# CONSTRUCTING NEW FAMILIES OF NESTED RECURSIONS WITH SLOW SOLUTIONS

A. ISGUR, R. LECH, S. MOORE, S. TANNY, Y. VERBERNE, AND Y. ZHANG

Abstract. A recurring theme in nested recursions research has been the search for recursion *families*. By a recursion family we mean a collection of recursions with a common or at least highly similar structure, and where, with appropriate (but usually different) initial conditions for each recursion, their respective solutions behave similarly in key respects. Our key result is a general method for generating a family of recursions with slow solutions from **any** nested recursion of the form either $R(n) = R(n - s_1 - R(n - a_1)) + R(n - s_2 - R(n - a_2))$ (a two term generalized Conolly recursion) or $R(n) = R(n - s_1 - R(n - a_1)) + R(-t_1 + R(n - b_1))$ (a generalized Conway recursion) so long as the recursion with which we start, together with its initial conditions, has a known slow solution. We apply this method to discover new families of recursions with slow solutions based on the well-known Hofstadter $V$ and Conway recursions, respectively.

## 1. Introduction

In this paper, all values of the parameters and variables are integers.

A nested recurrence relation (also called a meta-Fibonacci recursion) is any recursion where some argument contains a term of the recursion. Some early examples of nested recursions that have spawned considerable interest include: (1) Hofstadter's enigmatic $Q$ defined by $Q(n) = Q(n - Q(n-1)) + Q(n - Q(n-2))$ with $Q(1) = Q(2) = 1$ [18]; (2) Conway's[1] famous challenge recursion $A(n) = A(n - A(n-1)) + A(A(n-1)), A(1) = A(2) = 1$ [11, 28, 29]; and (3) Conolly's[2] sequence defined in [10] by the recursion $C(n) = C(n - C(n-1)) + C(n - 1 - C(n-2)), C(1) = C(2) = 1$.

A *solution* to a nested recursion is any sequence that satisfies the recursion together with its initial conditions. For each of the above examples a solution exists so long as all the arguments in the terms of the recursion remain positive for all $n$ past the initial conditions. For example, for $Q(n)$ we require that for every $n > 2$ both $n - Q(n-1) > 0$ and $n - Q(n-2) > 0$ so that the recursion remains well-defined and successive values of the recursion can be computed. If this turns out to be false then the recurrence has no solution and we say that it "terminates" (or "dies"). For the above examples it is evident that the solution, if it exists, is unique.[3] All the nested recursions we analyze later in this work have at most one solution.

For any nested recursion $R(n)$, we use either $R(n)$ or $R$ to refer both to the recursion *together* with its initial conditions and the (finite or infinite) sequence they generate. As

*Date*: Sep 18, 2015.

2000 *Mathematics Subject Classification.* Primary 11B37; Secondary 11B83.

*Key words and phrases.* nested recursion, Hofstadter sequence, V sequence, Conway sequence, slowly growing solution, family of nested recursions.

[1]Some authors, for example, [31], call this the Conway-Hofstadter sequence.

[2]This is sometimes called the Conolly-Hofstadter sequence; see [23].

[3]For some nested recursions this is not the case. For example, the very unusual nested recursion $g(g(n) + n) = 2g(n) + n, g(1) = 1, g(2) = 3$ defined by Golomb [15] has infinitely many solutions [6, 7].

we discuss further below, the initial conditions play an crucial role in whether or not the recursion has a solution, and if so, in the solution properties.

It is well known that the sequences generated by nested recursions can display a very wide range of behaviour.[4] In some cases the solutions are very well behaved with discernible structure (see, for example, [1, 4, 5, 8, 9, 14, 15, 20, 22, 24]). In others, the sequence generated by the nested recursion together with its initial conditions appears to be quite chaotic, but nonetheless displays some evidence of underlying structural regularities. Hofstadter's $Q$ is the most famous example of such a recursion (for details see [32] and [12]). It is still not known whether or not $Q$ has a solution, although the first twelve billion values of $Q$ have been computed.

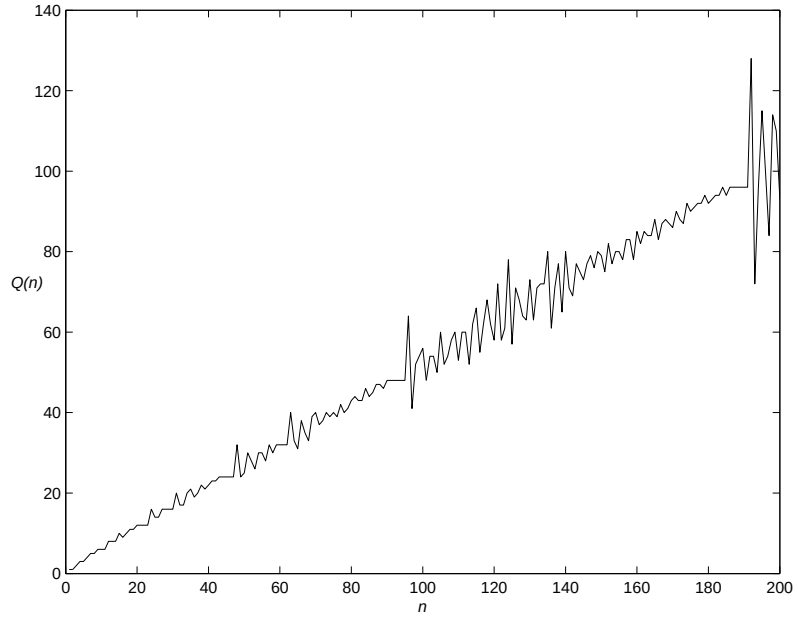


FIGURE 1.1. Graph of first 200 values of Hofstadter's $Q(n)$ with ICs 1, 1

By contrast, each of $A(n)$ and $C(n)$ has a beautiful monotone nondecreasing solution with the following properties: the solution begins with 1, all the differences between successive terms are either 0 or 1, and the solution tends to infinity. We call such a sequence *slowly growing* or *slow*.

It is natural to describe a slowly growing sequence $\sigma(n)$ by its *frequency sequence* $\phi_\sigma(w)$, which counts the number of times that $w > 0$ occurs in $\sigma(n)$. For the Conolly sequence, the frequency sequence $\phi_C(m)$ equals $r_m = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, \ldots$, the so-called "ruler function".[5] Observe (from $\phi_C(m)$) that for every $\alpha$ the value $2^\alpha$ appears $\alpha+1$ times in $C$. The frequency sequence of the Conway sequence $A(n)$ is more complex but nonetheless is fully understood. For details see [28, 29].

Over the past 25 years two related questions have been the focus of considerable interest in nested recursion research: given a nested recursion with a known solution, what kinds of changes to the parameters and/or initial conditions lead to a new recursion with a solution?

[4]See [20], especially chapter 1, for a detailed discussion and more comprehensive bibliography.

[5]The ruler function $r_m$ is defined as one plus the 2-adic valuation of $m$ (the exponent of 2 in the prime factorization of $m$).

In such a situation, how do the properties of the new solution relate to those of the solution to the original recursion?

The radically different behaviour of the Conolly $C$ and Hofstadter $Q$ recursions described above underscores the fact that sequences generated by very similar looking recursions together with identical initial conditions may behave extraordinarily differently. At the same time, this need not always be the case: for example, the solution to $T(n) = T(n - 1 - T(n-1)) + T(n-2-T(n-2)), T(0) = T(1) = T(2) = 1$, a close variant of the Conolly recursion, is almost identical to that for $C$. The only difference between the two slow solutions is that each power of 2 appears precisely one more time in $T(n)$ (see [35]).

It is well known that the behaviour of the sequence generated by a nested recursion is highly sensitive to the choice of the initial conditions. So a key element of these questions is the selection of the initial conditions for the modified recursion being explored. Sometimes, as in the above examples, the initial conditions for the original recursion may provide a useful guide. However, as we discuss further below, in many situations there is a wide range of plausible alternatives and not necessarily any "natural" choice.

Exploration of these questions has shifted the focus of nested recursion research to parameterized versions of individual recursions. In an attempt to understand $Q$, Hofstadter and Huber [19] define the two-parameter generalization $Q_{r,s}(n) = Q_{r,s}(n - Q_{r,s}(n-r)) + Q_{r,s}(n - Q_{r,s}(n-s))$, with $r < s$ and initial conditions $Q_{r,s}(1) = \cdots = Q_{r,s}(s) = 1$. Their empirical explorations led to their still unresolved conjecture that the only $(r, s)$ pairs for which $Q_{r,s}$ remains well-defined are (1,2) (the original $Q$), (2,4) (called $W$) and (1,4) (called $V$).[6] Further, they observe that each sequence behaves very differently: the $W$ sequence, if it exists, is much wilder than $Q$ while the $V$ sequence appears to be slow with a very complex frequency function (see [4] where this conjecture is proved).

Finding different recurrences with similar behaviour, such as $C$ and $T$, is very important and has been an essential key to the substantial progress that has been made in the relatively new field of nested recurrence relations. In some situations, analysis of the parameterized recursion that describes such a situation has led to improved understanding of the links between the behaviour of the solutions and the structure, parameters and choice of initial conditions.

In [27], Jackson and Ruskey generalize the Conolly recursion by introducing the shift parameter $s$: $C_s(n) = C_s(n-s-C_s(n-1)) + C_s(n-(s+1)-C_s(n-2))$, with $s+2$ initial conditions $C_s(1) = C_s(2) = \cdots = C_s(s+1) = 1, C_s(s+2) = 2$. They apply a "tree-based" solution method to prove that for every $s$ the $n^{th}$ term of $C_s(n)$ counts the number of leaves with label less than or equal to $n$ in a certain infinite, labeled binary tree (see Section 2 for further details). Their choice of the initial conditions for $C_s(n)$ necessarily derives directly from the structure and labeling of the infinite binary tree that provides the combinatorial interpretation for the solution to the recursion, since the initial conditions are the first few terms of the solution sequence. The counting interpretation for $C_s(n)$ explains why, as we indicated above, the solutions for $s = 0$ (the Conolly recursion $C$) and $s = 1$ (the $T$ recursion) are so similar.

$C_s(n)$ is an example of a collection of parameter-related recursions that have solutions that all behave very similarly in terms of the specified parameter. To emphasize this we call such a collection a *recursion family*.

Significant extensions of the tree-based solution method have been developed to prove the existence of other families of nested recursions, many of which are related to the Conolly

[6] They also investigate other choices for the initial conditions. We discuss this later in the Introduction.

recursion (see, for example, [26, 14, 23]). In each case the solution to a parameterized recursion is shown to have a combinatorial interpretation in terms of an infinite labelled tree whose structure and labeling is determined by the recursion and its parameters. As above, this counting interpretation for the $n^{th}$ term is typically in terms of the number of leaves or some variant, such as leaf labels, up to the label $n$. Because of the tree-based counting interpretation for the solution it follows that for different values of the parameters the solutions are slow and behave similarly. It follows that these recursions form a family with slow solutions. Finally, and very importantly, the tree interpretation for the solution makes completely transparent the choice of the initial conditions for the recursion: the initial values for the recursion can be read off the tree since the solution to the recursion counts certain features embedded in the tree.

In those situations where the tree-based solution method can be applied, it has proved a powerful technique for identifying and solving new families of recursions with slow solutions. Where no tree interpretation for the solution to a recursion is known, the search for recursion families related to that recursion has been far less fruitful. This is due, at least in part, to the difficulty in identifying the appropriate choice of initial conditions for the modified recursion.[7]

The following example is illustrative. In [19] Hofstadter and Huber examine empirically the recursion $V'(n) = V'(n - V'(n-2)) + V'(n - V'(n-8))$ in an attempt to identify a recursion family with solutions that behave like the solution to Hofstadter's $V$ recursion $V(n) = V(n - V(n-1)) + V(n - V(n-4)$.[8] The four initial conditions for $V$ are either $V(1) = ... = V(4) = 1$ or $V(1) = 1, V(2) = 2, V(3) = 3, V(4) = 4$ (with the second choice the resulting sequence is advanced three terms but otherwise the same). Not sure what initial conditions to use for $V'$ they report on the results from two alternatives considered: the eight initial conditions consisting of all 2s, that is, $V'(1) = ... = V'(8) = 2$ and the eight initial conditions 1,2,..,7,8.[9]

Neither of these two choices for the initial conditions for $V'$ results in a solution whose behaviour has interesting properties resembling those of $V$. With the first choice all the values are just twice those of $V$.[10] With the second choice the sequence that is generated seems to have some interesting properties but they are not like those of $V$ (for example, the sequence is not slow or even monotone); there is no known proof to date that a solution exists, that is, that this sequence doesn't terminate. In fact, a $V$-like sequence is generated by the $V'$ recursion together with either of the *nine* initial conditions 1,2,..,7,8,9 or 1,2,2,2,2,2,2,2,3, neither of which seem to be a priori natural choices. We discuss this unexpected result further in Section 4.

In this paper we describe a methodology for identifying new families of recursions in a very general situation. Our approach automatically specifies an appropriate choice for the initial conditions for the recursions in the new family that leads to the desired behaviour of the solutions.

More precisely, our key result is the following: suppose we have *any* nested recursion $R$ of the form either $R(n) = R(n - s_1 - R(n - a_1)) + R(n - s_2 - R(n - a_2))$ (we call this a two term generalized Conolly recursion) or $R(n) = R(n - s_1 - R(n - a_1)) + R(-t_1 + R(n - b_1))$ (a generalized Conway recursion), together with initial conditions so that the recursion has

[7]There have been some successes, however. See, for example, [15, 17, 9, 16].

[8]See [4] for details about the solution to $V$.

[9]Another natural choice for the initial conditions might be eight 1s, but this does not lead to a solution as the sequence terminates.

[10]This is a special case of Theorem 2.2 in [14].

a known slow solution. For any positive integer $j$, define the so-called "$j$-related" recursion $R_j$ obtained by multiplying all of the parameters of $R$ by $j$. Then we can specify appropriate initial conditions (determined by the initial conditions for $R$ and the value of $j$) so that $R_j$ defines a family of recursions whose solutions are slow and behave like the solution to $R$.[11]

Notice that while in general we don't have any knowledge of an underlying tree interpretation for the solution to $R$, the methodology for specifying the initial conditions for $R_j$ and identifying its solution is based upon a purely parametric tree-based formula, namely, the relation between the binary tree interpretation for the slow solution to the $j$-related version of the recursion $C_s$ (formed by multiplying all the parameters of the $C_s$ recursion by $j$) and that for the solution to $C_s$. In a sense it is almost as if $R$ has an unknown underlying "virtual" tree to which the parametric formula appeals. In our view this is a surprising and somewhat mysterious result.

The outline of the remainder of this paper is as follows: in the following section we derive the purely parametric formula for the relation between the binary tree interpretation for the slow solution to the $j$-related version of the recursion $C_s$ and that for the solution to $C_s$. In Section 3 we apply this parametric characterization to derive the slow solution to $R_j$, the $j$-related version of an *arbitrary* two term Conolly or Conway generalized recursion $R$ for which we have a known slow solution; in so doing the parametric characterization also specifies the appropriate initial conditions to use for $R_j$. That is, we start with any specific two term Conolly or Conway generalized recursion $R$, together with its initial conditions, for which a slow solution is known. To this slow solution for $R$ we apply the parametric map derived in Section 2. The resulting sequence solves the recursion $R_j$, together with initial conditions specified by the map. In this way we identify a new family of nested recursions with slow solutions related to $R$. We conclude in Section 4 by applying this methodology to identify a new family of recursions with slow solutions based on each of the well-known Hofstadter $V$ and Conway $A$ recursions, respectively.

## 2. A Parametric Formula from the Tree-Based Methodology

We begin with a very brief explanation of the tree-based solution methodology. For $s$, $n$ natural numbers, we define the *tree* $T_s(n)$ (or $T(n)$ when it won't cause confusion to omit the subscript) with an infinite number of nodes and a finite number $n$ of labels, as follows. First, draw an infinite binary tree in preorder (that is, from the bottom left). All nodes on the extreme left *except* the very first node on the bottom left are *supernodes*. All nodes on the bottom level of $T(n)$ (including the bottom leftmost node) are *leaves*. Any other node is a *regular* node. Place the natural numbers 1 through $n$ into the nodes of the $T(n)$ in preorder as labels, with one label going into each regular node and leaf, and $s$ labels going into each supernode. See Figure 2.1 for the case $s = 2$ and $n = 35$; in the diagram circles are used to depict the leaves and the regular nodes, while rectangles are used for the supernodes.

Note that if $n$ is a label on a supernode, that supernode might be only partly full (for example, if $s = 2$, and $n = 5$, then the second supernode in Figure 2.1 would have only one

[11] In fact our proofs extend naturally to the more general recursion $R(n) = \sum_{i=1}^{k_1} R(n - s_i - R(n - a_i)) + \sum_{i=1}^{k_2} R(-t_i + R(n - b_i))$ with arbitrary $k_1$ and $k_2$ and appropriate initial conditions. But to date, allowing for the more general $(k_1, k_2)$ values, the only recursions with slow solutions we know of have $(k_1 > 2, k_2 = 0)$. For such recursions a tree-based interpretation for the slow solution is already known, so a virtual tree interpretation is unnecessary. For this reason we restrict ourselves to the two above simpler recursion forms.

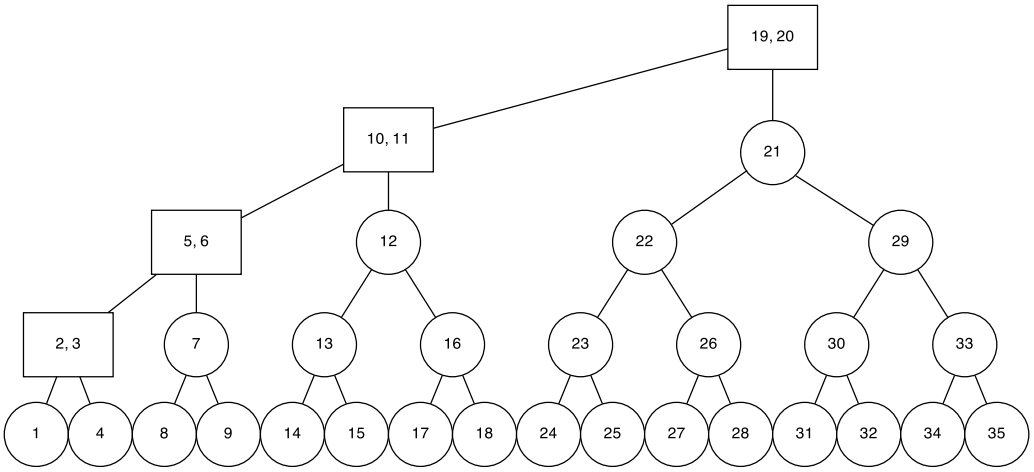


FIGURE 2.1. Solution of $C_2(n) = C_2(n-2-C_2(n-1))+C_2(n-3-C_2(n-2))$ with initial conditions 1, 1, 1, 2 counts labels in leaves of $T_2(n)$

label (namely, 5) so would be only partly full). When we want to refer to the unlabelled tree (that is, $T(n)$ without regard to the number or placement of labels), we call it the "skeleton."

Define the leaf label counting function $L_s(n)$ to be the number of labels in the leaves in $T_s(n)$. For example, in Figure 2.1, $L_2(35) = 16$ and $L_2(13) = 4$. Observe that by definition the label counting sequence for $T_s(n)$ is slow.

In [27] Jackson and Ruskey prove that the the $n^{th}$ term of the solution of the recursion $C_s(n) = C_s(n - s - C_s(n-1)) + C_s(n - (s+1) - C_s(n-2))$, with $s+2$ initial conditions $C_s(1) = C_s(2) = \cdots = C_s(s+1) = 1, C_s(s+2) = 2$ counts the number of leaves with label less than or equal to $n$ in the tree $T_2(n)$. That is, the label counting function $L_s(n)$ solves the recursion $C_s(n)$ with the specified initial conditions; for example, the reader can readily confirm from the recursion that $C_2(13) = 4$ and $C_2(35) = 16$, corresponding to the values $L_2(13)$ and $L_2(35)$, respectively. Observe that the first $s+2$ terms of the label counting function $L_s(n)$ match the given initial conditions for the recursion $C_s(n)$.

It is shown in [26] that the nested recursion $C_{s*j}(n) = C_{s*j}(n-sj-C_{s*j}(n-j))+C_{s*j}(n-(s+1)j-C_{s*j}(n-2j))$, which is obtained by multiplying all of the parameter values in $C_s(n)$ by $j$, defines a family of recursions (it is important to note that $C_{s*j} \neq C_{j*s}$ in most cases).[12] Each recursion in this family, with appropriate initial conditions, has a slow solution that is obtained by extending the Jackson-Ruskey tree-based methodology.[13] In this extended methodology the solution is shown to be the label counting function $L_{s*j}(n)$ for the tree $T_{s*j}(n)$, which is formed by replacing every label in $T_s(n)$ by $j$ labels. The (as yet unstated) initial conditions required for the recursion for $C_{s*j}(n)$ are the first $(s+2)j$ terms of the

[12]Observe that the $C_{s*j}$ notation for this recursion used here differs slightly from how this recursion would be identified in [26]. There we would write $C_{sj,j}$ rather than $C_{s*j}$. We adopt the alternate notation here to emphasize our focus in this work on multiplying all the parameters in the recursion of interest, here $C_s$, by $j$.

[13]See, in particular, Theorem 3.10 in [26], where the solutions to considerably more general recursions are derived.

label counting sequence $L_{s*j}(n)$ of the tree $T_{s*j}(n)$ (recall that $L_{s*j}(n)$ counts the number of labels less than or equal to $n$ in the leaves of $T_{s*j}(n)$). In a sense the specification of the initial conditions for which we can solve for $C_{s*j}(n)$ occurs after the fact. See Figure 2.2, where we illustrate the tree-based solution to $C_{s*j}(n)$ for $s = 2$ and $j = 3$.

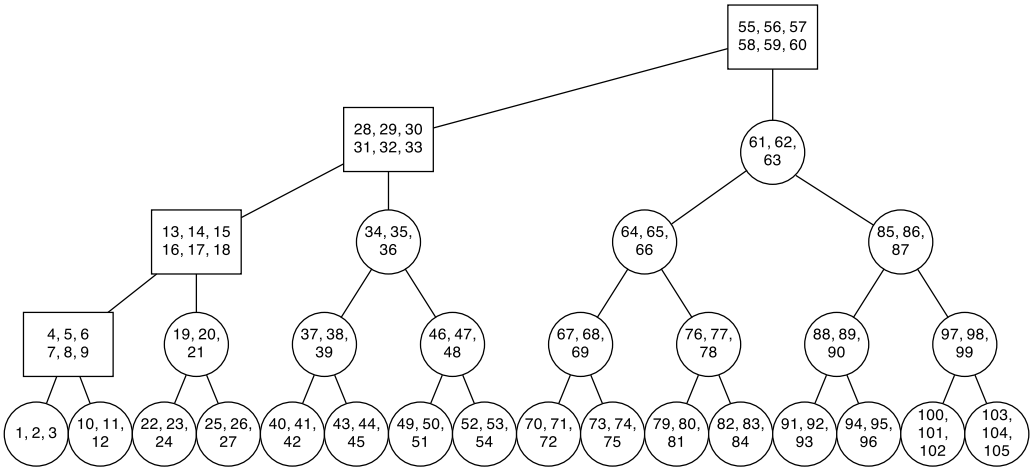


FIGURE 2.2. Solution of $C_{2*3}(n) = C_{2*3}(n-6-C_{2*3}(n-3)) + C_{2*3}(n-9-C_{2*3}(n-6))$ with 12 initial conditions 1, 2, 3, 3, 3, 3, 3, 3, 3, 4, 5, 6 counts labels in leaves of $T_{2*3}(n)$.

By comparing the labeling on each of the trees $T_s(n)$ and $T_{s*j}(n)$, both of which have identical skeletons, we observe a fundamental connection between the frequency functions of their respective label counting sequences $L_s(n)$ $(= C_s(n))$ and $L_{s*j}(n)$ $(= C_{s*j}(n))$.

**Proposition 2.1.** *Let $\phi_s(m)$ (respectively, $\phi_{s*j}(m)$) be the frequency of $m$ in the label counting sequence $L_s(n)$ (respectively, $L_{s*j}(n)$). Let $m = jw + a$, where $0 \le a < j$. For $a = 0$, $\phi_{s*j}(m) = \phi_{s*j}(jw) = j\phi_s(w) - (j-1)$. Otherwise $\phi_{s*j}(m) = 1$.*

*Proof.* In [20] a formula for the frequency function of generalized Conolly sequences is proved (see Theorem 4.4.2). Proposition 2.1 is a direct consequence of the explicit formulas for these frequency functions that can be derived from this formula. In order to emphasize the multiplicative relationship between the parameters and to provide additional insight into the relationship between the structures of the related trees we include an alternative direct proof.

We first prove the formula for $a = 0$. By the definition of the label counting function and the structure of the labeled tree $T_{s*j}(n)$ the first occurrence of the label count $jw$ in $L_{s*j}(n)$ is with the last label in the $w^{th}$ leaf node ($T_{s*j}(n)$ has $j$ labels in every leaf node). The skeletons of $T_{s*j}$ and $T_s$ are the same, so this node is also the $w^{th}$ leaf node $T_s$. It follows by the definition of the frequency function of the label counting sequence that in the tree $T_s$ there are $\phi_s(w) - 1$ labels, and thus nodes (since in $T_s$ there is one label in each node) following the $w^{th}$ leaf node before arriving at the next leaf node. Thus, in $T_{s*j}(n)$ there must be $j(\phi_s(w) - 1)$ labels following the $jw^{th}$ label before arriving at the first label in the next leaf node. It follows that in $T_{s*j}(n)$ the label count remains at $jw$ for all of these

$j(\phi_s(w)-1)$ labels. Counting the last label in the $w^{th}$ leaf node we get that the frequency for the value $jw$ is $\phi_{s*j}(jw) = j(\phi_s(w)-1)+1 = j\phi_s(w)-(j-1)$, as desired.

By the definition of the label counting function the only frequency counts that *can be* (but are not necessarily) greater than 1 are those associated with the last label on a leaf; for every other label the associated frequency count must be 1. For $a \neq 0$ the $m = (jw+a)^{th}$ label cannot be the last label on a leaf, from which we have $\phi_{s*j}(m) = \phi_{s*j}(jw+a) = 1$. □

We can use this relation between the frequency functions for the label counting sequences $C_s(n)$ and $C_{s*j}(n)$ to identify a formula linking the sequences themselves. To do so first we set $C_s(0) = 0$ and $C_{s*j}(0) = 0$.

**Proposition 2.2.** *Let $n = jz+b$, where $0 \leq b < j$. Then $C_{s*j}(n)$, the label counting sequence of the tree $T_{s*j}$ (with an initial 0 term appended to the sequence), can be expressed as the following linear combination of terms in $C_s(n)$, the label counting sequence (with an initial 0 term appended) for the tree $T_s$: $C_{s*j}(n) = C_{s*j}(jz+b) = jC_s(z) + b(C_s(z+1) - C_s(z))$.*

*Proof.* The skeleton of $T_{s*j}$ is the same as the skeleton of $T_s$. Every label in $T_s$ is replaced by $j$ labels in $T_{s*j}$. Hence we know that for any $z$, $C_{s*j}(jz) = jC_s(z)$.

Since $jz$ is a multiple of $j$, the label $jz$ in the tree $T_{s*j}$ must be either (i) the last label of some regular or leaf node, or (ii) the last label in a group of $j$ labels on a supernode. In case (i), as we move forward $1 \leq b < j$ labels in $T_{s*j}$, we move to and stay on the next node (since $b < j$); in case (ii) either we stay on the same supernode or move to the next regular node. It follows that these $b$ labels will add to the leaf label count if and only if this next node is a leaf node, so $C_s(z+1) - C_s(z) = 1$. Thus, $jC_s(z) + b(C_s(z+1) - C_s(z))$ is the label counting function $L_{s*j}(n)$ of $T_{s*j}$, which we know is $C_{s*j}(n)$. □

Observe that the formula in Proposition 2.2 that is derived from the tree-based interpretation of the solutions of the recursions $C_s(n)$ and $C_{s*j}(n)$ describes a purely parametric relationship between the slow solutions to two nested recursions in the same family with closely related parameters (each parameter of one recursion is the same multiple of the corresponding parameter of the other). It is therefore natural to ask whether the tree interpretation is necessary for the formula in Proposition 2.2 to hold, or whether some other proof can be found for comparably related pairs of recursions from the same family for which there is no known tree interpretation. In the next section we will show how to extend the formula in Proposition 2.2 to a more general class of recursions. In this way we provide a general method for identifying and solving new families of recursions with slow solutions.

## 3. Slow Solutions for New Recursion Families Via Virtual Trees

We begin by generalizing the purely parametric relation between the label counting sequences that we derived in Section 2 to a broad class of slow sequences.

**Definition 1.** Suppose $\{x(n)\}$ is any slow sequence with $x(0) = 0$ and $x(1) = 1$. For any fixed $j > 0$ define the functional $\Psi_j : \{x(n)\} \to \{y(n)\}$ as follows: for any $n = jz+b$ with $0 \leq b < j$, $y(n) = y(jz+b) = jx(z) + b(x(z+1) - x(z))$.

Note that as a result of the above definition $y(0) = 0$. If $\{x(n)\}$ is a finite sequence with $d+1$ terms, where $x(0) = 0, x(1) = 1$, then applying $\Psi_j$ to $\{x(n)\}$ results in a finite sequence $\{y(n)\}$ with $dj+1$ terms including $y(0)$. For example, for $j = 3$, applying $\Psi_3$ to

$\{x(n)\} = \{0, 1, 1, 2, 3\}$ with five terms yields $\{y(n)\} = \{0, 1, 2, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9\}$ with thirteen terms.

It is easy to see that the functional $\Psi_j$ maps slow sequences to slow sequences.

**Proposition 3.1.** *Suppose $\{x(n)\}$ is a slow sequence with $x(0) = 0$ and $x(1) = 1$. Then $\Psi_j(\{x(n)\}) = \{y(n)\}$ is a slow sequence with $y(0) = 0$ and $y(1) = 1$.*

*Proof.* By the definition of $\Psi_j$ and direct computation we get $y(0) = 0$ and $y(1) = 1$. Again by the definition of $\Psi_j$, for any slow sequence $\{x(n)\}$ and any $z$, $y(jz) = jx(z)$. If $x(z+1)-x(z) = 1$, then the functional $\Psi_j$ inserts precisely $j-1$ different values into $\{y(n)\}$ between $y(jz)$ and $y(j(z+1))$, where successive terms differ by 1. If $x(z+1) - x(z) = 0$, then the functional $\Psi_j$ inserts precisely $j-1$ copies of $y(jz)$ in $\{y(n)\}$ between $y(jz)$ and $y(j(z+1))$. Since $\{x(n)\}$ is slow it follows that $\{y(n)\}$ must also be a slow sequence. □

Evidently, $\Psi_j(\{C_s(n)\}) = \{C_{s*j}(n)\}$, that is, $\Psi_j$ is a natural generalization of the relationship proved in Proposition 2.2 between the label counting sequences for the trees $T_s$ and $T_{s*j}$. Further, Proposition 2.1 extends to the frequency functions for any slow $\{x(n)\}$ in place of $C_s(n)$ and $\{y(n)\} = \Psi_j(\{x(n)\})$ in place of $C_{s*j}(n)$. More precisely, we have:

**Proposition 3.2.** *Suppose $\{x(n)\}$ is a slow sequence with $x(0) = 0$ and $x(1) = 1$. Let $\{y(n)\} = \Psi_j(\{x(n)\})$. Suppose $m = jw + a$, where $0 \le a < j$. If $a = 0$ then the frequency of $m$ in $\{y(n)\}$ is $\phi_y(m) = \phi_y(jw) = j\phi_x(w) - (j-1)$; otherwise, $\phi_y(m) = 1$.*

*Proof.* This follows directly from the argument in the proof of Proposition 3.1.

□

For any slow sequence $\{x(n)\}$ with $x(1) = 1$ it is possible to construct a labeled tree $T_x$ for which $\{x(n)\}$ is the leaf label counting sequence. But in general we don't know how to describe the structure of this tree (namely, its skeleton and labeling) in any useful way that allows us to use this tree to relate $\{x(n)\}$ to the solution of some recursion as we did with labeled binary trees (where we showed that the label counting sequence for particular versions of the tree solved different generalizations of the Conolly recursion).

Still, in certain situations we can make use of the existence of this underlying "virtual" tree. Suppose that we can prove by some means (usually an induction argument) that a nested recursion $R$ with given initial conditions has a slow solution $\{x(n)\}$, with $x(1) = 1$. Where this is the case we can sometimes use this idea that there exists an underlying "virtual" tree $T_x$ for which $\{x(n)\}$ is the leaf label counting sequence of $T_x$. That's because we already know from the above arguments that $\Psi_j(\{x(n)\}) = \{y(n)\}$ is the slow sequence that is the leaf label counting sequence for the "virtual" tree $T_y$ obtained from $T_x$ by replacing all the labels in each node of $T_x$ by $j$ labels. And what we now show is that for certain types of recursions $R$ and any positive integer $j$ the sequence $\{y(n)\}$ solves the new recursion family $R_j$ obtained from $R$ by multiplying all its parameters by $j$.[14]

Thus, for every $j$ the solution sequence for $R_j$ is found by taking the solution sequence of $R$ and applying $\Psi_j$ to it, thereby yielding a new family of recursions with slow solutions. We make this precise in the following results.

**Theorem 3.3.** *Let $R$ be a two term generalized Conolly nested recursion of the form $R(n) = R(n - s_1 - R(n - a_1)) + R(n - s_2 - R(n - a_2))$, with $0 < a_1 \le a_2$. Suppose that for some set of $\alpha$ initial conditions (beginning at $n = 1$) $R$ has a slow solution with $R(1) = 1$,*

[14]See [23, 26] where a related and more general idea is explored in the case when the solution of the recursion $R$ is known to be related to a binary or $k$-ary tree.

*call it $R(n)$. Define $R(0) = 0$. For any fixed $j \geq 1$, and for all $n = jz + b$, with $z \geq 0$ and $0 \leq b < j$, define the sequence $\Psi_j(R(n)) = jR(z) + b(R(z+1) - R(z))$. Then $\Psi_j(R(n))$ is the slow solution of the recursion $R_j$ obtained by multiplying all the parameters of $R$ by $j$, that is, $R_j(n) = R_j(n - s_1 j - R_j(n - a_1 j)) + R_j(n - s_2 j - R(n - a_2 j))$, and where we take as the initial conditions of $R_j$ the first $(\alpha+1)j-1$ terms (beginning at $n = 1$) of $\Psi_j(R(n))$.*

*Proof.* From Proposition 3.1 we know that $\Psi_j(R(n))$ is slow. By the choice of the initial conditions it is immediate that our statement holds for the first $(\alpha + 1)j - 1$ values for $R_j(n)$.

We proceed by strong induction on $n$. Suppose the result is true up to $n-1 \geq (\alpha+1)j-1$. We show that it is true for $n$. Write $n = jz + b$. Then from the definition of $R_j$ we have:

$$\begin{aligned} R_j(jz+b) &= R_j(jz + b - s_1 j - R_j(jz + b - a_1 j)) + R_j(jz + b - s_2 j - R_j(jz + b - a_2 j)) \\ &= R_j(jz + b - s_1 j - R_j(j(z - a_1) + b)) + R_j(jz + b - s_2 j - R_j(j(z - a_2) + b)) \end{aligned}$$

Note that $j(z - a_1) + b < jz + b$ and $j(z - a_2) + b < jz + b$. We can therefore rewrite our statement so that we can use the induction hypothesis.

$$\begin{aligned} R_j(jz+b) = R_j(jz + b - s_1 j - [jR(z - a_1) + b(R(z - a_1 + 1) - R(z - a_1))]) \\ + R_j(jz + b - s_2 j - [jR(z - a_2) + b(R(z - a_2 + 1) - R(z - a_2))]) \end{aligned} \tag{3.1}$$

Since $R(n)$ is slow it is sufficient to examine four cases:

(1) $R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$
(2) $R(z - a_1) + 1 = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$
(3) $R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) + 1 = R(z - a_2 + 1)$
(4) $R(z - a_1) + 1 = R(z - a_1 + 1)$ and $R(z - a_2) + 1 = R(z - a_2 + 1)$

The proof of each case is similar.

**Case 1.** *$R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$*

With this assumption, (3.1) simplifies to:

$$\begin{aligned} R_j(jz+b) &= R_j(jz + b - s_1 j - jR(z - a_1)) + R_j(jz + b - s_2 j - jR(z - a_2)) \\ &= R_j(j(z - s_1 - R(z - a_1)) + b) + R_j(j(z - s_2 - R(z - a_2)) + b) \\ &= jR(z - s_1 - R(z - a_1)) + b(R(z - s_1 - R(z - a_1) + 1) - R(z - s_1 - R(z - a_1))) \\ &\quad + jR(z - s_2 - R(z - a_2)) + b(R(z - s_2 - R(z - a_2) + 1) - R(z - s_2 - R(z - a_2))) \\ &= j(R(z - s_1 - R(z - a_1)) + R(z - s_2 - R(z - a_2)) \\ &\quad + b((R(z - s_1 - R(z - a_1) + 1) + R(z - s_2 - R(z - a_2) + 1) \\ &\quad - R(z - s_1 - R(z - a_1)) - R(z - s_2 - R(z - a_2)))) \end{aligned}$$

By the assumptions in Case 1 this simplifies further to the desired result:

$$\begin{aligned} R_j(jz+b) &= j(R(z - s_1 - R(z - a_1)) + R(z - s_2 - R(z - a_2)) \\ &\quad + b((R(z - s_1 - R(z + 1 - a_1) + 1) + R(z - s_2 - R(z + 1 - a_2) + 1) \\ &\quad - R(z - s_1 - R(z - a_1)) - R(z - s_2 - R(z - a_2)))) \\ &= jR(z) + b(R(z+1) - R(z)). \end{aligned}$$

**Case 2.** $R(z-a_1)+1=R(z-a_1+1)$ *and* $R(z-a_2)=R(z-a_2+1)$

The argument is similar to that in Case 1 above. We use these assumptions in the initial and penultimate steps to simplify (3.1) to the desired result.

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-jR(z-a_1)-b)+R_j(jz+b-s_2j-jR(z-a_2))\\
&= R_j(jz-s_1j-jR(z-a_1))+R_j(j(z-s_2-R(z-a_2))+b)\\
&= R_j(j(z-s_1-R(z-a_1)))+R_j(j(z-s_2-R(z-a_2))+b)\\
&= jR(z-s_1-R(z-a_1))\\
&\quad +jR(z-s_2-R(z-a_2))+b(R(z-s_2-R(z-a_2)+1)-R(z-s_2-R(z-a_2)))\\
&= jR(z-s_1-R(z-a_1))+(b-b)R(z-s_1-R(z-a_1))\\
&\quad +jR(z-s_2-R(z-a_2))+b(R(z-s_2-R(z-a_2)+1)-R(z-s_2-R(z-a_2)))\\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1+1)+1)-R(z-s_1-R(z-a_1)))\\
&\quad +jR(z-s_2-R(z-a_2))+b(R(z-s_2-R(z-a_2)+1)-R(z-s_2-R(z-a_2)))\\
&= j(R(z-s_1-R(z-a_1))+R(z-s_2-R(z-a_2))\\
&\quad +b((R(z-s_1-R(z+1-a_1)+1)+R(z-s_2-R(z+1-a_2)+1)\\
&\quad -R(z-s_1-R(z-a_1)))-R(z-s_2-R(z-a_2))))\\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

**Case 3.** $R(z-a_1)=R(z-a_1+1)$ *and* $R(z-a_2)+1=R(z-a_2+1)$

The argument for Case 3 is the same as for Case 2 with the roles of $a_1$ and $a_2$ interchanged. We omit the details.

**Case 4.** $R(z-a_1)+1=R(z-a_1+1)$ *and* $R(z-a_2)+1=R(z-a_2+1)$

As above we use these assumptions to simplify (3.1) in the initial and penultimate steps to derive the desired result.

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-jR(z-a_1)-b)+R_j(jz+b-s_2j-jR(z-a_2)-b)\\
&= R_j(jz-s_1j-jR(z-a_1))+R_j(jz-s_2j-jR(z-a_2))\\
&= R_j(j(z-s_1-R(z-a_1)))+R_j(j(z-s_2-R(z-a_2)))\\
&= jR(z-s_1-R(z-a_1))+jR(z-s_2-R(z-a_2))\\
&= jR(z-s_1-R(z-a_1))+(b-b)R(z-s_1-R(z-a_1))\\
&\quad +jR(z-s_2-R(z-a_2))+(b-b)R(z-s_2-R(z-a_2))\\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1+1)+1)-R(z-s_1-R(z-a_1)))\\
&\quad +jR(z-s_2-R(z-a_2))+b(R(z-s_2-R(z-a_2+1)+1)-bR(z-s_2-R(z-a_2)))\\
&= j(R(z-s_1-R(z-a_1))+R(z-s_2-R(z-a_2))\\
&\quad +b((R(z-s_1-R(z+1-a_1)+1)+R(z-s_2-R(z+1-a_2)+1)\\
&\quad -R(z-s_1-R(z-a_1)))-R(z-s_2-R(z-a_2))))\\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

This completes the proof. □

Recall from the Introduction the generalized Conway nested recursion $R(n) = R(n - s_1 - R(n - a_1)) + R(-s_2 + R(n - a_2))$, with $s_1$ and $s_2$ nonnegative and $a_1$ and $a_2$ positive. For $s_1 = s_2 = 0$ and $a_1 = a_2 = 1$ this is the ordinary Conway recursion $A(n)$ that is known to have a slow solution [11, 28, 29].

The following result is analogous to that for Theorem 3.3.

**Theorem 3.4.** *Let $R$ be a two term generalized Conway nested recursion of the form $R(n) = R(n - s_1 - R(n - a_1)) + R(-s_2 + R(n - a_2))$, with $s_1$ and $s_2$ nonnegative and $a_1 > 0$ and $a_2 > 0$. Suppose that for some set of $\alpha$ initial conditions (beginning at $n = 1$) $R$ has a slow solution, call it $R(n)$, with $R(1) = 1$. Define $R(0) = 0$. For any fixed $j \geq 1$, and for all $n = jz + b$, with $z \geq 0$ and $0 \leq b < j$, define the sequence $\Psi_j(R(n)) = jR(z) + b(R(z + 1) - R(z))$. Then $\Psi_j(R(n))$ is the slow solution of the recursion $R_j(n) = R_j(n-s_1j-R_j(n-a_1j))+R_j(-s_2j+R_j(n-a_2j))$ obtained by multiplying all the parameters of $R$ by $j$, and where the initial conditions of $R_j$ are the first $(\alpha+1)j-1$ terms (beginning at $n = 1$) of $\Psi_j(R(n))$.*

*Proof.* From Proposition 3.1 we know that $\Psi_j(R(n))$ is slow. By the choice of the initial conditions it is immediate that our statement holds for the first $(\alpha + 1)j - 1$ values for $R_j(n)$.

We proceed by strong induction on $n$. Suppose the result is true up to $n-1 \geq (\alpha+1)j-1$. We show that it is true for $n$. Write $n = jz + b$. Then from the definition of $R_j$ we have:

$$\begin{aligned} R_j(jz + b) &= R_j(jz + b - s_1j - R_j(jz + b - a_1j)) + R_j(-s_2j + R_j(jz + b - a_2j)) \\ &= R_j(jz + b - s_1j - R_j(j(z - a_1) + b)) + R_j(-s_2j + R_j(j(z - a_2) + b)) \end{aligned}$$

Note that $j(z - a_1) + b < jz + b$ and $j(z - a_2) + b < jz + b$. We can therefore rewrite our statement so that we can use the induction hypothesis.

$$\begin{aligned} R_j(jz + b) = R_j(jz + b - s_1j - [jR(z - a_1) + b(R(z - a_1 + 1) - R(z - a_1))]) \\ + R_j(-s_2j + [jR(z - a_2) + b(R(z - a_2 + 1) - R(z - a_2))]) \end{aligned} \tag{3.2}$$

Since $R(n)$ is slow it is again sufficient to examine four cases:

(1) $R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$
(2) $R(z - a_1) + 1 = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$
(3) $R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) + 1 = R(z - a_2 + 1)$
(4) $R(z - a_1) + 1 = R(z - a_1 + 1)$ and $R(z - a_2) + 1 = R(z - a_2 + 1)$

**Case 1.** *$R(z - a_1) = R(z - a_1 + 1)$ and $R(z - a_2) = R(z - a_2 + 1)$*

With this assumption, (3.2) simplifies to:

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-[jR(z-a_1)+b(R(z-a_1+1)-R(z-a_1))]) \\
&\quad + R_j(-s_2j+[jR(z-a_2)+b(R(z-a_2+1)-R(z-a_2))]) \\
&= R_j(jz+b-s_1j-jR(z-a_1))+R_j(-s_2j+jR(z-a_2)) \\
&= R_j(j(z-s_1-R(z-a_1))+b)+R_j(j(-s_2+R(z-a_2)) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2)) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2)))+(b-b)R(-s_2+R(z-a_2))) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1+1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2)))+b(R(-s_2+R(z-a_2+1))-R(-s_2+R(z-a_2))) \\
&= j(R(z-s_1-R(z-a_1)+R(-s_2+R(z-a_2))) \\
&\quad + b(R(z-s_1-R(z-a_1+1)+1)+R(-s_2+R(z-a_2+1)) \\
&\quad - R(z-s_1-R(z-a_1))-R(-s_2+R(z-a_2))) \\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

**Case 2.** $R(z-a_1)+1=R(z-a_1+1)$ *and* $R(z-a_2)=R(z-a_2+1)$

The argument is similar to that in Case 1 above. We use these assumptions in the initial and penultimate steps to simplify (3.2) to the desired result.

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-[jR(z-a_1)+b(R(z-a_1+1)-R(z-a_1))]) \\
&\quad + R_j(-s_2j+[jR(z-a_2)+b(R(z-a_2+1)-R(z-a_2))]) \\
&= R_j(jz+b-s_1j-jR(z-a_1)-b)+R_j(-s_2j+jR(z-a_2)) \\
&= R_j(jz-s_1j-jR(z-a_1))+R_j(-s_2j+jR(z-a_2)) \\
&= R_j(j(z-s_1-R(z-a_1)))+R_j(j(-s_2+R(z-a_2))) \\
&= jR(z-s_1-R(z-a_1))+jR(-s_2+R(z-a_2)) \\
&= jR(z-s_1-R(z-a_1))+(b-b)R(z-s_1-R(z-a_1)) \\
&\quad + jR(-s_2+R(z-a_2))+(b-b)R(-s_2+R(z-a_2)) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1+1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2)))+b(R(-s_2+R(z-a_2+1))-R(-s_2+R(z-a_2))) \\
&= j(R(z-s_1-R(z-a_1)+R(-s_2+R(z-a_2))) \\
&\quad + b(R(z-s_1-R(z-a_1+1)+1)+R(-s_2+R(z-a_2+1)) \\
&\quad - R(z-s_1-R(z-a_1))-R(-s_2+R(z-a_2))) \\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

**Case 3.** $R(z-a_1)=R(z-a_1+1)$ *and* $R(z-a_2)+1=R(z-a_2+1)$

The argument is similar to that in Cases 1 an 2 above. We once again use these assumptions in the initial and penultimate steps to simplify (3.2) to the desired result.

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-[jR(z-a_1)+b(R(z-a_1+1)-R(z-a_1))]) \\
&\quad + R_j(-s_2j+[jR(z-a_2)+b(R(z-a_2+1)-R(z-a_2))]) \\
&= R_j(jz+b-s_1j-jR(z-a_1))+R_j(-s_2j+jR(z-a_2)+b) \\
&= R_j(j(z-s_1-R(z-a_1))+b)+R_j(j(-s_2+R(z-a_2))+b) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2))+b(R(-s_2+R(z-a_2)+1)-R(-s_2+R(z-a_2))) \\
&= j(R(z-s_1-R(z-a_1)+R(-s_2+R(z-a_2))) \\
&\quad + b(R(z-s_1-R(z-a_1+1)+1)+R(-s_2+R(z-a_2+1)) \\
&\quad - R(z-s_1-R(z-a_1))-R(-s_2+R(z-a_2))) \\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

**Case 4.** $R(z-a_1)+1=R(z-a_1+1)$ *and* $R(z-a_2)+1=R(z-a_2+1)$

As above we use these assumptions to simplify (3.2) in the initial and penultimate steps to derive the desired result.

$$\begin{aligned}
R_j(jz+b) &= R_j(jz+b-s_1j-[jR(z-a_1)+b(R(z-a_1+1)-R(z-a_1))]) \\
&\quad + R_j(-s_2j+[jR(z-a_2)+b(R(z-a_2+1)-R(z-a_2))]) \\
&= R_j(jz+b-s_1j-jR(z-a_1)-b)+R_j(-s_2j+jR(z-a_2)+b) \\
&= R_j(j(z-s_1-R(z-a_1)))+R_j(j(-s_2+R(z-a_2))+b) \\
&= jR(z-s_1-R(z-a_1))+jR(-s_2+R(z-a_2)) \\
&\quad + b(R(-s_2+R(z-a_2)+1)-R(-s_2+R(z-a_2))) \\
&= jR(z-s_1-R(z-a_1))+(b-b)R(z-s_1-R(z-a_1)) \\
&\quad + jR(-s_2+R(z-a_2))+b(R(-s_2+R(z-a_2)+1)-R(-s_2+R(z-a_2))) \\
&= jR(z-s_1-R(z-a_1))+b(R(z-s_1-R(z-a_1+1)+1)-R(z-s_1-R(z-a_1))) \\
&\quad + jR(-s_2+R(z-a_2))+b(R(-s_2+R(z-a_2)+1)-R(-s_2+R(z-a_2))) \\
&= j(R(z-s_1-R(z-a_1)+R(-s_2+R(z-a_2))) \\
&\quad + b(R(z-s_1-R(z-a_1+1)+1)+R(-s_2+R(z-a_2+1)) \\
&\quad - R(z-s_1-R(z-a_1))-R(-s_2+R(z-a_2))) \\
&= jR(z)+b(R(z+1)-R(z))
\end{aligned}$$

This completes the proof. □

Theorems 3.3 and 3.4 enable us to construct new families of nested recursions with slow solutions from a given nested recursion with a slow solution for which no tree structure is known. In the following section we do so for two well-known recursions, Hofstadter's $V$ and Conway's $A$.

## 4. Applications

We now apply the two preceding general results to identify a new family of recursions with slow solutions based on each of the well-known Hofstadter $V$ and Conway $A$ recursions, respectively. In each case we demonstrate how the properties of the original slow solution extend naturally to the solutions to the recursions in the new family that we identify.

The recursion $V(n) = V(n - V(n-1)) + V(n - V(n-4))$ with initial conditions $V(1) = V(2) = V(3) = V(4) = 1$, first discussed by Hofstadter and Huber [19], has a slow solution V with a complex structure (see [4] for details). It is readily seen that the same recursion with initial conditions $V(1) = 1, V(2) = 2, V(3) = 3, V(4) = 4$ yields essentially the same solution, excluding the initial four 1s. This latter version of the $V$ sequence is more convenient for our present purposes since by omitting the initial four 1s its frequency sequence consists only of 1s, 2s, and 3s (see [4]), so we use it in what follows[15]. Applying Theorem 3.3 to this $V$ sequence (again with the additional term $V(0) = 0$) we derive a new family of recursions $V_j$ with slow solutions.

**Corollary 4.1.** *For any fixed $j > 1$ let $V_j(n) = V_j(n - V_j(n-j)) + V_j(n - V_j(n-4j))$. Then for all $n = jz + b$, with $z \geq 0$ and $0 \leq b < j$, the sequence $\Psi_j(V(n)) = jV(z) + b(V(z+1) - V(z))$ is the slow solution of the recursion $V_j$, where the initial conditions of $V_j$ are the first $5j - 1$ terms (beginning at $n = 1$) of $\Psi_j(V(n))$.*

Each of the nested recursions in this new family has a solution with properties analogous to those for $V$. In particular, for every $j$ the frequency sequence of the solution sequence $V_j$ also consists only of three values, these being $\{1, j+1, 2j+1\}$. This follows immediately from Proposition 3.2 and the fact that the frequency sequence for the $V$ sequence consists only of 1s, 2s and 3s.

The following result summarizes this discussion and describes precisely how to construct the frequency sequence of $V_j$ from that of $V$.

**Corollary 4.2.** *The frequency sequence of $V_j$ consists of 3 elements $\{1, j+1, 2j+1\}$. To construct the frequency sequence of $V_j$ from that of $V$ first map each entry $x$ in the frequency sequence of $V$ to the value $jx - (j-1)$. Next, insert $j - 1$ 1s before the initial entry 1 in this sequence and between every pair of entries.*

Observe that the frequency sequence of $V$ begins 1,1,1,1,2,2,1,.... It follows from the above result that the frequency sequence of $V_j$ begins with $5j - 1$ 1s, corresponding to the $5j-1$ distinct values $1, 2, 3, ..., 5j-1$ which are the initial conditions, followed by a 4, which is the image of the first 2 in the frequency sequence of $V$.

In [4] it is shown that there are definite, highly complex rules that determine the occurrences of the 1s, 2s, and 3s in the frequency sequence of $V$. Using Corollary 4.2 we can specify analogous (but somewhat more complicated) rules for the occurrences of the values $\{1, j+1, 2j+1\}$ in the frequency sequence of the solution sequence $V_j$. These rules apply to the images $jx-(j-1)$ of the terms $x$ of the frequency sequence of $V$; all the other terms of the frequency sequence of $V_j$ are 1.

We illustrate what we mean by this with an example of such a rule. It is shown in [4] that if $\phi_V(a) = 1$ then $\phi_V(2a) = 2$ and $\phi_V(2a+1) = 2$; for example, the value 13 occurs once in $V$, so both 26 and 27 occur twice.

It turns out that the analogous rule for an arbitrary $j$ is: if $a \geq 4j$ and $a$ is a multiple of $j$, then if $\phi_{V_j}(a) = 1$ then $\phi_{V_j}(2a) = j+1$ and $\phi_V(2a+j) = j+1$.

[15]Either choice for the initial conditions leads to essentially the same solution other than some of the initial terms.

In a similar way we can enunciate analogues for each of the rules governing the frequency sequence of $V$ that are described in [4]. The proofs for these rules apply Corollary 4.2 and follow closely the arguments described in [4].[16,17]

We can generalize the above results further to identify new families of "$V$-like" sequences. First we introduce an $s$-parameter into the $V$ recursion:

$$\overline{V}_s(n) = \overline{V}_s(n - s - \overline{V}_s(n-1)) + \overline{V}_s(n - s - \overline{V}_s(n-4)),\ n > 4. \tag{4.1}$$

Note that the original $V$ sequence corresponds to $s = 0$.

With appropriate initial conditions we can prove that $\overline{V}_s$ has a slow solution sequence with highly analogous properties to those for the original $V$ sequence.[18] For $s = 1$ we use the initial conditions $(1, 1, 1, 2)$, while for $s = 2$ the initial conditions $(1, 1, 2, 2)$ are suitable.[19]

For any $s > 2$ we use the $4s - 6$ initial conditions $(1, 2, \ldots, s-3, s-2, s-1, s-1, s-1, s, s, s, s+1, s+1, s+1, \ldots, 2s-4, 2s-4, 2s-4, 2s-3, 2s-3)$; that is, each of the values from 1 to $s-2$ occurs once, then each of the $s-2$ values from $s-1$ to $2s-4$ inclusive occurs three times, and the final value $2s-3$ occurs twice. We write this more compactly using frequency sequence notation as $(1)^{s-2}(3)^{s-2}(2)$. For $s > 2$, $\overline{V}_s$ with these initial conditions always has a slow solution with "$V$-like" properties. For example, for $s = 4$ the ten initial conditions are the first ten terms in Table 4.1, which illustrates the behaviour of the first fifty terms of the sequence $\overline{V}_4$.

| | $n$ | | | | | | $n$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 |
| $\overline{V}_4(n+0)$ | 1 | 2 | 3 | 3 | 3 | $\overline{V}_4(n+25)$ | 11 | 12 | 12 | 12 | 13 |
| $\overline{V}_4(n+5)$ | 4 | 4 | 4 | 5 | 5 | $\overline{V}_4(n+30)$ | 13 | 14 | 15 | 15 | 15 |
| $\overline{V}_4(n+10)$ | 5 | 6 | 6 | 6 | 7 | $\overline{V}_4(n+35)$ | 16 | 16 | 16 | 17 | 17 |
| $\overline{V}_4(n+15)$ | 7 | 8 | 8 | 8 | 9 | $\overline{V}_4(n+40)$ | 18 | 19 | 19 | 19 | 20 |
| $\overline{V}_4(n+20)$ | 9 | 10 | 10 | 10 | 11 | $\overline{V}_4(n+45)$ | 20 | 20 | 21 | 21 | 22 |

TABLE 4.1.
First 50 terms of $\overline{V}_4(n)$ with initial conditions 1, 2, 3, 3, 3, 4, 4, 4, 5, 5

For any fixed $j > 1$ we apply Theorem 3.3 to $\overline{V}_s$ to obtain a new family of nested recursions with slow solutions which we can show are "$V$-like" in the same sense as the family $V_j$ discussed above:

**Corollary 4.3.** *For any fixed $j > 1$ let $\overline{V}_{s*j}(n) = \overline{V}_{s*j}(n - js - \overline{V}_{s*j}(n-j)) + \overline{V}_{s*j}(n - js - \overline{V}_{s*j}(n-4j))$. Then for all $n = jz + b$, with $z \geq 0$ and $0 \leq b < j$, the sequence $\Psi_j(\overline{V}_s(n)) = j\overline{V}_s(z) + b(\overline{V}_s(z+1) - \overline{V}_s(z))$ is the slow solution of the recursion $\overline{V}_{s*j}$ with the appropriate number of initial conditions depending on the values of $s$ and $j$. These initial conditions are given by the initial terms of the sequence $\Psi_j(\overline{V}_s(n))$.*

[16]For the sake of conciseness we omit the details. The interested reader may contact us for additional information.

[17]In [3] Allouche and Shallit use the existence of the rules for the frequency sequence of $V$ in [4] to prove that the $V$ sequence is 2-automatic. In a private communication Professor Shallit confirmed that a similar argument using the analogous rules for the frequency sequence of $V_j$ would show that the sequence $V_j$ is also 2-automatic for any $j$.

[18]The required induction arguments are fairly complicated, but mirror closely those that appear in [4]. The interested reader can contact us for the details.

[19]Other sets of initial conditions may also yield essentially the same sequence, perhaps with some of the initial terms omitted.

For $s = 1$ and $s = 2$ we noted above that we use four initial conditions to generate the slow solution $\overline{V}_s$, so $\alpha = 4$ and by Theorem 3.3 we use the initial $5j - 1$ terms of $\Psi_j(\overline{V}_s(n))$ as the initial conditions. For $s > 2$ we generate the slow solution of $\overline{V}_s$ using $\alpha = 4s - 6$ initial conditions. By Theorem 3.3 we use the initial $4js - 5j - 1$ initial conditions of $\Psi_j(\overline{V}_s(n))$ to generate the solution $\overline{V}_{s*j}$. We can show that these initial conditions begin with 1, and have frequency sequence that factors as $(1)^{sj-j-1}((1+2j)(1^{j-1}))^{s-2}(2j)$.

For the sake of greater clarity here is an example: for $s = 4$ and $j = 3$ there are $4js-5j-1 = 32$ initial conditions. These begin with 1 and have frequency sequence $(1)^8((7)(1^2))^2(6)$, so the initial conditions are: $1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9, 9, 9, 9, 9, 10, 11, 12, 12, 12, 12, 12, 12, 12, 13, 14, 15, 15, 15, 15, 15, 15, 15$.

We can apply Theorem 3.4 to derive a new family of nested recursions related to the Conway recursion. Recall from the Introduction that the Conway recursion [11, 28, 29] $A(n) = A(n - A(n-1)) + A(A(n-1)), A(1) = A(2) = 1$ is known to have a slow solution. Applying Theorem 3.4 we derive a new "Conway-type" family of nested recursions.

**Corollary 4.4.** *Let $A(n)$ be the Conway recursion $A(n) = A(n - A(n-1)) + A(A(n-1))$, $A(1) = A(2) = 1$. Fix $j > 0$ and let $n = jz + b$, $z \geq 0$ and $0 \leq b < j$. Then the sequence $\Psi_j(A(n)) = \Psi_j(A(jz + b)) = jA(z) + b(A(z + 1) - A(z))$ satisfies the recursion $A_j(n) = A_j(n - A_j(n - j)) + A_j(A_j(n - j))$ with initial conditions the first $3j - 1$ terms of the sequence $\Psi_j(A(n))$.*

For $j = 2$ and $j = 3$ selected data for $A_j(n)$ and its associated frequency sequence appears in the following tables.

| | $n$ | | | | | | $n$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 |
| $A_2(n+0)$ | 1 | 2 | 2 | 2 | 3 | $A_2(n+25)$ | 16 | 16 | 16 | 16 | 16 |
| $A_2(n+5)$ | 4 | 4 | 4 | 5 | 6 | $A_2(n+30)$ | 16 | 16 | 17 | 18 | 19 |
| $A_2(n+10)$ | 7 | 8 | 8 | 8 | 8 | $A_2(n+35)$ | 20 | 21 | 22 | 23 | 24 |
| $A_2(n+15)$ | 8 | 9 | 10 | 11 | 12 | $A_2(n+40)$ | 24 | 24 | 25 | 26 | 27 |
| $A_2(n+20)$ | 13 | 14 | 14 | 14 | 15 | $A_2(n+45)$ | 28 | 28 | 28 | 29 | 30 |

TABLE 4.2. First 50 terms of $A_2(n)$ with initial conditions $1, 2, 2, 2, 3$

| | $n$ | | | | | | $n$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 |
| $\phi_{A_2}(n+0)$ | 1 | 3 | 1 | 3 | 1 | $\phi_{A_2}(n+25)$ | 1 | 1 | 3 | 1 | 5 |
| $\phi_{A_2}(n+5)$ | 1 | 1 | 5 | 1 | 1 | $\phi_{A_2}(n+30)$ | 1 | 9 | 1 | 1 | 1 |
| $\phi_{A_2}(n+10)$ | 1 | 1 | 1 | 3 | 1 | $\phi_{A_2}(n+35)$ | 1 | 1 | 1 | 1 | 1 |
| $\phi_{A_2}(n+15)$ | 7 | 1 | 1 | 1 | 1 | $\phi_{A_2}(n+40)$ | 1 | 3 | 1 | 1 | 1 |
| $\phi_{A_2}(n+20)$ | 1 | 1 | 1 | 3 | 1 | $\phi_{A_2}(n+45)$ | 1 | 1 | 3 | 1 | 1 |

TABLE 4.3. First 50 terms of $\phi_{A_2}(n)$ with initial conditions $1, 2, 2, 2, 3$

| | $n$ | | | | | | $n$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 |
| $A_3(n+0)$ | 1 | 2 | 3 | 3 | 3 | $A_3(n+25)$ | 14 | 15 | 16 | 17 | 18 |
| $A_3(n+5)$ | 3 | 4 | 5 | 6 | 6 | $A_3(n+30)$ | 19 | 20 | 21 | 21 | 21 |
| $A_3(n+10)$ | 6 | 6 | 7 | 8 | 9 | $A_3(n+35)$ | 21 | 22 | 23 | 24 | 24 |
| $A_3(n+15)$ | 10 | 11 | 12 | 12 | 12 | $A_3(n+40)$ | 24 | 24 | 24 | 24 | 24 |
| $A_3(n+20)$ | 12 | 12 | 12 | 12 | 13 | $A_3(n+45)$ | 24 | 24 | 24 | 25 | 26 |

TABLE 4.4. First 50 terms of $A_3(n)$ with initial conditions $1, 2, 3, 3, 3, 3, 4, 5, 6$

| | $n$ | | | | | | $n$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 |
| $\phi_{A_3}(n+0)$ | 1 | 1 | 4 | 1 | 1 | $\phi_{A_3}(n+25)$ | 1 | 1 | 1 | 1 | 1 |
| $\phi_{A_3}(n+5)$ | 4 | 1 | 1 | 1 | 1 | $\phi_{A_3}(n+30)$ | 1 | 1 | 1 | 1 | 1 |
| $\phi_{A_3}(n+10)$ | 1 | 7 | 1 | 1 | 1 | $\phi_{A_3}(n+35)$ | 4 | 1 | 1 | 1 | 1 |
| $\phi_{A_3}(n+15)$ | 1 | 1 | 1 | 1 | 1 | $\phi_{A_3}(n+40)$ | 1 | 4 | 1 | 1 | 7 |
| $\phi_{A_3}(n+20)$ | 4 | 1 | 1 | 10 | 1 | $\phi_{A_3}(n+45)$ | 1 | 1 | 13 | 1 | 1 |

TABLE 4.5. First 50 terms of $\phi_{A_3}(n)$ with initial conditions $1, 2, 3, 3, 3, 3, 4, 5, 6$

The data in the tables suggests that the recursions $A_j(n)$ all behave in much the same way, with frequency functions that have strong analogues to the frequency function for the Conway recursion $A(n) = A_1(n)$; that is, that these recursions form a family in our sense and therefore share certain properties. To illustrate the similarities between recursions in this family two such generalized properties will be proved below using Corollary 4.4.

For the Conway sequence $A(n)$, it is well known (see, for example, [28]) that for any $k > 0$, $A(n) = 2^k$ for exactly $k + 1$ consecutive values of $n$ ending with $n = 2^{k+1}$. The following lemma together with its corollary provide a natural extension of this result, for any $j > 1, k > 0$, for $A_j(n)$.

**Lemma 4.5.** *Let $A(n)$ be the Conway recursion with $A(1) = A(2) = 1$. Fix $j, k > 0$ and let $n = j2^{k+1} + b$, $0 \le b < j$. Then $A_j(j2^{k+1} + b) = j2^k + b$.*

*Proof.* We apply Corollary 4.4 together with the known properties of $A(n)$: $A_j(j2^{k+1}+b) = jA(2^{k+1}) + b(A(2^{k+1}+1) - A(2^{k+1})) = j2^k + b(2^k + 1 - 2^k) = j2^k + b$.

□

**Corollary 4.6.** *Let $A(n)$ be the Conway recursion with $A(1) = A(2) = 1$. For any $j, k > 0$, $A_j(n) = j2^k$ for exactly $jk + 1$ consecutive values of $n$ ending with $n = j2^{k+1}$.*

*Proof.* Since $A_j(n)$ is slow it is sufficient to show that $n = j2^{k+1}$ is the largest index for which $A_j(n) = j2^k$ while $n = j2^{k+1} - jk$ is the smallest such index.

By Lemma 4.5, $A_j(j2^{k+1}) = j2^k$ and $A_j(j2^{k+1} + 1) = j2^k + 1$. Therefore, $n = j2^{k+1}$ is the largest index $n$ such that $A_j(n) = j2^k$.

Applying Corollary 4.4 to $j2^{k+1}-jk = j(2^{k+1}-k)$, we obtain $A_j(j(2^{k+1}-k)) = jA(2^{k+1}-k)$. From our observation above about $A(n)$ we know that the first time that $A(n) = 2^k$ occurs at $n = 2^{k+1} - k$, and since $A(n)$ is slow we know that $A(2^{k+1} - k - 1) = 2^k - 1$. Thus $A_j(j(2^{k+1} - k)) = jA(2^{k+1} - k) = j2^k$. Further,

$$\begin{aligned}
A_j(j2^{k+1} - jk - 1) &= A_j(j2^{k+1} - jk - j + j - 1) \\
&= A_j(j(2^{k+1} - k - 1) + (j - 1)) \\
&= jA(2^{k+1} - k - 1) + (j - 1)(A(2^{k+1} - k) - A(2^{k+1} - k - 1)) \\
&= j(2^k - 1) + (j - 1)(2^k - (2^k - 1)) \\
&= j(2^k - 1) + (j - 1) \\
&= j2^k - j + j - 1 \\
&= j2^k - 1
\end{aligned}$$

Therefore, $n = j2^{k+1} - jk$ is the smallest term that equals to $j2^k$. □

We illustrate Lemma 4.5 and Corollary 4.6 for $j = 3$ in Table 4.4 and Table 4.5. For $k = 2$, $jk = 3 \cdot 2^2 = 12$ and Table 4.4 contains a string of seven 12s ending at position $j2^{k+1} = 3 \cdot 2^{2+1} = 24$; this corresponds to the frequency $\phi_{A_3}(12) = 7$ in Table 4.5 which is equal to $3 \cdot 2 + 1 = jk + 1$. Similarly, for $k = 3$, $jk = 3 \cdot 2^3 = 24$ and the table contains a string of ten 24s ending at position $j2^{k+1} = 3 \cdot 2^{3+1} = 48$, corresponding to the frequency $\phi_{A_3}(24) = 10$ (equals $3 \cdot 3 + 1 = jk + 1$).

From Corollary 4.4 it follows readily that for fixed $j$ the asymptotic behaviour of $A_j(n)$ is identical to that for $A(n)$, which is well known. More precisely, we have:

**Corollary 4.7.** *Let $A(n)$ be the Conway recursion with $A(1) = A(2) = 1$. Then for any $j > 0$, $\lim_{n\to\infty} \frac{A_j(n)}{n} = \lim_{n\to\infty} \frac{A(n)}{n} = \frac{1}{2}$.*

*Proof.* Fix $j > 0$ and let $n = jz + b$, $z \geq 0$ and $0 \leq b < j$. Then:

$$\begin{aligned}
\lim_{n\to\infty} \frac{A_j(n)}{n} &= \lim_{z\to\infty} \frac{A_j(jz+b)}{jz+b} \\
&= \lim_{z\to\infty} \frac{(jA(z) + b(A(z+1) - A(z)))}{jz+b} \\
&= \lim_{z\to\infty} \frac{jA(z)}{jz+b} + b\left(\lim_{z\to\infty} \frac{A(z+1)}{jz+b} - \lim_{z\to\infty} \frac{A(z)}{jz+b}\right) \\
&= \frac{1}{2}
\end{aligned}$$

□

Although there is no tree interpretation of $A(n)$ comparable to that for $C(n)$ and its generalizations, Kubo and Vakil [28] demonstrate a very useful graphical interpretation of $A(n)$. It would be interesting to investigate if this approach can be extended to derive analogous results for $A_j(n)$.

We believe that analogous results to those in Section 3 can be developed for other nested recursions with slow solutions with somewhat different structures from those that we have examined above. In particular, a variation of the Conway recursion defined by Grytczuk [16] that contains additional nestings shows particular promise. We intend to report on this and related matters in a future communication.

## 5. Acknowledgement

The authors wish to thank Matthew Sunohara and Adam Wisniewski for helpful assistance in reviewing and clarifying material introduced in the final stages of the preparation of this paper.

## References

[1] R. Allenby and R. Smith, Some sequences resembling Hofstadter's, *J. Korean Math Soc.* 40 (2003), 921-932.

[2] Jean-Paul Allouche and Jeffrey Shallit, *Automatic Sequences: Theory, Applications, Generalizations*, Cambridge University Press, 2003.

[3] J.-P. Allouche and J. Shallit, A variant of Hofstadter's sequence and finite automata, *J. Aust. Math. Soc.* 93 (2012), 1-8.

[4] B. Balamohan, A. Kuznetsov, and S. Tanny, *On the Behaviour of a Variant of Hofstadter's Q-Sequence*, *Journal of Integer Sequences* 10 (2007), Article 07.7.1.

[5] B. Balamohan, Z. Li, and S. Tanny, A combinatorial interpretation for certain relatives of the Conolly sequence, *J. Integer Seq.* **11** (2008), Article 08.2.1.
[6] E. Barbeau and S. Tanny, On a Strange Recursion of Golomb, *Electronic J. of Combinatorics* 3 (1996), R8.
[7] E. Barbeau, J. Chew, and S. Tanny, A matrix dynamics approach to Golomb's recursion, *Electronic J. of Combinatorics* 4 (1) (1997), R16.
[8] M. Cai and S. Tanny, How the shift parameter affects the behavior of a family of meta-Fibonacci sequences, *J. of Integer Sequences* 11 (2008), Article 08.3.6.
[9] Joseph Callaghan, John J. Chew III, and Stephen M. Tanny, On the Behavior of a Family of Meta-Fibonacci Sequences, *SIAM J. Discrete Math.* 18(4) (2005), 794–824.
[10] B.W. Conolly, Fibonacci and meta-Fibonacci sequences, in: S. Vajda. ed., *Fibonacci & Lucas Numbers and the Golden Section: Theory and Applications*, E. Horwood Ltd., Chichester, 1989, 127–139.
[11] J. H. Conway, Some crazy sequences, videotaped talk at ATT Bell Labs, July 15, 1988.
[12] B. Dalton, M. Rahman and S. Tanny, Spot-based generations for meta-Fibonacci sequences, *Experimental Math*, 20 (2) (2011), 129-137.
[13] C. Deugau and F. Ruskey, Complete $k$-ary trees and generalized meta-Fibonacci sequences, *Fourth Colloquium on Mathematics and Computer Science: Algorithms, trees, Combinatorics and Probabilities*, DMTCS Proceedings Series, 2006 AG, pp. 203-214.
[14] A. Erickson, A. Isgur, B.W. Jackson, F. Ruskey and S. Tanny, Nested Recurrence Relations with Conolly-like Solutions, *Siam J. Discrete Math*, **26** (1) (2012), 206–238.
[15] Solomon W. Golomb, *Discrete chaos: sequences satisfying strange recursions*, preprint, undated.
[16] J. Grytczuk, Another variation on Conway's recursive sequence, *Discrete Mathematics* **282** (2004), 149–161.
[17] J. Higham and S. Tanny, More Well-Behaved Meta-Fibonacci Sequences, *Congressus Numerantium* 98 (1993), 3-17.
[18] D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Random House, 1979.
[19] D. Hofstadter and G. Huber, Private communications and seminar at University of Toronto, March 2000.
[20] A. Isgur, Solving nested recursions with trees, *Ph.D. thesis*, 2012, University of Toronto.
[21] A. Isgur, M. J. Kim, J. Milcak, and S. Tanny, "Golomb-like" nested recursions with Beatty function solutions, *J. of Difference Equations and Applications* 19 (3) (2013), 372-383.
[22] A. Isgur, V. Kuznetsov, and S. Tanny, Nested recursions with ceiling function solutions, *J. Difference Equations and Applications* 18 (6) (2012), 1015–1026. (DOI:10.1080/10236198.2012.662967).
[23] A. Isgur, V. Kuznetsov, M. Rahman, S. Tanny, Nested recursions, simultaneous parameters and tree superpositions, *Electronic J. of Combinatorics* 21 (1) (2014), P1.49.
[24] A. Isgur, V. Kuznetsov, and S. Tanny, A combinatorial approach for solving certain nested recursions with non-slow solutions, *J. Difference Equations and Applications*, 19 (4) (2013), 605–614.
[25] A. Isgur, M. Rahman, On variants of Conway and Conolly's Meta-Fibonacci recursions, *Electron. J. Combin.* 18 (1) (2011), Article P96.
[26] A. Isgur, D. Reiss, and S. Tanny, *Trees and Meta-Fibonacci Sequences* 16 (2009), #R129.
[27] B. Jackson and F. Ruskey, Meta-Fibonacci sequences, binary trees and extremal compact codes, *Electron. J. Combin.* **13** (2006), R26.
[28] T. Kubo and R. Vakil, On Conway's recursive sequence, *Discrete Mathematics* 152 (1996), 225-252.
[29] C. L. Mallows, Conway's challenge sequence, *American Math Monthly* 98 (1991), 5-20.
[30] D. Newman, Problem E3274, *American Math Monthly* 95 (1988), 555.
[31] John A. Pelesko, Generalizing the Conway-Hofstadter $10,000 sequence, *J. Integer Seq.* **7** (2004), Article 04.3.5.
[32] K. Pinn, Order and chaos in Hofstadter's Q(n) sequence, *Complexity* 4 (3), (1999), 41-46.
[33] F. Ruskey and C. Deugau, The combinatorics of certain $k$-ary meta-Fibonacci sequences, *J. Integer Seq.* **12** (2009), Article 09.4.3.
[34] N. J. A. Sloane, *Online Encyclopedia of Integer Sequences*, `https://oeis.org/`.
[35] S.M. Tanny, A well-behaved cousin of the Hofstadter sequence, *Discrete Mathematics*, 105 (1992) 227–239.

Department of Mathematics, University of Toronto, 40 St. George Street, Toronto, ON M5S 2E4, Canada

*E-mail address*, Abraham Isgur: umarovi@gmail.com

*E-mail address*, Robert Lech: robert.lech@mail.utoronto.ca

*E-mail address*, Scott Moore: scott.moore@mail.utoronto.ca

*E-mail address*, Stephen Tanny: tanny@math.utoronto.ca

*E-mail address*, Yvon Verberne: yvon.verberne@mail.utoronto.ca

*E-mail address*, Maria Yifan Zhang: mariayifan.zhang@mail.utoronto.ca