COUNTING PARSES*

ALAN PRINCE
Typohedron.org

Metrical theory allows for a rich if finite variety of ways that a string of syllables can be parsed by feet. We deploy a couple of different techniques to determine the number of parses admitted under various structural assumptions.

In so doing, we are led to effective procedures for constructing the entire set of parses. Since a claim of optimality refers to an entire candidate set, and not just to a handful of obvious competitors, these procedures provide a starting point for establishing the truth of any such claim in the realm of foot-level prosody.

1 At Issue

How many metrical parses are there for a string of *n* syllables?

1.1 Going Meta

Why would a linguist ask or seek to answer such a question, with no immediate empirical consequences in sight? No quick advantage to be claimed over a competitor? Curiosity is sufficient motive for some, as is an interest in exactitude. Both play a driving, behind-the-scenes role in investigations of all kinds: yesterday's math is tomorrow's physics, and vice versa; enough, even felt from a distance, to rattle the disciplinary cage. On the empirical side, the field's growth is a history of influx. The work of the dedicatees of this volume, central to modern prosodic theory, brings into play a diversity of illuminations coming from a vast, searching, and sometimes even playful exploration of phenomena and ideas that often lie well beyond the canonical foci and sources.

In the case at hand, we will find that asking a question about the theory, purely because of its formal interest, can lead us to useful insights or tools that can shape our understanding of the things we want to understand.[1] As an encouragement to the venture, very little specialized math is needed to reach the answer; all that's required is persistence with the familiar slightly beyond the bounds of familiarity.

1.2 Optimal

A candidate is optimal if there is nothing better in its candidate set.[2] *Nothing*. To establish optimality, then, requires that we control every candidate in the set.[3] Vast infinities of candidates may vanish at a glance, through harmonic bounding arguments. For example, Prince & Smolensky (1993/2004, ch. 6), in studying the Basic Syllable Theory system, quickly reduce all candidate sets to finitude by establishing the (few) conditions under which epenthetic material can appear in optimal forms.

But as Tesar has reminded us from time to time, and as this example shows, infinity is often the easy part.[4] The twists, imperspicuities, and surprisingly large numbers that arise from finite combinatorics

* Thanks to Brett Hyde for valuable suggestions; to Paul Smolensky, Naz Merchant, and an anonymous reviewer for useful comments; to Bruce Tesar, Jane Grimshaw, Paula Houghton and Sara O'Neill for comments and general discussion.

[1] This line is promoted in "The Pursuit of Theory" (Prince 2007).

[2] See "What is OT?" (Prince 2016) for a recent treatment, and Prince & Smolensky 1993/2004, ch. 5 for the first.

[3] It's a fact that the literature is not replete with arguments to the effect that claimed optima are in fact optimal. But this does not lessen the need: live by the heuristic, die by incomprehension. Theories, if not theorists, are remarkably immune to assertions of personal belief. On showing optimality, see Prince & Smolensky (1993/2004, ch. 7).

[4] Qualitatively speaking, one might conjecture that this is so because reaching infinity typically requires a kind of uniformity of structural possibilities that leads to the availability of broad generalizations.

can be daunting.[5] In some cases, it may be necessary to contend directly with exhaustive lists of candidates; and, even when broad generalizations *exist*, it may well be useful to have exhaustive lists to ponder as a lead-in to finding those generalizations.

To answer the *how many parses* question, we will construct a way (indeed: ways) to produce the exhaustive list of parses. We examine these methods of construction to determine the number of forms they generate. But it is only a matter of a change in perspective to be able to use these methods to generate the forms and thereby provide the analyst with the desired fodder for analysis.

1.3 The Parses

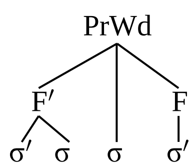
We work with an unremarkable conception of prosodic structure.[6] Feet are bisyllabic or monosyllabic, and do not overlap.[7] A licit metrical parse, for our purposes, is a PrWd consisting of a sequence of feet and unfooted syllables.

Any number of syllables may be left unfooted, including all of them. Every foot has one head; and one and only one foot may be distinguished as the head or (in a stress system) the main-stress-bearing constituent of the PrWd. For simplicity, we will refer to the head of a foot as a 'stress' and the head of the head foot as the 'main stress', bypassing questions of realization. We will occasionally abbreviate 'syllable' by σ.

The term *unit* will be used here to refer to any child node of a PrWd: a foot or an unfooted syllable; and used only to refer to those entities.

Here's an example of our assumptions and usage:

(1) A 4σ parse



The English word 'perigrinate' provides an instance. This parse has three units: two feet (of which the first is bisyllabic, the second, monosyllabic) and one unfooted syllable. In this illustration, we portray headship by marking a head category C as C′: hence F′ (head of PrWd) and σ′ (head of foot). This parse is of *length* 4. We reserve the term *length* to measure the size in syllables of the string being parsed.

In building the argument, we will proceed analytically from the simpler to the more complex by introducing distinctions into previously analyzed classes that lack them.

We separate out the Quantity Insensitive (QI) systems, in which metrical terminals (syllables) are treated as being metrically equivalent. These contrast with Quantity Sensitive (QS) systems, in which a relevant binary distinction exists between syllable types.[8] This move is fully justified because the QS parse count can be derived from the more basic QI count.

We also recognize a class of systems *with **no main** stress* (NM) where all feet are prominentially nondistinct, with the head of the PrWd ignored. We distinguish these from systems where the head of the PrWd is attended to; these are *systems recognizing **main** stress* (M). This move is analytically justified because the count of M(ain) systems can be determined directly from the count of N(o)M(ain) systems.

[5] In Harmonic Serialism, for example, candidate sets are strictly finite, but the plenitude and complexity of the derivations will (in certain perfectly ordinary cases) defeat some current software (Mullin et al., 2010, §1.2:7–11). The HS package in OTWorkplace (Prince et al 2007-2018) aims to adhere closely to the basic premises of the theory.

[6] Of course, it was remarkable at certain points in recent history and derives from inter alia Liberman 1975, Prince 1976, and more proximately Selkirk 1980 and Hayes 1981.

[7] Hyde 2002 et seq. finds a number of striking properties in an overlapping foot theory.

[8] For the terms abbreviated by QS and QI, and much else, see Hayes 1995.

The course of analysis will run from QI/NM, the simplest class, which honors the fewest structural distinctions, to QI/M, and thence to QS/NM and QS/M.

1.4 Strategies of Enumeration

We use two different strategies for enumerating parses, which we will name idiosyncratically: the *method of continuations*, and the *method of arrangements*. The first has a bottom-up flavor; the second, top-down.

- The method of continuations asks this question: given a (partially completed) structure, how many ways can we continue it one syllable further?[9]
- The method of arrangements asks: given that a parse has a certain number of units, how may we arrange them to form licit structures?

1.5 Preview of the Counting Results

Using the method of continuations, we will determine that $P_{NM}(n)$, the number of No Main QI parses of n syllables, $n > 0$, is as follows, where round(x) denotes 'the nearest integer to x':

(2) QI/NM
$$P_{NM}(n) = \text{round}\left(\frac{(1+\sqrt{3})^{n+1}}{2\sqrt{3}}\right)$$

Using the method of arrangements, we will find another expression for the same quantity, in which we write U for the number of units in the parse, B for the number of binary feet, and use the notation $[n/2]$ to mean 'the largest integer less than or equal to $n/2$'.

(3) QI/NM
$$P_{NM}(n) = \sum_{B=0}^{[n/2]} 2^U \binom{U}{B}$$

We'll see that $U = n - B$, and since we fix n, this relation will allow us to compute with equation (3).

Equation (3) uses the *binomial coefficient*, which has this interpretation:

(4) Binomial coefficient
$$\binom{U}{B} = \frac{U!}{B!(U-B)!}$$

This counts the number of ways of choosing B things out of a collection of size U, and hence would often be read 'U choose B'.[10]

---

[9] See Riggle 2004 for major development of the finite state machine idea of which this is an instance.

[10] Qualitatively speaking, the factor $B!$ shows up in the denominator because we don't care about the order of choosing the B things. Similarly, we don't care about the order of the things we don't choose, hence the appearance of $(U-B)!$. This entity is called the 'binomial coefficient' because it appears when we expand the expression $(1+x)^n$ as a sum of terms involving some number of times x^k, $0 \le k \le n$: that number is n-choose-k. This is because each x^k term arises from the choice of an element, either 1 or x, from each of the n factors in the product $(1 + x) \times \ldots \times (1 + x)$. We have to choose k x's and $n - k$ 1's to get x^k. Each such choice gives us one term x^k. The number of ways to do this is the total number of x^k terms we get, and this is just the number of ways we can choose k things from n possibilities.

Using the method of arrangements, we determine that the number of QI parses containing a main stress, $P_M(n)$, is as follows:

(5) QI/M

$$P_M(n) = \frac{n}{2} P_{NM}(n)$$

Quantity sensitive totals are obtained by noting that each QI parse of a string of *n* syllables gives rise to 2^n QS parses, since each QI syllable independently yields two QS syllables (light/heavy).

(6) QS

$$P_{QS}(n) = 2^n P_{QI}(n)$$

We will encounter various other expressions of interest along the way. In the end, the methods of arriving at these formulas may be of more interest than the formulas themselves.

1.6 By the Numbers

We close the preliminaries with a glance at the resulting numerics.

(7) Quantities of Parses

Sylls	QI No Main	QI w/ Main	QS No Main	QS w/ Main
1	2	1	4	2
2	6	6	24	24
3	16	24	128	192
4	44	88	704	1,408
5	120	300	3,840	9,600
6	328	984	20,992	62,976
7	896	3,136	114,688	401,408
8	2,448	9,792	626,688	2,506,752
9	6,688	30,096	3,424,256	15,409,152
10	18,272	91,360	18,710,528	93,552,640
11	49,920	274,560	102,236,160	562,298,880
12	136,384	818,304	558,628,864	3,351,773,184
13	372,608	2,421,952	3,052,404,736	19,840,630,784
14	1,017,984	7,125,888	16,678,649,856	116,750,548,992
15	2,781,184	20,858,880	91,133,837,312	683,503,779,840

Two things to note:

1. The ‘w/ Main’ category reckons only those parses that *actually have* a main-stressed syllable; footless forms are not included in this count. We amplify below, in §5.

2. The QS counts aggregate over all possible QS inputs, thereby summing all possible faithfully-parsed output candidates from any QS input string whatever. Each QI length has, of course, only one input, whereas under QS, for a string of *n* syllables, we have 2^n distinct inputs, namely all length-*n* sequences over {light, heavy}. See §6 below.

The rate of growth in the QI sector settles down so that each successive length provides approximately 2.7 times the number of parses of its predecessor. The QS sector ultimately grows at about twice this rate.

Given any OT system, of course, the *total number* of violation-distinct optima in any candidate set—forms that can be optimal under some ranking—is limited by the interactions of the constraint system, regardless of the number of candidates. It will therefore be capped, and must stop growing, even though the total number of parses grows, nay *explodes*, with candidate length. For example, a QI/NM version of the system studied in Alber 2005, with seven constraints, has just 9 even-length possible optima and 14 of odd-length, for any length above three syllables (Alber & Prince 2008, in prep). Indexing these findings against the table, we note that whereas about 20% of the length-4 candidates are optimal in some system, a mere 0.0005% of length-15 forms are. This forcefully illustrates the fact that, even in systems like this, where each candidate set is finite, almost all forms are harmonically bounded. And it highlights, on the one hand, the tremendous power of a constraint system to exclude, and on the other, the remarkable effectiveness in parsing obtained by researchers like Tesar 1995 and Riggle 2004.

2 Counting NM Parses by Arrangements

Let's begin with the method of arrangements, which is conceptually akin to the hierarchical way of thinking about metrical constituency and which uses familiar techniques to do its counting. We'll then move to the method of continuations, which yields a very simple and practical generation scheme.

A syllable string is exhaustively parsed into units, each of which is a foot or unfooted syllable. Consider all metrical parses that contain *U* units: how many of these are there? To answer, we need to distinguish the number of binary units, *B*, from the number of monosyllabic units, *M*. The total number of units is merely their sum:

(8) $U = M + B$

What we want to know first is how many distinct ways a collocation of $U = M + B$ units may be linearly arranged. This is simply a matter of taking *U* sequential units and choosing *B* of them to binary: *U*-choose-*B*, the binomial coefficient (see fn. 10 for a brief characterization), whose definition we repeat here:

(9) Number of ways of choosing *B* things out of *U* things.

$$\binom{U}{B} = \frac{U!}{B!(U-B)!}$$

Next, we ask how many distinct full structures there are on *U* units, distinguishing x ('unstressed syllable') from X ('stressed syllable'). Observe that each binary unit comes in two varieties, which we notate -Xx- and -xX-; and each monosyllabic unit comes in two varieties, which we notate -x- and -X-. With two independent choices for *each* unit, whether binary or monosyllabic, there are 2^U full parses for each distinct collocation of *U* units. Putting these observations together:

(10) Number of parses with *U* units, *B* binary: $2^U \binom{U}{B}$

To make use of this, we need to be able to go through the parses of a length-*n* string, classified by the number of units each parse contains. That is: we need to relate *U* to *B* and *n*. Straight from the definition of *M* and *B*, we have that the number of syllables must equal the number of monosyllabic units plus twice the number of bisyllabic units:

(11) $n = M + 2B$
$= (M + B) + B = U + B$

Rearranging, we have

(12) $U = n - B$

Observe that the number of binary feet in the parse of a length-*n* string runs from a minimum of zero, with all units monosyllabic, to [*n*/2], the greatest integer less than or equal to *n*/2, obtained when we deploy as many binary units as possible. (For example, a five-syllable string can host a maximum of two binary feet.) Putting this together with equations (10) and (12), we arrive at the desired expression for the total number of parses:

(13) QI/NM

$$P_{NM}(n) = \sum_{B=0}^{[n/2]} 2^U \binom{U}{B}$$

$$= \sum_{B=0}^{[n/2]} 2^{n-B} \binom{n-B}{B}$$

Let's do an explicit calculation for length 5, noting that [5/2] = 2.

(14) QI/NM: length 5

$$\begin{aligned} P_{NM}(5) &= \sum_{B=0}^{[5/2]} 2^{5-B} \binom{5-B}{B} \\ &= 2^5 \cdot \binom{5}{0} + 2^4 \cdot \binom{4}{1} + 2^3 \cdot \binom{3}{2} \\ &= 32 \cdot 1 + 16 \cdot 4 + 8 \cdot 3 \\ &= 32 + 64 + 24 \\ &= 120 \end{aligned}$$

3 Counting NM Parses by Continuations

For purposes of analysis, we introduce a convenient notation that refers to the structure of constituency and headship. We choose '||' as the edge-marker over '-'and '.' for reasons of visibility.

Unstressed syllable	x
Stressed syllable	X
Main-stressed syllable	Y
Unit edge marker	||

Here are some examples of usage:

‖Xx‖Xx‖	Two binary trochaic feet, No Main
‖Xx‖Yx‖	Two binary trochaic feet, of which the second is the PrWd head
‖x‖x‖X‖X‖	Two unfooted syllables followed by two monosyllabic feet, No Main
‖x‖x‖Y‖X‖	Ditto footwise, except the penultimate foot is the head of the PrWd
‖Yx‖x‖X‖	Example (1) above

NB: We are treating the *unfooted syllable* as a demarcated unit which is notationally on a par with a monosyllabic foot: ‖x‖ *vs.* ‖X‖, ‖Y‖.

To enrich to QS, when the time comes, we can regard x, X, and Y as denoting *light* syllables, and use h, H, and K to denote their *heavy*-syllable cognates, as in OTWorkplace's built-in systems.

The vocabulary of characters used to encode QI/NM parses has three members: {X, x, ‖}, of which the first two represent syllables. Generation starts from the symbol "‖". Assume that we have built all strings of length $n - 1$ syllables ending in one of these three characters. Let's consider how any such string may be continued, advancing to strings of length n syllables. (We work arbitrarily left-to-right.)

(15) **Table of Continuations**

	IN: Ends in	OUT: May be continued as	Yields: A string ending in
[1a]	… x	X‖	bisyllabic iamb
[1b]		‖x	unstressed syllable
[1c]		‖X	stressed syllable
[2a]	… X	x‖	bisyllabic trochee
[2b]		‖x	unstressed syllable
[2c]		‖X	stressed syllable
[3a]	… ‖	x	unstressed syllable
[3b]		X	stressed syllable

Examples:

- The 3σ parse ‖Xx‖x can continue one syllable further in three ways:
  - by [1a] to ‖Xx‖x**X‖**
  - by [1b] to ‖Xx‖x**‖x**
  - by [1c] to ‖Xx‖x**‖X**
- The first 4σ parse ‖Xx‖xX‖ can continue in two ways:
  - by [3a] to ‖Xx‖xX‖**x** or
  - by [3b] to ‖Xx‖xX‖**X**, and in no other ways.
- And so on.

Representing the continuations in this way has four useful properties:

a) The output continuations end only in symbols mentioned in the inputs.

b) Continuation advances by exactly one syllable.

c) The footing status of x is left open at stage $n-1$; it is determined at the next stage.
d) We may stop at any time and have a complete parse.[11]

Property (c) permits us to look at the *single* characters at the very end of the stage $n-1$ input. We never need to examine the footing status of a final x, which would require us to know what characters precede and follow it.

In this scheme, for non-0 lengths, a *final* "||" marks the end of a binary foot. Monosyllabic feet are demarcated at the next step, when there is a next step; or by quitting, leaving them final in the string.

The continuations therefore fall into two classes:

- those ending in the unit-boundary marker "||", indicating the end of a binary foot
- those ending in a syllabic symbol, x or X

Let's write b(n) for the number of parses of length n ending in the boundary marker ('b-parses'), and s(n) for the number of parses ending in a syllable character x or X ('s-parses').

Writing $P_{NM}(n)$ for the number of parses on n syllables, our first observation is simply that this quantity is the sum of the number of s-parses and the number of b-parses.

(16) $P_{NM}(n) = s(n) + b(n)$

Less trivially, an examination of table (15) discloses that there is exactly one b-parse of length n syllables for each s-parse of length $n-1$ syllables. These are shown in [1a] and [2a] of the table.

(17) $b(n) = s(n-1)$

From this, it follows that solving for either s(n) or b(n) will solve the whole problem. Returning to the table, we observe that each s-parse on length $n-1$ leads to *two* s-parses on length n, as shown in rows [1b,c] and [2b,c].

(18) a. $s(n) = 2\,P_{NM}(n-1)$
b. $= 2\,s(n-1) + 2\,b(n-1)$ from equation (16)
c. $= 2\,s(n-1) + 2\,s(n-2)$ from equation (17)

We have now obtained a linear recurrence relation that defines the value of s at length n in terms of its values at lengths $n-1$ and $n-2$. This kind of relation has a unique solution, once we fix its two initial conditions, the values of s(0) and s(1). The length 0 string has just one parse , namely "||" in the notation, to start continuation off properly, which does not end in a syllable; therefore s(0) = 0. The length 1 string has two parses, namely as an unfooted syllable "||x" and as a monosyllabic foot "||X", by rules [3a] and [3b]. We have exactly the following problem to solve: find the function s meeting these conditions:

(19) $s(n) = 2\,s(n-1) + 2\,s(n-2)$
$s(0) = 0$
$s(1) = 2$

[11] If we stop by just ceasing to continue, a final unit may be explicitly demarcated by "||" as in the example ||Xx||xX||, or not, as in ||Xx||x. This orthographic inhomogeneity is irrelevant to the counting project. To fix it in implementation, we need merely add a stopping step which affixes the edge-marking character || when necessary. This step does not affect the number of parses.

The usual methods[12] yield the following solution:

(20) QI/NM $$s(n) = \frac{(1+\sqrt{3})^n - (1-\sqrt{3})^n}{\sqrt{3}}$$

From equations (16) and (17), we have the following:

(21) $P_{NM}(n) = s(n) + b(n)$
$= s(n) + s(n - 1)$

As Paul Smolensky notes, equation (18)c gives us, by dividing out the 2 on its right-hand side:

(22) $$s(n) + s(n - 1) = ½ s(n + 1)$$

Thus from equations (21) and (22), we obtain

(23) $$P_{NM}(n) = ½ s(n + 1)$$

With equation (20) in hand, this yields a closed-form expression for the total number of QI parses with no main stress on a syllable string of length *n*:

(24) QI/NM $$P_{NM}(n) = \frac{(1+\sqrt{3})^{n+1} - (1-\sqrt{3})^{n+1}}{2\sqrt{3}}$$

Observe that the subexpression

(25) $$\frac{(1-\sqrt{3})^{n+1}}{2\sqrt{3}}$$

is small for $n = 1$, at approximately 0.155, and only gets smaller as *n* increases. For all values of $n > 0$, it cannot carry us far from the integer we are seeking. Therefore, we arrive at the following, using the function 'round(x)' to deliver 'the closest integer to x'.

(26) QI/NM $$P_{NM}(n) = \text{round}\left(\frac{(1+\sqrt{3})^{n+1}}{2\sqrt{3}}\right) \qquad n > 0$$

Returning to the full unrounded result in equation (24), we note that expressions of the form

(27) $$\frac{(1+x)^n - (1-x)^n}{x}$$

[12] See, for example, "Recurrence Relation" in Wikipedia if you want to work by hand; or search on "recurrence relation" to find a solver and take your pick. Finding the solution requires no more than solving a quadratic equation and a pair of linear equations. Another linguistic application to a prosodic theory is found in Prince 1993. The commenter "d**a**/dt" worries whether expression (20) always provides integer values. Observe (or accept) that expression (20) does satisfy the recurrence relation for s(*n*). It agrees with initial conditions s(0) = 0 and s(1) = 2, which completely determine all further values, which are thus integers. The same conclusion follows from a calculation like that outlined in exx. (27)-(28).

are ripe for simplification via expansion of the numerator's terms by the binomial theorem. Clearly, the constant terms and all terms containing x^{2k} will drop out and all the surviving numerator terms will contain x^{2k+1}, both parenthesized numerator terms in the above contributing one such, which will simplify, when divided by x, to a term containing x^{2k}. This is convenient when $x = \sqrt{3} = 3^{1/2}$, and the final result looks like this:

(28) QI/NM $$\mathrm{P}_{\mathrm{NM}}(n) = \sum_{k=0}^{\left[\frac{n+1}{2}\right]} 3^k \binom{n+1}{2k+1}$$

As above, we write $[q]$ for the greatest integer less than or equal to q, and we take the value of the binomial coefficient to be zero when the lower number exceeds the upper.

One might not imagine from looking that equation (28), with its powers of 3 from the method of continuations, and equation (13), with its powers of 2 from the method of arrangements, come to the same thing. But they both count identical sets, so we can be confident that they do.

To get a sense of the way this formula plays out, let's recalculate the length-5 example:

(29) QI/NM: length 5

$$\begin{aligned}
\mathrm{P}_{\mathrm{NM}}(5) &= \sum_{k=0}^{\left[\frac{5+1}{2}\right]} 3^k \binom{6}{2k+1} = \sum_{k=0}^{3} 3^k \binom{6}{2k+1} \\
&= 3^0 \cdot \binom{6}{1} + 3^1 \cdot \binom{6}{3} + 3^2 \cdot \binom{6}{5} + 3^3 \cdot \binom{6}{7} \\
&= 1 \cdot 6 \quad + 3 \cdot 20 \quad + 9 \cdot 6 \quad + 27 \cdot 0 \\
&= 6 + 60 + 54 + 0 \\
&= 120
\end{aligned}$$

4 Main Stress

Now that we have expressions for the number of mainstressless QI parses on an arbitrary syllable string of length n, we inquire as to the status of the next level of complexity: metrical parses containing a single head foot (giving us the *main stress* when footheads are interpreted as stresses).

Here's the result:

(30) QI/M $$\mathrm{P}_{\mathrm{M}}(n) = \frac{n}{2} P_{\mathrm{NM}}(n)$$

To show that this is correct, let us associate each parse π with (what we will call) its X/x-dual, $\bar{\pi}$, which is obtained from π by switching every x for X and every X for x. The X/x-dual swaps iamb and trochee, monosyllabic foot and unfooted syllable, uniformly throughout the string. Consider the entire collection DP(n) of dual *pairs* of parses of length n syllables, writing $\Pi(n)$ for the set of individual QI parses on a length-n string.

(31) Dual pairs $\mathrm{DP}(n) = \{ \{ \pi , \overline{\pi} \} | \ \pi \in \Pi(n) \}$

We make four observations:

I. $\bigcup \mathrm{DP}(n) = \Pi(n)$.
II. For every $\pi \in \Pi(n)$, there is exactly one $\delta \in \mathrm{DP}(n)$ such that $\pi \in \delta$.
III. There are ½ $P_{NM}(n)$ elements in $\mathrm{DP}(n)$.
IV. Each element $\{ \pi , \overline{\pi} \} \in \mathrm{DP}(n)$ contains a total of n X's.

To establish the last, consider any pair $\{ \pi , \overline{\pi} \}$ and say π contains k X's and $n-k$ x's, $k \geq 0$. Then $\overline{\pi}$ contains $n - k$ X's. Sum across the pair to obtain the total of $k + (n-k) = n$ X's.

To generate the totality of main stress possibilities, take each *pair* and produce from it all individual parses in which one of the X's in one of its members has been replaced by a Y. Each pair then produces exactly n parses with one syllable identified as the main stress. Take this with observation III, and we obtain the result claimed in equation (30).

(32) QI/M $$\mathrm{P}_{\mathrm{M}}(n) = n\left|\mathrm{DP}(n)\right| = \frac{n}{2} P_{\mathrm{NM}}(n)$$

This method of counting reckons only with those parses that contain at least one stress. If we include parses without feet, we add for each input exactly one parse with no stresses at all. Call the number of these inclusive parses $P_{M+\varnothing}(n)$. We have:

(33) *All* QI/M parses $$\mathrm{P}_{\mathrm{M}+\varnothing}(n) = \frac{n}{2} P_{\mathrm{NM}}(n) + 1$$

5 QS, All Types

Each QI parse, under either the NM or M regimes, blows up to a set of QS parses by taking each syllable independently to be either light or heavy. Since there are two independent choices for each of the n syllables in a length-*n* parse, we get the following counts:

(34) $P_{NM/QS}(n) = 2^n P_{NM/QI}(n)$

(35) $P_{M/QS}(n) = 2^n P_{M/QI}(n)$

(36) $P_{M+\varnothing/QS}(n) = 2^n P_{M+\varnothing/QI}(n)$

Observe that this covers all the possibilities of QS parses: no new groupings, or assignments of stressed/unstressed status, are made available when the quantity distinction is imposed. Recall that in the QS case we are lumping all parses together from every possible QS input.

6 Generative Schemes

The counting strategies can be turned into procedures that produce the parses.

6.1 *QI Generation*

The method of continuations can be put to use quite directly. Let K_1, K_2,… be sets of output parses, where K_n is based on input of length-*n* syllables. Let's set up K_1 by hand:

(37) K_1 = { ||x, ||X }

We notate carefully, so as to feed properly in the continuation recipe.
Now we iterate through this set, examining the final symbol of each parse, storing for each of them *all* of its licit continuations, following the recipe of table (15) above.[13]

(38) 1σ parses to 2σ parses

-x → ||x**X**||
||x||**x**
||x||**X**

-X → ||X**x**||
||X||**x**
||X||**X**

This gives us the six possible parses on a length-2 input. Continue in this fashion, iterating through each of these six, producing the continuations, and we'll get the 16 length-3 parses; and so on.

Let's lay out the results for the first half of the length-3 set:

(39) 2σ parses to 3σ parses (half)

a. ||xX|| → ||xX||**x**, ||xX||**X**

b. ||x||x → ||x||**xX**||, ||x||x||**x**, ||x||x||**X**

c. ||x||X → ||x||X**x**||, ||x||X||**x**, ||x||X||**X**

The remaining half, we see, consists of the X/x-duals of these forms.

6.2 *QS Generation: Copy & Change*

The basic problem here is to take a sequence of *n* identical characters and produce the full set of sequences in which each character freely takes on one of two distinct forms.

[13] Akers 2008 is the first work to convert the counting scheme of table (15) into a candidate generator.

Here's one way to do it. For purposes of illustration, let's take T and F as our two basic characters. Suppose we have a list containing a sequence of 3 T's: ⟨TTT⟩. The following procedure will generate every sequence of length 3 over {T,F}.

1a. Copy the list and attach it to the original, producing:

TTT
TTT

1b. Turn all *first* characters *in the copy* to their opposite value:

TTT
FTT

2a. Copy this whole list, and attach it to itself:

TTT
FTT
TTT
FTT

2b. Turn all *second* characters in the *copy* to their opposite value:

TTT
FTT
T***F***T
F***F***T

3a. Copy *this*, and attach:

TTT
FTT
TFT
FFT
TTT
FTT
TFT
FFT

3b. Now turn all third characters in the copy to their opposite value:

TTT
FTT
TFT
FFT
TT***F***
FT***F***
TF***F***
FF***F***

In this method, there are *n* steps for a length-*n* string. We start out at step 1 with a one-element list containing a single length-*n* string.

On the m^{th} step, we copy the result L_{m-1} of the $(m-1)^{th}$ step and subjoin the copy to the original, creating a list of the form $L_{m-1} + L_{m-1}$. Then we change each character in the m^{th} serial position in each string *of the copy* to its opposite (non-initial) value. That's it.

We will certainly want to obtain all faithful prosodic parses from a given input; in this case, the input must have the same quantitative profile as all of its output parses. To generate, we must therefore change the *input* and everything in its output-set appropriately and simultaneously. So we apply the method to a list structure that joins the input with its QI parses.

To illustrate, let's construct the QS parses on all inputs of length 2. In the table below, the start row contains the input /xx/ and its parses. The final block (2b) lists all QS inputs of 2 syllables in length, from /xx/ to /hh/. Each is associated with a row that contains all its parses.

This requires two steps of copy & change. We write ch(*k*) for the procedure changing the k^{th} character in the copy. We replace "||" with ".".

Start: QI parses	L_0	xx	.x.x.	.x.X.	.X.x.	.xX.	.Xx.
1a. Copy L_0	$L_0 + L_0$	xx	.x.x.	.x.X.	.X.x.	.xX.	.Xx.
		xx	*.x.x.*	*.x.X.*	*.X.x.*	*.xX.*	*.Xx.*
1b. Change copy	L_1: ch(1)	xx	.x.x.	.x.X.	.X.x.	.xX.	.Xx.
		hx	.**h**.x.	.**h**.X.	.**H**.x.	.**h**X.	.**H**x.
2a. Copy L_1	$L_1 + L_1$	xx	.x.x.	.x.X.	.X.x.	.xX.	.Xx.
		hx	.h.x.	.h.X.	.H.x.	.hX.	.Hx.
		xx	*.x.x.*	*.x.X.*	*.X.x.*	*.xX.*	*.Xx.*
		hx	*.h.x.*	*.h.X.*	*.H.x.*	*.hX.*	*.Hx.*
2b. Change copy	L_2: ch(2)	xx	.x.x.	.x.X.	.X.x.	.xX.	.Xx.
		hx	.h.x.	.h.X.	.H.x.	.hX.	.Hx.
		x**h**	.x.**h**.	.x.**H**.	.X.**h**.	.x**H**.	.X**h**.
		h**h**	.h.**h**.	.h.**H**.	.H.**h**.	.h**H**.	.H**h**.

We are now fully equipped to go all the way from a starting point "||", using the method of continuations to produce QI parses of whatever length we desire and then, by means of the copy & change procedure, to produce the full panoply of QS parses. Parses marked for main-stress can be produced by working through the NM parses, iteratively selecting every X or H for promotion.

6 Concluding Remarks

Deploying techniques to solve a natural formal question—*how many parses*?—has led us to simple, effective methods for *constructing* the parses in their entirety. These methods enable the analyst to conduct sound analysis. We gain knowledge not only of parsing numerics, but also of the entire range of structures implied by our structural assumptions.

References

Alber, B. 2005. Clash, lapse, and directionality. *NLLT* 23.3: 485–542.

Alber, B. and A. Prince. 2008. newBASTo: Analysis of an asymmetrical stress system. Ms. Rutgers, The State University of New Jersey and the University of Verona.

Alber, B. and A. Prince. In prep. *Typologies*. Ms. University of Verona and Typohedron.org.

Akers, C. 2008. Constituent-based alignment and the QI stress typology. Ms. Rutgers, The State University of New Jersey.

Hayes, B. 1980. *A metrical theory of stress rules*. PhD dissertation, MIT. Revised version, Garland Press, 1985.

Hayes, B. 1995. *Metrical stress theory*. University of Chicago Press: Chicago.

Hyde, B. 2002. A restrictive theory of metrical stress. *Phonology* 19: 313–339.

Liberman, M. 1975. *The intonational system of English*. PhD dissertation, MIT.

Mullin, K., B. W. Smith, J. Pater, and J. J. McCarthy. 2010. OT-Help 2.0 UserGuide. Ms. University of Massachusetts, Amherst.

Prince, A. 1976. Applying stress. Ms. UMass, Amherst.

Prince, A. 1993. In defense of the number *i*: analysis of a linear dynamical model of linguistic generalizations. RuCCS-TR-1.

Prince, A. 2007. The pursuit of theory. In P. de Lacy, ed. *The Cambridge handbook of phonology*, 33–60.

Prince, A. 2009. RCD—The movie. ROA-1057.

Prince, A. 2016. What is OT? ROA-1271.

Prince, A. & P. Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Blackwell. ROA-537.

Riggle, J. 2004. Generation, recognition, and learning in finite state Optimality Theory. PhD dissertation, UCLA. To appear revised as *Computing Optimality*, OUP.

Selkirk, E. O. 1980. The role of prosodic categories in English word stress. *Linguistic Inquiry* 11: 563–605.

Tesar, B. 1995. *Computational Optimality Theory*. PhD dissertation, University of Colorado at Boulder. ROA-90.