

UNIVERSITY AT STONY BROOK

CEAS Technical Report 797

GENERATING TRANSFINITE AND NONSTANDARD GRAPHS
BY APPENDING AND INSERTING BRANCHES

A.H. Zemanian

January 31, 2002

GENERATING TRANSFINITE AND NONSTANDARD GRAPHS BY APPENDING AND INSERTING BRANCHES

A. H. Zemanian

Abstract — Transfinite graphs can be obtained by “appending” and “inserting” branches infinitely many times into finite graphs. This is an alternative way of generating transfinite graph, which then appear as natural extensions of conventional graphs. This leads to the idea of nonstandard transfiniteness for graphs, electrical networks, and random walks, in which voltages, currents, and probabilities are hyperreal numbers and do not require the many restrictions imposed under standard analysis to ensure their existence.

Key Words: Building transfinite graphs, appending and inserting branches, nonstandard graphs, nonstandard electrical networks, nonstandard random walks.

1 Introduction

This is a companion report to [11]. It restates some results from the earlier work and presents several more ideas and examples beyond those appearing earlier.

Transfinite graphs arose from an attempt to define precisely how connections could be made to the infinite extremities of conventional graphs. This led to a hierarchy of transfinite graphs, with ranks of transfiniteness being indexed by the countable ordinals.

The present work shows that transfiniteness graphs can also be generated simply by “appending” and “inserting” branches infinitely many times. Not all transfinite graphs can be so obtained, but those that cannot are pathological in a way described below. Given a nonpathological transfinite graph, a procedure can be stated for building it through the appending and inserting of branches. On the other hand, the question arises as to what more can be obtained by appending and inserting branches arbitrarily without any guidance from a given transfinite graph. All this suggests nonstandard generalizations of transfinite

networks and random walks, which are then freed from the many restrictions that standard analysis requires.

We use the definitions and terminology found in any of the books [7], [8], [10] on transfinite graphs and networks. Since this new subject is rather unfamiliar, we include herein short comments and phrases to indicate the ideas and meanings of our transfinite terminology. When some terminologies of transfinite graphs are employed below, we will cite specific pages in [8] wherein their definitions can be found.

2 Appending and Inserting Edges

Let \mathcal{G}_0 be a finite graph. An alternative definition of it (one that has been essential for the generalization to transfinite graphs) starts by viewing each branch as being a set of two *elementary tips* (also called *tips of rank -1* or *$\vec{0}$ -tips*) [8, pages 4 and 9]. Then, the set of nodes of \mathcal{G}_0 is taken to be a partition of the set of all the elementary tips of the branches in \mathcal{G}_0 . Under this definition, there are no isolated nodes.

We *append* a new branch to \mathcal{G}_0 by adding each of its elementary tips to a vertex of \mathcal{G}_0 or to a newly created vertex.

We *insert* a new branch into \mathcal{G}_0 by choosing any vertex x of \mathcal{G}_0 having at least two elementary tips, by partitioning it into two nonempty sets of elementary tips to obtain from x two nodes, x_1 and x_2 , and then by adding one elementary tip of b to x_1 and one elementary tip to x_2 . We will say that b has been *inserted into* the vertex x . Thus, an insertion is the inverse of the contraction of a branch. This inverse operation is in general a multivalued function. In order to get a unique result, one must first specify a partition of the set of elementary tips in x into two nonempty sets.

3 Building a Transfinite Graph

We can build a transfinite graph by sequentially appending and inserting branches into finite graphs infinitely many times. If we only append them, but never insert them, we can only obtain a conventionally infinite graph having perhaps nodes of infinite degrees. On the other hand, if we insert infinitely many branches, we can obtain transfinite nodes. We shall

show this here by example and then will state some general theorems, whose proofs appear elsewhere.

Example 3.1. Consider the sequence of finite graphs shown in Fig. 1. We start with two branches, a and b_1 , connected in series as shown. We then insert the branch b_2 into the node between a and b_1 to get the shown series circuit of three branches. Recursively, having built a graph with the branches $a, b_n, b_{n-1}, \dots, b_1$ in series and in that order, we insert branch b_{n+1} into the node between a and b_n . We may think of this process as an example of Aristotle's potential infinity [6, page 3], one that never ends and never yields an infinite construct—as finitists would insist.

However, with apologies to finitists, let us complete the process to get an actual infinity, namely, the transfinite graph (of rank 1) shown at the bottom of Fig. 1. It consists of a one-ended path of branches, $b_1, b_2, b_3 \dots$, which is connected at its infinite extremity (a 0-tip) [8, page 30] to the branch a through the 1-node (represented by the circle) consisting of an elementary tip of a and the 0-tip of the said path.

Let us emphasize that no conventional node of finite graph theory exists at the infinite extremity of the path of b_k branches. Instead, that path has as its extremity a new entity, called a “0-tip” [8, page 20]. In particular, a 0-tip is an equivalence class of one-ended paths of branches, where two such paths are considered equivalent if they are identical except for finitely many branches. Thus, a conventionally infinite graph may (or may not) have a set of 0-tips. To make connections at those 0-tips, we partition the set of them into nonempty subsets. To each subset, we may (but need not) append exactly one 0-node (i.e., conventional node). The result is defined to be a node of rank 1, that is a 1-node [8, page 22]. Such a node is denoted by the small circle in the bottom graph of Fig. 1; it connects the 0-tip of the path of b_k branches to an elementary tip of branch a . That bottom graph is called a 1-graph [8, page 23]. In no way is this a conventional graph. It is a new kind of graph encompassing transfiniteness in an essential way; indeed, the branches a and b_1 are connected only through a transfinite path, not a finite one.

Let us also point out that transfinite graphs are different from other generalizations of graphs, such as hypergraphs and graphical matroids, although these latter standard

constructs may perhaps be generalized to relate to transfinite graphs someday. \square

Here we have a very simple example of how a transfinite graph can be built. Despite its simplicity, such graphs have been overlooked during the two and a half centuries of graph theory up to a dozen years ago.

Example 3.2. A transfinite graph of rank 2 can be built by inserting branches in a more complicated way. The final result is shown in Fig. 2. We start with the series connection of a and $b_{1,1}$. Then, we insert branches in accordance with a tracing of the array shown in Fig. 3, which in turn can be described by an index $m = p + q$, where each branch is indexed according to $b_{p,q}$. For $m = 3$, we insert $b_{1,2}$ and then $b_{2,1}$ to get the series connection $a, b_{2,1}, b_{1,2}, b_{1,1}$. For each $m = 3, 4, 5, \dots$ in sequence, we insert $m - 1$ branches as follows: $b_{1,m-1}, b_{2,m-2}, \dots, b_{m-1,1}$, this too being done sequentially. With respect to the branches that have been previously inserted, we insert each new branch in accordance with its location shown in Fig. 2. Upon completion of this infinite process, we get the 2-graph [8, page 34] of Fig. 2.

More specifically, the process ends in a series connection of infinitely many one-ended 0-paths (i.e., one-way infinite conventional paths). The 0-tip of each one-ended 0-path is shorted to the first 0-node of the next 0-path by means of a 1-node. That infinite series connection of 0-paths is a path of higher transfiniteness, a 1-path [8, page 28], and its extremity can be defined as a 1-tip [8, page 30]. Then, that 1-tip along with a conventional node of branch a make up a node of higher rank, a 2-node [8, page 30]. That 2-node is denoted by the double circle in Fig. 2. Altogether, we have a particular case of a 2-graph [8, page 31]. \square

Example 3.3. More substantial transfinite graphs can be obtained by appending branches as well as inserting them. Consider the ladder graph of Fig. 4 connected at its infinite extremity to a series connection of branches β_1 and β_0 . This graph has two kinds of infinities: a conventional node z^0 (i.e. a 0-node) of infinite degree and a transfinite node w^1 (i.e., a 1-node). To build this graph, we start with the loop b_0, β_1, β_0 . We then insert b_1 between b_0 and β_1 , and then append b_2 to the node between b_1 and β_1 and the node between b_0 and β_0 . Continuing in this way, we insert b_3 , append b_4 , etc. Here too, we only

obtain a finite graph at each step of this process, but, with the completion of the process, a transfinite graph springs into view. \square

Example 3.4. Not all transfinite graphs can be so built. Consider the 1-graph of Fig. 5. The 1-node x_0^1 consists of the 0-tip of the path of even-numbered branches and a 0-node of branch β_0 . The 1-node x_1^1 consists of the 0-tip of the path of odd-numbered branches and a 0-node of branch β_1 . (There are other 0-tips, uncountably many of them, corresponding to one-ended paths that pass through both even-numbered and odd-numbered branches infinitely often, but no connections are made to them in this graph.) If we attempt to build this graph through an expanding sequence of finite graphs, the branches β_0 and β_1 will at some step become incident to the same 0-node and will remain so for all subsequent steps. As a result, those branches will be incident to the same 1-node when the process is completed, as is shown in Fig. 6. The difficulty can be described as follows: The two paths considered here meet infinitely often (their 0-tips are said to be “nondisconnectable”), but their 0-tips are placed in different 1-nodes in Fig. 5. \square

Upon defining nondisconnectable tips as in [8, page 54] or [10, page 31], we can state a more general result, whose proof appears in [12]. Any graph that can be obtained by appending and inserting branches into finite graphs will be called *buildable*.

Theorem 3.5. *A necessary condition for a transfinite graph to be buildable is the following: If two tips of the transfinite graph are nondisconnectable, then they are either shorted together (i.e., are in the same node) or at least one of them is open (i.e., no connection is made to that tip).*

On the other hand, we can define appending and inserting branches into transfinite graphs in much the same way as was done in Sec. 2 for finite graphs. In this case, it is possible to generate a transfinite graph having two nondisconnectable tips that are neither shorted together nor open.

4 Restorable Transfinite Graphs

It is quite clear how to build the simple transfinite graphs of Examples 3.1, 3.2, and 3.3. More generally, however, we need a specific procedure for building an arbitrarily given

transfinite graph \mathcal{G}^ν from finite graphs, if indeed \mathcal{G}^ν can be so built. Here is a possible way.

First, we assign to every edge b of \mathcal{G}^ν a *conductance* g_b , which is a positive finite real number. We *short* b by replacing g_b by 0, and we *open* b by replacing g_b by ∞ . These replacements yield *shorted* and *opened* edges, respectively. On the other hand, we restore a shorted or opened edge by reassigning the original conductance g_b to b . An edge b for which $0 < g_b < \infty$ will be called *conductive*.

Procedure 4.1.

1. Number all the edges using the natural numbers $j = 0, 1, 2, \dots$
2. For each nonopen nonelementary tip,¹ choose one of its representative paths and short every edge in that path. Open all other edges.
3. Restore edges finitely many at a time, starting with the lowest numbered edges and restoring others in accordance with increasing edge numbers. Number these steps of restoring finitely many edges at a time using the natural numbers $n = 0, 1, 2, \dots$

There are many ways of following Procedure 4.1 because of the different ways of numbering edges, choosing shorted representative paths, and restoring edges finitely many at a time.

The reason for shorting representative paths of nonopen nonelementary tips is to maintain the connections provided by the transfinite nodes that embrace those tips once the restoration is completed. We say that a node *shorts together* the tips it embraces.

The difficulty that arises at this point is that more tips might be shorted together after restoration than were shorted together in the original transfinite graph. An illustration of this is provided by the graph of Fig. 5. Having shorted a representative path for the 0-tip of the path of even-numbered edges and also for the path of odd-numbered edges, those two tips will remain shorted together even after the restoration is completed, and the different 1-graph of Fig. 6 will result. This is another way of viewing the difficulty described

¹A tip of any rank is called *nonopen* if it is a member of a nonsingleton node, that is, if it is shorted to at least one other tip of the same rank and/or to a node of equal or lower rank. A *nonelementary tip* is a tip of rank 0 or higher.

in Example 3.4. If, on the other hand, no additional shortings of tips occur so that the restored graph is the same as the original graph, that graph will be called *restorable*.

A necessary and sufficient condition can be stated for the restorability of a graph as follows. At each step of the restoration process of Procedure 4.1.3 (i.e., part 3 of Procedure 4.1), we define a *proximity* to be a maximal set of nodes that are connected together through paths of shorted edges. Also, any node having no incident shorted edges will be called a *singleton proximity*. As edges are restored, the proximities shrink in general. Two proximities will be called *eventually disjoint* if at some restoration stage of Procedure 4.1.3, those proximities are disjoint (and thus disjoint at all later stages, too). The following can be proven [12].

Theorem 4.2. *\mathcal{G}^ν is restorable if and only if it is possible to choose shorted representative paths for every nonopen nonelementary tip in \mathcal{G}^ν such that, for every pair of maximal² nodes, the corresponding sequences of proximities in which those two nodes lie are eventually disjoint.*

We can derive a sequence $\{\mathcal{G}_n^0\}_{n=0}^\infty$ of finite graphs that fill out a restorable graph by doing the following at each stage of the restoration process of Procedure 4.1.3. Remove all open edges. Also, remove all shorted edges, but in this case coalesce the incident nodes of all shorted edges of each proximity into a single 0-node after those edges have been removed.³ The result is a finite graph \mathcal{G}_n^0 .

Obtaining a sequence $\{\mathcal{G}_n^0\}_{n=0}^\infty$ by appending and inserting edges to generate a transfinite graph as in Secs. 2 and 3 and, on the other hand, doing so by using Procedure 4.1 on a given transfinite graph are two different processes. However, the following can be proven by relating appending and inserting edges to shorting, opening, and restoring edges [12].

Theorem 4.3. *Every restorable graph is buildable.*

We have not asserted however that every buildable graph is restorable. Indeed, it seems possible that certain constructs might arise that cannot be obtained through the restrictive procedures employed in [7, Chap. 5], [8, Chap. 2], and [10, Chap. 2]. If we build a graph without referring to a particular transfinite graph for guidance, the result of the

²A node is called *maximal* if it is not a member of a node of higher rank [8, page 32].

³This is the transfinite generalization of a contraction of a graph.

completed infinite process is obscure. For instance, is it possible to obtain a transfinite graph of any countable rank no matter how large that rank? More generally, is there any way of constructing a transfinite graph of rank \aleph_1 ? After all, the natural numbers have been extended to the countable ordinals and then to the uncountable ordinals. We have been able to extend finite graphs to transfinite graphs of at least some countable ranks. How can transfinite graphs of uncountable ranks be built?

5 Nonstandard Graphs and Networks

In Sec. 4 we completed the potential infinity of an expanding sequence $\{\mathcal{G}_n^0\}_{n=0}^\infty$ of finite graphs, obtaining thereby a transfinite graph \mathcal{G}^ν . However, there are many ways of following Procedure 4.1 to obtain \mathcal{G}^ν through many different sequences of finite graphs. This suggests that nonstandard graphs can be defined in much the same way as the hyperreal numbers are defined as equivalence classes of sequences of real numbers; see, for instance, any of the books [1], [2], [3], [4], all of which are expositions of the seminal work [5] (In the following, we use the terminology employed in [1].)

Specifically, given a nonprincipal ultrafilter \mathcal{F} on the set of natural numbers $\{0, 1, 2, \dots\}$, two sequences of finite graphs, $\{\mathcal{G}_n^0\}_{n=0}^\infty$ and $\{\mathcal{H}_n^0\}_{n=0}^\infty$ will be called *equivalent* (modulo \mathcal{F}) if, for some $F \in \mathcal{F}$, $\mathcal{G}_n^0 = \mathcal{H}_n^0$ (i.e., \mathcal{G}_n^0 and \mathcal{H}_n^0 are isomorphic) for every $n \in F$. Then, such an equivalence class of sequences of graphs is taken to be a nonstandard graph $\mathcal{G}_{\text{ns}}^\nu$, and any sequence, say, $\{\mathcal{G}_n^0\}_{n=0}^\infty$ in that equivalence class can represent $\mathcal{G}_{\text{ns}}^\nu$.

However, the method of Sec. 4 produces many sequences of finite graphs that fill out \mathcal{G}^ν , and those sequences are not all equivalent. Instead, the set of all those sequences is partitioned as just stated into equivalence classes, and thus there are many nonstandard versions of \mathcal{G}^ν . Analogous with the halo of hyperreal numbers around each real number, we now have a “halo” of nonstandard graphs around \mathcal{G}^ν .

These ideas lead us on to nonstandard versions of transfinite electrical networks and transfinite random walks. We can convert any transfinite graph \mathcal{G}^ν into a transfinite network \mathcal{N}^ν by assigning conductance values to every edge along with sources inserted into some edges or applied to some nodes. The advantage of this is that all the many restrictions

needed to ensure the existence of standard voltages and currents in transfinite electrical networks or the existence of standard probabilities for random walks can now be discarded. These quantities will now always exist, albeit as hyperreal numbers. Furthermore, many standard theorems concerning electrical networks and random walks can be immediately lifted into a nonstandard setting. See [11] and [12]. The present method of generating nonstandard electrical networks is more general than that proposed in [9] and subsumes that former method.

References

- [1] R. Goldblatt, *Lectures on the Hyperreals*, New York: Springer, 1998.
- [2] A.E. Hurd and P.A. Loeb, *An Introduction to Nonstandard Real Analysis*, New York: Academic Press, 1985.
- [3] H.J. Keisler, *Foundations of Infinitesimal Calculus*, Boston: Prindle, Weber, and Schmidt, 1976.
- [4] T. Lindstrom, An invitation to nonstandard analysis, in *Nonstandard Analysis and Its Applications* (Editor: N. Cutland), New York: Cambridge University Press, 1988.
- [5] A. Robinson, *Nonstandard Analysis*, Amsterdam: North-Holland Publishing Co., 1966.
- [6] R. Rucker, *Infinity and the Mind*, Boston: Birkhauser, 1982.
- [7] A.H. Zemanian, *Infinite Electrical Networks*, New York: Cambridge University Press, 1991.
- [8] A.H. Zemanian, *Transfiniteness—for Graphs, Electrical Networks, and Random Walks*, Boston: Birkhauser, 1996.
- [9] A.H. Zemanian, Nonstandard Electrical Networks, *Graph Theory Notes of New York*, vol. XXX (1996), 30-34.
- [10] A.H. Zemanian *Pristine Graphs and Permissive Electrical Networks*, Boston: Birkhauser, 2000.

- [11] A.H. Zemanian, *Nonstandard Electrical Networks*, CEAS Technical Report 783, University at Stony Brook, August 15, 2000.
- [12] A.H. Zemanian, *Nonstandard Transfinite Graphs and Random Walks on Them*, CEAS Technical Report 785, University at Stony Brook, October 17, 2000.

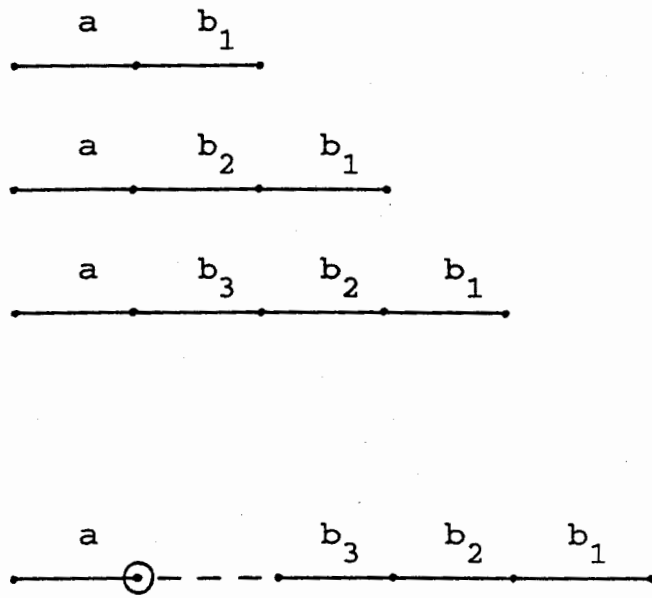


Fig. 1 - Zemanian

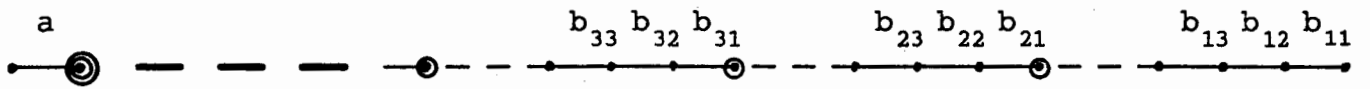
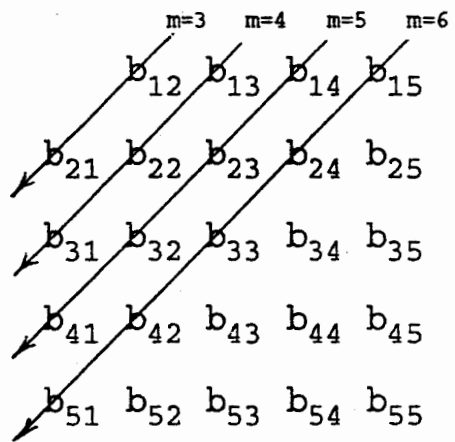


Fig. 2



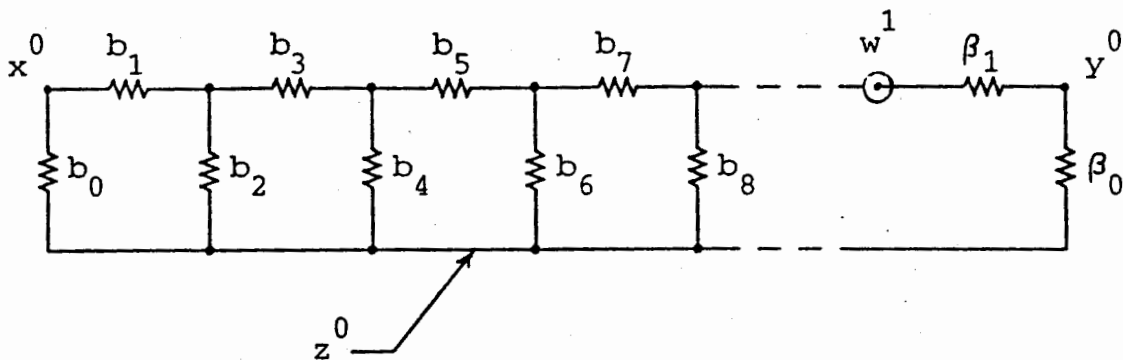


Fig. 4 - Zenerman

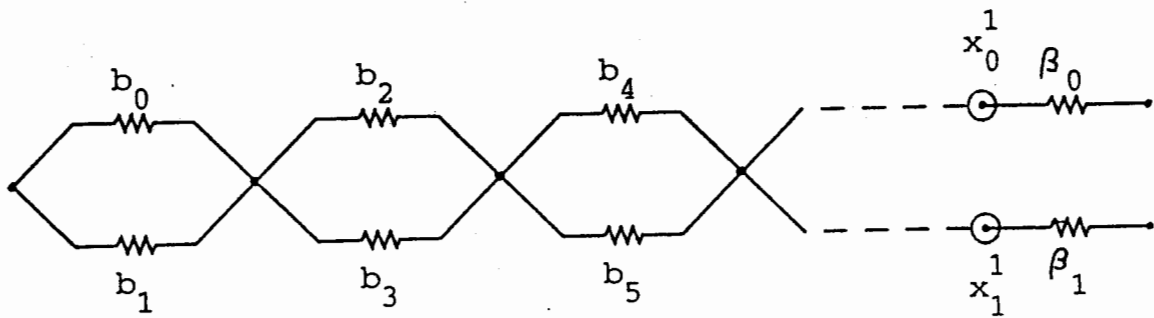


Fig. 5 - Zener

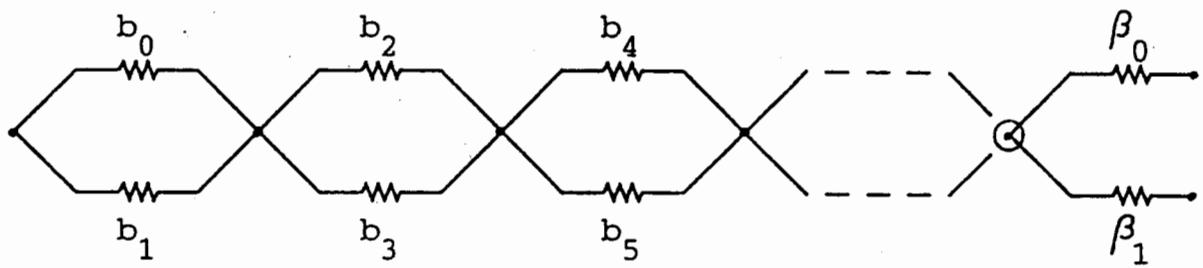


Fig-6 - Ladder network