Hofstadter's problem for curious readers

Pierre letouzey

pierre.letouzey@inria.fr
Team πr^2, PPS, Univ Paris Diderot & INRIA Paris-Rocquencourt
Technical report version 1.3, Sept 2015, license CC-BY

1 Introduction

This document summarizes the proofs made during a Coq development in Summer 2015. This development investigates the function G introduced by Hofstadter in his famous "Gödel, Escher, Bach" book[1], as well as a related infinite tree. The left/right flipped variant $\overline{G}$ of this G tree has also been studied here, following Hofstadter's "problem for the curious reader". The initial G function is refered as sequence A005206 in OEIS[4], while the flipped version $\overline{G}$ is the sequence A123070 [5].

The detailed and machine-checked proofs can be found in the files of this development[1] and can be re-checked by running Coq [2] version 8.4 on it. No prior knowledge of Coq is assumed here, on the contrary this document has rather been a "Coq-to-English" translation exercise for the author. Nonetheless, some proofs given in this document are still quite sketchy: in this case, the interested reader is encouraged to consult the Coq files given as references.

2 Prior art and contributions

Most of the results proved here were already mentioned elsewhere, e.g. on the OEIS pages for G and $\overline{G}$. These results were certified in Coq without consulting the existing proofs in the literature, so the proofs presented here might still be improved. To the best of my knowledge, the main novelties of this development are:

1. A proof of the recursive equation for $\overline{G}$ which is currently mentioned as a conjecture in OEIS:

$$\forall n > 3, \overline{G}(n) = n + 1 - \overline{G}(1 + \overline{G}(n - 1))$$

[1] See http://www.pps.univ-paris-diderot.fr/~letouzey/hofstadter_g

2. The statement and proof of another equation for $\overline{G}$:

$$\forall n > 3, \overline{G}(\overline{G}(n)) + \overline{G}(n-1) = n$$

   Although simpler than the previous one, this equation isn't enough to characterize $\overline{G}$ unless an extra monotonocity condition on $\overline{G}$ is assumed.

3. The statement and proof of a result comparing $\overline{G}$ and G: for all n, $\overline{G}(n)$ is either $G(n)+1$ or $G(n)$, depending on the existence of a decomposition of n as sum of Fibonacci numbers of the form $F_2 + F_{2p} + ...$ where $F_2 = 2$. Moreover these specific numbers where $\overline{G}$ and G differ are separated by either 5 or 8.

4. The studies of G and $\overline{G}$ "derivatives" ($\Delta G(n) = G(n+1) - G(n)$ and similarly for $\Delta\overline{G}$), leading to yet another characterization of G and $\overline{G}$.

3 Coq jargon

For readers without prior knowledge of Coq but curious enough to have a look at the actual files of this development, here comes a few words about the Coq syntax.

- By default, we're manipulating natural numbers only, corresponding to Coq type `nat`. Only exception: in file `Phi.v`, we switch to real numbers to represent the golden ratio.
- The symbol `S` denotes the successor of natural numbers. Hence `S(2)` is the same as 3.
- As in many modern functional languages such as OCaml or Haskell, the usage is to skip parenthesis whenever possible, and in particular around atomic arguments of functions. Hence `S(x)` will rather be written `S x`.
- The symbol `pred` is the predecessor for natural numbers. In Coq, all functions must be total, and `pred 0` has been chosen to be equal to 0. Similarly, the subtraction on natural numbers is rounded: `3 - 5 = 0`.
- Coq allows to define custom predicates to express various properties of numbers, lists, etc. These custom predicates are introduced by the keyword `Inductive`, followed by some "introduction" rules for the new predicate. In this development we use capitalized names for these predicates (e.g. `Delta`, `Low`, ...).
- Coq also accepts new definition of recursive functions via the command `Fixpoint`. But these definitions should satisfy some criterion

to guarantee that the new function is well-defined and total. So in this development, we also need sometimes to define new functions via explicit justification of termination, see for instance `norm` in `Fib.v` or `g_spec` in `FunG.v`.

4 Fibonacci numbers and decompositions

This section corresponds to file `Fib.v`.

Convention. In this document and in the Coq development, we took the following non-standard definition of Fibonacci numbers F_n:

$$F_0 = F_1 = 1$$

$$\forall n, \quad F_{n+2} = F_n + F_{n+1}$$

Please note the unusual $F_0 = 1$ instead of 0. The ranks are hence shifted by one when compared with, say, OEIS's sequence A000045 [3]. With this choice, we avoid considering zeros in the decompositions of numbers as sums of Fibonacci numbers (see below). In Coq, these F_n numbers correspond to the `fib` function, and we start by proving a few basic properties : strict positivity, monotony, strict monotony above 1, etc. We also prove the theorem `fib_inv` which states that any positive number can by bounded by consecutive Fibonacci numbers :

Theorem 1 (Fibonacci inverse)**.** $\forall n > 0, \exists k, F_k \leq n < F_{k+1}$.

4.1 Fibonacci decompositions

The rest of the file `Fib.v` deals with the decompositions of numbers as sums of Fibonacci numbers.

Definition 1. *A decomposition $n = \Sigma F_i$ is said to be* canonical *if:*

(a) F_0 does not appear in the decomposition

(b) A Fibonacci number appears at most once in the decomposition

(c) Fibonacci numbers in the decomposition aren't consecutive

A decomposition is said to be relaxed *if at least conditions (a) and (b) hold.*

For instance, 11 admits the canonical decomposition $F_3 + F_5 = 3 + 8$, but also the relaxed decomposition $F_1 + F_2 + F_3 + F_4 = 1 + 2 + 3 + 5$. On a technical level, we represented in Coq the decompositions as sorted lists of ranks, and we used a predicate `Delta` p l to express that any element in the list l exceeds the previous element by at least p. We hence use $p = 2$ for canonical decompositions, and $p = 1$ for relaxed ones. See file `DeltaList.v` for the definition and properties of this predicate.

Then we proved Zeckendorf's theorem (actually discovered earlier by Lekkerkerker):

Theorem 2 (Zeckendorf)**.** *Any natural number has a unique canonical decomposition.*

Proof. The proof of this theorem is quite standard : to decompose n, take the highest F_k which is less or equal to n (cf `fib_inv` below), and continue recursively with $n - F_k$. We cannot obtain this way two consecutive F_{k+1} and F_k, otherwise F_{k+2} could have been used in the first place. And F_0 isn't necessary in the decomposition since we could use $F_1 = 1$ instead. For proving the uniqueness, the key ingredient is that this kind of sum cannot exceed the next Fibonacci number. For instance $F_1 + F_3 + F_5 = 1 + 3 + 8 = 12 = F_6 - 1$. □

Definition 2. *For a non-empty decomposition, its* lowest rank *is the rank of the lowest Fibonacci number in this decomposition. By extension, the lowest rank $low(n)$ of a number $n \neq 0$ is the lowest rank of its (unique) canonical decomposition.*

For instance $low(11) = 3$. In Coq, $low(n) = k$ is written via the predicate `Low 2` n k. Note that this notion is in fact more general : `Low 1` n k says that n can be decomposed via a relaxed decomposition of lowest rank k.

Interestingly, a relaxed decomposition can be transformed into a canonical one (see Coq function `norm`):

Theorem 3 (normalization)**.** *Consider a relaxed decomposition of n, made of k terms and whose lowest rank is r. We can build a canonical decomposition of n, made of no more than k terms, and whose lowest rank can be written $r + 2p$ with $p \geq 0$.*

Proof. We simply eliminate the highest consecutive Fibonacci numbers (if any) by summing them : $F_m + F_{m+1} \to F_{m+2}$, and repeat this process. By dealing with highest numbers first, we avoid the apparition of duplicated terms in the decomposition: F_{m+2} wasn't already in the decomposition, otherwise we would have considered $F_{m+1} + F_{m+2}$ instead. Hence all the obtained decompositions during the process are correct relaxed decompositions. The number of terms is decreasing by 1 at each step, so the process is guaranteed to terminate, and will necessarily stops on a decomposition with no consecutive Fibonacci numbers : we obtain indeed the canonical decomposition of n. And finally, it is easy to see that the lowest rank is either left intact or raised by 2 at each step of the process. □

4.2 Classifications of Fibonacci decompositions.

The end of `Fib.v` deals with classifications of numbers according to the lowest rank of their canonical decomposition. In particular, this lowest rank could be 1, 2 or more. It will also be interesting to distinguish between lowest ranks that are even or odd. These kind of classifications and their properties will be heavily used during theorem 26 which compares $\overline{G}$ and G. For instance:

Theorem 4 (decomposition of successor)**.**

1. *$low(n) = 1$ implies $low(n+1)$ is even.*
2. *$low(n) = 2$ implies $low(n+1)$ is odd and different from 1.*
3. *$low(n) > 2$ implies $low(n+1) = 1$.*

Proof. Let r be $low(n)$. We can write $n = F_r + \Sigma F_i$ in a canonical way.

1. If $r = 1$, then $n+1 = F_2 + \Sigma F_i$ is a relaxed decomposition, and we conclude thanks to the previous normalization theorem.
2. If $r = 2$, then $n+1 = F_3 + \Sigma F_i$ is a relaxed decomposition, and we conclude similarly.
3. If $r > 2$, then $n+1 = F_1 + F_r + \Sigma F_i$ is directly a canonical decomposition.

□

We also studied the decomposition of the predecessor $n-1$ in function of n. This decomposition depends of the parity of the lowest rank in n, since for $r \neq 0$ we have:

$$F_{2r} - 1 = F_1 + F_3 + ... + F_{2r-1}$$
$$F_{2r+1} - 1 = F_2 + F_4 + ... + F_{2r}$$

Hence:

Theorem 5 (decomposition of predecessor)**.**

1. *If $low(n)$ is even then $low(n-1) = 1$.*
2. *If $low(n)$ is odd and different from 1, then $low(n-1) = 2$.*
3. *For $n > 1$, if $low(n) = 1$ then $low(n-1) > 2$.*

Proof.

1. Let the canonical decomposition of n be $F_{2r} + \Sigma F_i$. Then we have $n-1 = F_1 + ... + F_{2r-1} + \Sigma F_i$, which is also a canonical decomposition. Moreover $r \neq 0$ since F_0 isn't allowed in decompositions. Hence the decomposition of $n-1$ contains at least the term F_1, and $low(n-1) = 1$.

2. When $low(n) = 2r + 1$ with $r \neq 0$, we decompose similarly $F_{2r+1} - 1$, leading to a lowest rank 2 for $n - 1$.

3. Finally, when $n = F_1 + \Sigma F_i$, then the canonical decomposition of $n-1$ is directly the rest of the decomposition of n, which isn't empty (thanks to the condition $n > 1$) and cannot starts by F_1 nor F_2 (for canonicity reasons).

□

4.3 Subdivision of decompositions starting by F_2

When $low(n) = 2$ and $n > 2$, the canonical decomposition of n admits at least a second-lowest term: $n = F_2 + F_k + ...$ and we will sometimes need to consider the parity of this second-lowest rank.

Definition 3.

- *A number n is said* 2-even *when its canonical decomposition is of the form $F_2 + F_{2p} + ...$.*
- *A number n is said* 2-odd *when its canonical decomposition is of the form $F_2 + F_{2p+1} + ...$*

In Coq, this corresponds to the `TwoEven` and `TwoOdd` predicates. In fact, the 2-even numbers will precisely be the locations where G and $\overline{G}$ differs (see theorem 26). The first of these 2-even number is $7 = F_2 + F_4$, followed by $15 = F_2 + F_6$ and $20 = F_2 + F_4 + F_6$.

A few properties of 2-even and 2-odd numbers:

Theorem 6.

1. *A number n is 2-even if and only if it admits a* relaxed *decomposition of the form $F_2 + F_{2p} + ...$.*
2. *A number n is 2-even if and only if $low(n) = 2$ and $low(n-2)$ is even.*
3. *A number n is 2-odd if and only if $low(n) = 2$ and $low(n-2)$ is odd.*
4. *When $low(n)$ is odd and at least 5, then $n - 1$ is 2-even.*
5. *Two consecutive 2-even numbers are always apart by 5 or 8.*

Proof.

1. The left-to-right implication is obvious, since a canonical decomposition can in particular be seen as a relaxed one. Suppose now the existence of such a relaxed decomposition. During its normalization, the F_2 will necessarily be left intact, and the term F_{2p} can grow, but only by steps of 2.

2. Once again, the left-to-right implication is obvious. Conversely, we start with the canonical decomposition of $n-2 = F_{2p} + \Sigma F_i$. If $p \neq 1$, this leads to the desired canonical decomposition of n as $F_2 + F_{2p} + \Sigma F_i$. And the case $p = 1$ is impossible: assume $p = 1$, then we could turn the decomposition of $n-2$ into a canonical decomposition of $n-4$ whose lowest rank is at least 4. This gives us a relaxed decomposition of n of the form $F_1 + F_3 + \Sigma F_i$. After normalization, we would obtain that $low(n)$ is odd, which is contradictory with $low(n) = 2$.

3. We proceed similarly for $low(n) = 2$ and $low(n-2)$ odd implies that n is 2-odd. First we have a canonical decomposition of $n-2 = F_{2p+1} + \Sigma F_i$. Then $p = 0$ would give a relaxed decomposition of n starting by 3, hence $low(n)$ odd, impossible. And $p <> 0$ allows to write $n = F_2 + F_{2p+1} + \Sigma F_i$ in a canonical way.

4. When $n = F_{2p+1} + \Sigma F_i$ with $p > 1$, then $n - 1 = F_2 + F_4 + F_{2p} + \Sigma F_i$ and the condition on p ensures that the terms F_2 and F_4 are indeed present.

5. If n is 2-even, it could be written $F_2 + F_{2p} + \Sigma F_i$. When $p > 2$, then $n + 5 = F_2 + F_4 + F_{2p} + \Sigma F_i$ is also 2-even. When $p = 2$, we need to consider the third term in the decomposition (if any).

   - Either n is of the form $F_2 + F_4 + F_6 + ...$ Then $n + F_5 = F_2 + F_4 + F_7 + ...$ is a relaxed decomposition of $n+8$, hence $n+8$ is 2-even (see the first part of the current theorem above).
   - Either n is of the form $F_2 + F_4 + \Sigma F_i$ where all terms in ΣF_i are strictly greater than F_6. Then $n + F_5 = F_2 + F_6 + \Sigma F_i$ is a relaxed decomposition of $n+8$, hence $n+8$ is 2-even.

   We finish the proof by considering all intermediate numbers between n and $n+8$ and we show that $n+5$ is the only one of them that might be 2-even:

   - $low(n+1)$ is odd (cf. theorem 4).
   - $n+2 = F_1 + F_3 + F_{2p} + \Sigma F_i$, and normalizing this relaxed decomposition shows that $low(n+2) = 1$.
   - $n+3 = F_2 + F_3 + F_{2p} + \Sigma F_i$, and normalizing this relaxed decomposition will either combine F_3 with some higher terms (leading

to an odd second-lowest term and hence $n+3$ is 2-odd) or combine F_3 with F_2 (in which case $low(n+3) \geq 4$).

- $n+4 = F_1 + F_2 + F_3 + F_{2p} + \Sigma F_i$, hence $low(n+4)$ is odd.
- $n+6 = F_5 + F_{2p} + \Sigma F_i$ is a relaxed decomposition whose lowest rank is either 5 (when $p > 2$) or 4 (when $p = 2$). After normalization, we obtain $low(n+6) \geq 4$.
- Hence $low(n+7) = 1$ by last case of theorem 4.

□

5 The G function

This section corresponds to file `FunG.v`.

5.1 Definition and initial study of G

Theorem 7. *There exists a unique function $G : \mathbb{N} \to \mathbb{N}$ which satisfies the following equations:*

$$G(0) = 0$$

$$\forall n > 0, \quad G(n) = n - G(G(n-1))$$

Proof. The difficulty is the second recursive call of G on $G(n-1)$. A priori $G(n-1)$ could be arbitrary, so how could we refer to it during a recursive definition of G ? Fortunately $G(n-1)$ will always be strictly lower than n, and that will allow this recursive call. More formally, we prove by induction on n the following statement: for all n, there exists a sequence $G_0...G_n$ of numbers in $[0..n]$ such that $G_0 = 0$ and $\forall k \in [1..n], G_k = k - G_{k'}$ where $k' = G_{k-1}$.

- For $n = 0$, an adequate sequence is obviously $G_0 = 0$.
- For some n, suppose we have already proved the existence of an adequate sequence $G_0...G_n$. In particular $G_n \in [0..n]$, hence G_{G_n} is well-defined and is also in $[0..n]$. Finally $(n+1) - G_{G_n}$ is in $[1..(n+1)]$. We define G_{n+1} to be equal to this $(n+1) - G_{G_n}$ and obviously keep the previous values of the sequence. All these values are indeed in $[0..(n+1)]$, and the recursive equations are satisfied up to $k = n+1$.

All these finite sequences that extend each other lead to an infinite sequence $(G_n)_{n \in \mathbb{N}}$ of natural numbers, which can also be seen as a function $G : \mathbb{N} \to \mathbb{N}$, that satisfy the desired equations by construction.

For the uniqueness, we should first prove that any function f satisfying $f(0) = 0$ and our recursive equation above is such that $\forall n, 0 \leq f(n) \leq n$. This proof can be done by strong induction over n, and a bit of upper and lower bound manipulation. Then we could prove (still via strong induction over n) that $\forall n, f(n) = g(n)$ when f and g are any functions satisfying our equations. □

The initial values are $G(0) = 0, G(1) = G(2) = 1, G(3) = 2, G(4) = G(5) = 3$. We can then establish some basic properties of G:

Theorem 8.

1. *$\forall n, 0 \leq G(n) \leq n$.*
2. *$\forall n \neq 0, G(n) = G(n-1)$ implies $G(n+1) = G(n) + 1$.*
3. *$\forall n, G(n+1) - G(n) \in \{0, 1\}$.*
4. *$\forall n, m, n \leq m$ implies $0 \leq G(m) - G(n) \leq m - n$.*
5. *$\forall n, G(n) = 0$ if and only if $n = 0$.*
6. *$\forall n > 1, G(n) < n$.*

Proof.

1. Already seen during the definition of G.
2. $G(n+1) - G(n) = (n+1) - G(G(n) - n + G(G(n-1)) = 1 - (G(G(n)) - G(G(n-1)))$. If $G(n) = G(n-1)$ then the previous expression is equal to 1.
3. By strong induction over n. First, $G(1) - G(0) = 1 - 0$. Then for a given $n \neq 0$, we assume $\forall k < n, G(k+1) - G(k) \in \{0, 1\}$. We reuse the same formulation of $G(n+1) - G(n)$ as before. By induction hypothesis for $k = n - 1$, $G(n) - G(n-1) \in \{0, 1\}$. If it's 0, then $G(n+1) - G(n) = 1$ as before. If it's 1, we could use another induction hypothesis for $k = G(n-1)$ (and hence $k + 1 = G(n)$), leading to $G(G(n)) - G(G(n-1)) \in \{0, 1\}$ and the desired result.
4. Mere iteration of the previous result between n and m.
5. For all $n \geq 1$, $0 \leq G(n) - G(1)$ and $G(1) = 1$.
6. For all $n \geq 2$, $G(n) - G(2) \leq n - 2$ and $G(2) = 1$.

□

We can also say that $lim_{+\infty} G = +\infty$, since G is monotone and grows by at least 1 every 2 steps (any stagnation is followed by a growth). Taking into account this limit, and $G(0) = 0$, and the growth by steps of 1, we can also deduce that G is onto: any natural number has at least one antecedent by G. Moreover there cannot exists more than two antecedents for a given value: by monotonicity, these antecedents are neighbors, and having more than two would contradict the "stagnation followed by growth" rule. We can even provide explicitly one of these antecedent:

Theorem 9. $\forall n, G(n + G(n)) = n$.

Proof. We've just proved that n has at least one antecedent, and no more than two. Let k be the largest of these antecedents. Hence $G(k) = n$ and $G(k + 1) \neq n$, leading to $G(k + 1) = G(k) + 1$. If we re-inject this into the defining equation $G(k + 1) = k + 1 - G(G(k))$, we obtain that $G(k) + G(G(k)) = k$ hence $n + G(n) = k$, and finally $G(n + G(n)) = G(k) = n$. □

As shown during the previous proof, $n + G(n)$ is actually the largest antecedent of n by G. In particular $G(n + G(n) + 1) = n + 1$. And if n has another antecedent, it will hence be $n + G(n) - 1$.

From this, we can deduce a first relationship between G and Fibonacci numbers.

Theorem 10. *For all* $k \neq 0$, $G(F_k) = F_{k-1}$.

Proof. By induction over k. First, $G(F_1) = G(1) = 1 = F_0$ and $G(1 + F_2) = G(3) = 2 = 1 + F_1$. Take now a $k \neq 0$ and assume the result for k. Then $G(F_{k+1}) = G(F_k + F_{k-1}) = G(F_k + G(F_k)) = F_k$. □

Moreover, for $k > 1$, $F_k = F_{k-1} + G(F_{k-1})$ is the largest antecedent of F_{k-1}, hence $G(1 + F_k) = 1 + F_{k-1}$.

We could also establish a alternative equation for G, which will be used during the study of function $\overline{G}$:

Theorem 11. *For all* n *we have* $G(n) + G(G(n + 1) - 1) = n$.

Proof. First, this equation holds when $n = 0$ since $G(0) + G(G(1) - 1) = G(0) + G(1 - 1) = 0$. We consider now a number $n \neq 0$. Either $G(n + 1) = G(n)$ or $G(n + 1) = G(n) + 1$.

- In the first case, $G(n-1)$ cannot be equal to $G(n)$ as well, otherwise G would stay flat longer than possible. Hence $G(n-1) = G(n)-1$. Hence $G(n)+G(G(n+1)-1) = G(n)+G(G(n)-1) = G(n)+G(G(n-1)) = n$.
- In the second case: $G(n)+G(G(n+1)-1) = G(n+1)-1+G(G(n)) = (n + 1) - 1 = n$.

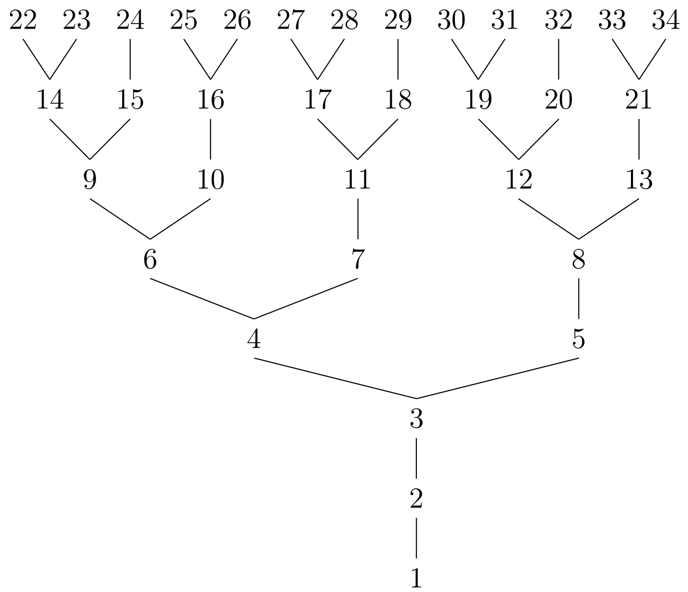
□

5.2 The associated G tree

Hofstadter proposed to associate the G function with an infinite tree satisfying the following properties:

- The nodes are labeled by all the positive numbers, in the order of a left-to-right breadth-first traversal, starting at 1 for the root of the tree.
- For $n \neq 0$, the node n is a child of the node $g(n)$.

In practice, the tree can be constructed progressively, node after node. For instance, the node 2 is the only child of 1, and 3 is the only child of 2, while 4 and 5 are the children of 3. Now come the nodes 6 and 7 on top of 4, and so on. The picture below represents the tree up to depth 7 (assuming that the root is at depth 0).



This construction process will always be successful : when adding the new node n, this new node n will be linked to a parent node already constructed, since $G(n) < n$ as soon as $n > 1$. Moreover, G is monotone and grows by at most 1 at each step, hence the position of the new node n will be compatible with the left-to-right breadth-first ordering: n has either the same parent as $n-1$, and we place n to the right of $n-1$, or $G(n) = G(n-1)+1$ in which case n is either to be placed on the right of $n-1$, or at a greater depth.

Moreover, since G is onto, each node will have at least one child, and we have already seen that each number has at most two antecedents by G, hence the node arities are 1 or 2.

Tree depth

On the previous picture, we can notice that the rightmost nodes are the Fibonacci numbers, while the leftmost nodes have the form $1 + F_k$. Before proving this fact, let us first define more properly a depth function.

Definition 4. *For a number $n > 0$, we note $depth(n)$ the least number d such that $G^d(n)$ (the d-th iteration of G on n) reaches 1. We complete this definition by choosing arbitrarily $depth(0) = 0$.*

For $n > 0$, such a number d is guaranteed to exists since $G(k) < k$ as long as k is still not 1, hence the sequence $G^k(n)$ is strictly decreasing as long as it hasn't reached 1. In particular, we have $depth(1) = 0$, and for all $n > 1$ we have $depth(n) = 1 + depth(G(n))$. Hence our depth function is compatible with the usual notion of depth in a tree.

Theorem 12.

1. *For all n, $depth(n) = 0$ if and only if $n \leq 1$.*
2. *For $n > 1$ and $k < depth(n)$, we have $G^k(n) > 1$.*
3. *For $n, m > 0$, $n \leq m$ implies $depth(n) \leq depth(m)$.*
4. *For $k > 0$, $depth(F_k) = k - 1$.*
5. *For $k > 0$, $depth(1 + F_k) = k$.*
6. *For $k > 0, n > 0$, $depth(n) = k$ if and only if $1 + F_k \leq n \leq F_{k+1}$.*

Proof.

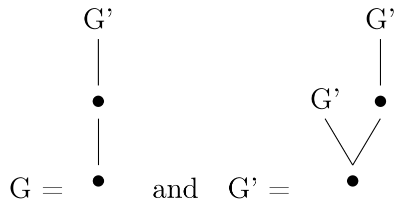
1. The statement is valid when $n = 0$. For a $n > 0$, $depth(n) = 0$ means $G^0(n) = 1$ i.e. $n = 1$.
2. Consequence of the minimality of depth.
3. Consequence of the monotony of G : for all k, $G^k(n) \leq G^k(m)$ hence the iterates of n will reach 1 faster than the iterates of m.
4. G maps a Fibonacci number to the previous one, $k - 1$ iterations of G on F_k leads to $F_1 = 1$.
5. G maps a successor of a Fibonacci number to the previous such number, $k - 1$ iterations of G on $1 + F_k$ leads to $1 + F_1 = 2$.
6. If $1 + F_k \leq n \leq F_{k+1}$, the monotony of depth and the previous facts gives $k \leq depth(n) \leq (k + 1) - 1$, hence $depth(n) = k$. Conversely, when $depth(n) = k$, we cannot have $n < 1 + F_k$ otherwise $n \leq F_k$ and hence $depth(n) \leq k - 1$, and we cannot have $n > F_{k+1}$ otherwise $n \geq 1 + F_{k+1}$ and hence $depth(n) \geq k + 1$.

□

The previous characterization of *depth*(n) via Fibonacci bounds shows that depth could also have been defined thanks to `fib_inv`$(n-1)$, see 1. It also shows that the number of nodes at depth k is $F_{k+1}-(1+F_k)+1 = F_{k-1}$.

The shape of the G tree

If we ignore the node labels and concentrate on the shape of the G tree, we encounter a great regularity. G starts with two unary nodes that are particular cases (labeled 1 and 2), and after that we encounter a sub-tree G' whose shape is obtained by repetitions of the same basic pattern:



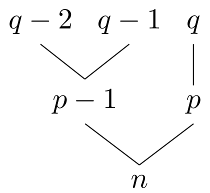
G' has hence a fractal shape: it appears as a sub-tree of itself. To prove the existence of such a pattern, we study the arity of children nodes.

Theorem 13. *In the G tree, a binary node has a left child which is also binary, and a right child which is unary, while the unique child of a unary node is itself binary.*

Proof. First, we show that the leftmost child of a node is always binary. Let n be a node, and p its leftmost child, i.e. $G(p) = n$ and $G(p-1) = n-1$. We already know one child $q = p + G(p)$ of p, let's now show that $q-1$ is also a child of p. For that, we consider $p-1$ and its rightmost child $q' = p-1+G(p-1)$. Rightmost means that $G(q'+1) = p$. So $G(q-1) = G(p+G(p)-1) = G(p+n-1) = G(p+G(p-1)) = G(q'+1) = p$ and we can conclude.

We can now affirm that the unique child of a unary node is itself binary, since this child is in particular the leftmost one.

Let's now consider a binary node n, with its right child $p = n + G(n)$ and its left child $p-1$: $G(p) = G(p-1) = n$.



The leftmost child $p-1$ is already known to be binary. Let's now show that the right child p is unary. If p has a second child, it will be $q-1$, but: $G(q-1) = G(p+G(p)-1) = G(p-1+G(p-1)) = p-1$. so $q-1$ is a child of $p-1$ rather than p, and p is indeed a unary node. □

5.3 G and Fibonacci decompositions

Theorem 14. *Let $n = \Sigma F_i$ be a relaxed Fibonacci decomposition. Then $G(n) = \Sigma F_{i-1}$, the Fibonacci decomposition obtained by removing 1 at all the ranks in the initial decomposition.*

For instance $G(11) = G(F_3 + F_5) = F_2 + F_4 = 7$. Note that the obtained decomposition isn't necessarily relaxed anymore, since a F_0 might have appeared if F_1 was part of the initial decomposition. The theorem also holds for canonical decompositions since they are a particular case of relaxed decompositions. Once again, the obtained decompositions might not be canonical since F_0 might appear, but in this case we could turn this F_0 into a F_1. This gives us a relaxed decomposition that we can re-normalize. For instance $G(12) = G(F_1 + F_3 + F_5) = F_0 + F_2 + F_4 = F_1 + F_2 + F_4 = F_3 + F_4 = F_5 = 8$.

Proof. We proceed by strong induction over n. The case 0 is obvious, since the only possible decomposition is the empty one, and the "shifted" decomposition is still empty, as required by $G(0) = 0$. We now consider a number $n \neq 0$, with a non-empty relaxed decomposition $n = F_k + \Sigma F_i$, and assume the statement to be true for all $m < n$. We will use the recursive equation $G(n) = n - G(G(n-1))$, and distinguish many cases according to the value of k.

- Case $k = 1$. Then $n - 1 = \Sigma F_i$. A first induction hypothesis gives $G(n-1) = \Sigma F_{i-1}$. Since the initial decomposition was relaxed, ΣF_i cannot contain F_1, hence ΣF_{i-1} is still a relaxed decomposition. As seen many times now, $G(n-1) < n$ here, and we're free to use a second induction hypothesis leading to $G(G(n-1)) = \Sigma F_{i-2}$. Finally $G(n) = 1 + \Sigma F_i - \Sigma F_{i-2} = 1 + \Sigma(F_i - F_{i-2}) = F_0 + \Sigma F_{i-1}$, as required.

- Case k even. Then k can be written $2p$ with $p > 0$. Then $n - 1 = F_{2p} - 1 + \Sigma F_i = F_1 + F_3 + ... + F_{2p-1} + \Sigma F_i$. The induction hypothesis on $n - 1$ gives:

  $$G(n-1) = F_0 + F_2 + ... + F_{2p-2} + \Sigma F_{i-1}$$

  We turn the F_0 above into a F_1, obtaining again a relaxed decomposition, for which a second induction hypothesis gives:

  $$G(G(n-1)) = F_0 + F_1 + ... + F_{2p-3} + \Sigma F_{i-2}$$

  $$G(G(n-1)) = 1 + (F_{2p-2} - 1) + \Sigma F_{i-2}$$

  And so:

  $$G(n) = F_{2p} - F_{2p-2} + \Sigma(F_i - F_{i-2}) = F_{2p-1} + \Sigma F_{i-1}$$

- Case k odd and distinct from 1. Then k can be written $2p+1$ with $p > 0$. Then $n-1 = F_{2p+1} - 1 + \Sigma F_i = F_2 + F_4 + ... + F_{2p} + \Sigma F_i$. The induction hypothesis on $n-1$ gives:

$$G(n-1) = F_1 + F_3 + ... + F_{2p-1} + \Sigma F_{i-1}$$

Since this is still a relaxed decomposition, we use a second induction hypothesis on it, hence:

$$G(G(n-1)) = F_0 + F_2 + ... + F_{2p-2} + \Sigma F_{i-2}$$

$$G(G(n-1)) = 1 + (F_{2p-1} - 1) + \Sigma F_{i-2}$$

And so:

$$G(n) = F_{2p+1} - F_{2p-1} + \Sigma(F_i - F_{i-2}) = F_{2p} + \Sigma F_{i-1}$$

□

Thanks to this characterization of the effect of G on Fibonacci decomposition, we can now derive a few interesting properties.

Theorem 15.

1. *$G(n+1) = G(n)$ if and only if $low(n) = 1$.*
2. *If $low(n)$ is even, then $G(n-1) = G(n)$.*
3. *If $low(n)$ is odd, then $G(n-1) = G(n) - 1$.*
4. *For $n \neq 0$, the node n of tree G is unary if and only if $low(n)$ is even.*

Proof.

1. If $low(n) = 1$, then we can write a canonical decomposition $n = F_1 + \Sigma F_i$, hence $n+1 = F_2 + \Sigma F_i$ is a relaxed decomposition. By the previous theorem, $G(n) = F_0 + \Sigma F_{i-1}$ while $G(n+1) = F_1 + \Sigma F_{i-1}$, hence the desired equality. Conversely, we consider a canonical decomposition of n as ΣF_i. If F_1 isn't part of this decomposition, then $n+1 = F_1 + \Sigma F_i$ is a correct relaxed decomposition, leading to $G(n+1) = 1 + \Sigma F_{i-1} = 1 + G(n)$. So $low(n) \neq 1$ implies $G(n+1) \neq G(n)$.

2. If $low(n)$ is even, we've already shown in theorem 5 that $low(n-1) = 1$ hence $G(n) = G(n-1)$ by the previous point.

3. If $low(n)$ is odd, the same theorem 5 shows that $low(n-1)$ is 2 or more, provided that $n > 1$. In this case the first point above allows to conclude. We now check separately the cases $n = 0$ (irrelevant here since $low(0)$ doesn't exists) and $n = 1$ (for which $G(1-1)$ is indeed $G(1) - 1$).

4. We've already seen that a node n is binary whenever $G(n+G(n)-1)=n$. When $low(n)$ is even, we have $G(n-1)=G(n)$, hence $G(n+G(n)-1)=G(n-1+G(n-1))=n-1$, hence n is unary. When $low(n)$ is odd, we have $n+G(n)-1=(n-1+G(n-1))+1$: this is the next node to the right after the rightmost child of $n-1$, its image by G is hence n, and finally n is indeed binary.

□

Theorem 16. *If $low(n)>1$, then $low(G(n))=low(n)-1$. In particular, if $low(n)$ is even, then $low(G(n))$ is odd, and if $low(n)$ is odd and different from 1, then $low(G(n))$ is even.*

Proof. This is a direct application of theorem 14: when $low(n)>1$, the decomposition we obtain for $G(n)$ is still canonical, and its lowest rank is $low(n)-1$. The statements about parity are immediate consequences. □

Theorem 17.

1. *If $low(n)=1$, then $low(G(n))$ is odd.*
2. *If $low(n)=2$, then $low(G(n))=1$.*
3. *If $low(n)>2$, then $low(G(n))>1$ and $low(G(n)+1)$ is odd.*

Proof.

1. Once again, we consider a canonical decomposition $n=F_1+\Sigma F_i$. Then $G(n)=F_0+\Sigma F_{i-1}$. If we turn the F_0 into a F_1, we obtain a relaxed decomposition, that can be normalized into a canonical decomposition whose lowest rank will hence be odd.
2. Direct application of previous theorem: $low(G(n))=low(n)-1$.
3. Here also, $low(G(n))=low(n)-1$, hence $low(G(n))$ cannot be 1 here. Then the theorem 4 implies that $low(G(n)+1)$ is odd.

□

Theorem 18.

1. *If n is 2-odd, then $G(n)+1$ is 2-even.*
2. *If n is 2-even, then either $G(n)+1$ is 2-odd or $low(G(n)+1)>2$.*

Proof.

1. Take a canonical decomposition $n=F_2+F_{2p+1}+\Sigma F_i$. Hence $G(n)+1=1+F_1+F_{2p}+\Sigma F_{i-1}=F_2+F_{2p}+\Sigma F_{i-1}$, and this decomposition is still canonical (for canonicity reasons, p cannot be 1).

2. Similarly, $n = F_2 + F_{2p} + \Sigma F_i$ implies $G(n)+1 = F_2 + F_{2p-1} + \Sigma F_{i-1}$. When $p > 1$, this decomposition is canonical and $G(n)+1$ is 2-odd. When $p = 1$, this decomposition is only a relaxed one, starting by $F_2 + F_3 + ...$ Its normalization will hence end with a lowest rank of at least 4.

□

5.4 G and its "derivative" ΔG

We consider now the "derivative" ΔG of G, defined via $\Delta G(n) = G(n+1) - G(n)$. We already know from theorem 8 that the output of ΔG is always either 0 or 1.

Theorem 19. *For all n, $\Delta G(n+1) = 1 - \Delta G(n).\Delta G(G(n))$.*

Proof. We already know that $\Delta G(n) = 0$ implies $\Delta G(n+1) = 1$: we cannot have $G(n) = G(n+1) = G(n+2)$. Consider now a n such that $G(n+1) - G(n) = 1$. By using the recursive definition of G, we have $G(n+2) - G(n+1) = (n+2-G(G(n+1))) - (n+1-G(G(n))) = 1-(G(G(n+1)) - G(G(n))$. Hence $\Delta G(n+1) = 1 - (G(G(n)+1) - G(G(n)) = 1 - \Delta G(G(n))$. □

This equation provides a way to express $G(n+2)$ in terms of $G(n+1)$ and $G(n)$ and $G(G(n))$ and $G(G(n)+1)$. Since $n+1$, n, $G(n)$ and $G(n)+1$ are all strictly less than $n+2$, we could use this equation as an alternative way to define recursively G, alongside two initial equations $G(0) = 0$ and $G(1) = 1$. We proved in Coq that G is indeed the unique function to satisfy these equations (see `GD_unique` and `g_implements_GD`).

6 The $\overline{G}$ function

This section corresponds to file `FlipG.v`. Here is a quote from page 137 of Hofstadter's book [1]:

> A problem for curious readers is: suppose you flip Diagram G around as if in a mirror, and label the nodes of the new tree so they increase from left to right. Can you find a recursive *algebraic* definition for this "flip-tree" ?

6.1 The $flip$ function

Flipping the G tree as if in a mirror is equivalent to keeping its shape unchanged, but labeling the nodes from right to left during the breadth-first traversal. Let us call $flip(n)$ the new label of node n after this transformation. We've seen in the previous section that given a depth $k \neq 0$, the nodes at this depth are labeled from $1+F_k$ till F_{k+1}. After the $flip$ transformation, the $1 + F_k$ node and the F_{k+1} node will hence have exchanged their label. Similarly, $2 + F_k$ will become $F_{k+1} - 1$ and vice-versa. More generally, if $depth(n) = k$, the distance between $flip(n)$ and the leftmost node $1 + F_k$ will be equal to the distance between n and the rightmost node F_{k+1}, hence:

$$flip(n) - (1 + F_k) = F_{k+1} - n$$

So:

$$flip(n) = 1 + F_{k+2} - n$$

We finally complete this definition to handle the case $n \leq 1$:

Definition 5. *We define the function $flip : \mathbb{N} \to \mathbb{N}$ in the following way:*

$$flip(n) = if\ \ (n \leq 1)\ then\ n\ else\ 1 + F_{2+depth(n)} - n$$

A few properties of this $flip$ function:

Theorem 20.

1. *For all $n \in \mathbb{N}$, depth(flip(n)) = depth(n).*
2. *For all n, $n > 1$ if and only if $flip(n) > 1$.*
3. *For $k > 0$ and $n < F_{k-1}$, we have $flip(1 + F_k + n) = F_{k+1} - n$.*
4. *$flip$ is involutive.*
5. *For all $n > 1$, if $depth(n+1) = depth(n)$ then $flip(n+1) = flip(n) - 1$.*
6. *For all $n > 1$, if $depth(n-1) = depth(n)$ then $flip(n-1) = flip(n) + 1$.*

Proof.

1. If $n \leq 1$, then $flip(n) = n$ and the property is obvious. For $n > 1$, if we name k the depth of n, we have $flip(n) = 1 + F_{k+2} - n$. Since $1 + F_k \leq n \leq F_{k+1}$, we hence have $1 + F_{k+2} - F_{k+1} \leq flip(n) \leq 1 + F_{k+2} - 1 - F_k$. And finally $1 + F_k \leq flip(n) \leq F_{k+1}$, and this characterizes the nodes at depth k, hence $depth(flip(n)) = k = depth(n)$.
2. We know that n and $flip(n)$ have the same depth, and we've already seen that being less or equal to 1 is equivalent to having depth 0.
3. For $k > 1$ and $n < F_{k-1}$, we know that $depth(1 + F_k + n) = k$ (still via the same characterization of nodes at depth k). Moreover $1 + F_k + n \geq 1 + F_k \geq 2$, so $flip(1 + F_k + n) = 1 + F_{k+2} - (1 + F_k + n) = F_{k+1} - n$.
4. If $n \leq 1$ then $flip(n) = n$ is still less or equal to 1, so $flip(flip(n)) = flip(n) = n$. Consider now $n > 1$. We already know that $flip(n) > 1$ and $flip(n)$ and n have same depth (let us name it k). Hence: $flip(flip(n)) = 1 + F_{k+2} - flip(n) = 1 + F_{k+2} - (1 + F_{2+k} - n) = n$.

5. Let us name k the common depth of $n+1$ and n. In these conditions $flip(n+1) = 1 + F_{k+2} - (n+1) = (1 + F_{k+2} - n) - 1 = flip(n) - 1$.

6. Similar proof.

□

In particular, the third point above shows that for $k \neq 0$ we indeed have $flip(1 + F_k) = F_{k+1}$ (and vice-versa since $flip$ is involutive).

6.2 Definition of $\overline{G}$ and initial properties

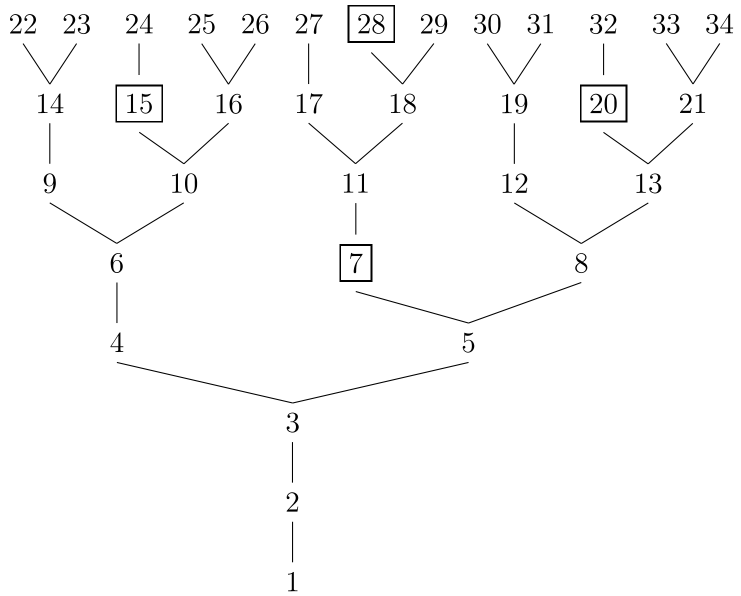
We can now take advantage of this $flip$ function to obtain a first definition of the $\overline{G}$ function, corresponding to the flipped G tree:

Definition 6. *The function* $\overline{G} : \mathbb{N} \to \mathbb{N}$ *is defined via:*

$$\overline{G}(n) = flip(G(flip(n))$$

We benefit from the involutive aspect of $flip$ to switch from right-left to left-right diagrams, use G, and then switch back to right-left diagram. The corresponding Coq function is named `fg`.

By following this definition, the initial values of $\overline{G}$ are $\overline{G}(0) = 0$, $\overline{G}(1) = \overline{G}(2) = 1$, $\overline{G}(3) = 2$. The first difference between $\overline{G}$ and G appears for $\overline{G}(7) = 5 = G(7)+1$. We'll dedicate a whole section later on the comparison between $\overline{G}$ and G. Here is the initial levels of this flipped tree $\overline{G}$, the boxed values being the places where $\overline{G}$ and G differ.



We now show that $\overline{G}$ enjoys the same basic properties as G:

Theorem 21.

1. *For all* $n > 1$, $depth(\overline{G}(n)) = depth(n) - 1$.
2. *For all* k, $\overline{G}(F_{k+1}) = F_k$.
3. *For all* $k \neq 0$, $\overline{G}(1 + F_{k+1}) = 1 + F_k$.
4. *For all* n, $\overline{G}(n+1) - \overline{G}(n) \in \{0, 1\}$.
5. *For all* n, m, $n \leq m$ *implies* $0 \leq \overline{G}(m) - \overline{G}(n) \leq m - n$.
6. *For all* n, $0 \leq \overline{G}(n) \leq n$.
7. *For all* n, $\overline{G}(n) = 0$ *if and only if* $n = 0$
8. *For all* $n > 1$, $\overline{G}(n) < n$.
9. *For all* $n \neq 0, \overline{G}(n) = \overline{G}(n-1)$ *implies* $\overline{G}(n+1) = \overline{G}(n) + 1$.
10. $\overline{G}$ *is onto.*

Proof.

1. Since $flip$ doesn't modify the depth, we reuse the result about the depth of $G(n)$.
2. $\overline{G}(F_{k+1}) = flip(G(flip(F_{k+1}))) = flip(G(1+F_k)) = flip(1+F_{k-1}) = F_k$.
3. Similar proof.
4. The property is true for $n = 0$ and $n = 1$. Consider now $n > 1$. Let k be $depth(n)$, which is hence different from 0. We have $1 + F_k \leq n \leq F_{k+1}$. If n is F_{k+1}, we have already seen that $\overline{G}(1 + F_{k+1}) = 1 + F_k = 1 + \overline{G}(F_{k+1})$. Otherwise $n < F_{k+1}$ and $n+1$ has hence the same depth as n. Then $flip(n+1) = flip(n) - 1$. Now, by applying G on the two consecutive values of same depth $flip(n)$ and $flip(n) - 1$, we obtain two results that are either equal or consecutive, and of same depth. By applying $flip$ again on these G results, we end on two consecutive or equal $\overline{G}$ results.
5. Iteration of the previous results between n and m.
6. Thanks to the previous point, $0 \leq \overline{G}(m) - \overline{G}(0) \leq m - 0$ and $\overline{G}(0) = 0$.
7. $\overline{G}(1) = 1$ and $0 \leq \overline{G}(m) - \overline{G}(1)$ as soon as $m \geq 1$.
8. $\overline{G}(2) = 1$ and $\overline{G}(m) - \overline{G}(2) \leq m - 2$ as soon as $m \geq 2$.

9. If $n = 1$, then the precondition $\overline{G}(1) = \overline{G}(0)$ isn't satisfied. If $n = 2$ or $n = 3$, the conclusion is satisfied: $\overline{G}(3) = 2 = \overline{G}(2) + 1$ and $\overline{G}(4) = 3 = \overline{G}(3) + 1$. We now consider $n > 3$ such that $\overline{G}(n) = \overline{G}(n-1)$. We have already seen that $\overline{G}(n+1) - \overline{G}(n)$ is either 0 or 1. If it is 1, we can conclude. We proceed by contradiction and suppose it is 0. We hence have $\overline{G}(n-1) = \overline{G}(n) = \overline{G}(n+1)$, or said otherwise $flip(G(flip(n-1))) = flip(G(flip(n))) = flip(G(flip(n+1)))$. Since $flip$ is involutive hence bijective, this implies that $G(flip(n-1)) = G(flip(n)) = G(flip(n+1))$. To be able to commute $flip$ and the successor/predecessor, we now study the depth of these values.

   - If $n+1$ has a different depth than n, then n is a Fibonacci number F_k, with $k > 3$ since $n > 3$. Then $flip(n+1) = F_{k+1}$ and $flip(n) = 1 + F_{k-1}$. These values aren't equal nor consecutive, hence G cannot give the same result for them, this situation is indeed contradictory.
   - Similarly, if $n-1$ has a different depth than n, then $n-1$ is a Fibonacci number F_k, with $k > 2$ since $n > 3$. Then $flip(n-1) = 1 + F_{k-1}$ while $flip(n) = F_{k+1}$. Once again, these values aren't equal nor consecutive, hence G cannot give the same result for them. Contradiction.
   - In the last remaining case, $depth(n-1) = depth(n) = depth(n+1)$. Then $flip(n+1) = flip(n) - 1$ and $flip(n-1) = flip(n) + 1$ and these value are at distance 2, while having the same result by G, which is contradictory.

10. We reason as we did earlier for G: $\overline{G}$ starts with $\overline{G}(0) = 0$, it grows by steps of 0 or 1, and it grows by at least 1 every two steps. Hence its limit is $+\infty$ and it is an onto function.

□

6.3 An recursive algebraic definition for $\overline{G}$

The following result is an answer to Hofstadter's problem. It was already mentioned on OEIS page [5], but only as a conjecture.

Theorem 22. *For all $n > 3$ we have $\overline{G}(n) = n + 1 - \overline{G}(1 + \overline{G}(n-1))$. And $\overline{G}$ is uniquely characterized by this equation plus initial equations $\overline{G}(0) = 0$, $\overline{G}(1) = \overline{G}(2) = 1$ and $\overline{G}(3) = 2$.*

Proof. We consider $n > 3$. Let k be its depth, which is hence at least 3. We know that $1 + F_k \leq n \leq F_{k+1}$. We will note $flip(n)$ below as $\overline{n}$. Roughly speaking, this proof is essentially a use of theorem 11 which implies that $G(G(\overline{n}+1) - 1) = \overline{n} - G(\overline{n})$, and then some play with flip and predecessors and successors, when possible, or direct particular proofs otherwise.

- We start with the most general case : we suppose here $depth(n-1) = depth(n)$ and $depth(\overline{G}(n-1)+1) = depth(\overline{G}(n-1)$. Let us shorten $\overline{G}(n-1)+1$ as p. In particular $depth(p) = depth(\overline{G}(n-1)) = depth(n-1) - 1 = depth(n) - 1 = k - 1$. Due to the equalities between depths, and the facts that $n > 1$ and $\overline{G}(n-1) > 1$, we're allowed to use the last properties of theorem 20 : $flip(n-1) = \overline{n} + 1$ and

$$\begin{aligned} flip(p) &= flip(\overline{G}(n-1)+1) \\ &= flip(\overline{G}(n-1)) - 1 \\ &= flip(flip(G(flip(n-1)))) - 1 \\ &= G(flip(n-1)) - 1 \\ &= G(\overline{n}+1) - 1 \end{aligned}$$

  We now exploit the definition of $flip$ three times:

  1. $depth(n) = k \neq 0$ implies $\overline{n} = 1 + F_{k+2} - n$
  2. $depth(G(flip(p))) = depth(p) - 1 = k - 2 \neq 0$ hence:

  $$\overline{G}(p) = flip(G(flip(p))) = 1 + F_k - G(flip(p))$$

  3. $depth(G(\overline{n})) = depth(\overline{n}) - 1 = depth(n) - 1 = k - 1 \neq 0$, so:

  $$\overline{G}(n) = flip(G(\overline{n})) = 1 + F_{k+1} - G(\overline{n})$$

  All in all:

$$\begin{aligned} n + 1 - \overline{G}(1 + \overline{G}(n-1)) &= n + 1 - \overline{G}(p) \\ &= n + 1 - (1 + F_k - G(flip(p))) \\ &= n - F_k + G(G(\overline{n}+1) - 1)) \\ &= n - F_k + (\overline{n} - G(\overline{n})) \\ &= n - F_k + (1 + F_{k+2} - n) - G(\overline{n}) \\ &= 1 + F_{k+1} - G(\overline{n}) \\ &= \overline{G}(n) \end{aligned}$$

- We now consider the case where $depth(n-1) \neq depth(n)$. So n is the least number of depth k, hence $n = 1 + F_k$. In this case:

$$\begin{aligned} 1 + n - \overline{G}(1 + \overline{G}(n-1)) &= 2 + F_k - \overline{G}(1 + \overline{G}(F_k)) \\ &= 2 + F_k - \overline{G}(1 + F_{k-1}) \\ &= 2 + F_k - (1 + F_{k-2}) \\ &= 1 + F_{k-1} \\ &= \overline{G}(n) \end{aligned}$$

- The last case to consider is $depth(n-1) = depth(n)$ but $depth(\overline{G}(n-1)+1) \neq depth(\overline{G}(n-1))$. As earlier, we prove that $depth(\overline{G}(n-1)) = depth(n-1) - 1 = depth(n) - 1 = k-1$. So $\overline{G}(n-1)$ is the greatest number of depth $k-1$, hence $\overline{G}(n-1) = F_k$. Since $\overline{G}(F_{k+1})$ is also F_k, then $n-1$ and F_{k+1} are at a distance of 0 or 1. But we know that $depth(n) = k$, hence $n \leq F_{k+1}$, so $n-1 < F_{k+1}$. Finally $n = F_{k+1}$ and:

$$\begin{aligned} 1 + n - \overline{G}(1 + \overline{G}(n-1)) &= 1 + F_{k+1} - \overline{G}(1 + F_k) \\ &= 1 + F_{k+1} - (1 + F_{k-1}) \\ &= F_k \\ &= \overline{G}(n) \end{aligned}$$

Finally, if we consider another function F satisfying the same recursive equation, as well as the same initial values for $n \leq 3$, then we prove by strong induction over n that $\forall n, F(n) = \overline{G}(n)$. This is clear for $n \leq 3$. Consider now some $n > 3$, and assume that $F(k) = \overline{G}(k)$ for all $k < n$. In particular $F(n-1) = \overline{G}(n-1)$ by induction hypothesis for $n-1 < n$. Moreover $n-1 > 1$ hence $\overline{G}(n-1) < n-1$ hence $\overline{G}(n-1)+1 < n$. So we could use a second induction hypothesis at this position: $F(\overline{G}(n-1)+1) = \overline{G}(\overline{G}(n-1)+1)$. This combined with the first induction hypothesis above gives $F(F(n-1)+1) = \overline{G}(\overline{G}(n-1)+1)$ and finally $F(n) = \overline{G}(n)$ thanks to the recursive equations for F and $\overline{G}$. □

6.4 The $\overline{G}$ tree

Since $\overline{G}$ has been obtained as the mirror of G, we already know its shape: it's the mirror of the shape of G. We'll nonetheless be slightly more precise here. The proofs given in this section will be deliberately sketchy, please consult theorem `unary_flip` and alii in file `FlipG.v` for more detailed and rigorous justifications.

First, a node n has a child p in the tree $\overline{G}$ is and only if the node $flip(n)$ has a child $flip(p)$ in the tree G. Indeed, $\overline{G}(p) = n$ iff $flip(G(flip(p))) = n$ iff $G(flip(p)) = flip(n)$. Moreover, a rightmost child in $\overline{G}$ corresponds by $flip$ with a leftmost child in G and vice-versa. Indeed, if p and n aren't on the borders of the trees, then the next node will become the previous by flip, and vice-versa (see theorem 20). And if p and/or n are on the border, they are Fibonacci numbers or successors of Fibonacci numbers, and we check these cases directly.

Similarly, the arity of n in $\overline{G}$ is equal to the arity of $flip(n)$ in G. For instance, if we take a unary node n and its unique child p in $\overline{G}$, this means that $\overline{G}(p+1) = n+1$ and $\overline{G}(p-1) = n-1$. In the most general case $flip$ will lead to similar properties about $flip(p)$ and $flip(n)$ in G, hence the fact that $flip(n)$ is unary in G. And we handle particular cases about Fibonacci numbers on the border as usual.

Theorem 23. *For all $n > 1$, $flip(flip(n) + G(flip(n)))$, which could also be written $flip(flip(n)+flip(\overline{G}(n)))$, is the leftmost child of n in the $\overline{G}$ tree.*

Proof. See the previous paragraph: $flip(n) + G(flip(n))$ is rightmost child of $flip(n)$ in G, hence the result about $\overline{G}$. ☐

Theorem 24. *For all $n > 1$, $n - 1 + \overline{G}(n+1)$ is the rightmost child of n in the $\overline{G}$ tree.*

Proof. We already know that all nodes n in the $\overline{G}$ tree have at least one antecedent, and no more than two. Take $n > 1$, and let k be the largest of its antecedents by $\overline{G}$. Hence $\overline{G}(k) = n$ and $\overline{G}(k+1) \neq n$, leading to $\overline{G}(k+1) = n+1$. If we re-inject this into the previous recursive equation $\overline{G}(k+1) = k+2-\overline{G}(\overline{G}(k)+1)$, we obtain that $n+1 = k+2-\overline{G}(n+1)$ hence $k = n-1+\overline{G}(n+1)$. ☐

Of course, for unary nodes, there is only one child, hence the leftmost and rightmost children given above coincide. Otherwise, for binary nodes, they are apart by 1.

Theorem 25. *In the $\overline{G}$ tree, a binary node has a right child which is also binary, and a left child which is unary, while the unique child of a unary node is itself binary.*

Proof. Flipped version of theorem 13. ☐

6.5 Comparison between $\overline{G}$ and G

We already noticed earlier that $\overline{G}$ and G produce very similar answers. Let us study this property closely now.

Theorem 26. *For all n, we have $\overline{G}(n) = 1 + G(n)$ whenever n is 2-even, and $\overline{G}(n) = G(n)$ otherwise.*

Proof. We proceed by strong induction over n. When $n \leq 3$, n is never 2-even, and we indeed have $\overline{G}(n) = G(n)$. We now consider $n > 3$, and assume that the result is true at all positions strictly less than n. For comparing $\overline{G}(n)$ and $G(n)$ we use their corresponding recursive equations: when n is 2-even we need to prove that $\overline{G}(\overline{G}(n-1)+1) = G(G(n-1))$, and otherwise we need to establish that $\overline{G}(\overline{G}(n-1)+1) = 1+G(G(n-1))$. So we'll need induction hypotheses (IH) for $n-1 < n$ and for $\overline{G}(n-1)+1$ (which is indeed strictly less than n: since $1 < n-1$, we have $\overline{G}(n-1) < n-1$). But the exact equations given by these two IH will depend on the status of $n-1$ and $\overline{G}(n-1)+1$: are these numbers 2-even or not ? For determining that, we'll consider the Fibonacci decomposition of n and study its classification.

- Case $low(n) = 1$. Hence n isn't 2-even and we try to prove $\overline{G}(\overline{G}(n-1)+1) = 1 + G(G(n-1))$. Theorem 5 implies that $low(n-1) > 2$, and in particular $n-1$ isn't 2-even. So the first IH is $\overline{G}(n-1) = G(n-1)$. By theorem 15, we also know that $G(n-1) = G(n) - 1$. So $\overline{G}(n-1)+1 = G(n-1)+1 = G(n)$ By theorem 17, we know that $G(n)$ has an odd lowest rank, so it cannot be 2-even, and the second IH is $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1)$. Since $low(G(n))$ is odd, we can use theorem 15 once more: $G(G(n)-1) = G(G(n)) - 1$. Finally: $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1) = G(G(n)) = 1 + G(G(n-1))$.

- Case n 2-even. We're trying to prove $\overline{G}(\overline{G}(n-1)+1) = G(G(n-1))$. Theorem 5 implies that $low(n-1) = 1$, and in particular $n-1$ isn't 2-even. So the first IH is $\overline{G}(n-1) = G(n-1)$. By theorem 15, we also know that $G(n-1) = G(n)$. Moreover theorem 18 shows that $\overline{G}(n-1)+1 = G(n)+1$ cannot be 2-even. So the second IH gives $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1)$. Now, $low(G(n)) = 1$ by theorem 17 so $G(G(n)+1) = G(G(n)$ by theorem 15, or equivalently $G(\overline{G}(n-1)+1) = G(G(n-1))$, hence the desired equation.

- Case n 2-odd. As in all the remaining cases, we're now trying here to prove $\overline{G}(\overline{G}(n-1)+1) = 1 + G(G(n-1))$. This case is very similar to the previous one until the point where we study $G(n)+1$ which is now 2-even (still thanks to theorem 18). So the second IH gives now $\overline{G}(\overline{G}(n-1)+1) = 1+G(\overline{G}(n-1)+1)$. And we conclude just as before by ensuring for the same reasons that $G(\overline{G}(n-1)+1) = G(G(n-1))$.

- Case $low(n) = 3$. Since 3 is odd, we also have $G(n-1) = G(n) - 1$. Hence $low(G(n)) = 2$ by theorem 16, and $G(G(n)-1) = G(G(n))$ and $G(G(n)+1) = 1+G(G(n))$, both by theorem 15. Finally $1+G(G(n-1)) = 1 + G(G(n)) = G(G(n)+1)$. Now, to determine whether $n-1$ is 2-even, we need to look deeper in the canonical decomposition of $n = F_3 + F_k + \Sigma F_i$.

  - If the second lowest rank k is even, then $n - 1 = F_2 + F_k + \Sigma F_i$ is hence 2-even, and the first IH is $\overline{G}(n-1) = 1 + G(n-1)$. So $\overline{G}(n-1)+1 = G(n)+1$, and this number has an odd lowest rank (theorem 17), it cannot be 2-even, and the second IH is: $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1)$. And this is known to be equal to $G(G(n)+1) = 1 + G(G(n-1))$.
  - Otherwise k is odd and $n-1$ is 2-odd and hence not 2-even, so the first IH is $\overline{G}(n-1) = G(n-1)$. Moreover, since $n-1$ is 2-odd then $\overline{G}(n-1)+1 = G(n-1)+1$ is 2-even by theorem 18. So the second IH is: $\overline{G}(\overline{G}(n-1)+1) = 1+G(\overline{G}(n-1)+1)$. And this is known to be equal to $1+G(G(n-1)+1) = 1+G(G(n)) = 1+G(G(n-1))$.

- Case $low(n) > 3$ and even. By theorem 5, $low(n-1) = 1$ hence the first IH is $\overline{G}(n-1) = G(n-1)$. We also know that $G(n-1) = G(n)$ by theorem 15, and that $low(\overline{G}(n-1)+1) = low(G(n)+1)$ is odd by theorem 17. The second IH is hence: $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1)$. Finally $G(G(n)+1) = 1 + G(G(n))$ since $low(G(n)) = 2 \neq 1$.

- Case $low(n) > 3$ and odd. In this case, $n-1$ is 2-even (theorem 6), so the first IH is $\overline{G}(n-1) = 1 + G(n-1)$. We also know that $G(n-1) = G(n) - 1$ by theorem 15. By theorem 16 we know that $low(G(n)) = low(n) - 1 > 2$, so by theorem 4 $low(G(n)+1) = 1$, and $\overline{G}(n-1) + 1 = 1 + G(n)$ cannot be 2-even : the second IH is hence $\overline{G}(\overline{G}(n-1)+1) = G(\overline{G}(n-1)+1)$. Moreover $low(G(n))$ is also known to be even, so by theorem 15 we have $G(G(n-1)) = G(G(n)-1) = G(G(n))$. The final step is $G(1+G(n)) = 1+G(G(n))$ also by theorem 15 (since $low(G(n)) \neq 1$).

□

As an immediate consequence, $\overline{G}$ is always greater or equal than G, but never more than $G+1$. And we've already studied the distance between 2-even numbers, which is always 5 or 8, while the first 2-even number is 7. So $\overline{G}$ and G are actually equal more than 80% of the time.

6.6 $\overline{G}$ and its "derivative" $\Delta\overline{G}$

We consider now the "derivative" $\Delta\overline{G}$ of $\overline{G}$, defined via $\Delta\overline{G}(n) = \overline{G}(n+1) - \overline{G}(n)$. This study will be quite similar to the corresponding section 5.4 for G. We already know from theorem 21 that the output of $\Delta\overline{G}$ is always either 0 or 1.

Theorem 27. *For all $n > 2$, $\Delta\overline{G}(n+1) = 1 - \Delta\overline{G}(n).\Delta\overline{G}(\overline{G}(n+1))$.*

Proof. We already know that $\Delta\overline{G}(n) = 0$ implies $\Delta\overline{G}(n+1) = 1$: we cannot have $\overline{G}(n) = \overline{G}(n+1) = \overline{G}(n+2)$. Consider now some $n > 2$ such that $\overline{G}(n+1) - \overline{G}(n) = 1$. By using the recursive equation of $\overline{G}$ for $n+1 > 3$ and $n+2 > 3$, we have as expected

$$\begin{aligned}
\Delta\overline{G}(n+1) &= \overline{G}(n+2) - \overline{G}(n+1) \\
&= (n+3-\overline{G}(\overline{G}(n+1)+1)) - (n+2-\overline{G}(\overline{G}(n)+1)) \\
&= 1 - (\overline{G}(\overline{G}(n+1)+1) - \overline{G}(\overline{G}(n)+1) \\
&= 1 - (\overline{G}(\overline{G}(n+1)+1) - \overline{G}(\overline{G}(n+1)) \\
&= 1 - \Delta\overline{G}(\overline{G}(n+1))
\end{aligned}$$

□

Note: when compared with theorem 19 about ΔG, the equation above looks really similar, but has a different inner call ($\Delta\overline{G}(\overline{G}(n+1))$ instead of $\Delta G(G(n))$), and doesn't hold for $n = 2$.

For $n > 2$, this equation provides a way to express $\overline{G}(n+2)$ in terms of $\overline{G}(n+1)$ and $\overline{G}(n)$ and $\overline{G}(\overline{G}(n+1))$ and $\overline{G}(\overline{G}(n+1)+1)$. For $n > 1$, we know that $G(n+1) < n+1$ hence $\overline{G}(n+1)+1 < n+2$. It is also clear that $n+1$ and n and $\overline{G}(n+1)$ are all strictly less than $n+2$. So we could use this equation as an alternative way to define recursively $\overline{G}$, alongside initial equations $\overline{G}(0) = 0$ and $\overline{G}(1) = \overline{G}(2) = 1$ and $\overline{G}(3) = 2$ and $\overline{G}(4) = 3$. We proved in Coq that $\overline{G}$ is indeed the unique function to satisfy these equations (see `FD_unique` and `fg_implements_FD`).

6.7 An alternative recursive equation for $\overline{G}$

During the search for an algebraic definition of $\overline{G}$, we first discovered and proved correct the following result, which unfortunately doesn't uniquely characterize $\overline{G}$. We mention it here nonetheless, for completeness sake.

Theorem 28. *For all $n > 3$ we have $\overline{G}(n-1) + \overline{G}(\overline{G}(n)) = n$.*

Proof. We proceed as for theorem 22, except that we use internally the original recursive equation for G instead of the alternative equation of theorem 11. We take $n > 3$, call k its depth (which is greater or equal to 3), and note $\overline{n} = flip(n)$. First:

$$\overline{G}(\overline{G}(n)) = flip(G(flip(flip(G(\overline{n}))))) = flip(G(G(\overline{n})))$$

The depth of $G(G(\overline{n}))$ is $k-2 \neq 0$, so the definition of $flip$ gives:

$$\overline{G}(\overline{G}(n)) = flip(G(G(\overline{n}))) = 1 + F_k - G(G(\overline{n}))$$

Now, thanks to the recursive definition of G at $\overline{n}+1$, we have

$$\overline{G}(\overline{G}(n)) = 1 + F_k + G(\overline{n}+1) - \overline{n} - 1$$

- If $depth(n-1) = k$ then $flip(n-1) = \overline{n}+1$ and we conclude by more uses of the $flip$ definition: $depth(G(\overline{n}+1)) = depth(G(flip(n-1)) = depth(n-1) - 1 = k-1$, so:

  $$\overline{G}(n-1) = flip(G(flip(n-1))) = flip(G(\overline{n}+1)) = 1+F_{k+1}-G(\overline{n}+1)$$

  And finally:

  $$\overline{G}(\overline{G}(n)) = F_k + (1+F_{k+1} - \overline{G}(n-1)) - (1+F_{k+2} - n) = n - \overline{G}(n-1)$$

- If $depth(n-1) \neq k$ then n is the least number at depth k, so $n = 1+F_k$, and:

  $$\overline{G}(n-1) + \overline{G}(\overline{G}(n)) = \overline{G}(F_k) + \overline{G}(\overline{G}(1+F_k)) = F_{k-1} + (1+F_{k-2}) = n$$

□

Note that the following function $f : \mathbb{N} \to \mathbb{N}$ also satisfies this equation, even with the same initial values than $\overline{G}$:

$$\begin{aligned} f(0) &= 0 \\ f(1) &= f(2) = 1 \\ f(3) &= 2 \\ f(4) &= f(5) = 3 \\ f(6) &= 5 \\ f(7) &= 3 \\ \forall n \leq 4,\ f(2n) &= n - 2 \\ \forall n \leq 4,\ f(2n+1) &= 4 \end{aligned}$$

This function isn't monotone. We actually proved in Coq that $\overline{G}$ is the only monotone function that satisfies the previous equation and the initial constraints $0 \mapsto 0,\ 1 \mapsto 1,\ 2 \mapsto 1,\ 3 \mapsto 2$ (see `alt_mono_unique` and `alt_mono_is_fg`). We will not detail these proofs here, the key ingredient is to prove that any monotone function satisfying these equations will grow by at most 1 at a time.

7 Conclusion

The proofs for theorems 22 and 26 are surprisingly tricky, all our attempts at simplifying them have been rather unsuccessful. But there's probably still room for improvements here, please let us know if you find or encounter nicer proofs.

Perhaps using the definition of G via real numbers could help shortening some proofs. We proved in file `Phi.v` this definition $\forall n, G(n) = \lfloor n/\varphi \rfloor$ where φ is the golden ratio $(1+\sqrt{5})/2$. But this has been done quite late in our development, and we haven't tried to use this fact for earlier proofs. Anyway, relating this definition with the flipped function $\overline{G}$ doesn't seem obvious.

Another approach might be to relate $\overline{G}$ more directly to some kind of Fibonacci decomposition. Of course, now that theorem 26 is proved, we know that $\overline{G}$ shifts the ranks of the Fibonacci decompositions just as G, except for 2-even numbers where a small +1 correction is needed. But could this fact be established more directly ? For the moment, we are only aware of the following "easy" formulation of $\overline{G}$ via decompositions: if n is written as $F_{k+1} - \Sigma F_i$ where $k = depth(n)$ and ΣF_i form a canonical decomposition of $F_{k+1} - n$, then $\overline{G}(n) = F_k - \Sigma F_{i-1} + \epsilon$, where $\epsilon = 1$ whenever the decomposition above includes F_1, and $\epsilon = 0$ otherwise. But this statement didn't brought any new insights for the proof of theorem 26, so we haven't formulated it in Coq.

As possible extensions of this work, we might consider later the other recursive functions proposed by Hofstadter :

- H defined via $H(n) = n - H(H(H(n-1)))$, or its generalized version with an arbitrary number of sub-calls instead of 2 for G and 3 for H.
- $\overline{H}$, the flipped version of H.
- M and F, the mutually recursive functions ("Male" and "Female").

We've already done on paper a large part of the analysis of M and F, and should simply take the time to certify it in Coq. The study of H and $\overline{H}$ remains to be done, it looks like a direct generalization of what we've done here, but surprises are always possible.

References

[1] Hofstadter, Douglas R., *Godel, Escher, Bach: An Eternal Golden Braid*, 1979, Basic Books, Inc, NY.

[2] *The Coq Proof Assistant Reference Manual*, 2015, `http://coq.inria.fr`.

[3] *The On-Line Encyclopedia of Integer Sequences*, Sequence A000045: Fibonacci numbers, `http://oeis.org/A000045`

[4] *The On-Line Encyclopedia of Integer Sequences*, Sequence A005206: Hofstadter G-sequence, `http://oeis.org/A005206`

[5] *The On-Line Encyclopedia of Integer Sequences*, Sequence A123070: Hofstadter Flip-G-sequence, `http://oeis.org/A123070`