaves as well. After countably many steps we have dealt with all red or blue leaves. We take these new trees to be S_{n+1} and T_{n+1}. They are non-isomorphic, since after removing all long non-branching paths, T_{n+1} contains T_n as a component, whereas S_{n+1} does not.

Figure 2 shows how T_{n+1} and S_{n+1} might appear. We have now fulfilled our intention of sticking the same thing onto all red leaves and the same thing onto all blue leaves, but we have also ensured that $T_{n+1} - v \cong S_{n+1} - w$, as desired.

3. Closure with respect to promises

In this section, we formalise the ideas set forth in the proof sketch of how to extend a graph so that it looks the same beyond certain sets of leaves.

Given a directed edge $\vec{e} = \vec{xy}$ in some forest $G = (V, E)$, we denote by $G(\vec{e})$ the unique component of $G - e$ containing the vertex y. We think of $G(\vec{e})$ as a rooted tree with root y. As indicated in the previous section, in order to make T and S hypomorphic at the end, we will often have to guarantee $S(\vec{e}) \cong T(\vec{f})$ for certain pairs of edges $\vec{e}$ and $\vec{f}$.

Definition 3.1 (Promise structure)**.** *A* promise structure $\mathcal{P} = \left(G, \vec{P}, \mathcal{L}\right)$ *consists of:*

- *a forest G,*
- *$\vec{P} = \{\vec{p}_i \colon i \in I\}$ a set of directed edges $\vec{P} \subseteq \vec{E}(G)$, and*
- *$\mathcal{L} = \{L_i \colon i \in I\}$ a set of pairwise disjoint sets of leaves of G.*

Often, when the context is clear, we will not make a distinction between $\mathcal{L}$ and the set $\bigcup_i L_i$, for notational convenience.

We will call an edge $\vec{p}_i \in \vec{P}$ a *promise edge*, and leaves $\ell \in L_i$ *promise leaves*. A promise edge $\vec{p}_i \in \vec{P}$ is called a *placeholder-promise* if the component $G(\vec{p}_i)$ consists of a single leaf $c_i \in L_i$, then called a *placeholder-leaf*. We write

$$\mathcal{L}_p = \{L_i \colon i \in I,\ \vec{p}_i \text{ a placeholder-promise}\} \text{ and } \mathcal{L}_q = \mathcal{L} \setminus \mathcal{L}_p.$$

Given a leaf ℓ in G, there is a unique edge $q_\ell \in E(G)$ incident with ℓ, and this edge has a natural orientation $\vec{q}_\ell$ towards ℓ. Informally, we think of the 'promise' $\ell \in L_i$ as saying that if we extend G to a graph $H \supset G$, we will do so in such a way that $H(\vec{q}_\ell) \cong H(\vec{p}_i)$. Given a promise structure $\mathcal{P} = \left(G, \vec{P}, \mathcal{L}\right)$, we would like to construct a graph $H \supset G$ which satisfies all the promises in $\mathcal{P}$. This will be done by the following kind of extension.

Definition 3.2 (Leaf extension)**.** *Given an inclusion $H \supseteq G$ of forests and a set L of leaves of G, H is called a* leaf extension*, or more specifically an* L-extension*, of G, if:*

- *every component of H contains precisely one component of G, and*
- *for every vertex $h \in H \setminus G$ and every vertex $g \in G$ in the same component as h, the unique $g - h$ path in H meets L.*

In the remainder of this section we describe a construction of a forest $\mathrm{cl}(G)$ which has the following properties.

Proposition 3.3. *Let G be a forest and let $\left(G, \vec{P}, \mathcal{L}\right)$ be a promise structure. Then there is a forest* $\mathrm{cl}(G)$ *such that:*

(cl.1) $\mathrm{cl}(G)$ *is an $\mathcal{L}_q$-extension of G, and*

(cl.2) *for every $\vec{p}_i \in \vec{P}$ and all $\ell \in L_i$,*

$$\mathrm{cl}(G)(\vec{p}_i) \cong \mathrm{cl}(G)(\vec{q}_\ell)$$

are isomorphic as rooted trees.

We first describe the construction of $\mathrm{cl}(G)$, and then verify the properties asserted in Proposition 3.3. Let us define a sequence of promise structures $\left(H^{(i)}, \vec{P}, \mathcal{L}^{(i)}\right)$ as follows. We set $\left(H^{(0)}, \vec{P}, \mathcal{L}^{(0)}\right) = \left(G, \vec{P}, \mathcal{L}\right)$. We construct a sequence of graphs

$$G = H^{(0)} \subseteq H^{(1)} \subseteq H^{(2)} \subseteq \cdots ,$$

where all the promise edges in each promise structure are edges in $\vec{P} \subseteq \vec{E}(G)$. Given $\left(H^{(n)}, \vec{P}, \mathcal{L}^{(n)}\right)$, we construct $H^{(n+1)}$ by gluing, for each i, at every promise leaf $\ell \in L_i^{(n)}$ a rooted copy of $G(\vec{p_i})$. We take the natural promise structure on $H^{(n+1)}$, consisting of all promise leaves from the newly added copies of $G(\vec{p_i})$. That is, if a leaf $\ell \in G(\vec{p_i})$ was such that $\ell \in L_j$, then every copy of that leaf will be in $L_j^{(n+1)}$.

Formally, suppose that $\left(G, \vec{P}, \mathcal{L}\right)$ is a promise structure. For each $\vec{p_i} \in \vec{P}$ let $C_i = G(\vec{p_i})$ and let c_i be the root of this tree. If U is a set and H is a graph, then we denote by $U \times H$ the graph whose vertices are pairs (u, v) with $u \in U$ and v a vertex of H, and with an edge from (u, v) to (u, w) whenever vw is an edge of H. Let $\left(H^{(0)}, \vec{P}, \mathcal{L}^{(0)}\right) = \left(G, \vec{P}, \mathcal{L}\right)$ and given $\left(H^{(n)}, \vec{P}, \mathcal{L}^{(n)}\right)$ let us define:

- $H^{(n+1)}$ to be the quotient of $H^{(n)} \oplus \bigoplus_{i \in I} (L_i^{(n)} \times C_i)$ w.r.t. the relation
$$l \sim (l, c_i) \text{ for } l \in L_i^{(n)} \in \mathcal{L}^{(n)}.$$
- $\mathcal{L}^{(n+1)} = \left\{L_i^{(n+1)} : i \in I\right\}$ with $L_i^{(n+1)} = \bigcup_{j \in I} L_j^{(n)} \times (C_j \cap L_i)$.

There is a sequence of natural inclusions $G = H^{(0)} \subseteq H^{(1)} \subseteq \cdots$ and we define $\mathrm{cl}(G)$ to be the direct limit of this sequence.

Definition 3.4 (Promise-respecting map)**.** *Let G be a forest, $F^{(1)}$ and $F^{(2)}$ be leaf extensions of G, and $\mathcal{P}^{(1)} = \left(F^{(1)}, \vec{P}, \mathcal{L}^{(1)}\right)$ and $\mathcal{P}^{(2)} = \left(F^{(2)}, \vec{P}, \mathcal{L}^{(2)}\right)$ be promise structures with $\vec{P} \subseteq \vec{E}(G)$. Suppose $X^{(1)} \subseteq V(F^{(1)})$ and $X^{(2)} \subseteq V(F^{(2)})$.*

A bijection $\varphi \colon X^{(1)} \to X^{(2)}$ is $\vec{P}$-respecting *(with respect to $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$) if the image of $L_i^{(1)} \cap X^{(1)}$ under φ is $L_i^{(2)} \cap X^{(2)}$ for all i.*

Since both promise structures $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ refer to the same edge set $\vec{P}$, we can think of them as defining a $|\vec{P}|$-colouring on some sets of leaves. Then a mapping is $\vec{P}$-respecting if it preserves leaf colours.

Lemma 3.5. *Let $\left(G, \vec{P}, \mathcal{L}\right)$ be a promise structure and let $G = H^{(0)} \subseteq H^{(1)} \subseteq \cdots$ be as defined above. Then the following statements hold:*

- *$H^{(n)}$ is an $\mathcal{L}_q$-extension of G for all n,*
- *$\Delta(H^{(n+1)}) = \Delta(H^{(n)})$ for all n, and*
- *For each $\ell \in L_i \in \mathcal{L}$ there exists a sequence of $\vec{P}$-respecting rooted isomorphisms $\varphi_{\ell,n} \colon H^{(n)}(\vec{p_i}) \to H^{(n+1)}(\vec{q_\ell})$ such that $\varphi_{\ell,n+1}$ extends $\varphi_{\ell,n}$ for all $n \in \mathbb{N}$.*

Proof. The first two statements are clear. We will prove the third by induction on n. To construct $H^{(1)}$ from G, we glued a rooted copy of $G(\vec{p_i})$ to each $\ell \in L_i$, keeping all copies of promise leaves. Hence, for any given $\ell \in L_i$, the natural isomorphism $\varphi_{\ell,0} \colon G(\vec{p_i}) \to H^{(1)}(\vec{q_\ell})$ is $\vec{P}$-respecting as desired.

Now suppose that $\varphi_{\ell,n}$ exists for all $\ell \in \mathcal{L}$. To form $H^{(n+1)}(\vec{p_i})$, we glued on a copy of $G(\vec{p_i})$ to each $\ell \in L_i^{(n)} \cap H^{(n)}(\vec{p_i})$, and to construct $H^{(n+2)}(\vec{q_\ell})$, we glued on a copy of $G(\vec{p_i})$ to each $\ell \in L_i^{(n+1)} \cap H^{(n+1)}(\vec{q_\ell})$, in both cases keeping all copies of promise leaves.

Therefore, since $\varphi_{\ell,n}$ was a $\vec{P}$-respecting rooted isomorphism from $H^{(n)}(\vec{p_i})$ to $H^{(n+1)}(\vec{q_\ell})$, we can combine the individual isomorphisms between the newly added copies of $G(\vec{p_i})$ with $\varphi_{\ell,n}$ to form $\varphi_{\ell,n+1}$. □

We can now complete the proof of Proposition 3.3.

Proof of Proposition 3.3. First, we note that $G \subseteq \mathrm{cl}(G)$, and since each $H^{(n)}$ is an $\mathcal{L}_q$-extension of G for all n, so is $\mathrm{cl}(G)$. Also, since each $H^{(n)}$ is a forest it follows that $\mathrm{cl}(G)$ is a forest.

Let us show that $\mathrm{cl}(G)$ satisfies property (cl.2). Since we have the sequence of inclusions $G = H^{(0)} \subseteq H^{(1)} \subseteq \ldots$, it follows that $\mathrm{cl}(G)(\vec{q_\ell})$ is the direct limit of the sequence $H^{(0)}(\vec{q_\ell}) \subseteq H^{(1)}(\vec{q_\ell}) \subseteq \cdots$ and also $\mathrm{cl}(G)(\vec{p_i})$ is the direct limit of the sequence $H^{(0)}(\vec{p_i}) \subseteq H^{(1)}(\vec{p_i}) \subseteq \cdots$. By Lemma 3.5 there is a sequence of rooted isomorphisms $\varphi_{\ell,n} \colon H^{(n)}(\vec{p_i}) \to H^{(n+1)}(\vec{q_\ell})$ such that $\varphi_{\ell,n+1}$ extends $\varphi_{\ell,n}$, so $\varphi_\ell = \bigcup_n \varphi_{\ell,n}$ is the required isomorphism. □

We remark that it is possible to show that $\mathrm{cl}(G)$ is in fact determined, uniquely up to isomorphism, by the properties (cl.1) and (cl.2). Also we note that since each $H^{(n)}$ has the same maximum degree as G, it follows that $\Delta(\mathrm{cl}(G)) = \Delta(G)$.

There is a natural promise structure on $\mathrm{cl}(G)$ given by the placeholder promises in $\vec{P}$ and their corresponding promise leaves. In the construction sketch from Section 2, these leaves corresponded the the yellow and green leaves, and we needed to keep track of the placeholder promises when we take the closure of a promise structure.

Note that if $\vec{p_i}$ is a placeholder promise, then for each $\left(H^{(n)}, \mathcal{P}, \mathcal{L}^{(n)}\right)$ we have $L_i^{(n)} \supseteq L_i^{(n-1)}$. Indeed, for each leaf in $L_i^{(n-1)}$ we glue a copy of the component c_i together with the associated promises on the leaves in this component. However, c_i is just a single vertex, with a promise corresponding to $\vec{p_i}$, and hence $L_i^{(n)} \supseteq L_i^{(n-1)}$. For every placeholder promise $\vec{p_i} \in \vec{P}$ we define $\mathrm{cl}(L_i) = \bigcup_n L_i^{(n)}$.

Definition 3.6 (Closure of a promise structure)**.** *The* closure *of the promise structure* $(G, \mathcal{P}, \mathcal{L})$ *is the promise structure* $\mathrm{cl}(\mathcal{P}) = \left(\mathrm{cl}(G), \mathrm{cl}(\vec{P}), \mathrm{cl}(\mathcal{L})\right)$*, where:*

- $\mathrm{cl}(\vec{P}) = \left\{\vec{p_i} \colon \vec{p_i} \in \vec{P} \text{ is a placeholder-promise}\right\}$*, and*
- $\mathrm{cl}(\mathcal{L}) = \{\mathrm{cl}(L_i) \colon \vec{p_i} \in \vec{P} \text{ is a placeholder-promise}\}$*.*

We note that, since each isomorphism $\varphi_{\ell,n}$ from Lemma 3.5 was $\vec{P}$-respecting, it is possible to strengthen Proposition 3.3 in the following way.

Proposition 3.7. *Let G be a forest and let $\left(G, \vec{P}, \mathcal{L}\right)$ be a promise structure. Then the forest* $\mathrm{cl}(G)$ *satisfies:*

(cl.3) *for every $\vec{p_i} \in \vec{P}$ and every $\ell \in L_i$,*

$$\mathrm{cl}(G)(\vec{p_i}) \cong \mathrm{cl}(G)(\vec{q_\ell})$$

are isomorphic as rooted trees, and this isomorphism is $\mathrm{cl}(\vec{P})$*-respecting with respect to* $\mathrm{cl}(\mathcal{P})$*.*

Proof. Since each isomorphism $\varphi_{\ell,n} \colon H^{(n)}(\vec{p_i}) \to H^{(n+1)}(\vec{q_\ell})$ in Proposition 3.5 is $\vec{P}$-respecting, we have

$$\varphi_{\ell,n}\Big(L_i^{(n)} \cap H^{(n)}(\vec{p_i})\Big) = L_i^{(n+1)} \cap H^{(n+1)}(\vec{q_\ell}).$$

For each placeholder promise we have that $\text{cl}(L_i) = \bigcup_n L_i^{(n)}$, and so it follows that

$$\text{cl}(L_i) \cap \text{cl}(G)(\vec{q_\ell}) = \bigcup_n \Big(L_i^{(n)} \cap H^{(n)}(\vec{q_\ell})\Big)$$

and

$$\text{cl}(L_i) \cap \text{cl}(G)(\vec{p_i}) = \bigcup_n \Big(L_i^{(n)} \cap H^{(n)}(\vec{p_i})\Big).$$

From this it follows that $\varphi_\ell = \bigcup_n \varphi_{l,n}$ is a $\text{cl}(\vec{P})$-respecting isomorphism between $\text{cl}(G)(\vec{p_i})$ and $\text{cl}(G)(\vec{q_\ell})$ as rooted trees. □

It is precisely this property (cl.3) of the promise closure that will allow us, in Claim 4.13 below, to maintain partial hypomorphisms during our recursive construction.

4. The construction

In this section we construct two hypomorphic locally finite trees neither of which embed into the other, establishing our main theorem announced in the introduction.

4.1. Preliminary definitions.

Definition 4.1 (Mii-path)**.** *A path $P = v_0, v_1, \dots, v_n$ in a graph G is called* internally isolated *if $\deg_G(v_i) = 2$ for all internal vertices v_i for $0 < i < n$. The path P is* maximal internally isolated *(or* mii *for short) if in addition $\deg_G(v_0) \neq 2 \neq \deg_G(v_n)$. An infinite path $P = v_0, v_1, v_2, \dots$ is* mii *if $\deg_G(v_0) \neq 2$ and $\deg_G(v_i) = 2$ for all $i \geq 1$.*

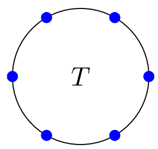
Lemma 4.2. *Let T be a tree and $e \in E(T)$. If every mii-path in T has length at most $k \in \mathbb{N}$, then every mii-path in $T - e$ has length at most $2k$.*

Proof. We first note that every mii-path in $T - e$ has finite length, since any infinite mii-path in $T_n - e$ would contain a subpath which is an infinite mii-path in T. If $P = \{x_0, x_1, \dots, x_n\}$ is an mii-path in $T - e$ which is not a subpath of any mii-path in T, then there is at least one $1 \leq i \leq n-1$ such that e is adjacent to x_i, and since T was a tree, x_i is unique. Therefore, both $\{x_0, x_1, \dots, x_i\}$ and $\{x_i, x_{i+1}, \dots, x_n\}$ are mii-paths in T. By assumption both i and $n - i$ are at most k, and so the length of P is at most $2k$, as claimed. □

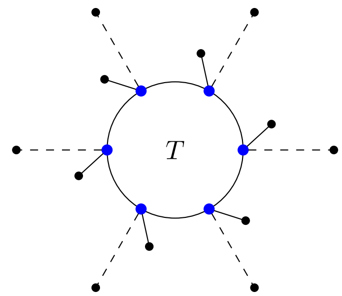
Definition 4.3 (Mii-extension)**.** *Given a forest G, a subset B of leaves of G, and a component T of G, we say that a tree $\hat{T} \supset T$ is an* mii-extension *of T at B to length k if $\hat{T}$ can be obtained from T by adjoining, at each vertex $l \in B \cap V(T)$, a new path of length k starting at l and a new leaf whose only neighbour is l.*

Note that the new leaves attached to each $l \in B$ ensure that the paths of length k are indeed mii-paths.

Definition 4.4 (k-ball)**.** *For G a subgraph of H, the k-ball* $\text{Ball}_H(G, k)$ *is the induced subgraph of H on the set of vertices within distance k of some vertex of G.*



(A) A tree T with designated leaf set B



(B) An mii-extension of T at B

FIGURE 3. Building an mii-extension of a tree T at B to length k. All dotted lines are mii-paths of length k.

Definition 4.5 (Regular binary tree)**.** *For $k \geq 1$, the* regular binary tree of height *k is the unique rooted tree on $2^k - 1 = 1 + 2 + \cdots + 2^{k-1}$ vertices such that the root has degree 2, there are 2^{k-1} leaves, and all other vertices have degree 3.*

The infinite regular binary tree *is the unique rooted tree such that the root has degree 2, and all other vertices have degree 3.*

By a regular binary tree *we mean any regular binary tree, either infinite or of height k for some $k \in \mathbb{N}$.*

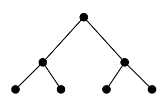


FIGURE 4. The regular binary tree of height 3.

4.2. **The back-and-forth construction.** Our aim is to prove the following theorem.

Theorem 1.6. *There are two hypomorphic infinite trees T and S with maximum degree 3 such that there is no embedding $T \hookrightarrow S$ or $S \hookrightarrow T$.*

To do this we shall recursively construct, for each $n \in \mathbb{N}$,

- disjoint (possibly infinite) rooted trees T_n and S_n,
- disjoint (possibly infinite) sets R_n and B_n of leaves of the forest $T_n \oplus S_n$,
- finite sets $X_n \subset V(T_n)$ and $Y_n \subset V(S_n)$, and bijections $\varphi_n \colon X_n \to Y_n$,
- a family of isomorphisms $\mathcal{H}_n = \{h_{n,x} \colon T_n - x \to S_n - \varphi_n(x) \colon x \in X_n\}$,
- strictly increasing sequences of integers $k_n \geq 2$ and $b_n \geq 3$,

such that (letting all objects indexed by -1 be the empty set) for all $n \in \mathbb{N}$:

(†1) $T_{n-1} \subset T_n$ and $S_{n-1} \subset S_n$ as induced subgraphs,
(†2) the vertices of T_n and S_n all have degree at most 3,
(†3) the root of T_n is in R_n and the root of S_n is in B_n,
(†4) all regular binary trees appearing as subgraphs of $T_n \oplus S_n$ are finite and have height at most b_n,
(†5) all mii-paths in $T_n \oplus S_n$ are finite and have length at most k_n,

(†6) $\mathrm{Ball}_{T_n}(T_{n-1}, k_{n-1}+1)$ is an mii-extension of T_{n-1} at $R_{n-1} \cup B_{n-1}$ to length $k_{n-1}+1$ and does not meet $R_n \cup B_n$,
(†7) $\mathrm{Ball}_{S_n}(S_{n-1}, k_{n-1}+1)$ is an mii-extension of S_{n-1} at $R_{n-1} \cup B_{n-1}$ to length $k_{n-1}+1$ and does not meet $R_n \cup B_n$,
(†8) there is no embedding from T_n into any mii-extension of S_n at $R_n \cup B_n$ to any length, nor from S_n into any mii-extension of T_n at $R_n \cup B_n$ to any length,
(†9) any embedding of T_n into an mii-extension of T_n at $R_n \cup B_n$ to any length fixes the root of T_n and has image T_n,
(†10) any embedding of S_n into an mii-extension of S_n at $R_n \cup B_n$ to any length fixes the root of S_n and has image S_n,
(†11) there are enumerations $V(T_n) = \{t_j : j \in J_n\}$ and $V(S_n) = \{s_j : j \in J_n\}$ such that
- $J_{n-1} \subset J_n \subset \mathbb{N}$,
- $\{t_j : j \in J_n\}$ extends the enumeration $\{t_j : j \in J_{n-1}\}$ of $V(T_{n-1})$, and similarly for $\{s_j : j \in J_n\}$,
- $|\mathbb{N} \setminus J_n| = \infty$,
- $\{0, 1, \dots, n\} \subset J_n$,

(†12) $\{t_j, s_j : j \leq n\} \cap (R_n \cup B_n) = \emptyset$,
(†13) the finite sets of vertices X_n and Y_n satisfy $|X_n| = n = |Y_n|$, and
- $X_{n-1} \subset X_n$ and $Y_{n-1} \subset Y_n$,
- $\varphi_n \restriction X_{n-1} = \varphi_{n-1}$,
- $\{t_j : j \leq n\} \subset X_{2n+1}$ and $\{s_j : j \leq n\} \subset Y_{2(n+1)}$,
- $(X_n \cup Y_n) \cap (R_n \cup B_n) = \emptyset$,

(†14) the families of isomorphisms $\mathcal{H}_n$ satisfy
- $h_{n,x} \restriction (T_{n-1} - x) = h_{n-1,x}$ for all $x \in X_{n-1}$,
- the image of $R_n \cap V(T_n)$ under $h_{n,x}$ is $R_n \cap V(S_n)$, and
- the image of $B_n \cap V(T_n)$ under $h_{n,x}$ is $B_n \cap V(S_n)$ for all $x \in X_n$.

4.3. **The construction yields the desired non-reconstructible trees.** By property (†1), we have $T_0 \subset T_1 \subset T_2 \subset \cdots$ and $S_0 \subset S_1 \subset S_2 \subset \cdots$. Let T and S be the direct limits of the respective sequences. It is clear that T and S are trees, and that as a consequence of (†2), both trees have maximum degree 3.

We claim that the map $\varphi = \bigcup_n \varphi_n$ is a hypomorphism between T and S. Indeed, it follows from (†11) and (†13) that φ is a well-defined bijection from $V(T)$ to $V(S)$. To see that φ is a hypomorphism, consider any vertex x of T. This vertex appears as some t_j in our enumeration of $V(T)$, so the map

$$h_x = \bigcup_{n>2j} h_{n,x} \colon T - x \to S - \varphi(x),$$

is a well-defined isomorphism by (†14) between $T - x$ and $S - \varphi(x)$.

Now suppose for a contradiction that $f \colon T \hookrightarrow S$ is an embedding of T into S. Then $f(t_0)$ is mapped into S_n for some $n \in \mathbb{N}$. Properties (†5) and (†6) imply that after deleting all mii-paths in T of length $> k_n$, the connected component of t_0 is an mii-extension of T_n to length 0. Further, it follows from (†7) that $\mathrm{Ball}_S(S_n, k_n+1)$ is an mii-extension of S_n at $R_n \cup B_n$ to length $k_n + 1$. But combining the fact that

$f(T_n) \cap S_n \neq \emptyset$ and the fact that T_n does not contain long mii-paths, it is easily seen that $f(T_n) \subset \mathrm{Ball}_S(S_n, k_n + 1)$, contradicting (†8).[2]

The case $S \hookrightarrow T$ yields a contradiction in a symmetric fashion, completing the proof.

4.4. **The base case: there are finite rooted trees T_0 and S_0 satisfying requirements (†1)–(†14).** Choose a pair of non-isomorphic, equally sized trees T_0 and S_0 of maximum degree 3, and pick a leaf each as roots $\mathrm{r}(T_0)$ and $\mathrm{r}(S_0)$ for T_0 and S_0, subject to conditions (†8)–(†10) when putting $R_0 = \{\mathrm{r}(T_0)\}$ and $B_0 = \{\mathrm{r}(S_n)\}$. A possible choice is given in Fig. 5. Here, (†8) is satisfied, because any embedding of T_0 into an mii-extension of S_0 has to map the binary tree of height 3 in T_0 to the binary tree in S_0, making it impossible to embed the middle leaf. Properties (†9) and (†10) are similar.

FIGURE 5. A possible choice for finite rooted trees T_0 and S_0.

Let $J_0 = \{0, 1, \dots, |T_0| - 1\}$ and choose enumerations $V(T_0) = \{t_j : j \in J_0\}$ and $V(S_0) = \{s_j : j \in J_0\}$ with $t_0 \neq \mathrm{r}(T_0)$ and $s_0 \neq \mathrm{r}(S_n)$. This takes care of (†11) and (†12). Finally, (†13) and (†14) are satisfied for $X_0 = Y_0 = \mathcal{H}_0 = \emptyset$. Set $k_0 = 2$ and $b_0 = 3$.

4.5. **The inductive step: Set-up.** Now, assume that we have constructed trees T_k and S_k for all $k \leq n$ such that (†1)–(†14) are satisfied up to n. If $n = 2m$ is even, then we have $\{t_j : j \leq m-1\} \subset X_n$ and in order to satisfy (†13) we have to construct T_{n+1} and S_{n+1} such that the vertex t_m is taken care of in our partial hypomorphism. Similarly, if $n = 2m+1$ is odd, then we have $\{s_j : j \leq m-1\} \subset Y_n$ and we have to construct T_{n+1} and S_{n+1} such that the vertex s_m is taken care of in our partial hypomorphism. Both cases are symmetric, so let us assume in the following that $n = 2m$ is even.

Now let v be the vertex with the least index in the set $\{t_j : j \in J_n\} \setminus X_n$, i.e.

$$v = t_i \quad \text{for} \quad i = \min\{\ell \colon t_\ell \in V(T_n) \setminus X_n\}. \tag{1}$$

Then by assumption (†13), v will be t_m, unless t_m was already in X_n anyway. In any case, since $|X_n| = |Y_n| = n$, it follows from (†11) that $i \leq n$, so by (†12), v does not lie in our leaf lets $R_n \cup B_n$, i.e.

$$v \notin R_n \cup B_n. \tag{2}$$

[2] To get the non-embedding property, we have used (†5)–(†8) at every step n. While at the first glance, properties (†4), (†9) and (†10) don't seem to be needed at this point, they are crucial during the construction to establish (†8) at step $n+1$. See Claim 4.10 below for details.

In the next sections, we will demonstrate how to to obtain trees $T_{n+1} \supset T_n$ and $S_{n+1} \supset S_n$ with $X_{n+1} = X_n \cup \{v\}$ and $Y_{n+1} = Y_n \cup \{\varphi_{n+1}(v)\}$ satisfying (†1)—(†10) and (†13)–(†14).

After we have completed this step, since $|\mathbb{N} \setminus J_n| = \infty$, it is clear that we can extend our enumerations of T_n and S_n to enumerations of T_{n+1} and S_{n+1} as required, making sure to first list some new elements that do not lie in $R_{n+1} \cup B_{n+1}$. This takes care of (†11) and (†12) and completes the recursion step $n \mapsto n+1$.

4.6. **The inductive step: Construction.** Given the two trees T_n and S_n, we extend each of them through their roots as indicated in Figure 6 to trees $\tilde{T}_n$ and $\tilde{S}_n$ respectively. The trees T_{n+1} and S_{n+1} will be obtained as components of the promise closure of the forest $G_n = \tilde{T}_n \oplus \tilde{S}_n$ with respect to the coloured promise edges.

Since v is not the root of T_n, there is a first edge e on the unique path in T_n from v to the root. Then $T_n - e$ has two connected components: one that contains the root of T_n which we name $T_n(r)$, and one that contains v which we name $T_n(v)$.

Since every mii-path in T_n has bounded finite length by (†5), it follows from Lemma 4.2 that all mii-paths in $T_n - e$, and so all mii-paths in $T_n(r)$ and $T_n(v)$, have bounded finite length as well. Let $k = \tilde{k}_n$ be twice the upper bound for the length of mii-paths in T_n, S_n, $T_n(r)$ and $T_n(v)$, which exists by (†5).

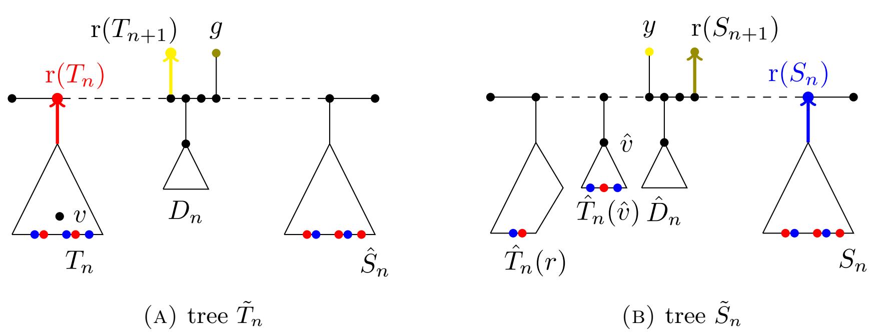


FIGURE 6. All dotted lines are mii-paths of length at least $k = \tilde{k}_n$. The trees D_n are regular binary trees of height $b_n + 3$, so that $D_n \not\to T_n$ or S_n.

To obtain $\tilde{T}_n$, we extend T_n through its root $\mathrm{r}(T_n) \in R_n$ by a path

$$\mathrm{r}(T_n) = u_0, u_1, \ldots, u_{p-1}, u_p = \mathrm{r}\left(\hat{S}_n\right)$$

of length $p = 4(\tilde{k}_n + 1) + 3$, where at its last vertex u_p we glue a rooted copy $\hat{S}_n$ of S_n (via an isomorphism $\hat{w} \leftrightarrow w$), identifying u_p with the root of $\hat{S}_n$.

Next, we add two additional leaves at u_0 and u_p, so that $\deg(\mathrm{r}(T_n)) = 3 = \deg\left(\mathrm{r}\left(\hat{S}_n\right)\right)$. Further, we add a leaf $\mathrm{r}(T_{n+1})$ at u_{2k+2}, which will be our new root for the next tree T_{n+1}; and another leaf g at u_{2k+5}. Finally, we take a copy D_n of a regular rooted binary tree of height $b_n + 3$ and connect its root via an edge to u_{2k+3}. This completes the construction of $\tilde{T}_n$.

The construction of $\tilde{S}_n$ is similar, but with a twist. For its construction, we extend S_n through its root $\mathrm{r}(S_n) \in B_n$ by a path

$$\mathrm{r}(S_n) = v_p, v_{p-1}, \dots, v_1, v_0 = \mathrm{r}\Big(\hat{T}_n(r)\Big)$$

of length p, where at its last vertex v_0 we glue a copy $\hat{T}_n(r)$ of $T_n(r)$, identifying v_0 with the root of $\hat{T}_n(r)$. Then, we take a copy $\hat{T}_n(\hat{v})$ of $T_n(v)$ and connect $\hat{v}$ via an edge to v_{k+1}.

Finally, as before, we add two leaves at v_0 and v_p so that $\deg\Big(\mathrm{r}\Big(\hat{T}_n(r)\Big)\Big) = 3 = \deg\left(\mathrm{r}(S_n)\right)$. Next, we add a leaf $\mathrm{r}(S_{n+1})$ to v_{2k+5}, which will be our new root for the next tree S_{n+1}; and another leaf y to v_{2k+2}. Finally, we take another copy $\hat{D}_n$ of a regular rooted binary tree of height $b_n + 3$ and connect its root via an edge to v_{2k+3}. This completes the construction of $\tilde{S}_n$.

By induction assumption, certain leaves of T_n have been coloured with one of the two colours $R_n \cup B_n$, and also some leaves of S_n have been coloured with one of the two colours $R_n \cup B_n$. In the above construction, we colour leaves of $\hat{S}_n$, $\hat{T}_n(r)$ and $\hat{T}_n(\hat{v})$ accordingly:

$$\begin{aligned} \tilde{R}_n &= \Big(R_n \cup \Big\{\hat{w} \in \hat{S}_n \cup \hat{T}_n(r) \cup \hat{T}_n(\hat{v}) \colon w \in R_n\Big\}\Big) \setminus \Big\{\mathrm{r}(T_n), \mathrm{r}\Big(\hat{T}_n(r)\Big)\Big\}, \\ \tilde{B}_n &= \Big(B_n \cup \Big\{\hat{w} \in \hat{S}_n \cup \hat{T}_n(r) \cup \hat{T}_n(\hat{v}) \colon w \in B_n\Big\}\Big) \setminus \Big\{\mathrm{r}(S_n), \mathrm{r}\Big(\hat{S}_n\Big)\Big\}. \end{aligned} \tag{3}$$

Now put $G_n := \tilde{T}_n \oplus \tilde{S}_n$ and consider the following promise structure $\mathcal{P} = \Big(G_n, \vec{P}, \mathcal{L}\Big)$ on G_n, consisting of four promise edges $\vec{P} = \{\vec{p}_1, \vec{p}_2, \vec{p}_3, \vec{p}_4\}$ and corresponding leaf sets $\mathcal{L} = \{L_1, L_2, L_3, L_4\}$, as follows:

(4)

- $\vec{p}_1$ pointing in T_n towards the root $\mathrm{r}(T_n)$, with promise leaves $L_1 = \tilde{R}_n$,
- $\vec{p}_2$ pointing in S_n towards the root $\mathrm{r}(S_n)$, with $L_2 = \tilde{B}_n$,
- $\vec{p}_3$ pointing in $\tilde{T}_n$ towards the root $\mathrm{r}(T_{n+1})$, with $L_3 = \{\mathrm{r}(T_{n+1}), y\}$,
- $\vec{p}_4$ pointing in $\tilde{S}_n$ towards the root $\mathrm{r}(S_{n+1})$, with $L_4 = \{\mathrm{r}(S_{n+1}), g\}$.

Note that our construction so far has been tailored to provide us with a $\vec{P}$-respecting isomorphism

$$h \colon \tilde{T}_n - v \to \tilde{S}_n - \hat{v}. \tag{5}$$

Consider the closure $\mathrm{cl}(G_n)$ with respect to the promise structure $\mathcal{P}$ defined above. Since $\mathrm{cl}(G_n)$ is a leaf-extension of G_n, it has two connected components, just as G_n. We now define

$$\begin{aligned} T_{n+1} = &\text{ the component containing } T_n \text{ in } \mathrm{cl}(G_n), \text{ and} \\ S_{n+1} = &\text{ the component containing } S_n \text{ in } \mathrm{cl}(G_n). \end{aligned} \tag{6}$$

It follows that $\mathrm{cl}(G_n) = T_{n+1} \oplus S_{n+1}$. Further, since $\vec{p}_3$ and $\vec{p}_4$ are placeholder promises, $\mathrm{cl}(G)$ carries a corresponding promise structure, cf. Def. 3.6. We define

$$R_{n+1} = \mathrm{cl}(L_3) \;\text{ and }\; B_{n+1} = \mathrm{cl}(L_4). \tag{7}$$

Lastly, we set

$$\begin{aligned} X_{n+1} &= X_n \cup \{v\}, \\ Y_{n+1} &= Y_n \cup \{\hat{v}\}, \text{ and} \\ \varphi_{n+1} &= \varphi_n \cup \{\langle v, \hat{v}\rangle\}, \end{aligned} \tag{8}$$

and put

$$k_{n+1} = 2\tilde{k}_n + 3 \text{ and } b_{n+1} = b_n + 3 \tag{9}$$

The construction of trees T_{n+1} and S_{n+1}, coloured leaf sets R_{n+1} and B_{n+1}, the bijection $\varphi_{n+1}\colon X_{n+1} \to Y_{n+1}$, and integers k_{n+1} and b_{n+1} is now complete. In the following, we verify that (†1)–(†14) are indeed satisfied for the $(n+1)^{\text{th}}$ instance.

4.7. **The inductive step: Verification.**

Claim 4.6. *T_{n+1} and S_{n+1} extend T_n and S_n. Moreover, they are rooted trees of maximum degree 3 such that their respective roots are contained in R_{n+1} and B_{n+1}. Hence,* (†1)–(†3) *are satisfied.*

Proof. Property (†1) follows from (cl.1), i.e. that $\mathrm{cl}(G_n)$ is a leaf-extension of G_n. Thus, T_{n+1} is a leaf extension of $\tilde{T}_n$, which in turn is a leaf extension of T_n, and similar for S_n. This shows (†1).

As noted after the proof of Proposition 3.3, taking the closure does not affect the maximum degree, i.e. $\Delta(\mathrm{cl}(G_n)) = \Delta(G_n) = 3$. This shows (†2).

Finally, (7) implies (†3), as $\mathrm{r}(T_{n+1}) \in R_{n+1}$ and $\mathrm{r}(S_{n+1}) \in B_{n+1}$. □

Claim 4.7. *All regular binary trees appearing as subgraphs of $T_{n+1} \oplus S_{n+1}$ have height at most b_{n+1}, and every such tree of height b_{n+1} is some copy D_n or $\hat{D}_n$. Hence, T_{n+1} and S_{n+1} satisfy* (†4).

Proof. We first claim that all regular binary trees appearing as subgraphs of $\tilde{T}_n \oplus \tilde{S}_n$ which are not contained in D_n or $\hat{D}_n$ have height at most $b_n + 1$. Indeed, note that any regular binary tree appearing as a subgraph of T_n, $\hat{T}_n(r)$, $\hat{T}_n(v)$, $\hat{S}_n$ or S_n has height at most b_n by the inductive hypothesis. Since the paths we added to the roots of T_n and $\hat{S}_n$ to form $\tilde{T}_n$ were sufficiently long, any regular binary tree appearing as a subgraph of $\tilde{T}_n$ can only meet one of T_n, $\hat{S}_n$ or D_n. Since the roots of T_n and $\hat{S}_n$ are adjacent to two new vertices in $\tilde{T}_n$, one of degree 1, any such tree meeting T_n or S_n must have height at most $b_n + 1$. By Figure 6 we see that any regular binary tree in $\tilde{T}_n$ which meets D_n but whose root lies outside of D_n has height at most $3 \leq b_n + 1$. Consider then, a regular binary tree whose root lies inside D_n, but that is not contained in D_n. Again, by Figure 6 we see that the root of D_n must lie in one of the bottom three layers of this binary tree. Hence, if the root of this tree lies on the kth level of D_n, then the tree can have height at most $\min\{b_n + 3 - k, k + 2\}$, and hence the tree has height at most $b_n/2 + 2 \leq b_n + 1$. Any other regular binary tree meeting D_n is then contained in D_n. It follows that the only regular binary tree of height $b_n + 3$ appearing as a subgraph of $\tilde{T}_n$ is D_n, and a similar argument holds for $\tilde{S}_n$ and $\hat{D}_n$.

Recall that T_{n+1} and S_{n+1} are the components of $\mathrm{cl}(\tilde{T}_n \oplus \tilde{S}_n)$ containing $\tilde{T}_n$ and $\tilde{S}_n$ respectively. If we refer back to Section 3 we see that T_{n+1} can be formed from $\tilde{T}_n$ by repeatedly gluing components isomorphic to $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$ to leaves. Consider a regular binary tree appearing as a subgraph of T_{n+1} which is contained in $\tilde{T}_n$ or one of the copies of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$. By the previous paragraph, this tree has height at most b_n+3, and if it has height b_n+3 it is a copy D_n or $\hat{D}_n$. Suppose then that there is a regular binary tree, of height b, whose root is in $\tilde{T}_n$, but is not contained in $\tilde{T}_n$. Such a tree must contain some vertex $\ell \in \tilde{T}_n$ which is adjacent to a vertex not in $\tilde{T}_n$. Hence, ℓ must have been a leaf in $\tilde{T}_n$ at which a copy of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$ was glued on. However, the roots of each of these components are adjacent to just two vertices, one of degree 1, and hence this leaf ℓ must either be in the bottom, or second to bottom layer of the binary tree. Therefore, $b \le b_n+2$. A similar argument holds when the root lies in some copy of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$, and also for S_{n+1}.

Therefore, all regular binary trees appearing as subgraphs of $T_{n+1} \oplus S_{n+1}$ have height at most b_n+3, and every such tree is some copy D_n or $\hat{D}_n$. Hence, since $b_{n+1} = b_n+3$, it follows that $b_{n+1} \ge b_n$ and T_{n+1} and S_{n+1} satisfy (†4). □

Claim 4.8. *Every mii-path in $T_{n+1} \oplus S_{n+1}$ has length at most k_{n+1} and hence T_{n+1} and S_{n+1} satisfy* (†5).

Proof. We first claim that all mii-paths in $\tilde{T}_n \oplus \tilde{S}_n$ have length at most $2\tilde{k}_n+3$. Firstly, we note that any mii-path which is contained in T_n or $\hat{S_n}$ has length at most $k_n \le \tilde{k}_n$ by the induction hypothesis. Also, since the roots of T_n and $\hat{S}_n$ have degree 3 in $\tilde{T}_n$, any mii-path is either contained in T_n or $\hat{S_n}$, or doesn't contain any interior vertices from T_n or $\hat{S_n}$. However, it is clear from the construction that any mii-path in $\tilde{T}_n$ that doesn't contain any interior vertices from T_n or $\hat{S_n}$ has length at most $2\tilde{k}_n+3$. Similarly, any mii-path which is contained in $\hat{T}_n(r)$, $\hat{T}_n(v)$, or S_n has length at most $\tilde{k}_n$ by definition. By the same reasoning as above, any mii-path in $\tilde{S}_n$ not contained in $\hat{T}_n(r)$, $\hat{T}_n(v)$, or S_n has length at most $2\tilde{k}_n+3$.

Again, recall that T_{n+1} can be formed from $\tilde{T}_n$ by repeatedly gluing components isomorphic to $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$ to leaves. Any mii-path in T_{n+1} which is contained in $\tilde{T}_n$ or one of the copies of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$ has length at most $2\tilde{k}_n+3$ by the previous paragraph. However, since every interior vertex in an mii-path has degree two, and the vertices in T_{n+1} at which we, at some point in the construction, stuck on copies of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$ have degree 3, any mii-path in T_{n+1} must be contained in $\tilde{T}_n$ or one of the copies of $\tilde{T}_n(\vec{p_1})$ or $\tilde{S}_n(\vec{p_2})$. Again, a similar argument holds for S_{n+1}. Hence, all mii-paths in $T_{n+1} \oplus S_{n+1}$ have length at most $2\tilde{k}_n+3$. Therefore, since $k_{n+1} = 2\tilde{k}_n+3$, it follows that $k_{n+1} \ge k_n$ and T_{n+1} and S_{n+1} satisfy (†5). □

Claim 4.9. $\mathrm{Ball}_{T_{n+1}}(T_n, k_n+1)$ *is an mii-extension of T_n at $R_n \cup B_n$ to length k_n+1 and does not meet $R_{n+1} \cup B_{n+1}$ and similarly for S_{n+1}. Hence, T_{n+1} and S_{n+1} satisfy* (†6) *and* (†7) *respectively.*

Proof. We will show that T_{n+1} satisfies (†6), the proof that S_{n+1} satisfies (†7) is identical. By Proposition 3.3, the tree T_{n+1} is an $\Big((\tilde{R}_n \cup \tilde{B}_n) \cap V(\tilde{T}_n)\Big)$-extension of $\tilde{T}_n$. Hence T_{n+1} is an

$$\Big(\Big((\tilde{R}_n \cup \tilde{B}_n) \cap V(T_n)\Big) \cup r(T_n)\Big) = \big((R_n \cup B_n) \cap V(T_n)\big)\text{-extension of } T_n. \quad (10)$$

By looking at the construction of $\mathrm{cl}(G)$ from Section 3, we see that T_{n+1} is also an L'-extension of the supertree $T' \supseteq T_n$ formed by gluing a copy of $\tilde{T}_n(\vec{p_1})$ to every leaf in $R_n \cap V(T_n)$ and a copy of $\tilde{S}_n(\vec{p_2})$ to every leaf in $B_n \cap V(T_n)$, where the leaves in L' are the inherited promise leaves from the copies of $\tilde{T}_n(\vec{p_1})$ and $\tilde{S}_n(\vec{p_2})$.

However, we note that every promise leaf in $\tilde{T}_n(\vec{p_1})$ and $\tilde{S}_n(\vec{p_2})$ is at distance at least $\tilde{k}_n + 1$ from the respective root, and so $\mathrm{Ball}_{T_{n+1}}(T_n, \tilde{k}_n) = \mathrm{Ball}_{T'}(T_n, \tilde{k}_n)$. However, $\mathrm{Ball}_{T'}(T_n, \tilde{k}_n)$ can be seen immediately to be an mii-extension of T_n at $R_n \cup B_n$ to length $\tilde{k}_n$, and since $\tilde{k}_n \geq k_n + 1$ it follows that $\mathrm{Ball}_{T_{n+1}}(T_n, k_n + 1)$ is an mii-extension of T_n at $R_n \cup B_n$ to length $k_n + 1$ as claimed.

Finally, we note that $R_{n+1} \cup B_{n+1}$ is the set of promise leaves $\mathrm{cl}(\mathcal{L}_n)$. By the same reasoning as before, $\mathrm{Ball}_{T_{n+1}}(T_n, k_n + 1)$ contains no promise leaf in $\mathrm{cl}(\mathcal{L}_n)$, and so does not meet $R_{n+1} \cup B_{n+1}$ as claimed. □

Claim 4.10. *Let U_{n+1} be an mii-extension of $\mathrm{cl}(G_n) = T_{n+1} \oplus S_{n+1}$ at $R_{n+1} \cup B_{n+1}$ to any length. Then any embedding of T_{n+1} or S_{n+1} into U_{n+1} fixes the respective root. This immediately implies* (†8)*.*

Proof. Recall that the promise closure was constructed by recursively adding copies of rooted trees C_i and identifying their roots with promise leaves. For the promise structure $\mathcal{P} = \left(G_n, \vec{P}, \mathcal{L}\right)$ on G_n we have $C_1 = \tilde{T}_n(\vec{p_1})$ and $C_2 = \tilde{S}_n(\vec{p_2})$.

Next note that no embedding of T_n into U_{n+1} can use any mii-path of length $k_n + 1$ because T_n does not contain such a path. Also, by construction every copy of T_n, S_n, $\hat{T}_n(r)$, or $\hat{T}_n(\hat{v})$ in T_{n+1} has the property that its $k_n + 1$-ball in T_{n+1} is an mii-extension to length $k_n + 1$ of this copy. Hence, if the root of T_n embeds into some copy of T_n, S_n, $\hat{T}_n(r)$, or $\hat{T}_n(\hat{v})$, then the whole tree embeds into an mii-extension of this copy. The same is true for S_n.

Observe that only two of the aforementioned embeddings are in fact possible:

- T_n embeds into an mii-extension of a copy of T_n (or S_n embeds into an mii-extension of a copy of S_n.). This can be done, but the root must be preserved, otherwise we contradict (†9) or (†10);
- There are no embeddings of T_n into an mii-extension of S_n, or of S_n into an mii-extension of T_n by (†8);
- Both $\hat{T}_n(r)$ and $\hat{T}_n(\hat{v})$ are strict subtrees of T_n. Hence there is no embedding of T_n or S_n into mii-extensions of them by (†8) and (†9).

Let $f\colon T_{n+1} \hookrightarrow U_{n+1}$ be an embedding. By Claim 4.7, U_{n+1} contains no binary trees of height $b_n + 3$ apart from D_n, $\hat{D}_n$, and the copies of those two trees that were created by adding copies of C_1 and C_2. Consequently f maps D_n to one of these copies. The neighbours of $\mathrm{r}(T_{n+1})$ and g must map to vertices of degree 3, showing that $f(\mathrm{r}(T_{n+1})) \in R_{n+1}$. If $f(\mathrm{r}(T_{n+1})) = \mathrm{r}(T_{n+1})$ then we are done.

Otherwise there are two possibilities for $f(\mathrm{r}(T_{n+1}))$. If $f(\mathrm{r}(T_{n+1}))$ is contained in a copy of C_1, then $\mathrm{r}(T_n)$ maps to a promise leaf other than the root in a copy of T_n, S_n, $\hat{T}_n(r)$, or $\hat{T}_n(\hat{v})$. If $f(\mathrm{r}(T_{n+1})) = y$ or $f(\mathrm{r}(T_{n+1}))$ is contained in a copy of C_2, then $\mathrm{r}(T_n)$ maps to a copy of $\mathrm{r}\left(\hat{T}_n(r)\right)$ or some vertex of $\hat{T}_n(\hat{v})$. In both cases the root of T_n does not map to the root of a copy of T_n, which is impossible.

Finally, let $f\colon S_{n+1} \hookrightarrow U_{n+1}$ be an embedding. By the same arguments as above $f(\mathrm{r}(S_{n+1})) \in B_{n+1}$. If f fixes $\mathrm{r}(G_{n+1})$, we are done.

Otherwise we have again two cases. If $f(\mathrm{r}(T_{n+1})) = g$, or $f(\mathrm{r}(T_{n+1}))$ is contained in a copy of C_1, then v_{k+1} (the neighbour of $\hat{v}$ on the long path) would have to map to a vertex of degree 2, giving an immediate contradiction. If $f(\mathrm{r}(S_{n+1}))$ is contained in a copy of C_2, then $\mathrm{r}(S_n)$ maps to a promise leaf other than the root in a copy of T_n, S_n, $\hat{T}_n(r)$, or $\hat{T}_n(\hat{v})$ which is also impossible. □

Claim 4.11. *Let U_{n+1} be as in the previous claim. Then there is no embedding of T_{n+1} or S_{n+1} into U_{n+1} whose image contains vertices outside of $\mathrm{cl}(G_n)$, i.e. vertices that have been added to form the mii-extension.*

Since a root-preserving embedding of a locally finite tree into itself must be an automorphism, this together with the previous claim implies (†9) *and* (†10).

Proof. We prove this claim for T_{n+1}, the proof for S_{n+1} is similar. Assume for a contradiction that there is a vertex w of T_{n+1} and an embedding $f\colon T_{n+1} \hookrightarrow U_{n+1}$ such that $f(w) \notin \mathrm{cl}(G_n)$. By definition of mii-extension, removing $f(w)$ from U_{n+1} splits the component of $f(w)$ into at most two components, one of which is a path.

Note first that w does not lie in a copy of D_n or $\hat{D}_n$, because these must map to binary trees of the same height by Claim 4.7. Furthermore, all vertices in $R_{n+1} \cup B_{n+1}$ have a neighbour of degree 3 whose neighbours all have degree ≥ 2, thus $w \notin R_{n+1} \cup B_{n+1}$. Finally, only one component of $T_{n+1} - w$ can contain vertices of degree 3. Consequently, w must lie in a copy C of T_n, S_n, $\hat{T}_n(r)$, or $\hat{T}_n(\hat{v})$.

All mii-paths in the image $f(C)$ have length at most $k = \tilde{k}_n$, so $f(C)$ cannot intersect any copies of T_n, S_n, $\hat{T}_n(r)$, or $(\hat{T}_n(\hat{v}) + v_{k+1})$. Let r be the root of C (where $r = \hat{v}$ in the last case). Now $f(r)$ must have the following properties: it is a vertex of degree 3, and the root of a nearest binary tree of height b_{n+1} not containing $f(r)$ lies at distance d from $f(r)$, where $5 \leq d \leq 2k+4$.

But the only vertices with these properties are contained in copies of T_n, $\hat{S}_n$, $\hat{T}_n(r)$, or $(\hat{T}_n(\hat{v}) + v_{k+1})$. This contradicts the fact that $f(C)$ does not intersect any of these copies. □

Claim 4.12. *The function φ_{n+1} is a well-defined bijection extending φ_n, such that its domain and range do not intersect $R_{n+1} \cup B_{n+1}$. Hence, property* (†13) *holds for $\varphi_{n+1}\colon X_{n+1} \to Y_{n+1}$.*

Proof. By the choice of x in (1) and the definition of $\varphi_{n+1}\colon X_{n+1} \to Y_{n+1}$ in (8), the first three items of property (†13) hold.

Since v does not lie in $R_n \cup B_n$ by (2), it follows by our construction of the promise structure $\mathcal{P} = \left(G_n, \vec{P}, \mathcal{L}\right)$ in (3) and (4) that neither v nor $\hat{v} = \varphi_{n+1}(v)$ appear as promise leaves in $\mathcal{L}$. Furthermore, by the induction hypothesis, $(X_n \cup Y_n) \cap (R_n \cup B_n) = \emptyset$, so no vertex in $(X_n \cup Y_n)$ appears as a promise leaf in $\mathcal{L}$ either. Thus, in formulas,

$$(X_{n+1} \cup Y_{n+1}) \cap \bigcup_{L \in \mathcal{L}} L = \emptyset. \tag{11}$$

In particular, since

$$(R_{n+1} \cup B_{n+1}) \cap G_n = (\mathrm{cl}(L_3) \cup \mathrm{cl}(L_4)) \cap G_n = L_3 \cup L_4,$$

and $X_{n+1} \cup Y_{n+1} \subset G_n$, we get $(X_{n+1} \cup Y_{n+1}) \cap (R_{n+1} \cup B_{n+1}) = \emptyset$. Thus, also the last item of (†13) is verified. □

Claim 4.13. *There is a family of isomorphisms $\mathcal{H}_{n+1} = \{h_{n+1,x}\colon x \in X_{n+1}\}$ witnessing that $T_{n+1} - x$ and $S_{n+1} - \varphi_{n+1}(x)$ are isomorphic for all $x \in X_{n+1}$, such that $h_{n+1,x}$ extends $h_{n,x}$ for all $x \in X_n$. Hence, property* (†14) *holds.*

Proof. There are four things to be verified for this claim. Firstly, we need an isomorphism $h_{n+1,v}$ witnessing that $T_{n+1} - v$ and $S_{n+1} - \hat{v}$ are isomorphic. Secondly, we need to *extend* all previous isomorphisms $h_{n,x}$ between $T_n - x$ and $S_n - \varphi_n(x)$

to $T_{n+1} - x$ and $S_{n+1} - \varphi_n(x)$. This will take care of the first item of (†14). To also comply with the remaining two items, we need to make sure that the isomorphisms

$$\mathcal{H}_{n+1} = \{h_{n+1,x} \colon x \in X_{n+1}\}$$

all map leaves in $R_{n+1} \cap V(T_{n+1})$ bijectively to leaves in $R_{n+1} \cap V(S_{n+1})$, and similarly for B_{n+1}.

To find the first isomorphism, note that by construction of the promise structure $\mathcal{P} = \left(G_n, \vec{P}, \mathcal{L}\right)$ on G_n in (3), and properties (cl.1) and (cl.3) of the promise closure, the trees T_{n+1} and S_{n+1} are obtained from $\tilde{T}_n$ and $\tilde{S}_n$ by attaching at every leaf $r \in \tilde{R}_n$ a copy of the rooted tree $\mathrm{cl}(G_n)(\vec{p}_1)$, and by attaching at every leaf $b \in \tilde{B}_n$ a copy of the rooted tree $\mathrm{cl}(G_n)(\vec{p}_2)$.

By (11), neither v nor $\varphi_{n+1}(v)$ are mentioned in $\mathcal{L}$. As observed in (5), there is a $\vec{P}$-respecting isomorphism

$$h \colon \tilde{T}_n - v \to \tilde{S}_n - \varphi_{n+1}(v).$$

In other words, h maps promise leaves in $L_i \cap V(\tilde{T}_n)$ bijectively to the promise leaves in $L_i \cap V(\tilde{S}_n)$ for all $i = 1, 2, 3, 4$. Our plan is to extend h to an isomorphism between $T_{n+1} - v$ and $S_{n+1} - \varphi_n(v)$ by mapping the corresponding copies of $\mathrm{cl}(G_n)(\vec{p}_1)$ and $\mathrm{cl}(G_n)(\vec{p}_2)$ attached to the various red and blue leaves to each other.

Formally, by (cl.3) there is for each $\ell \in \left(\tilde{R}_n \cup \tilde{B}_n\right) \cap V(T)$ a $\mathrm{cl}(\vec{P})$-respecting isomorphism of rooted trees

$$\mathrm{cl}(G_n)(\vec{q}_\ell) \cong \mathrm{cl}(G_n)(\vec{q}_{h(\ell)}).$$

Therefore, by combining the isomorphism h between $\tilde{T}_n - v$ and $\tilde{S}_n - \varphi_{n+1}(v)$ with these isomorphisms between each $\mathrm{cl}(G_n)(\vec{q}_\ell)$ and $\mathrm{cl}(G_n)(\vec{q}_{h(\ell)})$ we get a $\mathrm{cl}(\vec{P})$-respecting isomorphism

$$h_{n+1,v} \colon T_{n+1} - v \to S_{n+1} - \varphi_{n+1}(v).$$

And since R_{n+1} and B_{n+1} have been defined in (7) to be the promise leaf sets of $\mathrm{cl}(\mathcal{P})$, by definition of $\mathrm{cl}(\vec{P})$-respecting (Def. 3.4), the image of $R_{n+1} \cap V(T_{n+1})$ under $h_{n+1,v}$ is $R_{n+1} \cap V(S_{n+1})$, and similarly for B_{n+1}.

It remains to extend the old isomorphisms in $\mathcal{H}_n$. As argued in (10), both trees T_{n+1} and S_{n+1} are leaf extensions of T_n and S_n at $R_n \cup B_n$ respectively. By property (cl.3), these leaf extensions are obtained by attaching at every leaf $r \in R_n$ a copy of the rooted tree $\mathrm{cl}(G_n)(\vec{p}_1)$, and similarly by attaching at every leaf $b \in B_n$ a copy of the rooted tree $\mathrm{cl}(G_n)(\vec{p}_2)$.

By induction assumption (†14), for each $x \in X_n$ the isomorphism

$$h_{n,x} \colon T_n - x \to S_n - \varphi_n(x)$$

maps the red leaves of T_n bijectively to the red leaves of S_n, and the blue leaves of T_n bijectively to the blue leaves of S_n. Thus, by property (cl.3), there are $\mathrm{cl}(\vec{P})$-respecting isomorphisms of rooted trees

$$\mathrm{cl}(G_n)(\vec{q}_\ell) \cong \mathrm{cl}(G_n)(\vec{q}_{h_{n,x}(\ell)})$$

for all $\ell \in (R_n \cup B_n) \cap V(T_n)$. By combining the isomorphism $h_{n,x}$ between $T_n - x$ and $S_n - \varphi_n(x)$ with these isomorphisms between each $\mathrm{cl}(G_n)(\vec{q}_\ell)$ and $\mathrm{cl}(G_n)(\vec{q}_{h_{n,x}(l)})$, we obtain a $\mathrm{cl}(\vec{P})$-respecting extension

$$h_{n+1,x} \colon T_{n+1} - x \to S_{n+1} - \varphi_n(x).$$

As before, by definition of $\mathrm{cl}(\vec{P})$-respecting, the image of $R_{n+1} \cap V(T_{n+1})$ under $h_{n+1,x}$ is $R_{n+1} \cap V(S_{n+1})$, and similarly for B_{n+1}.

Finally, by construction we have $h_{n+1,x} \restriction (T_n - x) = h_{n,x}$ for all $x \in X_n$ as desired. The proof is complete. □

References

[1] T. Andreae. On the Reconstruction of Locally Finite Trees. *Journal of Graph Theory*, 5(2):123–135, 1981.

[2] J. A. Bondy and R. Hemminger. Reconstructing infinite graphs. *Pacific Journal of Mathematics*, 52(2):331–340, 1974.

[3] J. A. Bondy and R. L. Hemminger. Graph reconstruction – a survey. *Journal of Graph Theory*, 1(3):227–268, 1977.

[4] N. Bowler, J. Erde, P. Heinig, F. Lehner, and M. Pitz. Counterexamples to the reconstruction conjecture for graphs with one or countably many ends. In preparation.

[5] R. Diestel. *Graph Theory*. Springer, 4th edition, 2010.

[6] J. Fisher. A counterexample to the countable version of a conjecture of Ulam. *Journal of Combinatorial Theory*, 7(4):364–365, 1969.

[7] J. Fisher, R. L. Graham, and F. Harary. A simpler counterexample to the reconstruction conjecture for denumerable graphs. *Journal of Combinatorial Theory, Series B*, 12(2):203–204, 1972.

[8] F. Harary. On the reconstruction of a graph from a collection of subgraphs. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 47–52, 1964.

[9] F. Harary, A. J. Schwenk, and R. L. Scott. On the reconstruction of countable forests. *Publ. Inst. Math.*, 13(27):39–42, 1972.

[10] C. St. J. A. Nash-Williams. Reconstruction of locally finite connected graphs with at least three infinite wings. *Journal of Graph Theory*, 11(4):497–505, 1987.

[11] C. St. J. A. Nash-Williams. Reconstruction of infinite graphs. *Discrete Mathematics*, 95(1):221–229, 1991.

[12] C. St. J. A. Nash-Williams. Reconstruction of locally finite connected graphs with two infinite wings. *Discrete Mathematics*, 92(1):227–249, 1991.

[13] C. Thomassen. Reconstructing 1-coherent locally finite trees. *Commentarii Mathematici Helvetici*, 53(1):608–612, 1978.

University of Hamburg, Department of Mathematics, Bundesstrasse 55 (Geomatikum), 20146 Hamburg, Germany

E-mail address: nathan.bowler@uni-hamburg.de, joshua.erde@uni-hamburg.de, peter.heinig@uni-hamburg.de, florian.lehner@uni-hamburg.de, max.pitz@uni-hamburg.de