Generating Functions of Invariants in Numerical Semigroups

Jeske Glenn
Benjamin Sepanski

July 2017

1 Introduction

Given a Numerical Semigroup $N = \langle n_1, n_2, ..., n_k \rangle$, we define the Hilbert Series $\mathcal{H}(N;t)$ to be

$$\sum_{n \in N} t^n$$

We define the simplicial complex Δ_n of each element $n \in N$ to be simplicial complex on the vertices $\{n_1, n_2, ..., n_k\}$ where the d-dimensional face F of vertices $\{n_{r_1}, n_{r_2}, ..., n_{r_d}\}$ is in Δ_n if and only if $n - \sum_{i=1}^{d} n_{r_i} \in N$. It is known that one expression for the generating function of $\mathcal{H}(N;t)$ is

$$\frac{\mathcal{K}(N;t)}{\prod_{i=1}^{k}(1 - t^{n_i})}$$

where

$$\mathcal{K}(N;t) = \sum_{n \in N} \left(\sum_{F \in \Delta_n} (-1)^{|F|} \right) t^n = \sum_{n \in N} \chi(\Delta_n) t^n$$

where $\chi(\Delta_n)$ is the euler characteristic of the simplicial complex Δ_n. We introduce similar generating functions for other invariants on the semigroup N.

2 Max Length Factorization in Two Generated Numerical Semigroups

Let $S = \langle n_1, n_2 \rangle$, where $\{n_1, n_2\}$ is the minimal generating set for S. Now let $M(n)$ denote the maximum length of the factorizations of n for each $n \in S$. We denote

$$\mathcal{M}(N;t) = \sum_{n \in S} M(n) t^n$$

Theorem 1.

Classification of Generating Function for Two Generators

$$\mathcal{M}(N;t) = \frac{t^{n_1} + t^{n_2} - 2t^{n_1+n_2} - n_1 t^{n_1 n_2} + (n_1 - 1)t^{n_1 n_2 + n_1} + (n_1 - 1)t^{n_1 n_2 + n_2} - (n_1 - 2)t^{n_1 n_2 + n_1 + n_2}}{(1 - t^{n_1})^2 (1 - t^{n_2})^2} \tag{1}$$

Proof. We begin by noting that the right side of the above equation may be factored so that

$$\begin{aligned}
&\frac{(1-t^{n_1n_2})(t^{n_1}+t^{n_2}-2t^{n_1+n_2})-n_1t^{n_1n_2}(1-t^{n_1})(1-t^{n_2})}{(1-t^{n_1})^2(1-t^{n_2})^2}\\
&=\frac{(1-t^{n_1n_2})(t^{n_1}+t^{n_2}-2t^{n_1+n_2})}{(1-t^{n_1})^2(1-t^{n_2})^2}-n_1t^{n_1n_2}\frac{1}{(1-t^{n_1})(1-t^{n_2})}\\
&=\frac{(t^{n_1}+t^{n_2}-2t^{n_1+n_2})}{(1-t^{n_1})^2(1-t^{n_2})^2}-t^{n_1n_2}\frac{(t^{n_1}+t^{n_2}-2t^{n_1+n_2})}{(1-t^{n_1})^2(1-t^{n_2})^2}-n_1t^{n_1n_2}\frac{1}{(1-t^{n_1})(1-t^{n_2})}\\
&=\frac{t^{n_1}(1-t^{n_2})+t^{n_2}(1-t^{n_1})}{(1-t^{n_1})^2(1-t^{n_2})^2}-t^{n_1n_2}\frac{(t^{n_1}+t^{n_2}-2t^{n_1+n_2})}{(1-t^{n_1})^2(1-t^{n_2})^2}-n_1t^{n_1n_2}\frac{1}{(1-t^{n_1})(1-t^{n_2})} \qquad (2)
\end{aligned}$$

We now identify each of the three terms in the above equation. Firstly, consider

$$\begin{aligned}
\frac{1}{(1-t^{n_1})^2(1-t^{n_2})^2}&=\left(\frac{1}{1-t^{n_1}}\right)^2\left(\frac{1}{1-t^{n_2}}\right)^2\\
&=\left(\sum_{n\geq 0}t^{n\cdot n_1}\right)^2\left(\sum_{n\cdot n_2}t^{n\cdot n_2}\right)^2\\
&=\left(\sum_{n\geq 0}(n+1)t^{n\cdot n_1}\right)\left(\sum_{n\geq 0}(n+1)t^{n\cdot n_2}\right)
\end{aligned}$$

Now we see that

$$\frac{t^{n_1}(1-t^{n_2})+t^{n_2}(1-t^{n_1})}{(1-t^{n_1})^2(1-t^{n_2})^2}=\left(\frac{t^{n_1}}{(1-t^{n_1})^2}\right)\left(\frac{1}{(1-t^{n_2})}\right)+\left(\frac{1}{(1-t^{n_1})}\right)\left(\frac{t^{n_2}}{(1-t^{n_2})^2}\right)$$

$$\begin{aligned}
&=t^{n_1}\left(\sum_{n\geq 0}t^{n\cdot n_1}\right)^2\left(\sum_{n\geq 0}t^{n\cdot n_2}\right)+\left(\sum_{n\geq 0}t^{n\cdot n_2}\right)t^{n_2}\left(\sum_{n\geq 0}t^{n\cdot n_2}\right)^2\\
&=\left(\sum_{n\geq 1}nt^{n\cdot n_1}\right)\left(\sum_{n\geq 0}t^{n\cdot n_2}\right)+\left(\sum_{n\geq 0}t^{n\cdot n_1}\right)\left(\sum_{n\geq 1}nt^{n\cdot n_2}\right)\\
&=\left(\sum_{n\geq 0}nt^{n\cdot n_1}\right)\left(\sum_{n\geq 0}t^{n\cdot n_2}\right)+\left(\sum_{n\geq 0}t^{n\cdot n_1}\right)\left(\sum_{n\geq 0}nt^{n\cdot n_2}\right)
\end{aligned}$$

Note that we may re-index from $n \geq 1$ to $n \geq 0$ because the $n = 0$ term is 0. Now, the above equation is the sum of two products of sums. Consider the left product. For any $n \in S$, any factorization of n may be written in the form $an_1 + bn_2$ for some a and b (This is simply the $n = 2$ case of Lemma (1)). Note that, in the left product, the coefficient of t^{an_1} is, in fact, a. In particular, the coefficient of the t^n term in the left product is the sum of each value a over all factorizations $an_1 + bn_2$ of n. Analogously, the coefficient of t^n in the right product is the sum of each value b over all factorizations $an_1 + bn_2$ of n. Adding the two products of sums together, we find that the coefficient of t^n for $n \in S$ is equal to the sum of the lengths of each factorization of n in S. We have now characterized the power series generated by the first term in (2). In addition, the second term in (2) is simply the first term with each exponent shifted up by n_1n_2. In particular, the coefficient of t^n for $n - n_1n_2 \in S$ is the sum of the lengths of each factorization of $n - n_1n_2$ in S.

It remains to characterize the third term in (2). Define $\mathsf{z}_S(n)$ to be the set of factorizations of n in S, and to be 0 if $n \notin S$. Observe that

$$\frac{1}{(1-t^{n_1})}\frac{1}{(1-t^{n_2})}$$

is the generating function for

$$\sum_{n\in S} |\mathsf{z}_S(n)| t^n$$

It is then merely a matter of distributing $t^{n_1 n_2}$ into the sum to see that the third term in (2) generates

$$\sum_{n-n_1n_2\in S} |\mathsf{z}_S(n-n_1n_2)| t^n$$

For a given factorization $f \in \mathsf{z}_S(n)$ for some $n \in S$, let $l_S(f)$ denote the length of the factorization. We now have that 2 is equivalent to

$$\sum_{n\in S}\left(\sum_{f\in \mathsf{z}_S(n)} l_s(f)\right)t^n - \sum_{n-n_1n_2\in S}\left(\sum_{f\in \mathsf{z}_S(n-n_1n_2)} l_s(f)\right)t^n - n_1\sum_{n-n_1n_2\in S}|\mathsf{z}_S(n-n_1n_2)|t^n \quad (3)$$

It remains to show that $\mathcal{M}(N;t)$ is equal to (3). For $n < n_1n_2$, if $n \in S$ then n has exactly one factorization. Then, t^n term appears only in the leftmost sum, whose coefficient is the length of the only factorization of n. For $n \geq n_1n_2$, one may obtain all factorizations of n by first picking the factorization with the maximal number of n_1's, and then repeatedly exchanging n_2 n_1's for n_1 n_2's by noting that n_1n_2 is the least common multiple of n_1 and n_2.
From this process, one may see that the factorization using the most n_1's is the longest factorization. In particular, every other factorization has at least n_1 n_2's. Subtracting n_1 n_2's from any of these factorizations results in a unique factorization of $n - n_1n_2$, and adding n_1n_2 to any factorization of $n - n_1n_2$ creates a unique factorization of n. Thus, the summation of the second and third sums has, as the coefficient for t^n, the sum of the lengths of all non-maximal factorizations of n. Subtracting this from the first results in $\mathcal{M}(N;t)$.

□

The above proof separates the generating function for $\mathcal{M}(N;t)$ into parts and identifies what each part counts. In generalizing this result different methods of proof are used, but the basic components and the general idea of their function remains.

3 Max Length Factorization in Numerical Semigroups

Lemma 1.

Formula for the Sum of the Length of the Factorizations of n

$$\frac{\sum_{i=1}^{k}(-1)^{i-1}\, i \sum_{A\subseteq[k],|A|=i} t^{\sum_{j\in A} n_j}}{(1-t^{n_1})^2(1-t^{n_2})^2\cdots(1-t^{n_k})^2} = \frac{1}{\prod_{j=1,j\neq i}^{k}(1-t^{n_j})}\sum_{i=1}^{k}\frac{t^{n_i}}{(1-t^{n_i})^2} = \sum_{n\geq 0}\left(\sum_{f\in \mathsf{Z}_s(n)}|f|\right)t^n$$

where $[k] = \{1, 2, 3, \ldots, k\}$ and $\mathsf{Z}_s(n)$ is the set of factorizations of n in N.

Proof. To simplify, we will only look at the numerator momentarily.

$$\begin{aligned}
&\sum_{i=1}^{k}(-1)^{i-1}\, i \sum_{A\subseteq[k], i\in A} t^{\sum\limits_{j\in A} n_j} \\
&= \sum_{i=1}^{k}\left(\sum_{A\subseteq[k], i\in A}(-1)^{|A|-1} t^{\sum\limits_{j\in A} n_j}\right) \\
&= \sum_{i=1}^{k} t^{n_i}\left(\sum_{A\subseteq[k], i\in A}(-1)^{|A|-1} t^{\sum\limits_{j\in A, j\neq i} n_j}\right)
\end{aligned}$$

We may now rewrite the inner sum as a product by noting that given any subset A of $[k]$ such that $i \notin A$, we may obtain the term in the above inner sum for $A \cup \{i\}$ by multiplying together, for each generator n_j, 1 if $j \notin A$ and $(-1) \cdot t^{n_j}$ if $j \in A$. Thus, the above sum is equal to:

$$\sum_{i=1}^{k} t^{n_i} \prod_{j=1, j\neq i}^{k} (1 - t^{n_j})$$

Dividing by the denominator, we find

$$\begin{aligned}
&\sum_{i=1}^{k}\frac{t^{n_i}}{(1-t^{n_i})^2}\cdot\frac{1}{\prod\limits_{j=1, j\neq i}^{k}(1-t^{n_j})} \\
&= \sum_{i=1}^{k}\left(\left(\sum_{n} n t^{n\cdot n_i}\right)\prod_{j=1, j\neq i}^{k}\left(\sum t^{n\cdot n_j}\right)\right)
\end{aligned}$$

This is analogous to the sum of the lengths of the factorizations of n. For any $n \in S$, all factorizations of n are written in the form $an_1 + bn_2 + \cdots + kn_k$ for some $a, \ldots, k \in \mathbb{N}$. Consider the inner term:

$$\left(\sum_{n} n t^{n\cdot n_i}\right)\left(\prod_{j=1, j\neq i}^{k}\left(\sum t^{n\cdot n_j}\right)\right)$$

where the coefficient of t^{an_1} is a, and for t^n the coefficient is the sum of a's for all possible factorizations of n. When you sum up this product for every n_i we find that the coefficient of t^n is precisely the sum of the length of every factorization of n. $\square$

Definition 1. *Weighted Euler Characteristic*

Given a simplicial complex Δ on vertices $\{1, 2, 3, ..., n\}$, assign to each vertex i a weight $c_i \in \mathbb{Z}$ so that $c_1 \leq c_2 \leq c_3 \leq \cdots \leq c_n$, relabeling the vertices if necessary. Define

$$\chi_W(\Delta) = \sum_{F\in\Delta}(-1)^{|F|}\min_{i\in F}(c_i, c_n)$$

Lemma 2.

If Δ is the simplicial complex generated by the facet $\{1, 2, 3, ..., n\}$, then $\chi_W(\Delta) = 0$

Proof. Consider the faces of Δ which contain vertex 1. Each face is uniquely identified by some subset of the vertices $\{2, 3, 4, ..., n\}$, so there are exactly 2^{n-1} such faces. Since for each subset A of $\{2, 3, 4, ..., n\}$, the set $\{2, 3, 4, ..., n\} \setminus A$ has the opposite sign, exactly half of these faces are positive in the above sum, and exactly half are negative. In particular, the part of the above sum over these faces is 0. More generally, for i in $\{1, 2, 3, ..., n\}$, the number of faces containing the vertex i, but not any vertex in $\{1, 2, 3, ..., i-1\}$, is exactly 2^{n-i}. If $i < n$, then exactly half of these faces are positive in the above sum, and exactly half are negative. Thus, the sum over these faces is 0. If $i = n$, then there is only one such face, leaving a value of $-c_n$. The only remaining face, however, is the empty face, which restores the sum to 0. $\square$

Our definition for weighted euler characteristic is defined in order to work with the nabla complex Δ_n, the simplicial complex with each vertex being a factorization of n, and a face between a set of vertices if they have support in common, i.e. all of the factorizations use at least one of some common generator n_i.

Lemma 3.

Given a numerical semigoup N with minimal generating set $\{n_1, n_2, ..., n_k\}$, for $n \in N$, let $M(n)$ denote the maximum factorization length of n. Consider a face $F \in \nabla_n$. Each vertex in ∇_n is associated to some factorization of n. For a vertex $v \in F$, let $len(v)$ denote the length of its associated factorization. Define

$$g_n(F) = \underset{v \in F}{min}(len(v), M(n))$$

Now consider a face $f \in \Delta$ where Δ is the simplicial complex on vertices $\{1, 2, 3, ..., n\}$ generated by the facet $\{1, 2, 3, ..., n\}$. Define

$$h_n(f) = \begin{cases} M(n - \sum\limits_{i \in f} n_i) + |f| & n - \sum\limits_{i \in f} n_i \in N \\ 0 & n - \sum\limits_{i \in f} n_i \notin N \end{cases}$$

Then,

$$\chi_W(\nabla_n) = \sum_{F \in \nabla_n} (-1)^{|F|} g_n(F) = \sum_{f \in \Delta} (-1)^{|f|} h_n(f) = \chi_M(\Delta_n) \quad (4)$$

Note that the above equation defines χ_M.

Proof. Note that the first equality is simply the definition of χ_W. For some $A \subseteq \{1, 2, 3, ..., n\}$, put $S_A = \{F \in \nabla_n |$ the associated factorization of each vertex $v \in F$ uses at least one $n_i \forall i \in A\}$. Then, S_A is a simplicial complex which satisfies the requirements of Lemma (2), where the weight of each vertex $v \in S_A$ is $len(v)$. Since the only faces for which $\underset{i \in F}{min}(c_i, c_n) \neq g_n(F)$ are the empty face and possibly the vertex with the largest factorization in S_A,

$$\sum_{F \in S_{A_f}} (-1)^{|F|} g_n(F) = M(n) - \underset{v \in S_A}{max}(len(v))$$

Now, for each $f \in \Delta$, put $A_f = \{i \mid n_i \in f\}$. Note that for any subset of A of $\{1, 2, 3, ..., n\}$ there is some unique $f \in \Delta$ for which $A = A_f$. If $n - \sum\limits_{i \in f} n_i \notin N$, then $h_n(f) = 0$, and $S_{A_f} = \emptyset$. In this case,

$$\sum_{F \in S_{A_f}} (-1)^{|F|} g_n(F) = M(n)$$

Else, $n - \sum_{i \in f} n_i \in N$. In this case,

$$h_n(f) = M(n - \sum_{i \in f} n_i) + |f|$$

Now, consider S_{A_f}. For each $v \in S_{A_f}$, by definition its associated factorization must use at least one n_i for each $i \in f$. Now, note that

$$\{\text{factorizations of n which use } n_i \text{for each } i \in f\} = \{\text{factorizations of n } - \sum_{i \in f} n_i \text{with an added } n_i \text{for each } i \in f\}$$

Thus,

$$\underset{v \in S_{A_f}}{max}\,(len(v)) = M(n - \sum_{i \in f} n_i) + |f| = h_n(f)$$

In particular, for any A_f:

$$\sum_{F \in S_{A_f}} (-1)^{|F|} g_n(F) = M(n) - \underset{v \in S_A}{max}(len(v)) = M(n) - h_n(f)$$

Since $\nabla_n = \underset{|f|=1}{\cup} S_{A_f}$, and $S_{A_{f_1}} \cap S_{A_{f_2}} = A_{f_1 \cap f_2}$, one may apply inclusion-exclusion principle in summing over faces of ∇_n:

$$\sum_{F \in \nabla_n} (-1)^{|F|} g_n(F) = \sum_{f \in \Delta} (-1)^{|f|-1} \sum_{F \in S_{A_f}} (-1)^{|F|} g_n(F) = \sum_{f \in \Delta} (-1)^{|f|-1} (M(n) - h_n(f))$$
$$= -M(n) \sum_{f \in \Delta} (-1)^{|f|} + \sum_{f \in \Delta} (-1)^{|f|} h_n(f) = \sum_{f \in \Delta} (-1)^{|f|} h_n(f)$$

□

Definition 2. $\chi_M(\Delta_n) = \sum_{F \in \Delta_n} (-1)^{|F|} \left(M(n - \sum_{n_i \in F} n_i) + |F| \right)$

It is known that, for sufficiently large n, $M(n+n_1) = M(n)+1$. In order to fully characterize the generating function for $\mathcal{M}(N;t)$, we need the following definition.

Definition 3. Harmonic and Dissonant Semigroups
Given a Numerical Semigroup N with minimal generating set $\{n_1, n_2, n_3, ..., n_k\}$, we say N is harmonic *if*

$$M(n + n_1) = M(n) + 1 \ \forall \ n \in N$$

If N is not harmonic, we say N is dissonant. *If the above relation is satisfied for a particular element $n \in N$, we say that n has the* harmonic property.

This theorem gives three different characterizations for the generating function of $\mathcal{M}(N;t)$.

Theorem 2.

Generating Function for Generic Semigroup

(a)
$$\mathcal{M}(N;t) = zl \cdot \sum_{n\in N} \chi(\Delta_n)t^n + z \cdot \sum_{n\in N} \chi_M(\Delta_n)t^n$$
where $zl = \prod_{j=1}^{k} \frac{1}{(1-t^{n_j})} \left(\sum_{i=1}^{k} \frac{t^{n_i}}{(1-t^{n_i})} \right)$ and $z = \prod_{j=1}^{k} \frac{1}{(1-t^{n_j})}$
Note that zl is the function from Lemma (1) and z is the generating function for the number of factorizations of an element.

(b) Let $\frac{q(t)}{(1-t^{n_1})^2} = \sum_{n\in N} M(n)t^n$, then $q(t) = \sum_{n\geq 0} [M(n) - 2M(n-n_1) + M(n-2n_1)]\, t^n$
If $n - n_1$ and $n - 2n_1$ have the harmonic property and are in N, then the coefficient of t^n is 0.

(c) Let $\frac{r(t)}{(1-t^{n_1})(1-t)} = \sum_{n\in N} M(n)t^n$, then $r(t) = \sum_{n\geq 0} [M(n) - 2M(n-n_1) + M(n-n_1-1)]\, t^n$
If $n - n_1 - 1 \in N$ and $n - n_1$ and $n - n_1 - 1$ have the harmonic property, then the coefficient of t^n is 0.

Proof. (a) We begin by multiplying the right side of the equation by the denominator

$$\prod_{j=1}^{k}(1-t^{n_j}) \sum_{n\in N} M(n)t^n$$
$$= \sum_{n\in N} \sum_{A\subseteq[k]} (-1)^{|A|} t^{\sum_{i\in A} n_i} M(n)t^n \text{ where } [k] = \{1, 2, \ldots, k\}$$
let $m = n + \sum_{i\in A} n_i$
$$= \sum_{m\in N} \left(\sum_{\substack{A\subseteq[k], \\ m - \sum_{i\in A} n_i \in N}} (-1)^{|A|} M(m - \sum_{j\in A} n_j) \right) t^n$$

It is now important to note that

$$\sum_{n\in N} \sum_{A\in\Delta_n} (-1)^{|A|} \left(\sum_{i=1}^{k} \frac{t^{n_i}}{1-t^{n_i}} \right) t^n + \sum_{n\in N} \sum_{A\in\Delta_n} (-1)^{|A|} |A| t^n = 0 \tag{5}$$

Proof.

$$= \prod_{j=1}^{k}(1-t^{n_i})\sum_{n\in N}\sum_{A\in\Delta_n}(-1)^{|A|}\left(\sum_{i=1}^{k}\frac{t^{n_i}}{1-t^{n_i}}\right)t^n + \prod_{j=1}^{k}(1-t^{n_i})\sum_{n\in N}\sum_{A\in\Delta_n}(-1)^{|A|}|A|t^n$$

$$= \sum_{n\in N}\sum_{A\in\Delta_n}(-1)^{|A|}\left(\sum_{i=1}^{k}t^{n_i}\prod_{j=1,j\neq i}^{k}(1-t^{n_i})\right)t^n + \sum_{n\in N}\sum_{B\subseteq[n]}(-1)^{|B|}t^{\sum\limits_{j\in B}n_j}\sum_{A\in\Delta_n}(-1)^{|A|}|A|t^n$$

let $B\subseteq[n], m = n+\sum_{j\in B}n_j$

$$= \sum_{m\in N}\sum_{A\in\Delta_m}(-1)^{|A|}\sum_{i=1}^{k}\sum_{\substack{B\subseteq[m],\\ i\in B,\\ m-\sum_{j\in B}n_j\in N}}(-1)^{|B|-1}t^m + \sum_{m\in N}\sum_{\substack{B\subseteq[m],\\ m-\sum_{i\in B}n_i\in N}}(-1)^{|B|}\sum_{A\in\Delta_m}(-1)^{|A|}|A|t^m$$

$$= \sum_{m\in N}\sum_{A\in\Delta_m}(-1)^{|A|}\sum_{\substack{B\subseteq[m],\\ m-\sum_{j\in B}n_j\in N}}(-1)^{|B|-1}|B|t^m + \sum_{m\in N}\sum_{\substack{B\subseteq[m],\\ m-\sum_{i\in B}n_i\in N}}(-1)^{|B|}\sum_{A\in\Delta_m}(-1)^{|A|}|A|t^m$$

$$= \sum_{m\in N}\sum_{A\in\Delta_m}(-1)^{|A|}\sum_{B\in\Delta_m}(-1)^{|B|-1}|B|t^m + \sum_{m\in N}\sum_{B\in\Delta_m}(-1)^{|B|}\sum_{A\in\Delta_m}(-1)^{|A|}|A|t^m = 0$$

□

We will now use this claim by taking our current summation and adding on these two terms which sum to zero.

$$= \sum_{n\in N}\sum_{k\in\Delta_n}(-1)^{|A|}\left(\sum_{i=1}^{k}\frac{t^{n_i}}{1-t^{n_i}}\right)t^n + \sum_{m\in N}\left(\sum_{\substack{A\subseteq[k],\\ m-\sum_{i\in A}n_i\in N}}(-1)^{|A|}M(m-\sum_{j\in A}n_j)\right)t^n$$

$$+ \sum_{n\in N}\sum_{A\in\Delta_n}(-1)^{|A|}|A|t^n$$

$$= \sum_{n\in N}\left[\left(\sum_{k\in\Delta_n}(-1)^{|A|}\right)\left(\sum_{i=1}k\frac{t^{n_i}}{(1-t^{n_i})}\right) + \sum_{A\in\Delta_n}(-1)^{|A|}\left(M(n-\sum_{n_i\in A})+|A|\right)\right]t^n$$

$$= \prod_{j=1}^{k}(1-t^{n_j})\sum_{n\in N}\left(\chi(\Delta_n)zl + \chi_M(\Delta_n)z\right)t^n$$

(b) Here we extend $M(n)$ to $\mathbb{Z}$ by letting $M(n) = 0$ if $n \notin N$

$$(1-t^{n_1})^2 \sum_{n\in N} M(n)t^n = (1-t^{n_1}) \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^n$$
$$= \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^n - \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^{n+n_1}$$
$$= \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^n - \sum_{n\geq n_1} (M(n-n_1) - M(n-n_1)t^{n_1})\, t^n$$
$$= \sum_{n\geq 0} [M(n) - M(n-n_1) - (M(n) - M(n-n_1))\, t^{n_1}]\, t^n$$
$$= \sum_{n\geq 0} [M(n) - M(n-n_1) - (M(n-n_1) - M(n-2n_1))]\, t^n$$

And now we must observe the four possible cases of n's:

1. $n - 2n_1 \in N \to M(n) = M(n-n_1) + 1 + b$ and $M(n-n_1) = M(n-2n_1) + 1 + d$ so $[t^n]q(t) = b - d$
2. $n - 2n_1 \notin N$ but $n - n_1 \in N \to M(n) = M(n-n_1) + 1 + b$ and $M(n-2n_1) = 0$ so $[t^n]q(t) = 1 + b - M(n-n_1)$
3. $n - n_1 \notin N \to M(n-n_1) = M(n-2n_1) = 0$ so $[t^n]q(t) = M(n)$
4. $n \notin N \to M(n) = 0$ so $[t^n]q(t) = 0$

(c) Continuing the extension on $M(n)$ to $\mathbb{Z}$

$$(1-t^{n_1})(1-t) \sum_{n\in N} M(n)t^n = (1-t) \sum_{n\geq 0} \left(M(n) - M(n)t^{n-1}\right) t^n$$
$$= \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^n - \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^{n+1}$$
$$= \sum_{n\geq 0} (M(n) - M(n)t^{n_1})\, t^n - \sum_{n\geq 1} (M(n-1) - M(n-1)t^{n_1})\, t^n$$
$$= \sum_{n\geq 0} \left(M(n) - M(n-1) - (M(n) - M(n-1))\, t^{n-1}\right) t^n$$
$$= \sum_{n\geq 0} (M(n) - M(n-1) - (M(n-n_1) - M(n-n_1-1)))\, t^n$$

A similar argument of multiple cases of n will hold here as well.

1. $n - n_1 - 1 \in N$, $n - n_1$ and $n - n_1 - 1$ have the harmonic property:
   $n - n_1 - 1 \in N \to M(n-n_1) + 1 = M(n),\ M(n-n_1-1) + 1 = M(n-1)$ so $[t^n]r(t) = 0$
2. The Non-Zero Cases:
   - i Dissonance on either $n - n_1$ or $n - n_1 - 1$ so that $M(n-n_1) + 1 \neq M(n)$ or $M(n-n_1-1) + 1 \neq M(n-1)$ allowing $[t^n]r(t)$ to have terms which do not cancel completely.
   - ii Either n or $n - 1$ are in the apery set, meaning $n - n_1$ or $n - 1 - n_1$ not in N again leading to lack of cancellation and some $M(x) = 0$.
   - iii Either n or $n - 1$ are not in N which forces $n - n_1$ or $n - 1 - n_1$ to not be in N respectively, causing some $M(x) = 0$ and less cancellation.

□

The following lemma allows for some characterization of the numerator in part (a) of the previous theorem.

Lemma 4.

Let N be a numerical semigroup with minimal generating set $\{n_1, n_2, n_3, ..., n_k\}$. Suppose some $n \in N$ has the property that for any subset $A \subseteq \{1, 2, 3, ..., k\}$, if

$$m = n - \sum_{i \in A} n_i \ \in N$$

then m has the harmonic property, and that for each face F in Δ_n such that $n_1 \notin F$, $F \cup \{n_1\} \in \Delta_n$. Then,

$$\chi_M(\Delta_n) = 0$$

Proof. Choose a face F in Δ_n. If $n_1 \in F$, then by definition of Δ_n, $F \setminus \{n_1\} \in \Delta_n$. Else, $n_1 \notin F$, and by assumption $F \cup \{n_1\} \in \Delta_n$. In particular, we may group the faces of Δ_n into pairs in this manner. For such a pair of faces, let $n_1 \in F_1$ and let $F_2 = F_1 \setminus \{n_1\}$. Then, note that

$$|F_1| = |F_2| + 1$$

And that, using the definition of $h_n(F)$ from Lemma (3),

$$h_n(F_1) = M(n - \sum_{n_i \in F_1} n_i) + |F_1| = M(n - \sum_{n_i \in F_2} n_i - n_1) + |F_1| = M(n - \sum_{n_i \in F_2} n_i) + |F_1| - 1 = h_n(F_2)$$

Since $n - \sum_{n_i \in F_1} n_i$ is assumed to have the harmonic property. Thus,

$$(-1)^{|F_1|} h_n(F_1) = -(-1)^{|F_2|} h_n(F_2)$$

Since each face in Δ_n is a member of exactly one such pair, we see that

$$\chi_M(\Delta_n) = \sum_{f \in \Delta_n} (-1)^{|f|} h_n(f) = 0$$

□

Corollary 1.

For any Numerical Semigroup N and sufficiently large $n \in N$,

$$\chi_M(\Delta_n) = 0$$

Proof. Suppose $\{n_1, n_2, n_3, ..., n_k\}$ is the minimal generating set for N. It is known that $M(n)$ is a quasi-linear function in n for $n \gg 0$. Choose n such that

$$n - \sum_{i=1}^{k} n_i \ \in N$$

and such that $M(m)$ is quasi-linear for all values $m \geq n - \sum_{i=1}^{k} n_i$. Then, Lemma (4) holds for n. □

Corollary 2.

For a harmonic Numerical Semigroup N with minimal generating set $\{n_1, n_2, n_3, ..., n_k\}$, let $F(N)$ denote the Frobenius number of N. Let

$$n = F(N) + \sum_{i=1}^{k} n_i$$

Then

$$\chi_M(\Delta_n) = (-1)^{|k-1|} h_n(\{n_2, n_3, n_4, ..., n_k\})$$

In particular, $\chi_M(\Delta_n) \neq 0$.

Proof. Note that Δ_n contains every face except for $F_1 = \{n_1, n_2, n_3, ..., n_k\}$. Since this face, in the proof of Lemma (4), is paired with $F_2 = \{n_2, n_3, n_4, ..., n_k\}$, the sum over all other faces is 0. This leaves $\chi_M = (-1)^{|F_2|} h_n(F_2)$. □

Corollary 3.

For a dissonant harmonic Numerical Semigroup N, define the *dissonance point* $d(N)$ such that $d(N)$ does *not* have the harmonic property, but for $n \in N$, $n > d(N)$ implies that n has the harmonic property. Then, if $\{n_1, n_2, n_3, ..., n_k\}$ is the minimal generating set of N, and

$$m = d(N) + \sum_{i=1}^{k} n_i$$

Then

$$\chi_M(\Delta_m) \neq 0$$

Proof. Since m is defined so that $m - \sum_{i=1}^{k} n_i \in N$, Δ_m has every face on the vertices $\{n_1, n_2, n_3, ..., n_k\}$. Moreover, for every face F except for $F_1 = \{n_1, n_2, n_3, ..., n_k\}$, $m - \sum_{n_i \in F} n_i$ has the harmonic property. Put $F_2 = F_1 \setminus \{n_1\}$. As in the above proof, the sum to calculate χ_M over all faces other than F_1 and F_2 of Δ_m gives 0. Thus,

$$\begin{aligned}
\chi_M(\Delta_m) &= (-1)^{|F_1|} h_n(F_1) + (-1)^{|F_2|} h_n(F_2) = (-1)^{|F_1|}(h_n(F_1) - h_n(F_2)) \\
&= (-1)^{|F_1|}(M(d(N)) + |F_1| - M(d(N) + n_1) - |F_2|) \\
&= (-1)^{|F_1|}(M(d(N)) + 1 - M(d(N) + n_1)) \\
&\neq 0
\end{aligned}$$

By the definition of $d(N)$. □

Corollary 4.

Given a Numerical Semigroup $N = \langle n_1, n_2, ..., n_k \rangle$,

$$\deg \sum_{n \in N} \chi_M(\Delta_n) t^n = \max(F(N), d(N)) + \sum_{i=1}^{k} n_i$$

Proof. Put $m = \max(F(N), d(N)) + \sum_{i=1}^{k} n_i$. Note that $d(N) \in N$ and $F(n) \notin N$ by definition, so $d(N) \neq F(N)$. If $d(N) > F(N)$, then Lemma (4) holds for $n > m$, so for $\chi_{_M}(\Delta_n) = 0$ for $n > m$. In addition, we know $m \neq 0$ by Corollary (3), so the degree of $\sum_{n \in N} \chi_{_M}(\Delta_n) t^n$ must be m. Now suppose $F(n) > d(N)$. Lemma (4) still holds for $n > m$. Since $F(n) > d(N)$, $m - \sum_{n_i \in F} n_i$ for any $F \in \Delta_m$ is harmonic, so that Corollary (2) holds and $\chi_{_M}(\Delta_m) \neq 0$. □

Note:
The conditions of the second corollary may be loosened slightly to either N is harmonic, or $F(N) > d(N)$. In addition, if either of these hold, then $\chi_M(\Delta_n) = 0$ for all $n > F(N) + \sum_{i=1}^{k} n_i$.

Note:
Let $m(n)$ denote the minimal factorization length of $n \in N$. It is known that $m(n+n_k) = m(n)+1$ for $n \gg 0$. There are analogous results for each of the above statements in section 2, the proofs are also analogous.

Lemma 5.

If N is a dissonant numerical semigroup, then there exists some $n - n_1 \in N$ without the harmonic property such that the intersection of the supports of the maximal factorization of n and any factorization of n using at least one n_1 is empty.

Proof. If N is a dissonant numerical semigroup, then there exists some $n - n_1 \in N$ without the harmonic property. In particular,

$$M(n) \neq M(n - n_1) + 1$$

This implies that the factorization f_0 of n with maximal length does not use any n_1's. Now consider any factorization f of n that uses at least one n_1. If f and f_0 have support in common, then subtract off the intersection of their support to obtain a new n. This new n retains the property that

$$M(n) \neq M(n - n_1) + 1$$

Since f_0 minus the intersection in support will still be of greater length than any factorization f which uses an n_1 minus the intersection in support (if such a factorization is an appropriate factorization of the new n. Note that by construction one such f always exists). This is a process which will, in a finite number of steps, obtain some n such that $n - n_1$ is not harmonic and the intersection of the supports of the maximal factorization of n and any factorization of n using at least one n_1 is empty. □

Corollary 5.

Given a Numerical Semigroup N with minimal generating set $\langle n_1, n_2, n_3 \rangle$, then N is a dissonant semigroup if and only if there exist $a, b, c \in \mathbb{N}$ so that bn_2 has two factorizations: $(0, b, 0)$ and $(a, 0, c)$, and $bn_2 - n_1$ does *not* have the harmonic property

Proof. If N has any element without the harmonic property, it is dissonant by definition. Now suppose that N is dissonant. Then, by lemmma (5), there exists some $n - n_1 \in N$ without the harmonic property such that the intersection of the supports of the maximal factorization of n

and any factorization of n using at least one n_1 is empty. Now, if this element exists, it is of the form $n = (a, b, c)$ such that either

$$n = bn_2 + cn_3 = an_1$$
$$n = cn_3 = an_1 + bn_2$$
$$n = bn_2 = an_1 + cn_3$$

Now, in the first equation it is clear that $a \geq b + c$, and in the second equation that $c \leq a + b$ since $n_1 < n_2 < n_3$. In particular, since it is known that $M(n)$ is quasi-linear for $n > n_1n_3$, the above factorizations are the only factorizations of n, and a maximal factorization of n uses n_1 so $n - n_1$ is harmonic. Thus, the only possible such element n is of the form

$$bn_2 = an_1 + cn_3$$

Lemma (5) ensures that there is such an n so that $n - n_1$ is dissonant. □

4 Maximal Factorization Length Under Gluings

Given two numerical semigroups $S_1 = \langle s_{1,1}, s_{1,2}, ..., s_{1,k_1} \rangle$ and $S_2 = \langle s_{2,1}, s_{2,2}, ..., s_{2,k_2} \rangle$, choose $b \in S_2$, $c \in S_1$ such that $(b, c) = 1$, and let $G = \langle bs_{1,1}, bs_{1,2}, ..., bs_{1,k_1}, cs_{2,1}, cs_{2,2}, ..cs_{2,k_2} \rangle$. Let $g_1 < g_2 < \cdots < g_{k_1+k_2} = g_{k_G}$ be the ordered minimal generating set of G. Assume that, without loss of generality, $bs_{1,1} = g_1$. Let $M_1(n), M_2(n), M_G(n)$ be the maximal factorization length of n in S_1, S_2, and G respectively (assuming n is in the respective semigroup). We denote $G = bS_1 + cS_2$. Note that $M_1(s + c) > M_1(s)$ for all $s \in S_1$ since $c \in S_1$.

In general, characterizing $\chi_{M(G)}$ in terms of $\chi_{M(S_1)}$ and $\chi_{M(S_2)}$ is very difficult. Much of this arises from the fact that the gluing of two harmonic semigroups may be dissonant. Take, for example,

$$G = \langle 138, 230, 345, 135, 162 \rangle = 23 \cdot \langle 6, 10, 15 \rangle + 27 \cdot \langle 5, 7 \rangle$$

Both S_1 and S_2 are harmonic, while G has dissonant elements

$$\{831, 969, 993, 1061, 1131, 1155, 1199, 1223, 1291, 1293, 1317, 1361, 1385, 1429, 1453, 1455, 1479, \\ 1523, 1547, 1591, 1615, 1617, 1685, 1709, 1753, 1777, 1847, 1915, 1939, 2077\}$$

Both S_1 and S_2 have weighted euler characteristics very similar to the Hilbert Series Numerator,

$$\begin{array}{ll} \mathcal{K}(S_1; t) = 1 - 2t^{30} + t^{60} & \chi_{M(S_1)} = -5t^{30} + 5t^{60} \\ \mathcal{K}(S_2; t) = 1 - t^{30} & \chi_{M(S_2)} = -5t^{30} \end{array}$$

Whereas G's is quite dissimilar

$$\mathcal{K}(G; t) = 1 - t^{621} - 2t^{690} - t^{810} + 2t^{1311} + t^{1380} + t^{1431} + 2t^{1500} - t^{2001} - 2t^{2121} - t^{2190} + t^{2811}$$
$$\begin{aligned} \chi_{M(G)} - &-3t^{621} - 5t^{690} - 5t^{810} - t^{966} - 2t^{1242} - t^{1290} + 12t^{1311} + 5t^{1380} + 8t^{1431} + 15t^{1500} \\ &+ t^{1566} + 2t^{1587} + t^{1635} + t^{1656} + 2t^{1776} - t^{1911} + 2t^{1932} + t^{1980} - 9t^{2001} + t^{2052} - 23t^{2121} \\ &- 10t^{2190} - t^{2256} - 2t^{2277} - t^{2325} - t^{2397} - 2t^{2466} + t^{2601} - t^{2742} + 15t^{2811} \end{aligned}$$

One may often expect two harmonic semigroups to produce a dissonant one if some of their small generators wind up very close together in the gluing.

Definition 4. *Given $G = bS_1 + cS_2$, consider $n \in G$. We may write n as a decomposition into $bn' + cn''$ with $n' \in S_1$, $n'' \in S_2$. Let n_1 be the maximal such n', and n_2 be its corresponding n''. We say that G is a* harmonic gluing *if, for all $n \in G$, $M_1(n_1) + M_2(n_2) = M_G(n)$.*

Lemma 6.

Given $G = bS_1 + cS_2$ a harmonic gluing, let $bn_1 + cn_2$ be the decomposition of $n \in G$ with maximal factorization length. Then $g_2 = (g+b)_2$.

Proof.

$$\begin{aligned} n &\equiv n + b \bmod b \\ \Rightarrow\ bn_1 + cn_2 &\equiv b(n+b)_1 + c(n+b)_2 \bmod b \\ \Rightarrow\ cn_2 &\equiv c(n+b)_2 \bmod b \end{aligned}$$

Since $(b, c) = 1$, $c^{-1} \bmod b$ exists, and

$$\begin{aligned} n_2 &\equiv (n+b)_2 \bmod b \\ \Rightarrow n_2 &= (n+b)_2 + kb \text{ for some } k \in \mathbb{Z} \end{aligned}$$

Suppose $k > 0$. Then,

$$\begin{aligned} n &= bn_1 + cn_2 \\ &= bn_1 + c\left((n+b)_2 + kb\right) \\ &= b\left(n_1 + kc\right) + c(n+b)_2 \end{aligned}$$

Since $n_1 + kc \in S_1$ and $(n+b)_2 \in S_2$, this is a valid decomposition of n, contradicting the maximality of n_1.
Now suppose $k < 0$. Then write $n_2 + k'b = (n+b)_2$ with $k' = -k > 0$. Then,

$$\begin{aligned} n + b &= b(n+b)_1 + c(n+b)_2 \\ &= b(n+b)_1 + c\left(n_2 + k'b\right) \\ &= b\left((n+b)_1 + k'c\right) + cn_2 \end{aligned}$$

Since $(n+b)_1 + k'c \in S_1$ and $n_2 \in S_2$, this is a valid decomposition of $n + b$, contradicting the maximality of $(n+b)_1$.

$\square$

Corollary 6.

Suppose $G = bS_1 + cS_2$ is a harmonic gluing. Then,

$$\left(\sum_{n \in \mathrm{Ap}(S_2;b)} M_2(n)(t^c)^n\right) \sum_{m \in S_1} (t^b)^m + \sum_{n \in \mathrm{Ap}(S_2;b)} (t^c)^n \left(\sum_{m \in S_1} M_1(m)(t^b)^m\right) = \sum_{g \in G} M_G(g) t^g \quad (6)$$

This is equal to

$$\left(\sum_{n \in \mathrm{Ap}(S_2;b)} M_2(n)(t^c)^n\right) \mathcal{H}(S_1; t^b) + (1 - t^{bc})\mathcal{H}(S_2; t^c) \cdot \left(zl_1(t^b)\mathcal{K}(S_1; t^b) + z_1(t^b)\chi_{M(1)}(t^b)\right)$$

Proof. Consider $a \in \mathrm{Ap}(G; b)$. We must have $a_1 = 0$, so $a = ca_2$, and $a_2 \in \mathrm{Ap}(S_2; b)$. Suppose $n \in G$ such that $n \equiv a \bmod b$. Then $n = bn_1 + cn_2 = bn_1 + ca = bn_1 + ca_2$. In particular,

$$M_G(n) = M_1(n_1) + M_2(a_2)$$

Now consider the coefficient of t^n in the left-hand side of equation (6). In the left-hand product of sums, its coefficient must be $M_2(a_2)$ since n is uniquely obtained as an exponent by $bn_1 + cn_2 = bn_1 + ca_2$. In the right-hand product of sums, n is uniquely obtained in the same manner, and so has a coefficient of $M_1(n_1)$. In particular, its total coefficient is the sum of these two coefficients, and so is $M_1(n_1) + M_2(a_2) = M_G(n)$.
As to the second equation,

$$\mathcal{H}(S_1; t^b) = \sum_{m \in S_1} (t^b)^m$$

by definition, and it is known that

$$\frac{\sum\limits_{n \in \mathrm{Ap}(S_2;b)} (t^c)^n}{1 - (t^c)^b} = \mathcal{H}(S_2; t^c)$$

The substitution of

$$zl_1(t^b)\mathcal{K}(S_1; t^b) + z_1(t^b)\chi_(t^b)$$

for

$$\sum_{m \in S_1} M_1(m)(t^b)^m$$

comes from Theorem (2). □

Lemma 7.

Suppose $G = bS_1 + cS_2$, and that G and S_1 are harmonic. Then G is a harmonic gluing.

Proof. Consider $a \in \mathrm{Ap}(S_1; s_{1,1})$. Suppose $ba - bs_{1,1} = bn_1 + cn_2$ for some $n_1 \in S_1$, $n_2 \in S_2$. Then

$$b(a - s_{1,1} - n_1) = cn_2$$

Since $a - s_{1,1} \notin S_1$, and $n_1 \in S_1$, $a - s_{1,1} - n_1 \notin S_1$. Since $c \in S_1$, c does not divide $(a - s_{1,1} - n_1)$. Since $(b, c) = 1$, c does not divide b. Thus, $ck \neq b(a - s_{1,1} - n_1)$ for all $k \in \mathbb{Z}$ and $ba \in \mathrm{Ap}(G; g_1)$. In addition, each such ba has exactly one decomposition. Note then that $ba + ck$ for some $0 \leq k \leq b - 1$ must then also be in $\mathrm{Ap}(G; g_1)$ and have exactly one decomposition. We proceed by induction.

Suppose that the statement holds for $n < m$ for some $m \in G$ not in $\mathrm{Ap}(G; g_1)$. Consider $m - g_1$. Since G is harmonic, $M_G(m - g_1) = M_G(m) - 1$, so if $(m - g_1) = b(m - g_1)_1 + c(m - g_1)_2$ with $(m - g_1)_1$ maximal in factorization length, then $(m - g_1)_1 + s_{1,1}$ is the n_1 with maximal factorization length (note that S_1 is harmonic) in a decomposition $bn_1 + cn_2$ of n. By the inductive hypothesis and since G is harmonic and $bs_{1,1} = g_1$, this must give rise to the maximal factorization length of n. In addition, if there exists some $n_1' > n_1$, then $n = bn_1 + ck$ for some $k \in S_2$ with $k \geq b$. In particular, $n - g_1 = bn_1 + ck - g_1 = b(n - g_1)_1 + ck = b((n - g_1)_1 + c) + c(k - b)$ which contradicts the definition of $(n - g_1)_1$. □

5 General Invariants

Theorem 3.

Consider some mapping $f : \mathbb{R} \to \mathbb{R}$ such that $f(n) = 0\ \forall n \notin N$ where $N = \langle n_1, n_2, \ldots, n_k \rangle$. Let $[k] = \{1, 2, 3, \ldots, k\}$ Define $\widehat{\chi_f}(\Delta_n) = \sum\limits_{F \in n} (-1)^{|F|} f(n - \sum\limits_{j \in F} n_j)$

Then, $\sum f(n)t^n = \dfrac{\sum \widehat{\chi_f}(\Delta_n)t^n}{\prod_{i=1}^{k}(1-t^{n_i})}$

Proof.

$$\begin{aligned}
&\prod_{i=1}^{k}(1-t^{n_i})\left(\sum_{n\geq 0} f(n)t^n\right)\\
&= \sum_{n\geq 0}\left(\prod_{i=1}^{k}(1-t^{n_i})\right) f(n)t^n = \sum_{m\geq 0}\left(\sum_{A\subseteq[k]}(-1)^{|A|}f(n-\sum_{j\in A}n_j)\right)t^n\\
&= \sum_{m\geq 0}\left(\sum_{\substack{A\subseteq[k]\\ n-\sum\limits_{j\in A}n_j\in N}}(-1)^{|A|}f(n-\sum_{j\in A}n_j)\right)t^n = \sum_{m\geq 0}\left(\sum_{F\in\Delta_M}(-1)^{|F|}f(n-\sum_{j\in A}n_j)\right)t^n\\
&= \sum_{m\geq 0}\widehat{\chi_f}(\Delta_n)t^n
\end{aligned}$$

□

Corollary 7.

$$\textit{Define } \chi_f(\Delta_n) = \sum_{F\in\Delta_n}(-1)^{|F|}(f(n-\sum_{j\in F}n_j)+|F|)$$

$$\sum f(n)t^n = zl\sum_{n\geq 0}\chi(\Delta_n)t^n + z\sum_{n\geq 0}\chi_f(\Delta_n)t^n$$

Proof. We will now refer back to equation (5) and the definitions of zl and z from Theorem (2)

$$\begin{aligned}
\sum f(n)t^n &= \frac{\sum\limits_{n\in N}\sum\limits_{F\in\Delta_n}(-1)^{|F|}\left(\sum\limits_{i=1}^{k}\frac{t^{n_i}}{1-t^{n_i}}\right)t^n}{\prod\limits_{i=1}^{k}(1-t^{n_i})} + \frac{\sum\limits_{F\in\Delta_n}(-1)^{|F|}f(n-\sum\limits_{j\in F}n_j)}{\prod\limits_{i=1}^{k}(1-t^{n_i})} + \frac{\sum\limits_{n\in N}\sum\limits_{F\in\Delta_n}(-1)^{|F|}|F|t^n}{\prod\limits_{i=1}^{k}(1-t^{n_i})}\\
&= \frac{1}{\prod\limits_{i=1}^{k}(1-t^{n_i})}\left(\sum_{i=1}^{k}\frac{t^{n_i}}{1-t^{n_i}}\right)\sum_{n\in N}\sum_{F\in\Delta_n}(-1)^{|F|}t^n + \frac{1}{\prod\limits_{i=1}^{k}(1-t^{n_i})}\left(\sum_{n\in N}\sum_{F\in\Delta_n}(-1)^{|F|}(f(n-\sum_{j\in F}n_j)+|F|)\right)t^n\\
&= zl\sum_{n\geq 0}\chi(\Delta_n)t^n + z\sum_{n\geq 0}\chi_f(\Delta_n)t^n
\end{aligned}$$

□

Lemma 8.

If $\langle n_1, n_2, ..., n_k\rangle$ is the minimal generating set for some numerical semigroup N, and if f satisfies the conditions of Theorem (3), then if for $n \gg 0$

$$f(n+n_j) = f(n) + d$$

for some $d\in\mathbb{R}$ and some $1\leq j\leq k$, then $\widehat{\chi_f}(\Delta_n) = 0$ for sufficiently large n

Proof. Choose n large enough that Δ_n contains every face on k vertices, and so that $f(m+n_i) = f(m) + d$ for $m \geq n - \sum\limits_{i=0}^{k} n_i$. Then,

$$\begin{aligned}
\widehat{\chi_f}(\Delta_n) &= \sum_{F \in \Delta_n} (-1)^{|F|} f(n - \sum_{n_i \in F} n_i) \\
&= \sum_{\substack{F \in \Delta_n \\ n_j \in F}} (-1)^{|F|} f(n - \sum_{n_j \in F} n_i) + \sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|} f(n - \sum_{n_i \in F} n_i) \\
&= \sum_{\substack{F \in \Delta_n \\ n_j \in F}} (-1)^{|F|} (f(n - \sum_{\substack{n_i \in F \\ i \neq j}} n_i) + d) + \sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|} f(n - \sum_{n_i \in F} n_i) \\
&= \sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|+1} (f(n - \sum_{n_i \in F} n_i) + d) + \sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|} f(n - \sum_{n_i \in F} n_i) \\
&= d \sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|}
\end{aligned}$$

Now, note that the above sum is the euler characteristic of a simplicial complex with every face on the vertices $\{n_2, n_3, ..., n_k\}$. In particular,

$$\sum_{\substack{F \in \Delta_n \\ n_j \notin F}} (-1)^{|F|} = 0$$

□

Corollary 8.

If $f(n + p) = f(n) + d$ for some $p \in N$, $d \in \mathbb{Z}$, then

$$(1 - t^p) \sum_{n \in N} \widehat{\chi_f}(\Delta_n) t^n$$

is finite

Proof. We may define a simplicial complex $\Delta_{n\ (p)}$ on the non-minimal generating set $\{n_1, n_2, n_3, ..., n_k, p\}$. Then, $\sum\limits_{n \in N} \widehat{\chi_f}(\Delta_{n\ (p)}) t^n$ is finite. Moreover,

$$\widehat{\chi_f}(\Delta_{n\ (p)}) = \widehat{\chi_f}(\Delta_n) - \widehat{\chi_f}(\Delta_{n-p})$$

So $\sum\limits_{n \in N} \widehat{\chi_f}(\Delta_{n\ (p)}) t^n = (1 - t^p) \sum\limits_{n \in N} \widehat{\chi_f}(\Delta_n) t^n$ □

Note that this proof implies that treating p as a generator will give an expression in terms of $\mathcal{H}$ and χ_M using zl and z.

Lemma 9.

Suppose $f(n)$ satisfies the requirements for Theorem (3) for a numerical semigroup N with minimal generating set $\langle n_1, n_2, ..., n_k \rangle$, and that

$$f(n + p) = f(n) + \sum_{i=0}^{m} \lambda_i n_i$$

for some $p, m, \lambda_0, \lambda_1, ..., \lambda_m \in \mathbb{R}$. i.e. f is quasi-polynomial of degree m. Then,

$$\frac{(1-t^p)^{m+1} \sum_{n\geq 0} \widehat{\chi_f}(\Delta_n)t^n}{(1-t^p)^{m+1} \prod_{i=1}^{k}(1-t^{n_i})}$$

is a finite, rational generating function for $f(n)$.

Proof. It is known that quasi-polynomial sequences of degree m and period p have a generating function of the form

$$\frac{q(t)}{(1-t^p)^{m+1}}$$

For some finite polynomial $q(t)$. Theorem (3) gives us

$$\sum f(n)t^n = \frac{\sum \widehat{\chi_f}(\Delta_n)t^n}{\prod_{i=1}^{k}(1-t^i)} = \frac{q(t)}{(1-t^p)^{m+1}}$$

So that

$$\frac{(1-t^p)^{m+1} \sum \widehat{\chi_f}(\Delta_n)t^n}{(1-t^p)^{m+1} \prod_{i=1}^{k}(1-t^i)} = \frac{q(t) \prod_{i=1}^{k}(1-t^i)}{(1-t^p)^{m+1} \prod_{i=1}^{k}(1-t^i)}$$

Thus, $(1-t^p)^{m+1} \sum \widehat{\chi_f}(\Delta_n)t^n$ is finite. $\square$

Lemma 10.

Suppose $f(n)$ satisfies the requirements for Theorem (3) for a numerical semigroup N and f is eventually quasi-linear of period $p = n_i$ for some i. Then,

(a) Let $\dfrac{q(t)}{(1-t^p)^2} = \sum_{n\in N} f(n)t^n$, then

$$q(t) = \sum_{i=0}^{p-1} \left(f(n) - f(n)t^p\right)t^n - \sum_{n\geq p} \left(-f(n) + 2f(n-p) - f(n-2p)t^p\right)t^n$$

(b) Let $\dfrac{r(t)}{(1-t^p)(1-t)} = \sum_{n\in N} f(n)t^n$, then

$$r(t) = f(0) - f(0)t^p - \sum_{n\geq 1} \left(-f(n) + f(n-p) + f(n-1) - f(n-p-1)\right)t^n$$

Proof. (a)

$$
\begin{aligned}
&(1-t^p)^2\sum_{n\in N} f(n)t^n = (1-t^p)\sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n \\
&= \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^{n+p} \\
&= \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq p}\left(f(n-p)-f(n-p)t^p\right)t^n \\
&= \sum_{n=0}^{p-1}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq p}\left(-f(n)+f(n)t^p+f(n-p)-f(n-p)t^p\right)t^n \\
&= \sum_{n=0}^{p-1}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq p}\left(-f(n)+f(n-p)+f(n-p)-f(n-2p)t^p\right)t^n
\end{aligned}
$$

(b)

$$
\begin{aligned}
&(1-t)(1-t^p)\sum_{n\in N} f(n)t^n = (1-t)\sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n \\
&= \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^{n+1} \\
&= \sum_{n\geq 0}\left(f(n)-f(n)t^p\right)t^n - \sum_{n\geq 1}\left(f(n-1)-f(n-1)t^p\right)t^n \\
&= f(0)-f(0)t^p - \sum_{n\geq 1}\left(-f(n)+f(n)t^p+f(n-1)-f(n-1)t^p\right)t^n \\
&= f(0)-f(0)t^p - \sum_{n\geq 1}\left(-f(n)+f(n-p)+f(n-1)-f(n-p-1)\right)t^n
\end{aligned}
$$

□

5.1 General Comments on χ_f

Lemma 11. *$\widehat{\chi_f}$ and χ_f are Equivalent for Large n*

If $N = \langle n_1, n_2, ..., n_k\rangle$ with $k > 1$, then for $n \gg 0$, $\widehat{\chi_f}(\Delta_n) = \chi_f(\Delta_n)$

Proof.

$$\chi_f(\Delta_n) - \widehat{\chi_f}(\Delta_n) = \sum_{F\in\Delta_n}(-1)^{|F|}|F|$$

For $n \gg 0$, this is equal to

$$\sum_{A\subseteq[k]}(-1)^{|A|}|A| = \sum_{i=0}^{k}\binom{k}{i} i(-1)^i$$

Now, given x

$$(1+x)^k = \sum_{i=0}^{k}\binom{k}{i}x^i$$

With $k > 1$, we may differentiate and put $x = -1$ so that

$$0 = (1-1)^k = \sum_{i=0}^{k} \binom{k}{i} i(-1)^i$$

□

Definition 5.

Put

$$\chi_f^{(0)} = f$$

Then, given $\chi_f^{(i)}$, put $\chi_f^{(i+1)} = \chi_{\chi_f^{(i)}}$

Theorem 4.

Given a function f which satisfies the conditions of Theorem (3) and some $m \geq 0$,

$$\sum f(n)t^n = \frac{\sum \widehat{\chi_f}^{(m)}(\Delta_n)t^n}{\prod_{i=1}^{k}(1-t^{n_i})^m}$$

Proof. The result follows from m applications of Theorem (3) □

Lemma 12.

Given $m \geq 1$, $m!$ divides $\chi_f^{(m)}$ for $n \gg 0$.

Proof. Note that

$$\begin{aligned}\chi_f^{(m)} &= \sum_{F_m \in \Delta_n} (-1)^{|F_m|} \sum_{F_{m-1} \in \Delta_n} (-1)^{|F_{m-1}|} \ldots \sum_{F_1 \in \Delta_n} (-1)^{|F_1|} f(n - c_{F_1}) \\ &= \sum_{F_1, F_2, F_3, \ldots, F_m \in \Delta_n} (-1)^{\sum_{i=1}^{m} |F_i|} f(n - \sum_{i=1}^{k} c_{F_i})\end{aligned}$$

In particular, each term in the above sum appears exactly $m!$ times. □

Notation:
Given n, we may write $\chi_f(n)$ for $\chi_f(\Delta_n)$.

Lemma 13.

Suppose f is a function which satisfies the requirements of Theorem (3) for some Numerical Semigroup $N = \langle n_1, n_2, \ldots, n_k \rangle$ with $k > 1$, and that for $n \gg 0$, f is a quasi-polynomial of degree m and period p such that p is some n_j. Then, for sufficiently large n, χ_f is a polynomial in n, and $\deg\chi_f \leq \deg f - k$.

Proof. We begin by re-labeling $n_1, n_2, n_3, \ldots, n_k$ so that $p = n_1$. Put $N_i = \langle n_1, n_2, \ldots, n_i \rangle$, and write $\chi_{f\ (i)}$ for $\chi_{f\ (N_i)}$

$$\chi_{f\ (1)}(n) = f(n) - f(n - n_1) = f(n) - f(n-p)$$

Now, since f is a quasi-polynomial of period p for sufficiently large n,

$$\deg(f(n) - f(n-p)) \leq \deg f(n) - 1$$

Moreover,

$$f(n) - f(n-p) = f(n) - \left(f(n) - \sum_{i=0}^{m-1} \lambda_i n^i \right) = \sum_{i=0}^{m-1} \lambda_i n^i$$

For some constants $\lambda_1, \lambda_2, ..., \lambda_{m-1}$. In particular, $\chi_{f\ (1)}$ is a polynomial for $n \gg 0$. Now, suppose that for some $i < k$, $\chi_{f\ (i)}$ is a polynomial of degree less than or equal to $\deg f - i$ for $n \gg 0$. Note that

$$\chi_{f\ (i+1)} = \chi_{f\ (i)}(n) - \chi_{f\ (i)}(n - n_{i+1})$$

In particular, $\chi_{f\ (i)}(n) - \chi_{f\ (i)}(n - n_{i+1})$ is clearly still a polynomial, and of degree less than $\chi_{f\ (i)}$ for $n \gg 0$. The proof follows by induction. $\square$

6 Other Quasi-Linear Invariants

6.1 Introduction

It is known that Maximum Factorization Length, Minimum Factorization Length, and size of the Length set are all eventually quasi-linear with a period that is a multiple of one of the generators. This section identifies a few other invariants to which the results of the previous section may be applied

6.2 More Quasi-Linear Invariants

Definition 6. *Given n in some Numerical Semigroup N, let $\eta(n)$ denote the set of factorizations of n in N with maximal length*

Lemma 14.

For $n \gg 0$, $|\eta(n)|$ is periodic and its period divides n_1.

Proof. There exists m_1 such that $n \geq m$ implies $M(n+n_1) = M(n) + 1$. Let

$$m = \max\left(m_1, n_1 \sum_{i=2}^{k} n_i \right)$$

Now, for $n \in N$, let $z(n)$ denote the set of factorizations of n. Given a factorization $F = (a_1, a_2, ..., a_k)$, denote $(a_1, a_2, ..., a_{i-1}, a_i + 1, a_{i+1}, ..., a_k)$ by $F + n_i$. Consider $n \geq m$, and $F \in z(n)$. The $M(n+n_1) = M(n) + 1$ implies that if $F \in \eta(n)$ then $F + n_1 \in \eta(n+n_1)$, and that if $F \notin \eta(n)$ that $F + n_1 \notin \eta(n+n_1)$. The only factorizations $G \in z(n+n_1)$ that may not be written as $F + n_1$ for some $F \in z(n)$ are ones of the form $G = (0, b_2, b_3, ..., b_k)$. However, $n \geq n_1 \sum_{i=2}^{k} n_i$, so $b_i \geq n_1$ for some $2 \leq i \leq k$. Thus, $\widehat{G} = (n_i, b_2, b_3, ..., b_{i-1}, b_i - n_1, b_{i+1}, ..., b_k)$ is also a factorization of $n+n_1$, and the length of $\widehat{G}$ is strictly greater than that of G. In particular, $G \notin \eta(n)$. Thus, for $n \gg 0$, $|\eta(n)| = |\eta(n+n_1)|$. $\square$

Lemma 15.

Let $N = \langle n_1, n_2, n_3 \rangle$. Let

$$a = \frac{n_3 - n_2}{\gcd(n_3 - n_2, n_2 - n_1)}$$
$$c = \frac{n_2 - n_1}{\gcd(n_3 - n_2, n_2 - n_1)}$$
$$b = a + c$$

Then,

$$|\eta(n)| < 1 + \min\left(\frac{1}{b}\frac{n_1}{\gcd(n_1,n_2)}, \frac{1}{c}\frac{n_1}{\gcd(n_1,n_3)}, \frac{1}{c}\frac{n_2}{\gcd(n_2,n_3)}\right)$$

In addition, for some semigroups, this bound is tight.

Proof. Define a trade to be a set of two factorizations of the same element with an empty intersection of support. Suppose $T = \{(a_1, a_2, a_3), (b_1, b_2, b_3)\}$ such that both factorizations have the same length and are non-zero. Since $n_1 < n_2 < n_3$, we must have $T = \{(0, a_2, 0), (b_1, 0, b_3)\}$ with $a_2, b_1, b_3 \neq 0$. Now, we have

$$a_2 = b_1 + b_3$$
$$a_2 n_2 = b_1 n_1 + b_3 n_3$$

So that

$$b_1 n_2 + b_3 n_2 = b_1 n_1 + b_3 n_3$$
$$b_1(n_2 - n_1) = b_3(n_3 - n_2)$$

Now, we want $(b_1, b_3, a_2) = 1$, else the length preserving trade may be created by iterating a smaller trade. Now, since $a_2 = b_1 + b_3$, if any two have a common divisor, the other also has that divisor. In particular, we are looking for solutions b_1, b_3, a_2 that are pairwise relatively prime. Thus, b_1 must divide $n_3 - n_2$ and b_3 must divide $n_2 - n_1$. Note that, given any solution for b_1, b_3, $a|b_1$ and $c|b_3$. Since a and c are also solutions to the b_1 and b_3 respectively, we see that any length preserving trade $\widehat{T}$ may be recreating as iterations of $T = \{(0, b, 0), (a, 0, c)\}$.

Now consider any $F = (a_1, a_2, a_3) \in \eta(n)$. If there are any other factorizations of in $\eta(n)$, we may obtain them by performing the trade T iteratively on F. In particular,

$$\eta(n) = \{(a_1 + ka, a_2 - kb, a_3 + kc) \mid k \in \mathbb{Z}, a_1 + ka, a_2 - kb, a_3 + kc \geq 0\}$$

If F is chosen from $\eta(n)$ so that a_2 is maximal, then we may choose k exclusively from $\mathbb{Z}_{\geq 0}$. Then, the maximal valid k is equal to $|\eta(n)| - 1$. We must have

$$a_2 \geq kb$$

and, since the factorization is maximal in length

$$a_2 < \frac{n_1}{(n_1, n_2)}$$

Thus,

$$b(|\eta(n)| - 1) < \frac{n_1}{(n_1, n_2)}$$

Similarly,

$$a_3 + kc < \frac{n_1}{(n_1, n_3)}, \frac{n_2}{(n_2, n_3)}$$

Since $kc \leq a_3 + kc$,

$$c(|\eta(n)| - 1) < \frac{n_1}{(n_1, n_3)}, \frac{n_2}{(n_2, n_3)}$$

This bound is tight for $N = \langle 9, 10, 23\rangle$. $b = 14$, $\frac{n_1}{(n_1,n_2)} = 9$, so

$$|\eta(n)| < 1 + \frac{9}{14} < 2$$

Since $1 \leq |\eta(n)| < 2$ and $|\eta(n)| \in \mathbb{Z}$, $|\eta(n)| = 1$. ∎

Note: This bound is not tight for $N = \langle 14, 23, 35 \rangle$. For $n \gg 0$, $|\eta(n)| = 1$, but the best bound given by this Lemma is $\eta(n) < 3$.

nb: A family of numerical semigroups which allows for the number of maximal factorizations to be sufficiently large is $N = \langle a, a+1, a+2 \rangle$ where one chooses a to be a large integer. Note that the length preserving trade $(1,0,1|0,2,0)$ must be in this semigroup. Note also that the maximal factorization of an element in the semigroup with value $a-1$ mod a must use at least $\frac{a}{2}$ of n_2 and n_3. In particular, one may apply the above length-preserving trade at least $\frac{a}{2}$ times, so as a increases, the maximal $|\eta(n)|$ also increases.

nb: Another family which restricts the maximal factorization to be unique ($|\eta(n)| = 1 \ \forall_{n \in N}$) is $N = \langle a, 2a+1, 3a+3 \rangle$ with $n > 3$. Note that $(n_1, n_2) = 1$, $(n_3 - n_2, n_2 - n_1) = (a+2, a+1) = 1$, and that $b = \frac{3a+2-a}{(n_3-n_2, n_2-n_1)} = 2a+2 > \frac{n_1}{(n_1,n_2)} = n_1$.

Lemma 16.

Let the Length set $\mathcal{L}(n)$ for $n \in N$ be the set $\{\ell_1(n), \ell_2(n), ..., \ell_j(n)\}$ of lengths of factorizations of n, where $\ell_1(n) > \ell_2(n) > \cdots > \ell_j(n)$. Then, for sufficiently large n, and fixed i, $\ell_i(n + n_1) = \ell_i(n) + 1$.

Proof. Since $\ell_1(n) = M(n)$, it is known that the statement holds for $i = 1$. The proof proceeds by induction. Since $|\mathcal{L}(n)|$ grows eventually quasi-linearly, if N has more than one generator then there exists m_1 such that $n > m_1$ for $n \in N$ implies that $|\mathcal{L}(n)| \geq i$. Suppose that, for some m_2, $n > m_2$ implies that $\ell_{i-1}(n+n_1) = \ell_{i-1}(n) + 1$. Let $m = \max(m_1, m_2)$ and consider the sequence $a_n = \ell_{i-1}(n) - \ell_i(n)$. Clearly, $a_n > 0$. In addition, for $n \geq m$, since there is a factorization of length $\ell_i(n)$, adding n_1 to that factorization gives a factorization of length $\ell_i(n) + 1$ for $n + n_1$. In addition, since $n \geq m$, $\ell_{i-1}(n+1) = \ell_{i-1}(n) + 1 \geq \ell_i(n) + 1$, we must have that $\ell_i(n+n_1) \geq \ell_i(n)+1$. Thus, $a_{n+n_1} = \ell_{i-1}(n+n_1) - \ell_i(n+n_1) \leq \ell_{i-1}(n) - \ell_i(n) = a_n$, and $\{a_n\}$ is a monotonically decreasing sequence of integers which is bounded below, so must be eventually constant. In particular, for $n \gg 0$, $\ell_{i-1}(n) - \ell_i(n) = k$ for some $k \in \mathbb{Z}$, and so we must have $\ell_i(n + n_1) = \ell_i(n) + 1$. □

Lemma 17.

Suppose that, for some $n \in N$, $|\mathcal{L}(n)| \geq i$ for $i \in \mathbb{N}$. Let $\eta_i(n)$ be the set of factorizations of n with length $\ell_i(n)$. For $n \gg 0$, $|\eta_i(n)|$ is periodic, and its period divides p.

Proof. There exists m_1 such that $n \geq m_1$ implies $\ell_i(n + n_1) = M(n) + 1$. Let

$$m = \max\left(m_1, i \cdot n_1 \sum_{i=2}^{k} n_i\right)$$

Now, for $n \in N$, let $z(n)$ denote the set of factorizations of n. Given a factorization $F = (a_1, a_2, ..., a_k)$, denote $(a_1, a_2, ..., a_{i-1}, a_i + 1, a_{i+1}, ..., a_k)$ by $F + n_i$. Consider $n \geq m$, and $F \in z(n)$. The $\ell_i(n+n_1) = \ell_i(n)+1$ implies that if $F \in \eta_i(n)$ then $F + n_1 \in \eta_i(n+n_1)$, and that if $F \notin \eta_i(n)$ that $F + n_1 \notin \eta_i(n + n_1)$. The only factorizations $G \in z(n + n_1)$ that may not be written as $F + n_1$ for some $F \in z(n)$ are ones of the form $G = (0, b_2, b_3, ..., b_k)$. However, $n \geq i \cdot n_1 \sum_{i=2}^{k} n_i$, so $b_i \geq n_1$ for some $2 \leq i \leq k$. Thus, $G^{(1)} = (n_i, b_2, b_3, ..., b_{i-1}, b_i - n_1, b_{i+1}, ..., b_k)$ is also a factorization of $n + n_1$, and the length of $G^{(1)}$ is strictly greater than that of G. Having formed the factorization $G^{(j)}$ such that $j < i$, and the length of $G^{(j)}$ is strictly greater than that of $G^{(j-1)}$ in such a way that for some i_j, $n_{i_j} \geq n_1$, form $G^{(j+1)}$ by $G^{(j)} + (n_{i_j})n_1 - (n_1)n_{i_j}$. This is possible for $j < i$ by the pigeonhole principle, since $n \geq i \cdot n_1 \sum_{i=2}^{k} n_i$. In particular, $G \notin \eta_i(n)$ since there are i $G^{(j)}$s that are factorizations of n and strictly increasing length. Thus, for $n \gg 0$, $|\eta_i(n)| = |\eta_i(n + n_1)|$.

$\square$

Lemma 18.

Given a factorization $F = (a_1, a_2, ..., a_k)$, define $|F|_\infty$ to be $\max_{1 \leq i \leq k}(a_i)$. Then, for $n \in N$, let $|n|_\infty = \max_{F \in z(n)}(|F|_\infty)$. For $n \gg 0$, $|n|_\infty$ is periodic, and its period divides n_1.

Proof. Let $F_i(n)$ be some factorization of n such that it has the maximal number of n_i's out of all factorizations of n. In particular, $|n|_\infty = \max_{1 \leq i \leq k}(|F_i(n)|_\infty)$. Now, for each n_i, and given r mod n_i, consider the set of elements $n \in N$ such that $n \equiv r \bmod n_i$. There exists a least element F_r since this set is bounded below, and so for sufficiently large n, $|F_i(n+n_i)|_\infty = |F_i(n)|_\infty + 1$, since one may obtain F_i by $F_r + kn_1$ for some $k \in \mathbb{N}$. In addition, for large enough n, $|F_1(n)|_\infty > |F_i(n)|_\infty$ for $2 \leq i \leq k$, since $|F_1|_\infty$ grows more quickly. $\square$

Note:
There is strong evidence to the fact that in a Numerical Semigroup N, the Median Factorization length of an element n, which we will denote as $\zeta(n)$, is not eventually quasi-linear, but most likely very close to eventually quasi-linear.
If there are 2n factorizations ordered by length, we define the lower and upper medians to be the n and n + 1 lengths in the list, the median to be their mean. If there are 2n+1 factorizations ordered by length, we define the median, upper median, and lower median to be the n+1 length in the list.
On $\langle 3, 4, 5\rangle$, we tested a large range of periods and found rather strange periods to hold. For instance, every 483 increasing the median by 125 had only 45 non-harmonic points on the set of medians from $n = 0$ to $n = 20,000 - 483$.
In addition, we tested periods ranging from 2 to 1000, and to each assigned the value d such that $\zeta(n+d) - \zeta(n)$ was a constant for the maximal number of values in the range $n = 0$ to $n = 20,000 - period$. Across all of these periods, the values $[1755, 2238, 2721, 4568, 5449, 8660, 11388]$ were non-harmonic in every period.
Even considering lower and upper medians, similarly strange periods occur.

Note: There is relatively convincing evidence that the mode factorization length is eventually quasi-linear. On 3-generated semigroups, a period dividing $(n_3 - n_1)n_2$ sufficed in every example, and on any number of generators an arithmetic progression produced small periods that were clearly eventually quasi-linear.

7 Useful Code

```
load('/home/sage/NumericalSemigroup.sage')


S.<t> = PowerSeriesRing(QQ,default_prec=10000)


"""
```

```
                        giveNumerator(vals, period)

description: Returns the numerator for the generating function of a power series
             whose coefficients are eventually quasi-linear using a denominator
             of (1 - t^period)^2
parameters:  vals: A list containing all the coefficients of the power series
                   before they becomes quasi-linear as well as the first period
                   of quasi-linear values
           period: The period of the quasi-linear coefficients
return value: The numerator for the generating function if the denominator is
              (1-t^period)^2
"""
def giveNumerator(vals, period):
    #The number of coefficients before the coefficients become quasi-linear
    A = len(vals) - period
    #The numerator to be returned
    numerator = 0
    #current "denomionator" is (1-t^period)

    #Sum the first period of quasi-linear coefficients into the numerator
    for i in [A..A+period-1]:
        numerator += vals[i]*t^i

    #Convert the current "denominator" (1-t) * (1-t^period)
    numerator *= (1-t)

    #add to the numerator
    numerator += t^(period+A)

    #Convert the current "denominator" to (1-t^period)^2
    g = 0
    for j in [0..period-1]:
        g += t^j
    numerator *= g

    #add up the terms before the coefficients become quasi-linear
    h = 0
    for k in[0.. A-1]:
        if k in N:
            h += vals[k]*t^k
    #convert them to the proper denominator
    h *= (1-t^period)^2
    #add them to the numerator
    numerator += h

    return numerator



"""
                        prepareMaxVals(N)
```

```
description: Returns a list of the first n_1 * n_k + n_1 maximum factorization
             lengths of values in N (or 0 if not in N) ready to be used
             by the function giveNumerator
parameters:  N: The Numerical Semigroup to compute the maximum factorizations
                in
return value: A list of the first n_1 * n_k + n_1 maximum factorization
             lengths of values in N (or 0 if not in N)
"""
def prepareMaxVals(N):
    #the generators of N
    gens = N.gens
    #the last element whose maximum factorization must be recorded
    lastelem = gens[0] * gens[-1] + gens[0]

    #prepare maximum factorizations and store in vals
    N.LengthSetsUpToElement(lastelem)
    vals = [max(N.LengthSet(i)) if i in N else 0 for i in [0..lastelem]]

    return vals




"""
                convertNumeratorTwotoThree(generators, f)

description: Returns f converted from a numerator over (1 - t^n_1)^2 to the
             equivalent numerator over the product of (1 - t^n_i)^2 over all
             generators n_i
parameters:  generators: The list of all generators (Including n_1)
                     f : The numerator over (1 - t^n_1)^2
return value: The equivalent numerator over the product of (1 - t^n_i)^2 over all
             generators n_i
"""
def convertNumeratorTwotoThree(generators, f):
    numerator = f
    for i in [1..len(generators) - 1]:
        numerator *= (1 - t^generators[i])^2
    return numerator



"""
                        giveEC(N,n)

description: Returns the Euler Characteristic of the Simplicial Complex generated
             by n in N
parameters:  N: The Numerical Semigroup to compute the Simlicial Complex in
             n: The element of whose Simplicial Complex's Euler Characteristic
                will be computed
return value: The Euler Characteristic of the Simplicial Complex generated
```

```
             by n in N (note EC defined as 1 - (f0 - f1 + f2 - f3 + ...))
"""
def giveEC(N,n):
    #list of faces
    faceList = []
    #prepare factorizations
    N.FactorizationsUpToElement(n)
    #for each factorization of n, store the maximal face it encodes in facelist
    for factor in  N.Factorizations(n):
        face = []
        for i in [0..len(factor) - 1]:
            if factor[i] != 0:
                face.append(i)
        faceList.append(face)
    #generate the Simplicial Complex of n from the list of faces
    S = SimplicialComplex(faceList)
    #return our definition of the euler characterisitic
    return 1 - S.euler_characteristic()




"""
                       giveHil(N)

description: Returns the numerator of the Hilbert Series of N over the
             denominator which is the product of (1 - t^n_i) over all
             minimal generators n_i of N
parameters:  N: The Numerical Semigroup to compute the Hilbert Series in
return value: The Euler Characteristic of the Simplicial Complex generated
              by n in N (note EC defined as 1 - (f0 - f1 + f2 - f3 + ...))
"""
def giveHil(N):
    numerator = 0
    generators = N.gens
    N.FactorizationsUpToElement(N.frob)

    #compute sum of generators:
    sumOfGenerators = 0
    for gen in generators:
        sumOfGenerators += gen

    #from 0 to the last potential non-zero coefficient on the numerator
    #(that of N.Frob + the sum of the generators) add each term to the
    #numerator
    for i in [0.. N.frob + sumOfGenerators] :
        if i in N:
            numerator += giveEC(N,i) * t^i

    return numerator
```

```
import itertools
"""
                        giveSumOfFactLen(generators)

description: Returns the numerator of the generating function for the sum of
             lengths of factorizations over all factorizations for a numerical
             semigroup generated by the given generators assuming the denominator
             is the product of (1 - t^n_i)^2 ranging all minimal generators n_i
             of N
parameters:  generators: The minimal generating set of the numerical semigroup
return value: Returns the numerator of the generating function for the sum of
              lengths of factorizations over all factorizations for the given
              numerical semigroup
"""
def giveSumOfFactLen(generators):
    numerator = 0
    #sum over all subsets A of {1,2,...,k} where k is the number of generators
    #(-1)^(|A| - 1) * |A| * t ^ (sum of generators whose indices are in A)
    for i in [1..len(generators)]:
        for subset in itertools.combinations([0..len(generators) - 1], i):
            exp = 0
            for k in subset:
                exp += generators[k]
            numerator += (-1)^(i - 1) * i * t^exp

    return numerator


"""
                       factorByCorrectionTermsToString(N)

description: Returns the numerator of the generating function of the
             maximum length factorization in the semigroup N (assuming
             denominator of the product of (1 - t^n_i)^2 for each minimal
             generator n_i of N) factored by correction terms, i.e.
             in the form zl - \sum t^n*(k * zl + c * z) for some finite n
parameters:  N: The Numerical Semigroup to compute the factorization in
return value: The numerator of the generating function of the maximum length
              factorization in the semigroup N factored by correction terms
              AS A STRING
"""
def factorByCorrectionTermsToString(N):
    #minimal generators of N
    generators = N.gens
    vals = prepareMaxVals(N)
    #period of N
    period = generators[0]
    #Numerator of generating function of max factorization lengths of N
    answer = giveNumerator(vals, period)
    answer = convertNumeratorTwotoThree(generators, answer)
```

```
#The numerator of the generating function for the sum of factorization
#lengths over all factorizations
zl = giveSumOfFactLen(generators)
#the numerator of the generating function for the hilbert series
K = giveHil(N)
#The numerator of the generating function for the number of factorizations
z = 1
for i in generators:
    z *= (1-t^i)

#the current guess as to the best numerator
guess = zl
#a string of the current guess broken up into correction terms
guessStr = "zl"

#For each exponent exp in K = \sum a_n * t^n, compute the correction term
# sgn(a_n) * t^exp * (abs(a_n) * zl + sigmaCk * z) and add the correction
# to guess
for exp in K.exponents():
    if exp > 0:
        coeff = K.padded_list()[exp]
        sign = coeff / abs(coeff)

        if sign == 1:
            guessStr += " + "
        else:
            guessStr += " - "
        guessStr += "t^" + str(exp) + "*(" + str(abs(coeff)) + "*zl + "

        if exp in guess.exponents():
            sigmaCk = sign * (answer.padded_list()[exp] - guess.padded_list()[exp])
        else:
            sigmaCk = sign * answer.padded_list()[exp]

        guessStr += str(sigmaCk) + "*z)"
        guess += sign * t^exp*((abs(coeff)) * zl + sigmaCk * z)

#Now go back through exponents in the answer and guess, calculating similar
#correction terms wherever needed (ie the exponent for t^n in the guess
#                                   is wrong)
#(each correction term written t^n * (k * zl + c * z))
#This is done by noting that z has a non-zero constant term and zl has a
#zero constant term, so calculate the coefficient for z based on t^n
#and calculate the coefficient for zl based on t^(n + n_1)
exp = 0
while exp < max([max(answer.exponents()), max(guess.exponents())]) and \
                                   exp < generators[0] * generators[-1]:
    #coefficient of t^n in answer
    answerCoeff = 0
    if exp in answer.exponents():
        answerCoeff = answer.padded_list()[exp]
```

```
        #coefficient of t^(n + n_1) in answer
        nextAnswerCoeff = 0
        if exp + generators[0] in answer.exponents():
            nextAnswerCoeff = answer.padded_list()[exp + generators[0]]

        #coefficient of t^n in guess
        guessCoeff = 0
        if exp in guess.exponents():
            guessCoeff = guess.padded_list()[exp]

        #coefficient of t^(n + n_1) in guess
        nextGuessCoeff = 0
        if exp + generators[0] in guess.exponents():
            nextGuessCoeff = guess.padded_list()[exp + generators[0]]

        if answerCoeff != guessCoeff:
            c = answerCoeff - guessCoeff
            k = nextAnswerCoeff - nextGuessCoeff + c
            guess += t^exp*(k * zl + c * z)

            #if k is negative factor out the negative for the string
            if(k >= 0):
                guessStr += " + t^" + str(exp) + "(" + str(k) + "*zl + " + str(c) + "*z)"
            else:
                guessStr += " - t^" + str(exp) + "(" + str(-k) + "*zl + " + str(-c) + "*z)"

        #increment exp
        exp += 1

    return guessStr


"""

Prepares a list of the multiplicity of the maximum factorizations for each element in
the semigroup N up until the lastelem where the index of the list is the element and
the value in the list is the multiplicity of that element

"""
def prepareMultiplicityOfMaxVals(N, lastelem):
    #the generators of N
    gens = N.gens
    #the last element whose maximum factorization must be recorded
    vals =[]
    #prepare maximum factorizations and store in vals
    N.FactorizationsUpToElement(lastelem)
    for n in [0..lastelem]:
        if n in N:
            facts=[]
            for fact in N.Factorizations(n):
                sum = 0
                for i in fact:
```

```
                    sum +=i
                facts.append(sum)
            facts = sorted(facts)
            maxx = facts[-1]
            multiplicity = 0
            for length in facts:
                if length == maxx:
                    multiplicity += 1
            vals.append(multiplicity)
        else:
            vals.append(0)
    return vals


"""
                       factorByCorrectionTerms(N)

description: Returns the numerator of the generating function of the
             maximum length factorization in the semigroup N (assuming
             denominator of the product of (1 - t^n_i)^2 for each minimal
             generator n_i of N) factored by correction terms, i.e.
             in the form zl - \sum t^n*(k * zl + c * z) for some finite n
parameters:  N: The Numerical Semigroup to compute the factorization in
return value: The numerator of the generating function of the maximum length
              factorization in the semigroup N factored by correction terms
              AS THE VALUE OF THE NUMERATOR
"""
def factorByCorrectionTerms(N):
    #minimal generators of N
    generators = N.gens
    vals = prepareMaxVals(N)
    #period of N
    period = generators[0]
    #Numerator of generating function of max factorization lengths of N
    answer = giveNumerator(vals, period)
    answer = convertNumeratorTwotoThree(generators, answer)

    #The numerator of the generating function for the sum of factorization
    #lengths over all factorizations
    zl = giveSumOfFactLen(generators)
    #the numerator of the generating function for the hilbert series
    K = giveHil(N)
    #The numerator of the generating function for the number of factorizations
    z = 1
    for i in generators:
        z *= (1-t^i)

    #the current guess as to the best numerator
    guess = zl

    #For each exponent exp in K = \sum a_n * t^n, compute the correction term
    # sgn(a_n) * t^exp * (abs(a_n) * zl + sigmaCk * z) and add the correction
```

```
# to guess
for exp in K.exponents():
    if exp > 0:
        coeff = K.padded_list()[exp]
        sign = coeff / abs(coeff)

        if exp in guess.exponents():
            sigmaCk = sign * (answer.padded_list()[exp] - guess.padded_list()[exp])
        else:
            sigmaCk = sign * answer.padded_list()[exp]

        guess += sign * t^exp*((abs(coeff)) * zl + sigmaCk * z)

#Now go back through exponents in the answer and guess, calculating similar
#correction terms wherever needed (ie the exponent for t^n in the guess
#                                   is wrong)
#(each correction term written t^n * (k * zl + c * z))
#This is done by noting that z has a non-zero constant term and zl has a
#zero constant term, so calculate the coefficient for z based on t^n
#and calculate the coefficient for zl based on t^(n + n_1)
exp = 0
while exp < max([max(answer.exponents()), max(guess.exponents())]) and \
                                   exp < generators[0] * generators[-1]:
    #coefficient of t^n in answer
    answerCoeff = 0
    if exp in answer.exponents():
        answerCoeff = answer.padded_list()[exp]

    #coefficient of t^(n + n_1) in answer
    nextAnswerCoeff = 0
    if exp + generators[0] in answer.exponents():
        nextAnswerCoeff = answer.padded_list()[exp + generators[0]]

    #coefficient of t^n in guess
    guessCoeff = 0
    if exp in guess.exponents():
        guessCoeff = guess.padded_list()[exp]

    #coefficient of t^(n + n_1) in guess
    nextGuessCoeff = 0
    if exp + generators[0] in guess.exponents():
        nextGuessCoeff = guess.padded_list()[exp + generators[0]]

    #if error term, add correction term
    if answerCoeff != guessCoeff:
        if exp == 90:
            print "answerCoeff = ", answerCoeff
            print "guessCoeff = ", guessCoeff
            print "nextAnswerCoeff = ", nextAnswerCoeff
            print "nextGuessCoeff = ",nextGuessCoeff
        c = answerCoeff - guessCoeff
        k = nextAnswerCoeff - nextGuessCoeff + c
```

```
            guess += t^exp*(k * zl + c * z)

        #increment exp
        exp += 1

    return guess


"""
                        isGreatSemigroup(N)

description: Returns true if for the given Numerical Semigroup, the maximal
             length factorization is quasi-linear for all values of n in N, not
             just eventually
parameters:  N: The Numerical Semigroup to check for
return value: Returns true if for the given Numerical Semigroup,
             M(n + n_1) = M(n) + 1 for ALL n in N
"""
def isHarmonicSemigroup(N):
    isHarmonic = True
    #it suffices to check the first n_1 * n_k elements
    generators = N.gens
    n_1 = generators[0]
    n_k = generators[-1]
    N.LengthSetsUpToElement(n_1 * n_k)
    for i in [0..n_1 * n_k]:
        if i in N and i - n_1 in N:
            if max(N.LengthSet(i)) != max(N.LengthSet(i - n_1)) + 1:
                isHarmonic = False

    return isHarmonic


"""

Calculates the polynomial of the numerator for max factorization length using the zl,z form

"""
def calcNumerator(N):
    generators = N.gens
    zl = giveSumOfFactLen(generators)
    z = 1
    for gen in generators:
        z *= (1 - t^gen)

    #store maximum length factorizations for numbers 0 to n_1 * n_k + n_1, 0 if not in N
    maxLengths =  calcMaxLenFact(N)

    numerator = 0

    N.FactorizationsUpToElement(generators[0] * generators[-1])
    for n in [0..generators[0] * generators[-1]]:
```

```
        if n not in N:
            continue
        weightedEulerChar = maxLengths[n]
        #for each face, if m = n minus the sum of generators whose vertices is in the face is in N
        #then add (-1)^(dim(face) + 1) * (maxFact(m) + dim) to the euler characteristic
        for dim in [0..len(generators) - 1]:
            sign = (-1) ^ (dim + 1)
            for subset in itertools.combinations([0..len(generators) - 1], dim + 1):
                sumOfGenerators = 0
                for elem in subset:
                    sumOfGenerators += generators[elem]
                if n - sumOfGenerators in N:
                    weightedEulerChar += sign * (maxLengths[n - sumOfGenerators] + dim + 1)

        numerator += t^n * (giveEC(N,n) * zl + weightedEulerChar * z)

    return numerator



"""

Prints the numerator for max factorization length in string form so it prints
as zl*( ____ ) + z*( ____ )

"""
def calcNumeratorToString(N):
    generators = N.gens
    N.LengthSetsUpToElement(generators[0] * generators[-1])
    zl = giveSumOfFactLen(generators)
    z = 1
    for gen in generators:
        z *= (1 - t^gen)

    K = giveHil(N)

    #store maximum length factorizations for numbers 0 to n_1 * n_k, 0 if not in N
    maxLengths = []
    for i in [0..generators[0] * generators[-1]]:
        if i in N:
            maxLengths.append(max(N.LengthSet(i)))
        else:
            maxLengths.append(0)

    numeratorStr = ""

    for n in [0..generators[0] * generators[-1]]:
        if n not in N:
            continue
        weightedEulerChar = max(N.LengthSet(n))
        #for each face, if m = n minus the sum of generators whose vertices
        #is in the face is in N,
```

```
        #then add (-1)^(dim(face) + 1) * (maxFact(m) + dim) to the euler characteristic
        for dim in [0..len(generators) - 1]:
            sign = (-1) ^ (dim + 1)
            for subset in itertools.combinations([0..len(generators) - 1], dim + 1):
                sumOfGenerators = 0
                for elem in subset:
                    sumOfGenerators += generators[elem]
                if n - sumOfGenerators in N:
                    weightedEulerChar += sign * (max(N.LengthSet(n - sumOfGenerators)) + dim + 1)

        if n in K.exponents():
            zlCoeff = K.padded_list()[n]
        else:
            zlCoeff = 0

        if zlCoeff != 0 or weightedEulerChar != 0:
            numeratorStr += "t^" + str(n) + "*(" + str(zlCoeff) + \
                        "*zl + " + str(weightedEulerChar) + "*z) + "

    return numeratorStr


"""

Will return a list of elements, n, in the semigroup N where M(n+n_1) != M(n) + 1
REMINDER this is essentially the element that maps to the dissonant point

"""
def HarmonicSemigroupViolators(N):
    violators = []
    #it suffices to check the first n_1 * n_k elements
    generators = N.gens
    n_1 = generators[0]
    n_k = generators[-1]
    maxFacts = calcMaxLenFact(N)
    for i in [0..n_1 * n_k]:
        if i in N:
            if maxFacts[i] != maxFacts[i+n_1] - 1:
                violators.append(i)

    return violators


"""

Calculates the weighted euler characteristic on max factorization for elements in a semigroup N
NOTE: this function only works for max factorization

"""
def weightedEC(N,n):
    maxLengths =  calcMaxLenFact(N)
    weightedEulerChar = maxLengths[n]
```

```
    generators= N.gens
    #for each face, if m = n minus the sum of generators whose vertices is in the face is in N,
    #then add (-1)^(dim(face) + 1) * (maxFact(m) + dim) to the euler characteristic
    for dim in [0..len(generators) - 1]:
        sign = (-1) ^ (dim + 1)
        for subset in itertools.combinations([0..len(generators) - 1], dim + 1):
            sumOfGenerators = 0
            for elem in subset:
                sumOfGenerators += generators[elem]
            if n - sumOfGenerators in N:
                weightedEulerChar += sign * (maxLengths[n - sumOfGenerators] + dim + 1)
    return weightedEulerChar




"""

Returns the chi HAT for an element in the numerical semigroup with the invariant f
This function is called by the chiHatNumerator function and rarely used on its own

"""
def chiHatF(N,f,n):
    chiHatF = f(N,n)
    gens=N.gens
    for dim in [0..len(gens) - 1]:
        sign = (-1) ^ (dim + 1)
        for subset in itertools.combinations(gens, dim + 1):
            sumOfGenerators = sum(subset)
            chiHatF += sign * f(N,n-sumOfGenerators)
    return chiHatF




"""

Returns the weighted chi for an element in the numerical semigroup with the invariant f
This function is called by chiNumerator function and rarely used on its own

"""
def chiF(N,f,n):
    chiF = f(N,n)
    gens=N.gens
    for dim in [0..len(gens) - 1]:
        sign = (-1) ^ (dim + 1)
```

```
        for subset in itertools.combinations(gens, dim + 1):
            sumOfGenerators = sum(subset)
            if n - sumOfGenerators in N:
                chiF += sign * (f(N,n-sumOfGenerators) + dim + 1)
    return chiF




"""

Returns the chi HAT numerator on a semigroup N,
with the invariant f, calculated on all elements through the stop

Stop is usually chosen to be large in order to see if the numerator will
eventually zero out, or if it is infinite

"""
def chiHatNumerator(N,f,stop):
    num =0
    #N.LengthSetsUpToElement(stop)
    N.FactorizationsUpToElement(stop)
    for i in [0.. stop]:
        coef = chiHatF(N,f,i)
        num += coef*t^i
    return num




"""

Returns the chi numerator on a semigroup N, with the invariant f,
calculated on all elements through the stop

Stop is usually chosen to be large in order to see if the numerator
will eventually zero out, or if it is infinite

"""
def chiNumerator(N,f,stop):
    num =0
    #N.LengthSetsUpToElement(stop)
    N.FactorizationsUpToElement(stop)
    for i in [0.. stop]:
        coef = chiF(N,f,i)
        num += coef*t^i

    return num
```

```
"""

Calculates the minimum infinite norm for the factorizations of n in the semigroup N

"""
def minInfNorm(N,n):
    if n not in N:
        return 0
    norm = max(N.Factorizations(n)[0])
    for fact in N.Factorizations(n):
        norm = min(norm,max(fact))
    return norm



"""

Calculates the chi for an element, n, in the semigroup N on the invariant of
min infinity norm (this is a slightly modified
weighted chi and therefore needs its own function)

"""
def chiMinInf(N,n):
    chiF = minInfNorm(N,n)
    gens=N.gens
    for dim in [0..len(gens) - 1]:
        sign = (-1) ^ (dim + 1)
        for subset in itertools.combinations([0..len(gens)-1], dim + 1):
            validFactorizations = []
            for fact in N.Factorizations(n):
                validFact = True
                for slot in subset:
                    if fact[slot] == 0:
                        validFact = False
                if validFact:
                    validFactorizations.append(max(fact))
            if len(validFactorizations) > 0:
                norm = min(validFactorizations)
                chiF += sign * norm
    return chiF


"""

Returns the numerator of chi N up until the stop on the invariatn min infinity norm

"""
def chiMinInfNumerator(N,stop):
    num =0
    N.FactorizationsUpToElement(stop)
```

```
    for i in [0.. stop]:
        coef = chiMinInf(N,i)
        num += coef*t^i

    return num



"""

Calculates the mode of the factorization length of each element in the semigroup N
Returns a list where the index is the element and the value is the mode of that element

"""
def calcModes(N,stop):
    modes = []
    N.FactorizationsUpToElement(stop)
    for i in [0..stop]:
        if i not in N:
            modes.append(0)
            continue
        lengths = [0 for j in range(i//gens[0]+gens[0] +1)]

        for fact in N.Factorizations(i):
            lengths[sum(fact)] += 1

        mode = max(lengths)
        modes.append(lengths.index(mode)
    return modes
```

7.1 Example

```
#create Numerical Semigroup on generators
gens = [3,4,5]
N = Numerical Semigroup(gens)

#print whether or not it is harmonic
isGreatSemigroup(N)

#compute \chi_f for the minimum infinity norm
stop = 2000
chiN = chiNumerator(N, minInfNorm, stop)

#compute hilbert series, z, zl, and hilbert series
denom = 1
z = 1
for g in gens:
    denom *= (1 - t^g)^2
    z /= (1 - t^g)

zl = giveSumOfFactLen(gens)
```

```
zl /= denom

hilN = giveHil(N)
hilN *= z

#print generating function for min infinity norm
print zl * hilN + z * chiN
```