Counting Parses [1]

Alan Prince, Rutgers University
8-24-2010

0 At Issue

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

0.1 Going Meta

Why would a linguist ask or seek to answer such a question, there being no immediate empirical consequences in sight? No quick advantage to be claimed for a favored theory? Idle curiosity is sufficient motive for some.[2] Beyond that, we might hope that asking a fundamental formal question, purely because of its formal interest, would lead us to useful insights or tools that can shape our understanding of the things we want to understand.[3]

0.2 Optimal

A candidate is optimal if there is *nothing better* in its candidate set.[4] To establish optimality, then, requires that we control *every* candidate in the set.[5] 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, 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, infinity is often the easy part.[6] The twists, imperspicuities, and surprisingly large numbers that arise from finite combinatorics can be daunting.[7] In some cases, it may be necessary to contend directly with exhaustive lists of candidates; and, even when broad generalizations *exist*, it may be well 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] Thanks to Brett Hyde and Naz Merchant for valuable suggestions, and to Paul Smolensky, Bruce Tesar, Jane Grimshaw, and Sara O'Neill for useful comments and general discussion.

[2] The author admits to membership in this group.

[3] A line that is promoted in "The Pursuit of Theory" (Prince 2007).

[4] See Prince & Smolensky 1993/2004, ch.5, and for a recent re-telling, Prince 2009, sheet "Optimality Defined."

[5] 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).

[6] 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.

[7] 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 current software (Mullin et al., 2010, §1.2, 7–11).

0.3 The Parses

We work with an unremarkable conception of prosodic structure.[8] Feet are bisyllabic or monosyllabic, and do not overlap.[9] A licit metrical parse, for our purposes, is a PrWd (*Prosodic Word*) 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

```
     PrWd
    /  |  \
   F′  |   F
  / \  |   |
σ′  σ  σ   σ′
```

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 parse-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.[10] 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. These we distinguish from systems where the head of the PrWd is attended to: *systems recognizing main stress* (M). This move is analytically justified because the count of M systems can be determined directly from the count of NM systems.

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.

[8] 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 1980.
[9] Hyde 2002 finds a number of striking properties in an overlapping foot theory.
[10] For the terms abbreviated by QS and QI, and much else, see Hayes 1995.

0.4 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?[11]

- The method of arrangements asks: given that a parse has a certain number of units, how may we arrange them to form licit structures?

0.5 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
$$P_{NM}(n) = \mathrm{round}\left(\frac{(1+\sqrt{3})^{n+1}}{2\sqrt{3}}\right)$$

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 eq. (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'.[12]

---

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

[12] 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 the sum of terms $a_k x^k$, each of which involves some number a_k times x^k, $0 \le k \le n$: that number a_k is n-choose-k. This is so because any single x^k arises from the selection 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 x^k. The number of ways to make the choice 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.

0.6 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 §4.

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 §5 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 approximately twice this rate.

Given any OT system, of course, the *total number* of violation-distinct optima in any candidate set—forms 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). Indexing these findings against the table, we note that whereas about 20% of the length-4 candidates are optimal in some language, a mere 0.0005% of length-15 forms are. And so on, downhill all the way. This forcefully illustrates the fact that, even in systems like those studied here , where each candidate set is finite, almost all forms are harmonically bounded. And it highlights the tremendous power of a constraint system to exclude, as well as the remarkable effectiveness in plucking optima from the formal welter of possibilities obtained by the diverse computational methods of Tesar 1995 and Riggle 2004.

1 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?[13] 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. 12 for a brief characterization), whose definition we repeat here:

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

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

[13] Where have all the syllables gone? No worries: we'll re-connect with them shortly.

Next, we ask how many distinct full structures there are on U units, distinguishing among the types of units. Observe that each binary unit comes in two varieties, $(\sigma'\sigma)_F$ and $(\sigma\ \sigma')_F$, and each monosyllabic unit comes in two varieties, $(\sigma')_F$ and unfooted σ. With two independent choices for *each* unit, whether binary or monosyllabic, there are 2^U full parses for each distinct sequence 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) $n = M + 2B$

Subtracting eq. (11) from eq. (8) and rearranging, we obtain

(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 eqs. (10) and (12), we arrive at the desired expression for the total number of parses:

(13) QI/NM

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

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

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

(14) QI/NM: length 5

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

2 Counting NM Parses by Continuations

For purposes of analysis, we introduce a convenient notation that refers to the structure of constituency and headship:

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

Here are some examples of usage:

ǁXxǁXxǁ	Two binary trochaic feet, No Main
ǁXxǁYxǁ	Two binary trochaic feet, of which the second is the PrWd head
ǁxǁxǁXǁXǁ	Two unfooted syllables followed by two monosyllabic feet, No Main
ǁxǁxǁYǁXǁ	Ditto footwise, except that the penultimate foot is the head of the PrWd
ǁYxǁxǁXǁ	Example (1) above

NB: in accord with our descriptive assumptions, we notate the *unfooted syllable* as a demarcated unit which is structurally on a par with a monosyllabic foot: ǁxǁ *vs.* ǁXǁ, ǁYǁ.

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.

The vocabulary of characters used to encode QI/NM parses has three members: {X, x, ǁ}, of which the first two represent syllables. 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 continue with:	Yielding 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 would continue via clause 1a to the 4σ parse ‖Xx‖xX‖.
- This 4σ parse can continue to 5σ in only two ways:
  - ‖Xx‖xX‖x (by 3a)
  - ‖Xx‖xX‖X (by 3b).

Representing the continuations in this manner has several useful properties:

(a) The output continuations end only in symbols mentioned in the inputs.
(b) Continuation advances by exactly one syllable.
(c) The foot status of final x is left open at stage $n-1$, and determined at the next stage.
(d) We may stop at any time and have a complete parse.[14]

Property (c) permits us to look at the *single* character lying at the right edge of the stage *n* 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, a *final* "‖" turns out to mark the end of a binary foot. Monosyllabic feet are demarcated at the next step, when there is one; 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').

[14] 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 and will be ignored. To fix it, we need merely add a stopping step which affixes the edge-marking character ‖ when necessary.

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

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 rows 1a and 2a.

(17) $\mathsf{b}(n) = \mathsf{s}(n-1)$

From this, it follows that solving for either $\mathsf{s}(n)$ or $\mathsf{b}(n)$ will solve the whole problem. Returning to the table, we observe that each parse of length $n - 1$ leads to *two* s-parses of length *n*, as shown in rows 1bc, 2bc, 3ab.

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

We have now obtained a linear recurrence relation defining the value of the function 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 $\mathsf{s}(0)$ and $\mathsf{s}(1)$. Since the length-0 string has just one parse (which would be "‖" in the notation, to start continuation off properly) and the length-1 string has two, namely as an unfooted syllable "‖x‖" and as a monosyllabic foot "‖X‖", we have exactly the following problem to solve: find the function s meeting these conditions:

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

The usual methods[15] yield the following solution:

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

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

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

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

(22) $\mathsf{s}(n) + \mathsf{s}(n-1) = \tfrac{1}{2}\, \mathsf{s}(n+1)$

[15] See, for example, "Recurrence Relations" in Wikipedia; or search on "recurrence relation" 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.

Thus from eqs. (21) and (22), we obtain

(23) $P_{NM}(n) = ½\ s(n + 1)$

With eq. (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 $$P_{NM}(n) = \mathrm{round}\left(\frac{(1+\sqrt{3})^{n+1}}{2\sqrt{3}}\right) \qquad n > 0$$

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

(27) $$\frac{(1+x)^n - (1-x)^n}{2x}$$

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}, with both parenthesized numerator terms in the above contributing one such, which will simplify, when divided by $2x$, to a term containing x^{2k}. This is convenient when $x = \sqrt{3} = 3^{½}$, and the final result looks like this:

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

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

Looking at eq. (28), with its powers of 3 from the method of continuations, and at eq. (13), with its powers of 2 from the method of arrangements, one might not imagine that they come to the same thing. Since both count the same set, we can be quite confident that 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

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

3 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 single head foot (giving us the *main stress* when footheads are interpreted as stresses).

Here's the result:

(30) QI/M $\quad \mathrm{P_M}(n) = \dfrac{n}{2} P_{\mathrm{NM}}(n)$

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. Call the set $\{\pi, \bar{\pi}\}$ a *dual pair* (NB: unordered). Consider the entire collection DP(*n*) of dual pairs of parses of length *n* syllables, writing Π(*n*) for the set of individual QI parses on a length-*n* string.

(31) Set of Dual Pairs $\quad \mathrm{DP}(n) = \{\ \{\pi, \bar{\pi}\} \mid \pi \in \Pi(n)\ \}$

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 $\mathrm{P_{NM}}(n)$ elements in DP(*n*).

IV. Each element $\{\pi, \bar{\pi}\} \in \mathrm{DP}(n)$ contains a total of *n* X's.

With regard to II and III, note that π and $\bar{\pi}$ give rise to just one dual pair.

To establish IV, consider any pair $\{\pi, \bar{\pi}\}$ and say π contains k X's and $(n-k)$ x's, $k \geq 0$. Then $\bar{\pi}$ 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 dual 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, which has *n* X's by IV, then produces exactly *n* parses with one syllable identified as the main stress. Since there are half as many dual pairs as parses (noted in III), we obtain eq. (30).

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 length exactly one parse with no stresses at all. Call the number of these inclusive parses $P_{M+\varnothing}(n)$. We have:

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

4 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:

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

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

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

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

5 Generative Schemes

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

5.1 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:

(36) $K_1 = \{ \|x, \|X \}$

We notate carefully, so as to feed properly into 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.[16]

(37) 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:

(38) 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.

5.2 QS Generation: Copy & Change

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

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 three T's: TTT. The following procedure will generate every sequence of length 3 over {T,F}.

(39) Generation of all 3-character sequences 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

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

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.

Here's the algorithm. Let L_0 be the original string. On the m^{th} step, copy the result L_{m-1} of the $(m-1)^{\text{th}}$ step and subjoin the copy to the original, creating a list of the form $L_{m-1} + L_{m-1}$. Then change each character in the m^{th} serial position in each string *of the copy* to its opposite value. Perform as many steps as there are characters to be targeted in the string. 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 attaches the input to its QI parses.

To illustrate, let's construct the QS parses on all inputs of length 2. This will require two steps of copy & change. We write ch(k) for the procedure that changes the copy's k^{th} syllabic character.

(40) Generation of all 2σ QS parses

0. L_0 xx → ||x||x||, ||x||X||, ||X||x||, ||xX||, ||Xx||

1a. $L_0 + L_0$
xx → ||x||x||, ||x||X||, ||X||x||, ||xX||, ||Xx||
xx → ||*x*||*x*||, ||*x*||*X*||, ||*X*||*x*||, ||*xX*||, ||*Xx*||

1b. 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. $L_1 + L_1$
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. 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**||

There is now a clear path all the way from a starting point{||x, ||X}[17] to the full panoply of QS parses. The method of continuations produces the QI/NM parses up to any desired length, and the copy & change procedure expands to QS. Parses marked for main-stress can be constructed by working through the NM parses, iteratively selecting each X or H for promotion to Y or K.

6 Concluding Remarks

Resolving a natural formal question—*how many parses*?—has led to simple, effective methods for *constructing* the parses in their entirety. Knowledge of parsing numerics emerges along with control of the entire range of forms admitted by the structural assumptions. Thus equipped, the analyst may turn to the conduct of sound analysis.

References

Alber, B. 2005. Clash, Lapse, and Directionality. *NLLT* 23.3: 485–542.

Alber, B. and A. Prince. 2008. Class notes. Phonology III, Rutgers University.

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

[17] If we want to go aggressively minimal, we can start out with {||}, which parses the length-0 string.

Hayes, B. 1980. *A Metrical Theory of Stress Rules*. Ph.D. 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*. Ph.D. dissertation, MIT.

Mullin, K., B.W. Smith, J. Pater, and J.J. McCarthy. 2010. OT-Help 2.0 UserGuide. Ms. University of Massachusetts, Amherst. http://web.linguist.umass.edu/~OTHelp/OTHelp2man.pdf

Prince, A. 1976. Applying Stress. Ms. UMass, Amherst. http://ruccs.rutgers.edu/~prince/hold/app_str.pdf

Prince, A. 1993. In defense of the number *i*: Analysis of a linear dynamical model of linguistic generalizations. RuCCS-TR-1. http://equinox.rutgers.edu/gamma/dynlinmodel.pdf

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. http://roa.rutgers.edu/view.php3?id=1524

Prince, A. & P. Smolensky. 1993/2004. *Optimality Theory*: *Constraint Interaction in Generative Grammar*. Blackwell. ROA-537. http://roa.rutgers.edu/view.php3?id=845

Riggle, J. 2004. *Generation, Recognition, and Learning in Finite State Optimality Theory*. Ph.D. dissertation, UCLA. http://hum.uchicago.edu/~jriggle/riggleDiss.html. 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*. Ph.D. dissertation, University of Colorado at Boulder. ROA-90. http://roa.rutgers.edu/view.php3?id=548