Generalizing some Hofstadter functions

G, H and beyond

Pierre Letouzey, IRIF, U. Paris Cité

Joint work with Shuo Li (U. Winnipeg) & Wolfgang Steiner (IRIF)

One World Combinatorics on Words Seminar

March 11, 2025

Online ressources

Main page: `https://github.com/letouzey/hofstadter_g`

Includes these slides, the Coq files and links to two preprints:

Preprint about part 1: `https://hal.science/hal-04715451`
Preprint about part 2: `https://hal.science/hal-04948022`
(also on arXiv: 2410.00529 and 2502.12615)

Some nested recursions

From the book "Gödel,Escher,Bach":

Definition (Hofstadter's G function)

$$\begin{cases} G(0) = 0 \\ G(n) = n - G(G(n-1)) & \text{for all } n \in \mathbb{N}_*. \end{cases}$$

Some nested recursions

From the book "Gödel,Escher,Bach":

Definition (Hofstadter's G function)

$$\begin{cases} G(0) = 0 \\ G(n) = n - G(G(n-1)) \end{cases} \qquad \text{for all } n \in \mathbb{N}_*.$$

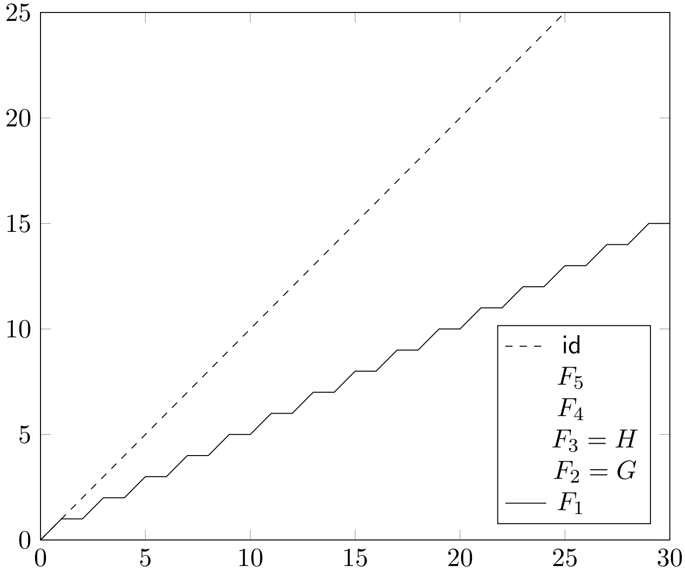
For $k \in \mathbb{N}$, we generalize to k nested recursive calls:

Definition (the F_k functions)

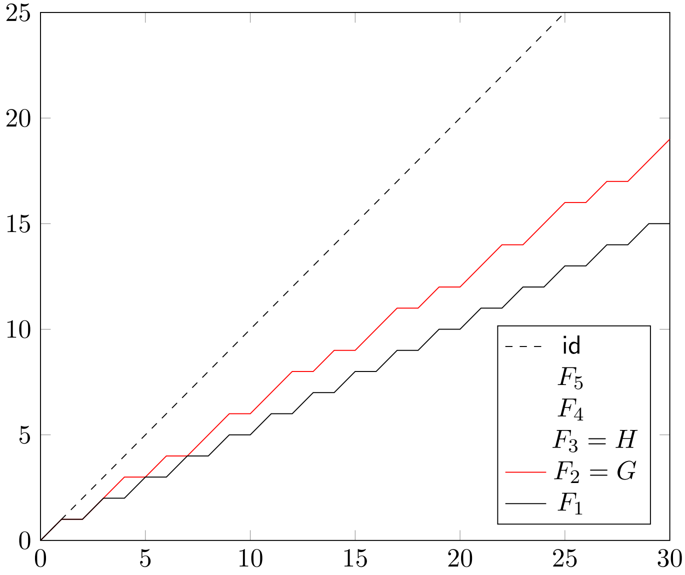
$$\begin{cases} F_k(0) = 0 \\ F_k(n) = n - F_k^{(k)}(n-1) \end{cases} \qquad \text{for all } n \in \mathbb{N}_*$$

where $F_k^{(k)}$ is the k-th iterate $F_k \circ F_k \circ \cdots \circ F_k$.

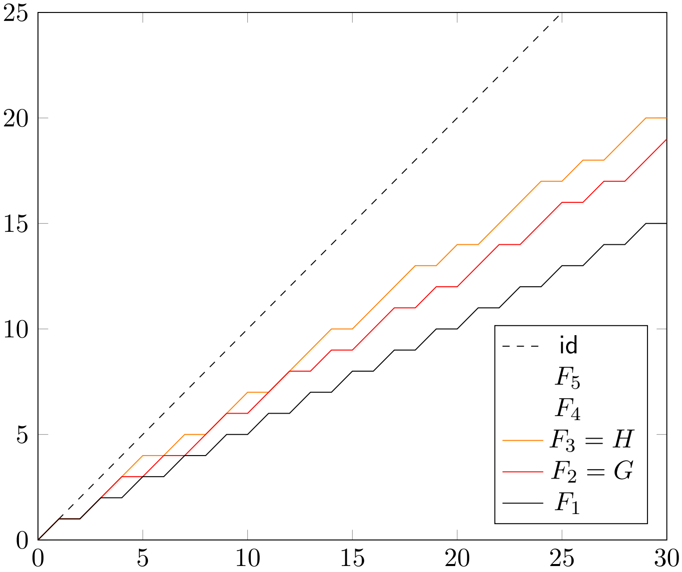
Plotting the early F_k



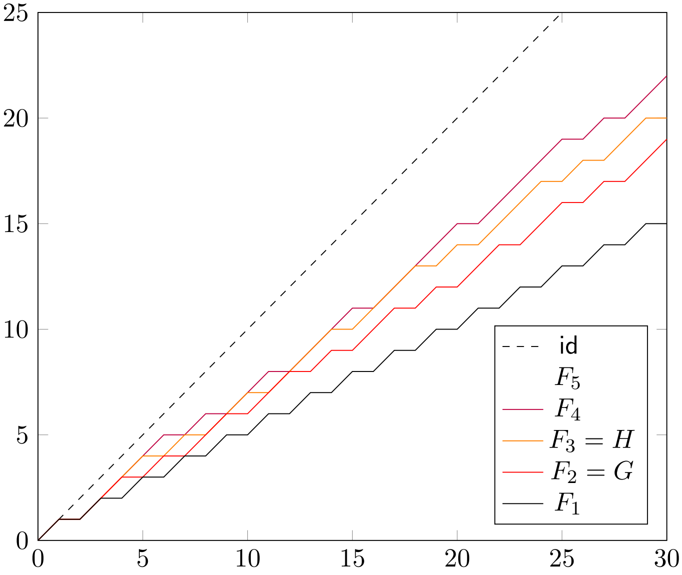
Plotting the early F_k



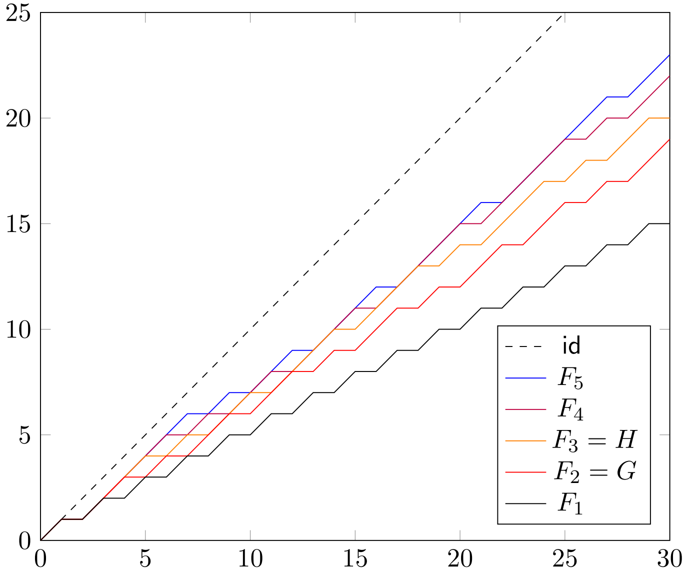
Plotting the early F_k



Plotting the early F_k



Plotting the early F_k



Outline

1. Morphic words and pointwise monotonicity
2. Numerical systems and discrepancy
3. The Coq formalization

Part 1

Morphic words and pointwise monotonicity

What about F_0 and F_1 and F_2 ?

- F_0 won't be considered : non-recursive, flat, boring.
  - For the rest of this talk, we assume $k \geq 1$.

What about F_0 and F_1 and F_2 ?

- F_0 won't be considered : non-recursive, flat, boring.
  - For the rest of this talk, we assume $k \geq 1$.
- F_1 is simply a division by 2 (rounded) :
  - $F_1(n) = n - F_1(n-1) = 1 + F_1(n-2)$ when $n \geq 2$
  - Hence $F_1(n) = \lfloor (n+1)/2 \rfloor = \lceil n/2 \rceil$.

What about F_0 and F_1 and F_2 ?

- F_0 won't be considered : non-recursive, flat, boring.
  - For the rest of this talk, we assume $k \geq 1$.
- F_1 is simply a division by 2 (rounded) :
  - $F_1(n) = n - F_1(n-1) = 1 + F_1(n-2)$ when $n \geq 2$
  - Hence $F_1(n) = \lfloor (n+1)/2 \rfloor = \lceil n/2 \rceil$.
- $F_2 = G$ is already well studied, see OEIS A5206.
  In particular $F_2(n) = \lfloor (n+1)/\varphi \rfloor$ where φ is the Golden Ratio.

Basic properties

$$\begin{cases} F_k(0) = 0 & \\ F_k(n) = n - F_k^{(k)}(n-1) & \text{for all } n > 0 \end{cases}$$

- Well-defined since $0 \leq F_k(n) \leq n$
- $F_k(0) = 0$, $F_k(1) = 1$ then $n/2 \leq F_k(n) < n$
- $F_k(n{+}1) - F_k = 0$ or 1 : a succession of flats and steps
- Hence each F_k is increasing, onto, but not one-to-one
- Never two flats in a row
- At most k steps in a row

Basic properties

$$\begin{cases} F_k(0) = 0 & \\ F_k(n) = n - F_k^{(k)}(n-1) & \text{for all } n > 0 \end{cases}$$

- Well-defined since $0 \leq F_k(n) \leq n$
- $F_k(0) = 0$, $F_k(1) = 1$ then $n/2 \leq F_k(n) < n$
- $F_k(n+1) - F_k = 0$ or 1 : a succession of flats and steps
- Hence each F_k is increasing, onto, but not one-to-one
- Never two flats in a row
- At most k steps in a row

F_k may be coded as an infinite word of flats and steps, e.g.

$$F_3 = \oslash\ominus\oslash\oslash\oslash\ominus\oslash\ominus\oslash\oslash\ominus\oslash\oslash\oslash\ominus\cdots$$

Too coarse, no nice properties for $k > 2$.

A letter substitution and its morphic word

We use $\mathcal{A} = \{1..k\}$ as alphabet.

Definition (substitution τ_k and morphic word x_k)

$$\begin{aligned} \mathcal{A} &\to \mathcal{A}^* \\ \tau_k : \; k &\mapsto k1, \\ i &\mapsto i{+}1 \quad \text{for } 1 \leq i < k. \end{aligned}$$

From letter k, τ_k leads to an infinite morphic word x_k, fixed-point of τ_k.

A letter substitution and its morphic word

We use $\mathcal{A} = \{1..k\}$ as alphabet.

Definition (substitution τ_k and morphic word x_k)

$$\begin{aligned} & \mathcal{A} \to \mathcal{A}^* \\ \tau_k : \ & k \mapsto k1, \\ & i \mapsto i{+}1 \quad \text{for } 1 \le i < k. \end{aligned}$$

From letter k, τ_k leads to an infinite morphic word x_k, fixed-point of τ_k.

For instance:

- $x_2 = 2122121221221212212 \cdots$ (Fibonacci word)
- $x_3 = 3123313123123312331 \cdots$

A letter substitution and its morphic word

We use $\mathcal{A} = \{1..k\}$ as alphabet.

Definition (substitution τ_k and morphic word x_k)

$$\begin{aligned} & \mathcal{A} \to \mathcal{A}^* \\ \tau_k :\ & k \mapsto k1, \\ & i \mapsto i{+}1 \quad \text{for } 1 \le i < k. \end{aligned}$$

From letter k, τ_k leads to an infinite morphic word x_k, fixed-point of τ_k.

For instance:

- $x_2 = 21221212212212212\cdots$ (Fibonacci word)
- $x_3 = 31233131231233123312331\cdots$

Spoiler: the previous word $F_k = \oslash\ominus\oslash\cdots$ is actually a projection of x_k where letter 1 becomes $\ominus$ and any other letter becomes $\oslash$.

Prior work

Note that τ_k is not novel, it can be seen as:

- A particular *modified Jacobi-Perron substitution* (see Pytheas Fogg)
- The substitution associated with the *Rényi expansion* of 1 in base $\beta_k = \mathrm{root}(X^k - X^{k-1} - 1)$ (see Frougny et al, 2004)
  - Hence the factor complexity of x_k is $n \mapsto (k-1)n+1$.

Length of substituted prefix

A useful notion relating F_k and x_k:

Definition (length L_k of substituted prefix)

$$L_k(n) := \left|\tau_k(x_k[0..n{-}1])\right|$$

where $x_k[0..n{-}1]$ is the prefix of size n of x_k.

Interestingly, the j-th iterate of L_k satisfies $L_k^j(n) = \left|\tau_k^j(x_k[0..n-1])\right|$.

Theorem

For $k, n, j > 0$, the antecedents of n by F_k^j are $L_k^j(n{-}1){+}1, \ldots, L_k^j(n)$.

The proof is pretty technical (thanks Wolfgang).
Corollary: $F_k(L_k(n)) = n \leq L_k(F_k(n)) \in \{n, n{+}1\}$ (Galois connection).
Prop: $L_k(n) = n + F_k^{k-1}(n)$.

More relations between x_k and F_k

Consequences of the previous theorem:

- The letter $x_k[n]$ is 1 whenever $F_k(n{+}1) - F_k(n)$ is 0.
- Counting letter 1 in $x_k[0..n{-}1]$ gives $n - F_k(n)$.
- For $1 \leq p \leq k$, counting letters p and more gives F_k^{p-1}.
  In particular the count of letter k is F_k^{k-1}.
- Another point of view: $x_k[n] = \min(j, k)$ where j is the least value such that F_k^j is flat at n.

Letter frequencies

Let α_k be the positive root of $X^k + X - 1$
and $\beta_k = 1/\alpha_k$, positive root of $X^k - X^{k-1} - 1$.

Theorem

$$\begin{aligned} \lim_{n\to\infty} \tfrac{1}{n} F_k(n) &= \alpha_k, \\ \lim_{n\to\infty} \tfrac{1}{n} L_k(n) &= \beta_k, \\ \text{frequency}(x_k, i) &= \alpha_k^{k+i-1} && \textit{for } 1 \le i < k, \\ \text{frequency}(x_k, k) &= \alpha_k^{k-1} = \beta_k - 1. \end{aligned}$$

Said otherwise, $F_k(n) = \alpha_k\, n + o(n)$ when $n \to \infty$.
See Dilcher 1993 for a proof without morphic words.

Corollary

When n is large enough, $F_k(n) < F_{k+1}(n)$.

Monotony of the F_k family

Definition

Pointwise order for functions : $f \leq h \iff \forall n \geq 0, f(n) \leq h(n)$

Theorem

For all k*,* $F_k \leq F_{k+1}$

Monotony of the F_k family

Definition

Pointwise order for functions : $f \leq h \iff \forall n \geq 0, f(n) \leq h(n)$

Theorem

For all k, $F_k \leq F_{k+1}$

- Conjectured in 2018.
- First proof by Shuo Li (Nov 2023).
- Improved version by Wolfgang Steiner.
- The key lemma proves simultaneously $L_k(n) \geq L_{k+1}(n)$ and $L_k^j(n) < L_{k+1}^{j+1}(n)$ for $k, n \geq 1$ and $j \leq k$.

Some key steps in the key lemma

When proving $L_k^j(n) < L_{k+1}^{j+1}(n)$ by induction on n, simultaneously with $L_k(n) \geq L_{k+1}(n)$:

- We deal with $j = k$ via an ad-hoc equation:

$$L_{k+1}^{k+1}(n) - L_k^k(n) = L_{k+1}^k(n) - L_k^{k-1}(n).$$

- When $j < k$, consider the last letter on the right $x_{k+1}[n{-}1]$:
  - If it is $k{+}1$, even a k letter on the left leads to a smaller quantity:

$$|\tau_k^j(k)| < |\tau_{k+1}^{j+1}(k{+}1)|.$$

  - If it is not $k{+}1$, the whole prefix $x_{k+1}[0..n{-}1]$ is the image by τ_{k+1} of a smaller prefix. Let m be its size. Induction hypothesis on this m: $L_k^{j+1}(m) < L_{k+1}^{j+2}(m)$ and $L_k(m) \geq L_{k+1}(m) = n$. And finally:

$$L_k^j(n) \leq L_k^j(L_k(m)) = L_k^{j+1}(m) < L_{k+1}^{j+2}(m) = L_{k+1}^{j+1}(n).$$

Pointwise monotonicity, summarized

Below, $f \to g$ whenever $f \leq g$ pointwise:

$$
\begin{tikzcd}
 & k=1 & k=2 & k=3 & k=4 & k=5 \\
j=1 & F_1 = \lceil \frac{n}{2} \rceil \arrow[r] & F_2 = G \arrow[r] & F_3 \arrow[r] & F_4 \arrow[r] & F_5 \\
j=2 & F_1^2 = \lceil \frac{n}{4} \rceil \arrow[r] \arrow[u] & F_2^2 \arrow[r] \arrow[u] \arrow[ul] & F_3^2 \arrow[r] \arrow[u] \arrow[ul] & F_4^2 \arrow[r] \arrow[u] \arrow[ul] & F_5^2 \arrow[u] \arrow[ul] \\
j=3 & F_1^3 = \lceil \frac{n}{8} \rceil \arrow[r] \arrow[u] & F_2^3 \arrow[r] \arrow[u] & F_3^3 \arrow[r] \arrow[u] \arrow[ul] & F_4^3 \arrow[r] \arrow[u] \arrow[ul] & F_5^3 \arrow[u] \arrow[ul] \\
j=4 & F_1^4 = \lceil \frac{n}{16} \rceil \arrow[r] \arrow[u] & F_2^4 \arrow[r] \arrow[u] & F_3^4 \arrow[r] \arrow[u] & F_4^4 \arrow[r] \arrow[u] \arrow[ul] & F_5^4 \arrow[u] \arrow[ul] \\
j=5 & F_1^5 = \lceil \frac{n}{32} \rceil \arrow[r] \arrow[u] & F_2^5 \arrow[r] \arrow[u] & F_3^5 \arrow[r] \arrow[u] & F_4^5 \arrow[r] \arrow[u] & F_5^5 \arrow[u] \arrow[ul]
\end{tikzcd}
$$

Pointwise monotonicity, summarized

Below, $f \to g$ whenever $f \leq g$ pointwise:

$$
\begin{tikzcd}
 & k=1 & k=2 & k=3 & k=4 & k=5 \\
j=1 & \color{red}{F_1 = \lceil \frac{n}{2} \rceil} \arrow[r, red] & \color{red}{F_2 = G} \arrow[r, red] & \color{red}{F_3} \arrow[r, red] & \color{red}{F_4} \arrow[r, red] & \color{red}{F_5} \\
j=2 & F_1^2 = \lceil \frac{n}{4} \rceil \arrow[u] \arrow[r] & \color{red}{F_2^2} \arrow[u] \arrow[ul, red] \arrow[r] & F_3^2 \arrow[u] \arrow[ul] \arrow[r] & F_4^2 \arrow[u] \arrow[ul] \arrow[r] & F_5^2 \arrow[u] \arrow[ul] \\
j=3 & F_1^3 = \lceil \frac{n}{8} \rceil \arrow[u] \arrow[r] & F_2^3 \arrow[u] \arrow[ul, dotted, "?"] \arrow[r] & \color{red}{F_3^3} \arrow[u] \arrow[ul, red] \arrow[r] & F_4^3 \arrow[u] \arrow[ul] \arrow[r] & F_5^3 \arrow[u] \arrow[ul] \\
j=4 & F_1^4 = \lceil \frac{n}{16} \rceil \arrow[u] \arrow[r] & F_2^4 \arrow[u] \arrow[r] & F_3^4 \arrow[u] \arrow[ul, dotted, "?"] \arrow[r] & \color{red}{F_4^4} \arrow[u] \arrow[ul, red] \arrow[r] & F_5^4 \arrow[u] \arrow[ul] \\
j=5 & F_1^5 = \lceil \frac{n}{32} \rceil \arrow[u] \arrow[r] & F_2^5 \arrow[u] \arrow[r] & F_3^5 \arrow[u] \arrow[r] & F_4^5 \arrow[u] \arrow[ul, dotted, "?"] \arrow[r] & \color{red}{F_5^5} \arrow[u] \arrow[ul, red]
\end{tikzcd}
$$

Pointwise monotonicity, summarized

Below, $f \to g$ whenever $f \le g$ pointwise:

$$
\begin{tikzcd}
 & k=1 & k=2 & k=3 & k=4 & k=5 \\
j=1 & \underset{0.5}{F_1 = \lceil \frac{n}{2} \rceil} \arrow[r] & \underset{0.618}{F_2 = G} \arrow[r] & \underset{0.682}{F_3} \arrow[r] & \underset{0.724}{F_4} \arrow[r] & \underset{0.754}{F_5} \\
j=2 & \underset{0.25}{F_1^2 = \lceil \frac{n}{4} \rceil} \arrow[r] \arrow[u] & \underset{0.381}{F_2^2} \arrow[r] \arrow[u] \arrow[ul, red] & \underset{0.465}{F_3^2} \arrow[r] \arrow[u] \arrow[ul] & \underset{0.524}{F_4^2} \arrow[r] \arrow[u] \arrow[ul] & \underset{0.569}{F_5^2} \arrow[u] \arrow[ul] \\
j=3 & \underset{0.125}{F_1^3 = \lceil \frac{n}{8} \rceil} \arrow[r] \arrow[u] & \underset{0.236}{F_2^3} \arrow[r] \arrow[u] \arrow[ul, dotted, "?"] & \underset{0.317}{F_3^3} \arrow[r] \arrow[u] \arrow[ul, red] & \underset{0.380}{F_4^3} \arrow[r] \arrow[u] \arrow[ul] & \underset{0.430}{F_5^3} \arrow[u] \arrow[ul] \\
j=4 & \underset{0.062}{F_1^4 = \lceil \frac{n}{16} \rceil} \arrow[r] \arrow[u] & \underset{0.145}{F_2^4} \arrow[r] \arrow[u] & \underset{0.216}{F_3^4} \arrow[r] \arrow[u] \arrow[ul, dotted, "?"] & \underset{0.275}{F_4^4} \arrow[r] \arrow[u] \arrow[ul, red] & \underset{0.324}{F_5^4} \arrow[u] \arrow[ul] \\
j=5 & \underset{0.031}{F_1^5 = \lceil \frac{n}{32} \rceil} \arrow[r] \arrow[u] & \underset{0.090}{F_2^5} \arrow[r] \arrow[u] & \underset{0.147}{F_3^5} \arrow[r] \arrow[u] & \underset{0.199}{F_4^5} \arrow[r] \arrow[u] \arrow[ul, dotted, "?"] & \underset{0.245}{F_5^5} \arrow[u] \arrow[ul, red]
\end{tikzcd}
$$

Remaining conjectures

The last n with $F_k(n) = F_{k+1}(n)$ seems $N_k := (k+1)(k+6)/2$.

- We proved $F_k(N_k) = F_{k+1}(N_k)$
- We proved $F_k(n) < F_{k+1}(n)$ for all $n > N_k$ and $k \leq 5$.
- We conjecture $F_k(n) < F_{k+1}(n)$ for all $n > N_k$ and any k.

Remaining conjectures

The last n with $F_k(n) = F_{k+1}(n)$ seems $N_k := (k+1)(k+6)/2$.

- We proved $F_k(N_k) = F_{k+1}(N_k)$
- We proved $F_k(n) < F_{k+1}(n)$ for all $n > N_k$ and $k \leq 5$.
- We conjecture $F_k(n) < F_{k+1}(n)$ for all $n > N_k$ and any k.

N_k also appears to be the last contact between L_{k+1} and L_{k+2}.

Part 2

Numeration systems and discrepancy

A Fibonacci-like family of sequences

Quiz ! $S \subset \mathbb{N}$ is said k-*sparse* if two elements of S are always separated by at least k. How many k-sparse subsets of $\{1..n\}$ could you form ?

A Fibonacci-like family of sequences

Quiz ! $S \subset \mathbb{N}$ is said k-*sparse* if two elements of S are always separated by at least k. How many k-sparse subsets of $\{1..n\}$ could you form ?

$$\begin{cases} A_{k,n} &= n+1 & \text{when } n \leq k \\ A_{k,n} &= A_{k,n-1} + A_{k,n-k} & \text{when } n \geq k \end{cases}$$

A Fibonacci-like family of sequences

Quiz ! $S \subset \mathbb{N}$ is said k-*sparse* if two elements of S are always separated by at least k. How many k-sparse subsets of $\{1..n\}$ could you form ?

$$\begin{cases} A_{k,n} = n+1 & \text{when } n \leq k \\ A_{k,n} = A_{k,n-1} + A_{k,n-k} & \text{when } n \geq k \end{cases}$$

- $A_{1,n}$: 1 2 4 8 16 32 64 128 256 512 ... (Powers of 2)
- $A_{2,n}$: 1 2 3 5 8 13 21 34 55 89 ... (Fibonacci)
- $A_{3,n}$: 1 2 3 4 6 9 13 19 28 41 ... (Narayana's Cows)
- $A_{4,n}$: 1 2 3 4 5 7 10 14 19 26 ...

A Fibonacci-like family of sequences

Quiz ! $S \subset \mathbb{N}$ is said k-*sparse* if two elements of S are always separated by at least k. How many k-sparse subsets of $\{1..n\}$ could you form ?

$$\begin{cases} A_{k,n} = n+1 & \text{when } n \leq k \\ A_{k,n} = A_{k,n-1} + A_{k,n-k} & \text{when } n \geq k \end{cases}$$

- $A_{1,n}$: 1 2 4 8 16 32 64 128 256 512 ... (Powers of 2)
- $A_{2,n}$: 1 2 3 5 8 13 21 34 55 89 ... (Fibonacci)
- $A_{3,n}$: 1 2 3 4 6 9 13 19 28 41 ... (Narayana's Cows)
- $A_{4,n}$: 1 2 3 4 5 7 10 14 19 26 ...

Actually: $A_{k,n} = L_k^n(1) = |\tau_k^n(k)|$

Not new: Meek & Van Rees (84), Dilcher (93), Kimberling (95), Eriksen & Anderson (2012), ...

F_k is a bitwise right shift

Theorem (Zeckendorf)

Fix a $k > 0$. All natural number can be written as a sum of $A_{k,i}$ numbers. This decomposition is unique when its indices i form a k-sparse set.

Theorem

F_k is a right shift for such a decomposition : $F_k(\Sigma A_{k,i}) = \Sigma A_{k,i-1}$ (with the convention $A_{k,0-1} = A_{k,0} = 1$)

- Beware, this shifted decomposition might not be k-sparse anymore
- Not so new: a variant of F_k is already known to be a right shift on these decompositions (Meek & van Rees, 1984).
- Key property: F_k is flat at n iff the decomposition of n uses $A_{k,0} = 1$.
- More generally, F_k^j is flat at n iff $j > \mathrm{rank}(n)$ where $\mathrm{rank}(n)$ is the smallest index in the decomposition of n.

Discrepancy

Definition (Discrepancy)

$\Delta_k := \sup_n \ |F_k(n) - \alpha_k\, n|$

- $F_1(n) = \lfloor (n+1)/2 \rfloor = \lceil n/2 \rceil$ hence $\Delta_1 = 0.5$
- $F_2(n) = \lfloor \alpha_2\, (n+1) \rfloor$ with $\alpha_2 = \varphi - 1 \approx 0.618...$ hence $\Delta_2 = \varphi - 1$

New results:

- $\Delta_3 < 1$
- $\Delta_4 < 2$
- For $k \geq 5$, $\sup_n (F_k(n) - \alpha_k\, n) = +\infty$ and $\inf_n (F_k(n) - \alpha_k\, n) = -\infty$ ($\Delta_k = \infty$ was already in Dilcher 1993).

Discrepancy

Definition (Discrepancy)

$\Delta_k := \sup_n \; |F_k(n) - \alpha_k\, n|$

- $F_1(n) = \lfloor (n+1)/2 \rfloor = \lceil n/2 \rceil$ hence $\Delta_1 = 0.5$
- $F_2(n) = \lfloor \alpha_2\,(n+1) \rfloor$ with $\alpha_2 = \varphi - 1 \approx 0.618...$ hence $\Delta_2 = \varphi - 1$

New results:

- $\Delta_3 < 1$
- $\Delta_4 < 2$
- For $k \geq 5$, $\sup_n(F_k(n) - \alpha_k\, n) = +\infty$ and $\inf_n(F_k(n) - \alpha_k\, n) = -\infty$ ($\Delta_k = \infty$ was already in Dilcher 1993).

This proves two conjectures of OEIS:

- $F_3(n) \in \lfloor \alpha_3\, n \rfloor + \{0, 1\}$
- $F_4(n) \in \lfloor \alpha_4\, n \rfloor + \{-1, 0, 1, 2\}$

Solving the Fibonacci-like recurrences

Theorem

For all n:

$$A_{k,n} = \sum_{i=0}^{k-1} c_{k,i}\, r_{k,i}^n$$

where $r_{k,i}$ are the roots of $X^k - X^{k-1} - 1$ and $c_{k,i} := r_{k,i}^k / (k r_{k,i} - (k-1))$.

- This generalizes the Binet formula.
- Coefficients obtained by inversing a Vandermonde matrix.
- Trick: temporary consider $\tilde{A}_{k,n}$ with the same recursion but initial values $0\ 0\ \cdots\ 0\ 1$.
- Formula already known to Dilcher (1993).

Computing discrepancies

With $D_k(n)$ the Zeckendorf k-decomposition of n and $r_{k,i}$ the roots of $X^k - X^{k-1} - 1$ and $d_{k,i}$ suitable coefficients:

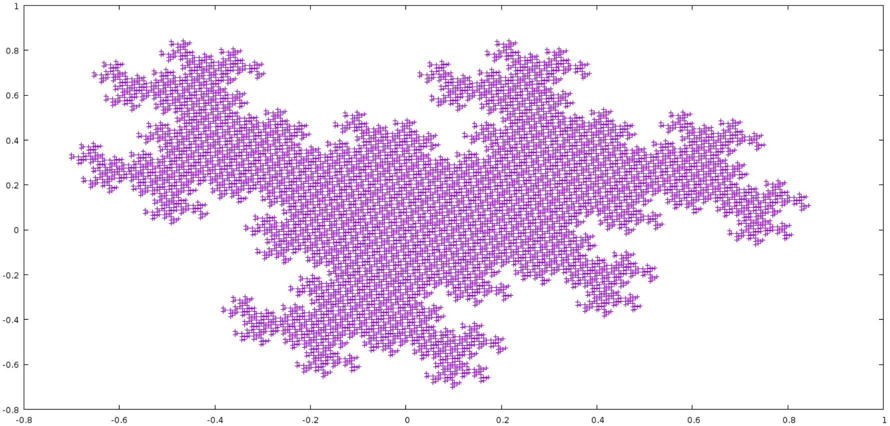
Theorem

$$F_k(n) - \alpha_k\, n = \sum_{q \in D_k(n)} \sum_{i=0}^{k-1} d_{k,i}\, r_{k,i}^q = \sum_{i=0}^{k-1} \left(d_{k,i} \sum_{q \in D_k(n)} r_{k,i}^q \right)$$

- One coefficient $d_{k,i}$ is null (the one for the positive root)
- For $k < 5$, all other roots have modulus strictly less than 1, leading to a finite discrepancy.
- For proving $\Delta_3 < 1$ and $\Delta_4 < 2$ we follow Rauzy and regroup some root powers together (up to k terms together). In these groups, a lot of cancellation happens.
- For $k \geq 5$, at least one non-real root has modulus 1 or more, leading to infinite discrepancy.

Serendipity : a Rauzy fractal

Let $\delta(n) := F_3(n) - \alpha_3\, n$, then plot $(\delta(i), \delta(F_3(i)))$ for many i:



Summary

	F_1	F_2	F_3	F_4	F_5	F_k $k \geq 6$
Hofstadter's name		G	H			
Mean slope α_k = $\mathrm{root}(X^k+X-1)$	0.5	$\varphi-1$	≈ 0.682	≈ 0.724	≈ 0.754	α_k
Discrepancy $\sup \lvert F_k(n) - \alpha_k n \rvert$	0.5	$\varphi-1$	< 1	< 2	$O(ln(n))$	$O(n^a)$, $0<a<1$
Exact expression	$\lfloor \frac{n+1}{2} \rfloor$	$\lfloor \frac{n+1}{\varphi} \rfloor$	$\times$	$\times$	$\times$	$\times$
Almost expression			$\lfloor \alpha_3 n \rfloor +$ $\{0,1\}$	$\lfloor \alpha_4 n \rfloor +$ $\{-1,0,1,2\}$	$\times$	$\times$
Almost additive	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\times$
$\beta_k = \frac{1}{\alpha_k}$ is Pisot	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$!	$\times$

Here $\beta_2 = \varphi \approx 1.618$ is the Golden Ratio.
And $\beta_5 \approx 1.324$ is the Plastic Ratio, root of $X^3 - X - 1$, smallest Pisot number.

Part 3

The Coq/Rocq formalization

Current status of the Coq formalization

- Freely accessible : `https://github.com/letouzey/hofstadter_g`
- Formalization of all results of part 1 and 2, and more.
- Quite large, about 30 000 lines. Lots of cruft, experiments, etc.
- Once installed, Coq rechecks the whole in 3 minutes.
- The discrete part is self-contained (`nat`, `list`, ...), no axioms.
- The part using $\mathbb{R}$ and $\mathbb{C}$ relies on 4 standard logical axioms (e.g. Excluded Middle) and two external libraries (Coquelicot, QuantumLib), with personal contributions and extensions.

Fibonacci-like recurrence

An example of direct definition (NB: "S" is Coq jargon for +1):

```
Fixpoint A (k n : nat) :=
  match n with
  | 0 ⇒ 1
  | S m ⇒ A k m + A k (m−(k−1))
  end.

Compute A 1 8. (* 256 *)
Compute A 2 8. (* 55 *)
```

Fibonacci-like recurrence

An example of direct definition (NB: “S” is Coq jargon for +1):

```
Fixpoint A (k n : nat) :=
  match n with
  | O => 1
  | S m => A k m + A k (m-(k-1))
  end.
```

```
Compute A 1 8. (* 256 *)
Compute A 2 8. (* 55 *)
```

Actually, some magic ensures that `m-(k-1)` is less than `n` hence a legal recursive call.

The `nat` datatype computes horribly slowly (in unary!), but mimics the Peano induction, which is handy in symbolic reasoning.

Fibonacci-like recurrence

Some corresponding proofs:

```
Lemma A_base k n : n ≤ k → A k n = n+1.
Proof.
 induction n; auto.
 simpl. intros.
 replace (n−(k−1)) with 0 by lia. simpl.
 rewrite IHn; lia.
Qed.

Lemma A_rec k n : 1 ≤ k → 1 ≤ n → A k n = A k (n−1) + A k (n−k).
Proof.
 intros. destruct n; try lia.
 simpl (A k (S n)).  f_equal; f_equal; lia.
Qed.
```

Defining F_k

Coq rejects the $f(f(\cdots))$ subcalls (no "structural decrease").
To overcome this, we count "generations" via an extra parameter p.

```
Notation "f ^^ n" := (Nat.iter n f) (at level 30, right associativity).
Fixpoint recf k p n :=
 match p, n with
 | S p, S n => S n - ((recf k p)^^k) n
 | _, _ => 0
 end.
Definition f k n := recf k n n.

Compute f 1 256. (* 128 *)
Compute f 2 89.  (* 55 *)
```

Then we ensure correctness by proving base and step equations for this f.

Alternative: predicative or inductive definitions, much more flexible but no computation.

A main result

```
Theorem f_grows k n : f q k ≤ f (S k) n.
Proof.
 ...
Qed.

Print Assumptions f_grows.
(* Closed under the global context *)
```

In the actual development, see theorem `Thm_7_4'` in `Article1.v`.

Alternative definition of F_k

More involved but faster: binary arithmetic plus memoization.
Can be proved equivalent to f.

```
Definition f_array (k n : N) :=
 N.peano_rect _
  (FlexArray.singleton 0)
  (fun n t ⇒ FlexArray.snoc t (N.succ n − N.iter k (FlexArray.get t) n))
  n.

Definition f_opt (k n : N) := FlexArray.get (f_array k n) n.

Compute f_opt 3 100000. (* = 68233 in less than 1s. *)
```

Finite and infinite words

Relatively generic definitions:

```
Notation letter := nat (only parsing).
Definition word := list letter. (* finite word *)
Definition sequence := nat → letter. (* infinite word *)
Definition subst := letter → word.

Definition apply : subst → word → word := ·flat_map _ _.
Definition napply (s:subst) n w := (apply s ^^n) w.

Definition subst2seq s a := fun n ⇒ nth n (napply s n [a]) a.
```

Examples τ_k and x_k:

```
Definition qsubst q (n:letter) := if n =? q then [q; 0] else [S n].
Definition qseq q := subst2seq (qsubst q) q.
```

Example of a larger proof I

```
Lemma Lq_LSq q n :
 L (S q) 1 n ≤ L q 1 n
 ∧ (0<n → forall j, j≤ S q → L q j n < L (S q) (S j) n).
Proof.
 induction n as [n IH] using lt_wf_ind.
 destruct (Nat.eq_dec n 0) as [→ |N0]; [easy|].
 destruct (Nat.eq_dec n 1) as [→ |N1].
 { clear N0 IH. split; intros;
   rewrite !L_S, !L_0, !qseq_q_0, !qnsub_qword, !qword_len, !A_base; lia. }
 split.
 - rewrite !Lq1_Cqq, ← !fs_count_q, ← Nat.add_le_mono_l.
   set (c := fs q q n).
   set (c' := fs (S q) (S q) n).
   destruct (Nat.eq_dec c' 0); try lia.
   replace c' with (S (c'-1)) by lia. change (c'-1 < c).
   apply (incr_strmono_iff _ (L_incr (S q) (S q))).
   apply Nat.lt_le_trans with n; [apply steiner_thm; lia|].
   transitivity (L q q c); [apply steiner_thm; lia|].
   destruct (Nat.eq_dec q 0) as [→ |Q].
   + rewrite L_q_0. apply L_ge_n.
   + apply Nat.lt_le_incl, IH; try apply fs_lt; try apply fs_nonzero; lia.
 - intros _. destruct n; try easy.
   destruct (Nat.eq_dec (qseq (S q) n) (S q)) as [E|N].
   + intros j Hj. rewrite !L_S, E.
     rewrite qnsub_qword, qword_len.
     assert (Hx := qseq_letters q n).
     set (x := qseq q n) in *.
     generalize (qnsub_len_le q j x Hx). rewrite !A_base by lia.
     destruct (IH n lia) as (_,IH').
     specialize (IH' lia j Hj).
     lia.
```

Example of a larger proof II

```
  + destruct (qsubst_prefix_inv (S q) (qprefix (S q) (S n)))
      as (v & w & Hv & E & Hw); try apply qprefix_ok.
    destruct Hw as [→ | → ].
    2:{ rewrite take_S in Hv; apply app_inv' in Hv; trivial;
        destruct Hv as (_,[= E']);  lia. }
    rewrite app_nil_r in Hv.
    red in E.
    set (l := length v) in *.
    assert (E' : L (S q) 1 l = S n).
    { now rewrite ← (qprefix_length (S q) (S n)), Hv, E. }
    assert (Hl0 : l <> 0). { intros → . now rewrite L_0 in E'. }
    assert (Hl : l < S n).
    { rewrite ← E'. rewrite Lq1_Cqq.
      generalize (Cqq_nz (S q) l). lia. }
    destruct (IH l Hl) as (IH5,IH6). clear IH. rewrite E' in IH5.
    specialize (IH6 lia).
    assert (LT : forall j, j ≤ q → L q j (S n) < L (S q) (S j) (S n)).
    { intros j Hj. specialize (IH6 (S j) lia).
      rewrite ← E' at 2. rewrite L_add, Nat.add_1_r.
      eapply Nat.le_lt_trans; [|apply IH6].
      rewrite ← (Nat.add_1_r j), ← L_add. apply incr_mono; trivial.
      apply L_incr. }
    intros j Hj. destruct (Nat.eq_dec j (S q)) as [→ |Hj'].
    * generalize (steiner_trick q (S n)).
      specialize (LT q (Nat.le_refl _)). lia.
    * apply LT. lia.
Qed.
```

Reals, Matrix, Polynomial, etc

- Due to the Coq standard definition of reals, this part uses 4 logical axioms (incl. excluded middle and functional extensionality).
- Example of library complement: the Vandermonde determinant.

```
Lemma Vandermonde_det n (l : list C) :
 length l = n ->
 Determinant (Vandermonde n l) = multdiffs l.
```

- A bit of interval arithmetic for computing bounds reliably (via rational approximation). For instance $\Delta_3 \leq 0.9959 < 1$.
- The proofs about Δ_3 automatically enumerate "sparse" subsets and find bounds for the corresponding 13 cases. Same for Δ_4 (69 cases).

Conclusion

About the Hofstadter F_k functions:

- Some remaining conjectures, seem to require new ideas.
- Is this specific to this particular recursion $X^k + X - 1$? What about similar functions e.g. for Rauzy's Tribonacci ? Unclear.

About the Coq/Rocq proof assistant:

- Proving this kind of study in Coq is doable, but still tricky and time consuming.
- Still critical parts to review manually : early definitions, final theorem statements, used axioms.
- Difficulty: lack of results in libraries (e.g. on complex, matrices, power series).