

Counting Melodies: Generating Functions, Regular Languages, and the Combinatorics of Pitch, Duration, and Silence

Christopher Yamba

September 2026

Revision 2 — 2026-09-20

Abstract

We study a deliberately small but musically natural enumeration problem: how many monophonic melodies of fixed total duration can be written when notes carry pitch, rests carry no pitch, durations are integer multiples of a smallest temporal unit, and adjacent rests are identified with the single rest obtained by merging them? The model leads naturally to integer compositions, regular languages, and ordinary generating functions. For a pitch alphabet of size p and unrestricted positive integral event durations, the generating function reduces to a rational function

$$M_p(q) = \frac{1 - q}{1 - (p + 2)q + q^2},$$

giving the recurrence $a_n = (p + 2)a_{n-1} - a_{n-2}$ with $a_0 = 1$ and $a_1 = p + 1$. A direct combinatorial formula is also obtained. For a 4/4 bar quantized into sixteen sixteenth-note units and two chromatic octaves ($p = 24$), the model gives 41,006,928,746,285,492,812,225 admissible melodies. We then lift the commutative counting series to a noncommutative formal series that retains individual melodic identity, illustrate the encoding on the subject of Bach's C-minor fugue BWV 847 with engraved scores, ribbon, Young and Ferrers diagrams, and outline a Lean 4 formalization. The purpose of the note is both enumerative and conceptual: to show how a musical notation problem turns into a compact piece of combinatorics.

1. The question

A bar of music looks finite. The staff has a fixed width; a time signature fixes a duration; a chosen smallest note value fixes a temporal grid. Yet even in a monophonic line, the number of possible bars becomes enormous as soon as rhythm, silence, and pitch are allowed to interact.

The point of this note is not to claim that a melody is exhausted by pitch and duration. It plainly is not. Articulation, dynamics, phrasing, harmony, timbre, voice leading, meter, notation, and musical expectation all matter. The point is narrower and, mathematically, cleaner:

Given a fixed amount of time, how many legal monophonic event sequences are possible under a specified set of notation rules?

The problem began from a very concrete representation. Take a 4/4 bar, measure time in sixteenths, and write the bar initially as sixteen atomic temporal units. Grouping consecutive units creates event durations. If order matters, these groupings are compositions rather than partitions. Pitch then colors note events, while rests behave differently: two adjacent rests are not musically distinct from their merged rest in the duration model used here.

That last observation is the small irregularity that makes the problem interesting. Notes may legitimately occur consecutively, even at the same pitch. Rests, in our canonical representation, do not: adjacent rest events are merged.

The resulting structure is simple enough to solve exactly and rich enough to connect three viewpoints:

1. integer compositions;
2. regular languages and formal words;
3. ordinary generating functions.

The same model also has a natural noncommutative refinement in which an individual melody, not merely its count, is retained algebraically.

2. Basic definitions

Fix a smallest temporal unit. In the running example this unit is a sixteenth note.

Let P be a finite pitch alphabet with

$$|P| = p.$$

For two chromatic octaves without duplicating the upper endpoint, for example C4 through B5, one has $p = 24$.

An **event** is one of two kinds:

- a **note event** (x, d) , where $x \in P$ is a pitch and $d \geq 1$ is its duration in atomic units;
- a **rest event** (R, d) , where $d \geq 1$ is its duration.

A **melody** is a finite ordered sequence of events satisfying two conditions:

1. the sum of all event durations is the required total duration n ;
2. no two rest events are adjacent.

The second condition is a canonicalization rule. A rest of duration 2 followed immediately by a rest of duration 3 is represented instead by one rest of duration 5. Thus every sounding pattern of silence has a unique rest decomposition.

We write $a_{n,p}$ for the number of such melodies of total duration n with p available pitches. The empty melody is allowed at duration zero, so

$$a_{0,p} = 1.$$

For $n = 1$, there are p one-unit notes and one one-unit rest:

$$a_{1,p} = p + 1.$$

3. Duration alone: compositions

Before pitch and silence are introduced, consider only durations.

A **composition** of a positive integer n is an ordered sequence of positive integers

$$(d_1, d_2, \dots, d_m)$$

such that

$$d_1 + d_2 + \dots + d_m = n.$$

For a small but musically meaningful example, take $n = 6$: one bar of $3/8$ measured in sixteenths. Its compositions include 6 (a single dotted quarter), $3 + 3$ (two dotted eighths), $2 + 2 + 2$ (three eighths), and $2 + 1 + 3$ (an eighth, a sixteenth and a dotted eighth). There are 32 in all; Figure 2 lists every one of them, both as a sum and as a rhythm in standard notation.

There are 2^{n-1} compositions of n . The familiar proof uses the $n - 1$ internal gaps between n atomic units. At each gap we choose either to retain a separator or remove it. Thus a composition corresponds bijectively to a subset of the set of internal gaps.

Figure 1 makes this bijection visible for $n = 4$. Draw the n atomic units as cells in temporal order. At each internal gap, either the next cell continues the current event, drawn as a step to the right, or a new event begins, drawn as a step straight down. The resulting shape is the **ribbon** (or border strip) of the composition: a connected skew shape with no 2×2 square, whose row lengths, read from top to bottom, are the parts. Up to the choice of drawing convention (English versus French), these are the ribbons that index ribbon Schur functions. The $n - 1$ steps are independently $\rightarrow$ or $\downarrow$, which is the count 2^{n-1} seen at a glance.

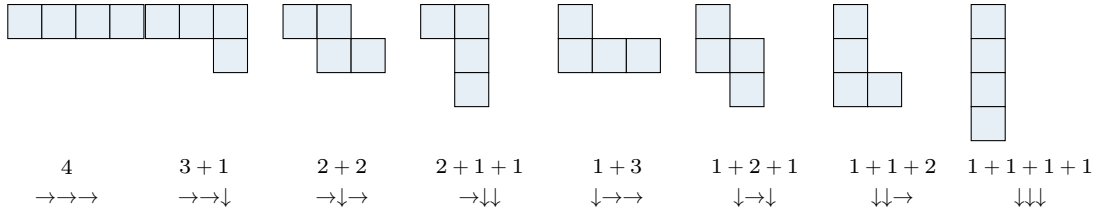


Figure 1: The eight compositions of 4 as ribbons. Each internal gap between units is either a step right (same event continues) or a step down (a new event begins), so ribbons of n cells correspond bijectively to words in $\{\rightarrow, \downarrow\}^{n-1}$.

Musically, the composition is the **duration skeleton**. It says how a fixed time span is partitioned into consecutive events before we decide which events are notes, which are rests, and which pitches the notes carry.

For a $4/4$ bar quantized into sixteenths, $n = 16$, so duration alone gives

$$2^{15} = 32768$$

ordered duration skeletons.

That is only the first layer.

Young and Ferrers diagrams: forgetting the order

Sorting the parts of a composition into weakly decreasing order forgets *when* each duration occurs and keeps only *which* durations occur and how often. The result is a **partition** $\lambda \vdash n$. A partition is drawn as a **Young diagram** (left-justified rows of boxes, longest row on top,

in English convention) or, equivalently, as a **Ferrers diagram** (the same arrangement with dots).

Figure 2 organises all 32 compositions of 6 this way. Each row is one partition, drawn as its Young diagram. The row lists every composition that sorts to that partition, together with its gap word and its notation as one bar of 3/8. A partition with ℓ parts, of which m_k are equal to k , is the image of exactly

$$\frac{\ell!}{\prod_k m_k!}$$

compositions. The eleven partitions of 6 have fibres of sizes 1; 2, 2, 1; 3, 6, 1; 4, 6; 5; 1, which sum to $32 = 2^5$.

The rows are grouped by the number of events m , and the group sizes 1, 5, 10, 10, 5, 1 are the binomial coefficients $\binom{5}{m-1}$. Choosing a composition with m parts means choosing which $m-1$ of the 5 internal gaps keep a separator, so the gap words in group m are exactly those with $m-1$ ones. This is the first factor $\binom{n-1}{m-1}$ of Theorem 2 in Section 7.

The notation column also previews Section 13. Five of the six durations have a single written note value: sixteenth, eighth, dotted eighth, quarter, and dotted quarter. A duration of five sixteenths does not, and must be written with a tie. In the two compositions containing a 5, the tied value is split at the eighth-note pulse: $5+1$ becomes an eighth tied to a dotted eighth, then a sixteenth, and $1+5$ becomes a sixteenth followed by a sixteenth tied to a quarter. If ties were forbidden, the note-duration series would be $A(q) = q + q^2 + q^3 + q^4 + q^6$, and the count would drop from 32 to $[q^6](1 - A(q))^{-1} = 30$.

Musically, the partition λ is the *rhythmic inventory* of a bar, while the composition is its *rhythm*. Transposing a diagram across its main diagonal gives the **conjugate partition** λ' , with

$$\lambda'_k = \#\{i : \lambda_i \geq k\}.$$

This too has a direct musical reading: λ'_k is the number of events lasting at least k units. With sixteenth units, λ'_1 is the number of events, λ'_2 the number lasting at least an eighth, and λ'_4 the number lasting at least a quarter. We use both pictures for the examples in Sections 10 and 12.

4. A formal-language description

Let N denote the class of single note events and R the class of single rest events. Let ε denote the empty word.

A legal melody has a unique decomposition

$$\boxed{(\varepsilon + R)(N(\varepsilon + R))^*}.$$

This expression is worth reading literally.

- $(\varepsilon + R)$ means: either nothing, or one initial rest.
- $N(\varepsilon + R)$ means: one note, optionally followed by one rest.
- The Kleene star $*$ means: repeat that note-plus-optional-rest block zero or more times.

This construction permits

$$N, \quad NN, \quad NR, \quad RN, \quad NRN, \quad RNR, \dots$$

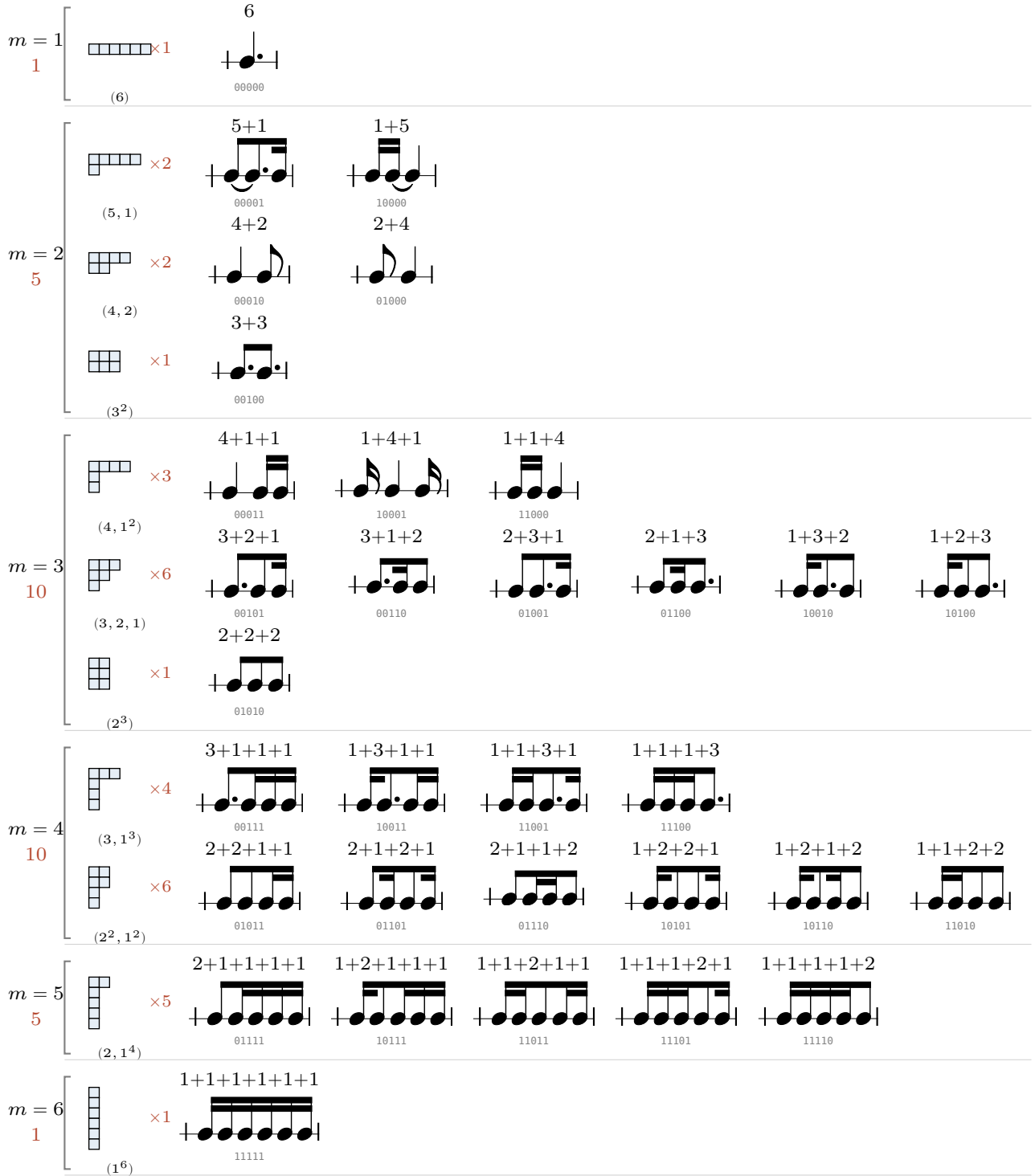


Figure 2: All $2^5 = 32$ rhythms of one $3/8$ bar, measured in sixteenth units. Rows are the 11 partitions of 6, drawn as Young diagrams, with fibre sizes $\ell! / \prod_k m_k!$ in red. Groups collect rows with the same number of events m , with group sizes $\binom{5}{m-1}$ in red. Each cell shows a composition, its rhythm on a one-line staff, and its gap word (1 = separator kept). Within a row, compositions are listed in decreasing lexicographic order.

but never permits RR .

The role of the three appearances of an empty choice should not be conflated. The optional initial rest has an empty alternative; the optional rest after a note has an empty alternative; and the Kleene star has its own zero-repetition case. At the language level the empty object is ε . Only after passing to generating functions does its weight become the scalar $1 = q^0$.

Lemma 1. Unique block decomposition

Every legal melody has a unique representation in the language

$$(\varepsilon + R)(N(\varepsilon + R))^*.$$

Proof. If a melody begins with a rest, remove that initial rest; otherwise choose the empty initial factor. Every remaining event begins with a note. After each note, either the next event is a rest, in which case pair the two, or it is another note or the melody ends, in which case pair the note with the empty choice. Since adjacent rests are forbidden, no rest can belong to two blocks and no rest can be left unpaired. The decomposition is therefore both exhaustive and unique. $\square$

This small lemma is the structural heart of the generating function.

5. The general generating function

The most useful formulation allows note durations and rest durations to have different admissible notation systems.

Let

$$A(q) = \sum_{d \geq 1} \alpha_d q^d$$

be the duration generating function for a **single note before pitch is chosen**, where α_d is the number of allowed written note forms of duration d .

Similarly let

$$B(q) = \sum_{d \geq 1} \beta_d q^d$$

be the duration generating function for a single rest.

The coefficients α_d and β_d are useful when one wants to distinguish notation rather than only sounding duration. For example, if two different permitted spellings of duration 3 are to count separately, then the appropriate coefficient can be 2 rather than 1.

Since each note can have any of p pitches,

$$N(q) = pA(q),$$

while

$$R(q) = B(q).$$

Theorem 1. Melody generating function

Under the canonical rule forbidding adjacent rest events, the ordinary generating function for legal melodies is

$$M_p(q) = \frac{1 + B(q)}{1 - pA(q)(1 + B(q))}.$$

Here $[q^n]M_p(q)$ is the number of legal melodies of total duration n .

Proof. Translate the unique language decomposition of Lemma 1 into generating functions. Disjoint choice becomes addition, concatenation becomes multiplication, and the Kleene star becomes a geometric series:

$$M_p(q) = (1 + R(q)) \sum_{k \geq 0} (N(q)(1 + R(q)))^k.$$

Substituting $N(q) = pA(q)$ and $R(q) = B(q)$ gives

$$M_p(q) = (1 + B(q)) \sum_{k \geq 0} (pA(q)(1 + B(q)))^k.$$

Using $\sum_{k \geq 0} X^k = (1 - X)^{-1}$ as an identity of formal power series,

$$M_p(q) = \frac{1 + B(q)}{1 - pA(q)(1 + B(q))}.$$

The exponent of q records total duration because multiplication adds exponents. Therefore the coefficient of q^n counts exactly the legal melodies of duration n . $\square$

The theorem is more general than the particular one-bar problem. Change the pitch inventory, the temporal unit, the allowed note values, the allowed rest values, or the number of orthographically distinct ways to write a duration, and only p , A , and B change.

6. The unrestricted positive-duration model

The cleanest algebra occurs when every positive integral duration is allowed, with one representation per duration for both notes and rests. Then

$$A(q) = B(q) = q + q^2 + q^3 + \dots = \frac{q}{1 - q}.$$

For a fixed target duration n , this infinite series causes no difficulty: terms of degree greater than n can never contribute to $[q^n]$. Equivalently one may truncate at q^n .

Substituting into Theorem 1 gives

$$M_p(q) = \frac{1 + \frac{q}{1-q}}{1 - p \frac{q}{1-q} \left(1 + \frac{q}{1-q}\right)}.$$

Since

$$1 + \frac{q}{1-q} = \frac{1}{1-q},$$

we obtain the unexpectedly compact form

$$M_p(q) = \frac{1-q}{1-(p+2)q+q^2}.$$

This is the central closed form of the model.

Corollary 1. Recurrence

Writing

$$M_p(q) = \sum_{n \geq 0} a_{n,p} q^n,$$

the coefficients satisfy

$$a_{n,p} = (p+2)a_{n-1,p} - a_{n-2,p} \quad (n \geq 2),$$

with

$$a_{0,p} = 1, \quad a_{1,p} = p+1.$$

Proof. Multiply the series by the denominator:

$$(1 - (p+2)q + q^2) \sum_{n \geq 0} a_{n,p} q^n = 1 - q.$$

Equating coefficients of q^n for $n \geq 2$ gives the recurrence. The coefficients of q^0 and q^1 give the stated initial values. $\square$

For $p = 24$ the recurrence is

$$a_n = 26a_{n-1} - a_{n-2},$$

with $a_0 = 1$ and $a_1 = 25$.

The first few values are

n	$a_{n,24}$
0	1
1	25
2	649
3	16,849
4	437,425
5	11,356,201

n	$a_{n,24}$
6	294,823,801

For a 4/4 bar of sixteen sixteenth units,

$$a_{16,24} = 41,006,928,746,285,492,812,225.$$

This is about 4.10×10^{22} possibilities, or about 75.1 bits of combinatorial choice. As a scale comparison rather than a mathematical ingredient, it is of the same order as the number of molecules in roughly a millilitre of water.

7. A direct combinatorial formula

The generating function is compact, but it is useful to see the same count without any formal power series.

Fix total duration n and suppose a melody contains exactly m events. There are

$$\binom{n-1}{m-1}$$

ways to choose the ordered positive durations of those events, because these are compositions of n into exactly m parts.

Now suppose exactly r of the m events are rests. Since rests may not be adjacent, the number of ways to choose their positions is

$$\binom{m-r+1}{r}.$$

To see this, first place the $m-r$ note events. They create $m-r+1$ gaps: before the first note, between consecutive notes, and after the last note. At most one rest may occupy each gap. Choose r of these gaps.

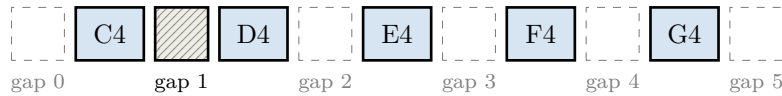


Figure 3: Rest placement for the bar of Section 10: $m = 6$ events, $r = 1$ rest. The $m-r = 5$ notes create $m-r+1 = 6$ gaps, and the rest occupies gap 1. Any choice of r distinct gaps gives a legal pattern, which is where $\binom{m-r+1}{r}$ comes from.

The remaining $m-r$ events are notes. Each has p pitch choices, giving

$$p^{m-r}$$

pitch assignments.

Multiplying the independent choices and summing gives the following result.

Theorem 2. Explicit counting formula

For $n \geq 1$,

$$a_{n,p} = \sum_{m=1}^n \binom{n-1}{m-1} \sum_{r=0}^{\lfloor (m+1)/2 \rfloor} \binom{m-r+1}{r} p^{m-r}.$$

Together with $a_{0,p} = 1$, this counts exactly the same objects as Theorem 1.

Proof. Every legal melody has a unique number m of events and a unique number r of rest events. The first binomial coefficient chooses its ordered positive durations, the second chooses a non-adjacent rest pattern among the m event positions, and the power of p chooses pitches for all remaining events. The decomposition by (m, r) is disjoint and exhaustive. $\square$

This proof is perhaps the most immediately combinatorial one: the object being counted never disappears behind a recurrence symbol.

8. A second combinatorial derivation of the recurrence

The recurrence can also be derived directly from the objects.

Let U_n be the number of legal duration- n melodies whose final event is a note, and V_n the number whose final event is a rest. Then

$$a_n = U_n + V_n.$$

To construct a melody counted by U_n , choose the duration of its final note. If the prefix has duration $j \in \{0, \dots, n-1\}$, then the final note has the uniquely determined duration $n-j$ and one of p pitches. Hence

$$U_n = p \sum_{j=0}^{n-1} a_j.$$

Therefore

$$U_n - U_{n-1} = pa_{n-1}.$$

For a melody ending in a rest, either the whole melody is the single rest of duration n , or a legal melody ending in a note is followed by one rest. Thus

$$V_n = 1 + \sum_{j=1}^{n-1} U_j,$$

so

$$V_n - V_{n-1} = U_{n-1}.$$

Adding the two differences gives

$$a_n - a_{n-1} = pa_{n-1} + U_{n-1}.$$

Applying the corresponding relation one step earlier and using

$$U_{n-1} = U_{n-2} + pa_{n-2}$$

eliminates the auxiliary state U and yields

$$a_n = (p+2)a_{n-1} - a_{n-2}.$$

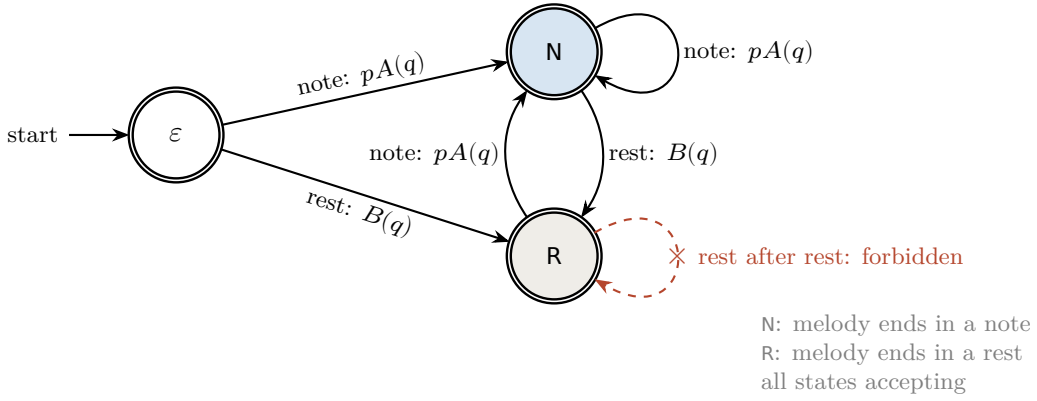


Figure 4: The two-state automaton behind the recurrence. Edges are labelled by the generating function of the event they read. The only missing transition is $R \rightarrow R$: this single forbidden edge is the whole content of the canonical rest rule.

The automaton in Figure 4 also gives a third route to Theorem 1. Writing $N(q)$ and $R(q)$ for the generating functions of nonempty melodies ending in a note and in a rest, the edges translate into the linear system

$$N = pA(1 + N + R), \quad R = B(1 + N), \quad M_p = 1 + N + R.$$

Substituting the first two equations into the third gives $M_p = (1 + B) + pA(1 + B)M_p$, which is Theorem 1 again.

This is a useful proof because the two-state structure is visible: the only memory needed is whether the melody currently ends in a note or a rest.

9. Closed form and growth

Let

$$\Delta = \sqrt{p(p+4)}$$

and

$$\lambda_{\pm} = \frac{p+2 \pm \Delta}{2}.$$

These are the roots of

$$\lambda^2 - (p+2)\lambda + 1 = 0.$$

The recurrence therefore has the closed form

$$a_{n,p} = \frac{p+\Delta}{2\Delta} \lambda_+^n + \frac{\Delta-p}{2\Delta} \lambda_-^n.$$

Since $\lambda_+ \lambda_- = 1$ and $\lambda_+ > 1$ for $p > 0$, the count grows asymptotically like a constant multiple of λ_+^n . For $p = 24$, $\lambda_+ = 13 + 2\sqrt{42} \approx 25.96$, so each additional sixteenth multiplies the number of melodies by about twenty-six, and $\log_{10} a_{n,p}$ is asymptotically a straight line of slope $\log_{10} \lambda_+$ (Figure 5).

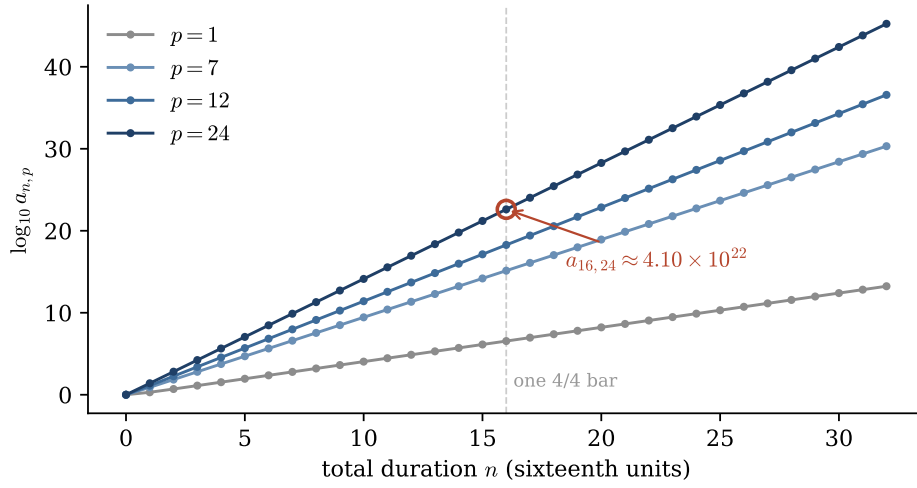


Figure 5: Growth of $a_{n,p}$ on a logarithmic scale for several pitch alphabets. The ringed point is the one-bar, two-octave count $a_{16,24}$.

A particularly pleasing special case is $p = 1$. Then

$$a_n = 3a_{n-1} - a_{n-2}, \quad a_0 = 1, \quad a_1 = 2,$$

which gives

$$1, 2, 5, 13, 34, 89, 233, 610, \dots$$

These are the odd-indexed Fibonacci numbers:

$$a_{n,1} = F_{2n+1}.$$

This is one bridge between the melody model and recent combinatorics-of-rhythm work based on words.

10. One concrete bar

Consider the following one-bar example in 4/4. Durations are measured in sixteenth units.

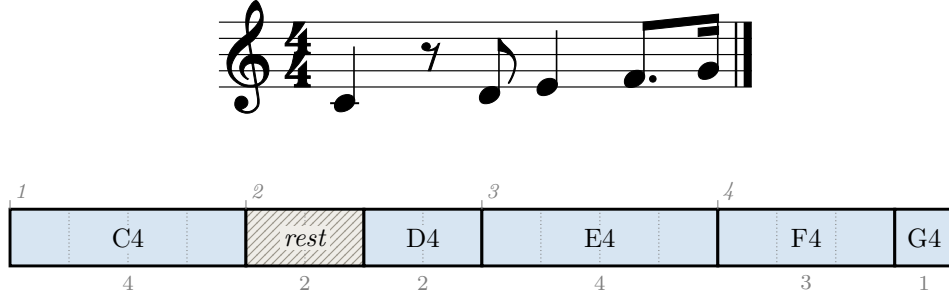


Figure 6: A one-bar toy melody. Top: engraved score. Bottom: the same bar on the sixteenth-note grid. Beat numbers are shown above the grid, durations below, and dotted lines mark the atomic units merged into each event.

Its event representation is

$$(C4, 4), (R, 2), (D4, 2), (E4, 4), (F4, 3), (G4, 1).$$

The durations sum to

$$4 + 2 + 2 + 4 + 3 + 1 = 16.$$

At the purely duration level this is the composition

$$4 + 2 + 2 + 4 + 3 + 1.$$

At the exact-melody level, however, the pitch labels and the position of the rest matter. The generating-function coefficient $[q^{16}]M_{24}(q)$ does not identify this particular melody; rather, it counts this melody together with every other legal melody of the same total duration.

Figure 7 shows the successive shadows of this bar. Its ribbon records the composition exactly (with the rest shaded). Sorting gives the rhythmic inventory $\lambda = (4, 4, 3, 2, 2, 1)$: two quarters, one dotted eighth, two eighths (one of them the rest), and one sixteenth. The conjugate $\lambda' = (6, 5, 3, 2)$ reads: six events, five lasting at least an eighth, three at least a dotted eighth, and two at least a quarter. Exactly $6!/(2!2!) = 180$ compositions of 16 share this inventory.

That distinction motivates a richer algebraic representation.

11. From counts back to individual melodies

The ordinary generating function is intentionally forgetful. It remembers total duration and counts how many objects have that duration, but it does not retain the identity or order of the pitches of one particular melody.

To retain exact melodies, introduce a noncommuting alphabet

$$\Sigma = \{n_{x,d} : x \in P, d \geq 1\} \cup \{r_d : d \geq 1\}.$$

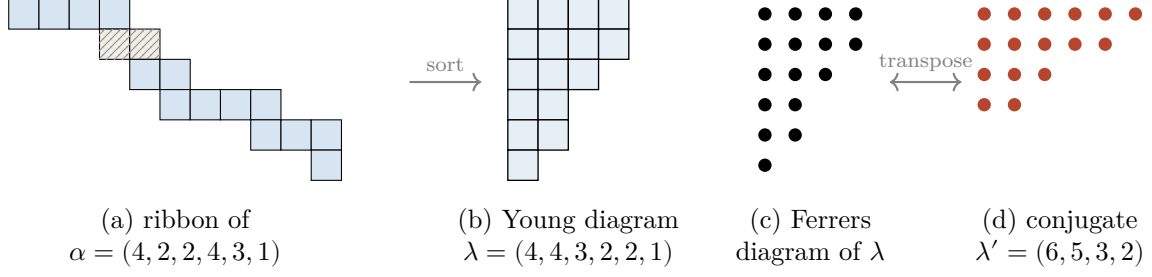


Figure 7: Four views of the rhythm of the toy bar, from most to least informative: (a) the ribbon of its composition, with the rest hatched; (b) the Young diagram of its rhythmic inventory; (c) the same partition as a Ferrers diagram; (d) the conjugate partition, obtained by transposing (c).

Here $n_{x,d}$ means a note of pitch x and duration d , while r_d means a rest of duration d .

Because the symbols do not commute,

$$n_{C,4}r_2n_{D,2} \neq n_{D,2}r_2n_{C,4}.$$

Order is therefore preserved.

Define the noncommutative event sums

$$\mathbf{N} = \sum_{x \in P} \sum_{d \geq 1} n_{x,d}, \quad \mathbf{R} = \sum_{d \geq 1} r_d.$$

Then the entire legal melodic language is represented formally by

$$\boxed{\mathbf{M} = (1 + \mathbf{R})(\mathbf{N}(1 + \mathbf{R}))^*}.$$

An individual melody appears as one word, hence one noncommutative monomial, in this formal series.

Now define a weight map ω by

$$\omega(n_{x,d}) = q^d, \quad \omega(r_d) = q^d.$$

When ω is extended multiplicatively and linearly, all pitch symbols of the same duration collapse onto the same power of q . Summing over the p possible pitches produces the factor p . Thus the ordinary generating function is the commutative image of the exact melodic language:

$$\omega(\mathbf{M}) = M_p(q).$$

Conceptually, this gives a useful hierarchy:

individual melodies $\longrightarrow$ weighted words $\longrightarrow$ counts by duration.

The generating function is not the melody universe itself. It is a deliberately compressed shadow of that universe.

12. Bach's C-minor fugue as an encoding example

A historically familiar example makes the distinction concrete. The subject of J. S. Bach's Fugue in C minor, BWV 847, begins in the middle voice. Figure 8 shows its opening two complete measures.

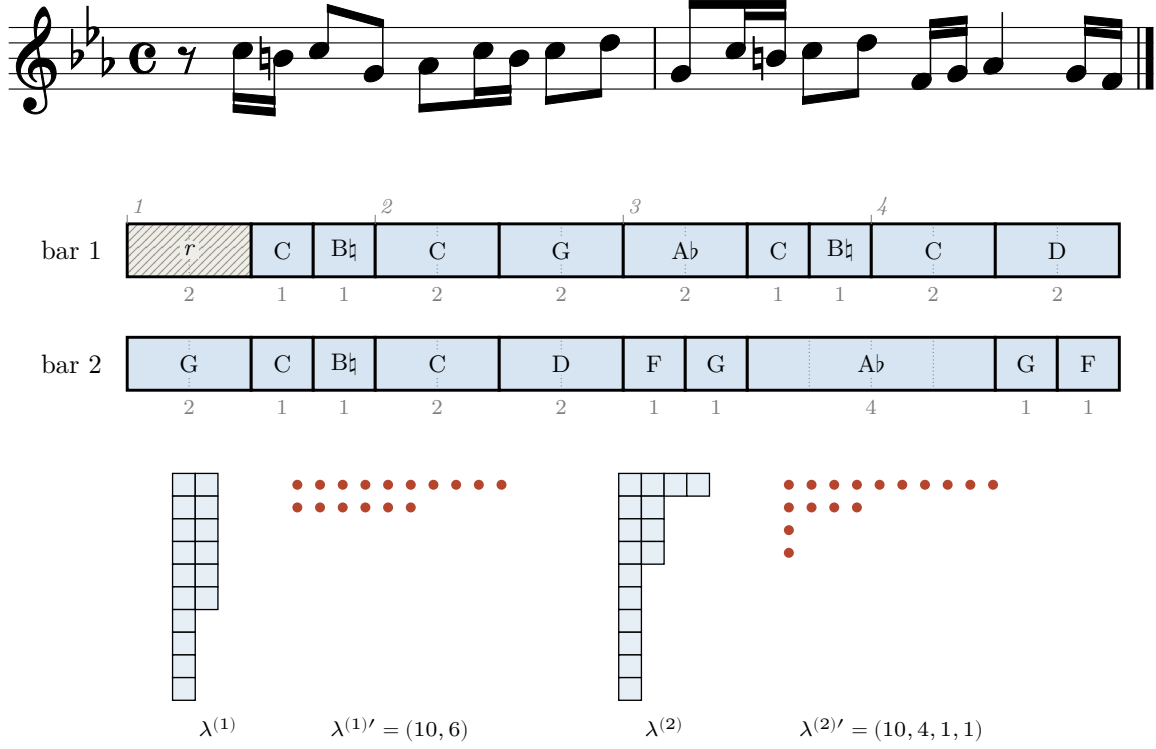


Figure 8: Opening two measures of the subject of J. S. Bach's Fugue in C minor, BWV 847 (*Well-Tempered Clavier*, Book I). Top: engraved score. Middle: the same bars on the sixteenth-note grid. Bottom: Young diagrams of each bar's rhythmic inventory and Ferrers diagrams of their conjugates. The source work is public domain; the event sequence was checked against public-domain/CC score sources.

Suppressing octave register in the text notation for readability, the first complete bar has event word

$$r_2 n_{C,1} n_{B\flat,1} n_{C,2} n_{G,2} n_{A\flat,2} n_{C,1} n_{B\flat,1} n_{C,2} n_{D,2}.$$

Its duration check is

$$2 + 1 + 1 + 2 + 2 + 2 + 1 + 1 + 2 + 2 = 16.$$

The second bar has event word

$$n_{G,2} n_{C,1} n_{B\flat,1} n_{C,2} n_{D,2} n_{F,1} n_{G,1} n_{A\flat,4} n_{G,1} n_{F,1}$$

with duration check $2 + 1 + 1 + 2 + 2 + 1 + 1 + 4 + 1 + 1 = 16$.

The diagrams in Figure 8 separate the two bars even though both have ten events ($\lambda'_1 = 10$ in each case). Bar 1 uses only eighths and sixteenths, $\lambda^{(1)} = (2^6, 1^4)$, so its conjugate $(10, 6)$

has just two columns. Bar 2 contains the syncopated Ab quarter, visible as the single long top row of $\lambda^{(2)} = (4, 2^3, 1^6)$ and as the tail $(\dots, 1, 1)$ of its conjugate $(10, 4, 1, 1)$. The inventories are realised by 210 and 840 compositions of 16 respectively; Bach's rhythms are one choice from each fibre.

As an exact noncommutative word, this bar is one object. Under the duration map ω , the product becomes simply

$$q^{16}.$$

That collapse is not a defect. It is precisely what coefficient extraction needs: every legal exact bar of duration sixteen maps to degree sixteen, and the coefficient records how many distinct exact bars arrived there.

The example also illustrates why the noncommutative lift can be useful computationally. One may retain exact words for motif searching, interval transforms, transposition classes, or pattern avoidance, and only later project onto a coarser generating function.

13. Dots, ties, and written notation

There are two different counting questions that should not be confused.

1. **Sounding-duration model.** A duration of three sixteenths is simply the integer 3.
2. **Written-notation model.** Distinct spellings of the same duration may count as different objects.

For example, a dotted eighth has duration three sixteenths. If the model also allows an eighth tied to a sixteenth and regards that spelling as distinct, then duration 3 has more than one notational realization.

This is exactly why Theorem 1 was stated with coefficients α_d and β_d . If there are α_3 legal note spellings of duration 3, then the note-duration series contains

$$\alpha_3 q^3.$$

The same framework therefore handles conventional note values, dotted notes, selected ties, or other notation systems by changing the duration series rather than rebuilding the combinatorics.

For example, if one permits only undotted powers of two in sixteenth units,

$$A(q) = q + q^2 + q^4 + q^8 + q^{16}.$$

If dotted eighths are added, one adds q^3 ; if dotted quarters are added, one adds q^6 ; and so on. If notes and rests use different conventions, take $A \neq B$.

This separation between **structural combinatorics** and **notation policy** is one of the useful features of the model.

14. Relation to earlier work

The basic combinatorial ingredients are classical. Integer compositions, colored compositions, formal languages, and rational generating functions are established subjects. The musical counting question also has prior treatments.

The earliest antecedent of the present model is the author’s own 2005 manuscript *Combinatorial Mathematical Music Theory* [1]. That work develops music theory from discrete mathematics, taking music to be a mapping from time units to sound, where the sound is either pitch or silence. It already distinguishes the rhythm of a pitch from the rhythm of a silence, which is the note/rest distinction used here; it constructs compositions of n through the bijection with subsets of $[n - 1]$, the same gap argument used in Section 3; and its Section 6.2 treats partitions and compositions of n in connection with the enumeration of melodies, alongside q -analogues, multiset permutations, descent statistics, and Parikh matrices. The present note takes a narrower slice of that programme and pushes it to a closed form: it adds the canonical no-adjacent-rest rule, the resulting regular language, and the rational generating function $M_p(q)$.

A second close antecedent is the 2018 essay *How many melodies are there?* on the blog *Possibly Wrong*. It considers notes and rests in a bar and derives the same structural language idea: an optional rest followed by a sequence of notes each optionally followed by a rest. Its note-duration generating function uses conventional powers-of-two values, while its rest generating function permits arbitrary integral lengths. In its notation,

$$f_n(x) = \frac{1 + h(x)}{1 - g_n(x)(1 + h(x))}.$$

Thus the structural generating-function identity of Theorem 1 should not be presented as a novelty claim.

More recently, Romero-Garcia, Tarsissi, Najman, and Agon (2026) formalized rhythms as words over an alphabet encoding onsets, sustains, and silences. Their regular-language description leads to rational generating functions and, for their rhythm model, the odd-indexed Fibonacci numbers. The specialization $p = 1$ of the present unrestricted-duration model yields the same Fibonacci subsequence, making the relationship especially suggestive.

There is also a broader combinatorial literature on colored and restricted compositions, where parts carry labels or colors and restrictions are imposed on their positions.

Accordingly, the present manuscript is best viewed in its current form as an **expository synthesis and generalization**, with several useful explicit consequences: the compact p -pitch rational function, the recurrence, the direct double-sum formula, the notational multiplicity extension, and the noncommutative exact-melody lift. A stronger claim of journal-level novelty would require a more exhaustive literature search and a precise comparison theorem with existing composition and rhythm models.

15. Toward a bijective proof

The direct formula in Theorem 2 is already combinatorial, while the recurrence follows elegantly from either the generating function or the two-state ending classification.

A stronger aesthetic goal would be a direct bijection explaining

$$a_n + a_{n-2} = (p + 2)a_{n-1}.$$

That identity asks for an explicit correspondence between

$$\mathcal{M}_n \sqcup \mathcal{M}_{n-2}$$

and

$$\{1, \dots, p + 2\} \times \mathcal{M}_{n-1},$$

where $\mathcal{M}_k$ is the set of legal melodies of duration k .

The generating function proves that such sets have the same cardinality, but cardinal equality is not yet an illuminating bijection. Constructing a natural musical bijection - one that explains the two extra states beyond the p pitches without artificial casework - remains a worthwhile refinement of the argument.

This is a good example of a broader principle: an algebraic recurrence may be easy to discover, while the bijection that makes the recurrence feel inevitable can contain the deeper combinatorial insight.

16. A Lean 4 formalization plan

A formal proof assistant can separate two levels of the argument.

First, the algebraic recurrence can be encoded immediately. Second, one can formalize the combinatorial model itself and prove that its cardinalities satisfy that recurrence or equal the explicit binomial sum.

A compact Lean 4 / Mathlib core is as follows:

```
import Mathlib

namespace MelodyCount

/-- Algebraic sequence predicted by the generating function. -/
def a (p : Int) : Nat -> Int
| 0      => 1
| 1      => p + 1
| n + 2 => (p + 2) * a p (n + 1) - a p n

@[simp] theorem a_zero (p : Int) : a p 0 = 1 := rfl

@[simp] theorem a_one (p : Int) : a p 1 = p + 1 := rfl

@[simp] theorem a_rec (p : Int) (n : Nat) :
  a p (n + 2) = (p + 2) * a p (n + 1) - a p n := by
  rfl

-- The concrete two-octave, one-bar value.
example : a 24 16 = 41006928746285492812225 := by
```

```
native_decide
```

```
end MelodyCount
```

This verifies the recurrence as a definition and makes the numerical claim executable. It does **not yet** prove that the sequence counts the musical objects. That bridge is the important formalization task.

A natural next definition is the explicit combinatorial count

```
open scoped BigOperators
```

```
open Finset
```

```
/-- Direct formula from Theorem 2; blueprint. -/
```

```
def directCount (p n : Nat) : Nat :=
```

```
  if n = 0 then 1 else
```

```
     $\sum m \text{ in } \text{Icc } 1 \ n,$ 
```

```
    Nat.choose (n - 1) (m - 1) *
```

```
     $\sum r \text{ in } (\text{range } (m + 1)).\text{filter } (\text{fun } r \Rightarrow 2 * r \leq m + 1),$ 
```

```
    Nat.choose (m - r + 1) r * p ^ (m - r)
```

The central theorem to formalize is then

```
/-- Main enumerative bridge to be proved in Mathlib. -/
```

```
theorem directCount_eq_recurrence (p n : Nat) :
```

```
  (directCount p n : Int) = a p n := by
```

```
  -- Suggested route:
```

```
  -- 1. prove the direct count has initial values 1 and p+1;
```

```
  -- 2. derive the same second-order recurrence using Pascal identities;
```

```
  -- 3. conclude by induction on n.
```

```
  sorry
```

An even more faithful formalization would define an inductive `Event`, define legal melodies as lists with positive durations and a no-adjacent-rest predicate, define total duration recursively, and prove that the finite subtype of duration- n legal melodies has cardinality `directCount p n`. That route is longer but mathematically satisfying because the theorem would connect the recurrence all the way back to the semantic object.

A possible datatype skeleton is

```
inductive Event (Pitch : Type)
```

```
  | note : Pitch -> Nat -> Event Pitch
```

```
  | rest : Nat -> Event Pitch
```

```
open Event
```

```
def duration : Event Pitch -> Nat
```

```
  | note _ d => d
```

```
  | rest d   => d
```

```
def isRest : Event Pitch -> Bool
```

```
  | rest _   => true
```

```
  | note _ _ => false
```

One would then require positive durations, define `totalDuration`, and formalize the prohibition of adjacent rests. The eventual theorem would assert that for a finite pitch type of cardinality p , the cardinality of legal melodies of total duration n equals $a_{n,p}$.

This division of labor is useful: the handwritten proof explains the mathematics; Lean checks that no case or algebraic identity has been smuggled past us.

17. Computational realization

The mathematics suggests a very small software tool.

An event can be represented as a pair

`(pitch-or-rest, duration)`

such as

`(C4, 4), (R, 2), (D4, 2), (E4, 4), (F4, 3), (G4, 1).`

A parser from MusicXML, MIDI plus quantization, or a small hand-entered notation can map a bar into this sequence. From the same structure one can compute three different representations:

- the exact ordered event word;
- the noncommutative monomial;
- the commutative duration weight q^n .

One can then reverse the perspective and enumerate or sample legal words from the grammar. More interestingly, additional quotient maps can identify melodies up to transposition, inversion, rhythmic scaling, or other equivalence relations. The algebraic representation therefore becomes a bridge from elementary counting toward computational music analysis.

18. Scope and limitations

The model intentionally ignores many musical distinctions. Among them are:

- polyphony and chords;
- articulation, dynamics, timbre, and phrasing;
- tuplets and irrational subdivisions unless explicitly inserted into the duration alphabet;
- ties as independent notation unless counted through α_d and β_d ;
- enharmonic spelling;
- pitch-range constraints that depend dynamically on previous notes;
- tonal, harmonic, stylistic, or perceptual constraints;
- equivalence under transposition, inversion, retrograde, or temporal scaling.

These are not defects so much as directions in which the model can be refined. The virtue of Theorem 1 is modularity: many refinements can be implemented by replacing the event generating functions or by replacing the regular language.

The most important conceptual distinction is between **possibility** and **musical plausibility**. A coefficient can count every legal string under a formal grammar; it says nothing by itself about which strings sound coherent, memorable, beautiful, or stylistically meaningful.

19. Conclusion

The original problem sounds almost recreational: count the melodies in a bar. But it quickly reveals a clean mathematical anatomy.

Duration produces compositions. Pitch labels the note parts. Silence introduces a local exclusion rule. That exclusion rule is naturally a regular language. The language becomes a rational generating function. In the unrestricted positive-duration model the rational function collapses to

$$M_p(q) = \frac{1-q}{1-(p+2)q+q^2},$$

and the entire enumeration is governed by

$$a_{n,p} = (p+2)a_{n-1,p} - a_{n-2,p}.$$

For $p = 24$ and $n = 16$, this gives

$$41,006,928,746,285,492,812,225$$

one-bar melodies under the stated rules.

The number itself is entertaining. The structure that produces it is the more interesting result. The same object can be viewed as a composition, a constrained colored word, a regular language, a rational generating function, a recurrence, a binomial sum, or a noncommutative formal series containing each individual melody as a word.

That multiplicity of viewpoints is exactly the sort of thing that makes a small combinatorial problem worth keeping. One begins with a bar of music and ends with a piece of algebra that can be unfolded, generalized, mechanized, and - if one wishes - played.

References

1. C. Yamba, *Combinatorial Mathematical Music Theory*, unpublished manuscript, 2005. Develops musical sets and mappings for pitch and silence, melodies as permutations of multisets, descent and Parikh statistics, and (Section 6.2) partitions and compositions of n in the enumeration of melodies.
2. P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, 2009. See especially the symbolic method and rational languages.
3. “Possibly Wrong,” *How many melodies are there?*, 24 April 2018. <https://possiblywrong.wordpress.com/2018/04/24/how-many-melodies-are-there/>
4. G. Romero-Garcia, L. Tarsissi, L. Najman, and C. Agon, “Counting Rhythms Using Combinatorics on Words,” in *Mathematics and Computation in Music - MCM 2026*, LNCS/LNAI 16609, pp. 225-238, 2026. DOI: 10.1007/978-3-032-27827-2_13.
5. R. Sachdeva and A. K. Agarwal, “Combinatorics of certain restricted n -color composition functions,” *Discrete Mathematics* 340(3), 361-372, 2017. DOI: 10.1016/j.disc.2016.09.009.
6. J. S. Bach, *The Well-Tempered Clavier, Book I*, Fugue in C minor, BWV 847. Public-domain score sources include IMSLP; the subject transcription used for checking the example is also available through Wikimedia Commons.

Note on status and revisions

This manuscript is a first draft and is undergoing ongoing work. Additional ongoing work includes an improved version with sharpening generalization, the noncommutative lift, equivalence-class enumeration, bijective proofs, and more formal verification directions. The improvements will be published as things are added or revised through some coherent steps, and will be posted with dates.

- **Revision 1 — 2026-09.** First draft: model, generating function, recurrence, explicit formula, noncommutative lift, Lean 4 plan.
- **Revision 2 — 2026-09-20** (current). Added engraved score examples, the ribbon/Young/Ferrers picture of compositions, the full catalogue of the 32 rhythms of a 3/8 bar, the two-state automaton and its linear system, the growth plot, and the 2005 reference.