Harmony at Base Omega:
Utility Functions for OT

Alan Prince, Rutgers University/ New Brunswick. January 13, 2006.

1. Background

A constraint hierarchy imposes a 'lexicographic order' on a candidate set. Alphabetic ordering is the commonly-cited analogy, but the positional system of numerical representation provides the closest parallel. Imagine a collection of numbers with, say, 4 digits; we can pad with zeros at the beginning to bring in everything below 10,000:

(1) **Positional representation**

0000
0009
0023
0798
1000
3362
3378
6409

Read in the usual way, this list describes various numbers ranging from 0 to 6,409. Under the usual ordering we have:

$0 < 9 < 23 < 798 < 1000 < 3362 < 3378 < 6409$

One number is smaller than another if and only if it is smaller in first digit on which the two differ — this is lexicographic order. Thus, 9 (0009) is strictly smaller than 23 (0023), because of the comparison in the penultimate digit, despite the fact that 9 is *not* smaller than 2 or 3, not smaller than 2+3, $2 \cdot 3$, 3^2, 2^3, *etc*., *etc*. Similarly, 1000 is bigger than 9 (0009). The same considerations apply if a decimal point is introduced anywhere: we have .0009 <.0023 under the lexicographic ordering principle.

Take these as OT violation profiles over a four-constraint set, and we get exactly the same ordering, where 'better than' mirrors 'less than'. The violation profile '0009' is better than '0023' because it is better in the highest-ranked constraint on which they differ, namely the third.

The decimal reading of the violation profiles (1) in yields a quantitative interpretation that perfectly tracks the 'harmonic ordering of forms' (Prince & Smolensky 2004: 85-86) and thereby makes concrete the notion of 'harmony'. In the economics literature, any such faithful translation of a preference ordering into quantities is known as a 'utility function', where 'utility' reifies preference. (More exactly, perhaps, an 'inutility' function, since we seek a minimum.) In the case of OT, it's clear that the decimal interpretation runs out of steam at 9 violations per constraint. Since the number of possible violations a form may accrue is unlimited (just as the

length of a possible form is unlimited), it follows that no positional interpretation, on any base, can adequately serve as a utility function for general OT. (If there is a cut-off point, as there will necessarily be in any finite collection of candidates, then a representation on the base m+1 will suffice, for m the maximum number of violations on any constraint, as has often been noted in the literature; see e.g. Prince & Smolensky 2004:236, Prince 2002).

It may seem mildly perverse to seek a utility function for OT, since the logic of the theory is so deeply comparative. Raw number of violations is, in isolation, indicative of nothing. But assigning a single quantity to a complex violation profile changes little in this regard; the 'utility' remains meaningless except in its order relations with other utilities, in exactly the same way that violation profiles are meaningless outside of comparison. A utility function may have its uses in establishing properties of the system, and, if any such functions exist, we should know about them. In this note, I identity a class of utility functions that work for any set of OT constraints.

2. A Utility Function for OT

Suppose we have a collection of n constraints Σ, individually designated $C_1,\ldots,C_n$, with arbitrary indices, each susceptible in principle of an unlimited number of violations. Each constraint is a function from the universal set of candidates U into $\mathbb{N}$, the set of non-negative integers.

(2) $C_k: U \rightarrow \mathbb{N}$

This merely says that a constraint assigns some nonnegative number of violations to each candidate.

A constraint hierarchy H is obtained by imposing a linear order on Σ, yielding a list or 'vector' of constraint functions: this then acts as a function from U to $\mathbb{N}^n$. We apply H to a candidate in the usual coordinatewise fashion, writing $\mathbf{x}[i]$ for the i^{th} coordinate of any vector $\mathbf{x}$.

(3) $\mathsf{H}(u)[i] = \mathsf{H}[i](u)$

This expression just records the fact the i^{th} value in the violation profile of candidate u, namely $\mathsf{H}(u)[i]$, is equal to the number of violations assigned by the i^{th} constraint in the ranking, $\mathsf{H}[i]$.

As a first step, we seek an appropriate representation for the *first* coordinate — one whose properties can be recursively replicated when we turn to the second coordinate, and so on. With this in mind, let us map $\mathbb{N}$ into the half-open interval [0,1), which consists of the set of all numbers from 0 to 1, including 0, but excluding 1. The following scheme will do it handily:

(4)

$\mathbb{N}$		[0,1)
0	$\rightarrow$	0
1	$\rightarrow$	$^1/_2$
2	$\rightarrow$	$^2/_3$
3	$\rightarrow$	$^3/_4$
4	$\rightarrow$	$^4/_5$
k	$\rightarrow$	k/(k+1)

Call this function f.

(5) $f(x) = x/(x+1)$

The function f fits every number from $\mathbb{N}$ into [0,1). We now face the further task of fitting every vector *beginning with 0* into the interval [0, ½), every vector beginning with 1 into [$^1/_2$, $^2/_3$), every vector beginning with 2 into [$^2/_3$, $^3/_4$) and so on. This will ensure the lexicographic result that every vector beginning with 0 is ordered ahead of every vector beginning with 1, and so on.

To step into the recursive world, let us consider how to deal with a two-coordinate vector. Coordinate 1 is handled by f alone. To handle coordinate 2, we must anchor the set of vectors with first coordinate v_1 at $v_1/(v_1+1)$; that is, we map $\langle v1, 0\rangle$ to the quantity $f(v_1)$. To deal with the remaining vectors $\langle v_1, v_2\rangle$, we must shrink all of $\mathbb{N}$ into the interval $[f(v_1), f(v_1+1))$, *i.e.*

$$[\frac{v_1}{v_1+1}, \frac{v_1+1}{v_1+2})$$

Both tasks are easily accomplished: we already know how to shrink $\mathbb{N}$ into [0,1) via f itself:

$$f(\mathbb{N}) \subseteq [0,1)$$

We multiply f by an appropriate shrinkage factor Δ, which we must yield the following inequality, no matter what the value of v_2:

$$f(v_1) + \Delta \cdot f(v_2) < f(v_1+1)$$

Equivalently,

$$\Delta \cdot f(v_2) < f(v_1+1) - f(v_1)$$

Since $f(v_2)$ is always less than one, we can get the desired inequality if we set Δ equal to the right hand side:

$$\Delta = f(1+v_1) - f(v_1)$$

A quick computation shows that

$$\Delta = 1/ (v_1+1)(v_1+2)$$

Thus, we reach the following expression for the value assigned to $\langle v_1, v_2\rangle$

$$f(v_1) + 1/(v_1+1)(v_1+2) \cdot f(v_2)$$

In this way, we fit a shrunken version of $\mathbb{N}$ into each of the subintervals [f(n), f(n+1)), for all $n \in N$, treating each subinterval as an exact miniature copy of [0,1). Of course, these are 'miniature' only in the sense of length, not cardinality: each copy contains exactly as many points as the original. This generous fact allows us to repeat the procedure recursively. Handling the third coordinate will mean stuffing a suitably shrunken copy of $\mathbb{N}$ into the subintervals carved out by the insertion of the second coordinate into the representation of the first, and so on.

In general, then, we can recursively define a utility function, $F: \mathbb{N}^n \rightarrow [0,1)$, as follows. We use these notational conventions:

- $\mathbf{v} \in \mathbb{N}^n$
- $v_k = v[k]$, the k^{th} coordinate of $\mathbf{v}$
- $\mathbf{v}\backslash 1$ = the list $\mathbf{v}$ with the first coordinate removed.
- $\boldsymbol{\lambda}$ the empty list

(6) **A Utility Function for OT**

$f(x) = x/(x+1)$

$F(\boldsymbol{\lambda}) = 0$

$F(\mathbf{v}) = f(v_1) + 1/((v_1+1)(v_1+2)) \cdot F(\mathbf{v}\backslash 1)$.

This function chews through a vector from first coordinate to last, shrinking any violation value via the function f into [0,1), then into the niche determined by the preceding values.

Here are some typical examples, to give a sense of how this operates:

C_1	C_2	C_3	F exact	F decimally
0	0	1	$^0/1 + ½ \cdot ^0/1 + ½ \cdot ½ \cdot ½ = {}^1/8$	0.125
0	0	10	$^0/1 + ½ \cdot ^0/1 + ½ \cdot ½ \cdot (^{10}/11) = {}^5/22$	0.227….
0	1	1	$^0/1 + ½ \cdot ½ + ½ \cdot ^1/6 \cdot ½ = ¼ + {}^1/24 = {}^7/24$	0.292…
2	1	1	$^2/3 + {}^1/12 \cdot ½ + {}^1/12 \cdot {}^1/6 \cdot ½ = {}^2/3 + {}^1/24 + {}^1/144 = {}^{103}/144$	0.715…

Observe that a 0 coordinate is followed by a shrinkage factor of 1/(0+1)(0+2) = ½; a 1 coordinate by a shrinkage factor of 1/(1+1)(1+2) = $^1/6$; a 2 coordinate by a factor of 1/(3·4) = $^1/12$, and so on. Each factor shrinks all that follows, so they mount up multiplicatively.

A diagrammatic overview of this construction is presented in the Appendix.

3. A Class of Utility Functions

The shrinkage term in (6) is merely the *length* of the interval $[v_1, v_1+1)$, equivalent to the fraction of the length of [0,1) which it occupies.

$$\frac{x+1}{x+2} - \frac{x}{x+1} = \frac{1}{(x+1)(x+2}$$

The length is obtained by subtracting the endpoints; we articulate the Δ-notation to represent it:

(7) $\Delta_k = f(v_k+1) - f(v_k)$

We may now re-write the recursion term as follows:

(8) $F(\mathbf{v}) = f(v_1) + \Delta_1 \cdot F(\mathbf{v}\backslash 1)$

This suggests a more general class of functions to which (6) belongs: those in which the base function $f:\mathbb{N} \rightarrow [0,1)$ is anything at all so long as it is *strictly increasing*, so that i.e. Δ_k is always positive.

(9) **Utility Functions for OT**.

Let f: $\mathbb{N} \rightarrow [0,1)$ be strictly increasing.

$F(\boldsymbol{\lambda}) = 0$

$F(\mathbf{v}) = f(v_1) + \Delta_1 \cdot F(\mathbf{v}\backslash 1)$

The results of recursion can be written out directly in closed form. Adopting the convention that $\Delta_0 = 1$, we have

$$F(\mathbf{v}) = \Delta_0 f(v_1) + \Delta_0\Delta_1 f(v_2) + \Delta_0\Delta_1\Delta_2 f(v_3) + \ldots \Delta_0\Delta_1\ldots\Delta_{n-1} f(v_n)$$

$$= \sum_{k=1}^{n} \left(\prod_{j=0}^{k-1} \Delta_j\right) f(v_k)$$

Positional numerical representation falls under this type of description as well. Decimal notation for numbers q in the interval $0 \leq q < 1$ is obtained when the base function is defined as $f(v_i) = v_i/10$. The v_i must of course be integers chosen from $0 \leq v_i \leq 9$ in order to ensure that $f(v_i)$ stays in [0,1). The Δ_k term, as defined in (7), is constant at $1/10$ for all k. The decimal representation comes out, unsurprisingly, as this:

$$F(\mathbf{v}) = \Sigma_k \left(1/10\right)^{k-1} (v_k/10)$$

The kind of function we are looking at crucially generalizes positional representation through its sensitivity, in the Δ_k terms, to the contents of the list. It's a kind of dynamic or context-sensitive positional interpreter, which flexibly accommodates individual coordinates of any size.

Let us now demonstrate that the defined function faithfully reproduces lexicographic order, as claimed. First, we establish a useful preliminary result: $F(\mathbf{v})$ is not only always less than 1, it must also always be less than $f(v_1+1)$.

(10) **Lemma.** Let F,f be any functions meeting the requirements of (9). Let $\mathbf{v} \in \mathbb{N}^n$. Then $F(\mathbf{v}) < f(v_1+1)$.

Proof. By induction on the number of coordinates in a vector. Suppose **v** has one coordinate, i.e. is of 'length 1'. Then $F(v)=f(v_1)< f(v_1+1)$, since f is strictly increasing. Now assume the lemma holds for all vectors of length <n. Let **v** be a vector of length n. We have

$$(*) \qquad F(\mathbf{v}) = f(v_1) + \Delta_1 F(\mathbf{v}\backslash 1)$$

$$= f(v_1) + (f(v_1+1) - f(v_1)) \cdot F(\mathbf{v}\backslash 1)$$

By the induction hypothesis, we have $F(\mathbf{v}\backslash 1)<1$. Since the Δ_1 term is positive, we may replace $F(\mathbf{v}\backslash 1)$ with its upper bound 1 in the expression (*) to create the following inequality:

$$F(\mathbf{v}) < f(v_1) + f(v_1+1) - f(v_1) = f(v_1+1)$$

This establishes the desired result. □

We may now show that that if **v** is lexicographically less than **w**, then $F(\mathbf{v})<F(\mathbf{w})$.

(11) **Proposition 1**. Let F,f be any functions meeting the requirements of (9). Let $\mathbf{v},\mathbf{w} \in \mathbb{N}^n$ be such that for some $k \leq n$, we have $v_k < w_k$ and for all $j<k$, we have $v_j=w_j$. Then $F(\mathbf{v})<F(\mathbf{w})$.

Proof. In the general case described in the statement, **v** and **w** are equal, coordinatewise, up to (but not including) coordinate k. This initial stretch of both vectors contributes exactly the same amount to F(**v**) and F(**w**) and does not affect their relative magnitude. We may therefore simply ignore the initial equal stretch and focus our attention, without loss of generality, on the computation from coordinate k on. This is equivalent to examining vectors which differ in the *first* coordinate.

From the hypothesis of the Proposition, we have **v**,**w** such that for some coordinate k, $v_k<w_k$, with all preceding coordinate equal. As per our remark, we may assume for convenience and without loss of generality that k=1.

From Lemma (10), we have this relation.

$$(*)\ F(\mathbf{v}) < f(v_1+1)$$

We argue that this is sufficient to establish the claim.

Remark 1. We know that $v_1<w_1$, i.e. that $w_1\geq v_1+1$. To reach our desired conclusion, consider first the set of vectors which have 0's in every coordinate but the first. For any such vector **x**, $F(\mathbf{x}) = f(x_1)$. Thus, the relative magnitude of the utilities of any such vectors is determined entirely by the first coordinate, and by f. Since f is strictly increasing, the relative magnitude of the utilities is the same as the relative magnitude of their first coordinates.

Remark 2. Now consider the set of vectors **y** which all share the same first coordinate, y_1. Clearly, the minimum utility in this set is obtained by $\langle y_1, 0, 0,\ldots,0\rangle$, the vector that has 0's everywhere outside the first coordinate. The reason is that in the definition of utility

$$F(\mathbf{y}) = f(y_1) + \Delta_1 \cdot F(\mathbf{y}\backslash 1)$$

the second term on the right side is positive iff the coordinates of **y**\1 are non-zero.

It follows from Remark 2 that

$$F(\langle w_1, 0, 0, \ldots,0\rangle) \leq F(\mathbf{w})$$

From Remark 1 it follows that

$$F(\langle v_1+1, 0, 0, \ldots 0\rangle) \leq F(\langle w_1, 0, 0, \ldots,0\rangle)$$

Putting these two observations together, we have

$$F(\langle v_1+1, 0, 0, \ldots 0\rangle) \leq F(\mathbf{w})$$

That is,

$$f(v_1+1) \leq F(\mathbf{w}).$$

Recalling the inequality (*) from the Lemma, we have

$$F(\mathbf{v}) < f(v_1+1) \leq F(\mathbf{w}).$$ □

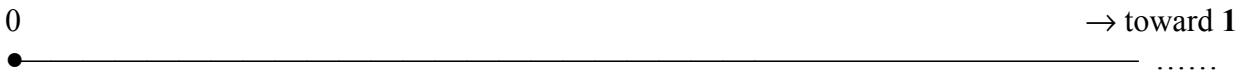
We now show that the relation between utility and lexicographic order holds in the other direction as well: a relatively smaller utility guarantees a better lexicographic performance.

(12) **Proposition 2**. Let F,f be any functions meeting the requirements of (9).Let $\mathbf{v},\mathbf{w}\in\mathbb{N}^n$ be such that $F(\mathbf{v})<F(\mathbf{w})$. Then for some $k\leq n$, we have $v_k < w_k$ and for all $j<k$, we have $v_j=w_j$.

Proof. Vectors **v**, **w** must differ in at least one coordinate, else $F(\mathbf{v})=F(\mathbf{w})$. Let k be the first coordinate in which they differ. By Proposition 1, if $w_k>v_k$, then $F(\mathbf{w})>F(\mathbf{v})$, contrary to assumption. Since $w_k\neq v_k$, it can only be that $v_k<w_k$. □

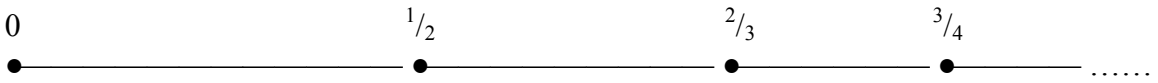
Appendix. Diagrammatic development of the Utility function of §2

¶ Take the half-open interval [0,1) which starts off at 0 and runs up to but *doesn't include* 1.

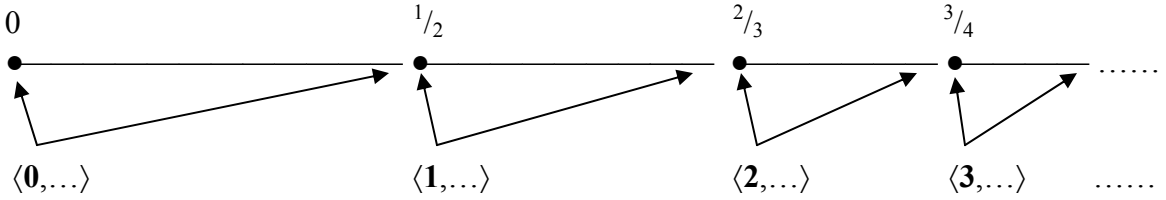


¶ Divide it up into disjoint half-open segments: **[**0, $^1/_2$ **)**,
[$^1/_2$, $^2/_3$**)**,
[$^2/_3$, $^3/_4$**)**, [$^3/_4$,$^4/_5$), [$^5/_6$,$^6/_7$), etc., etc.

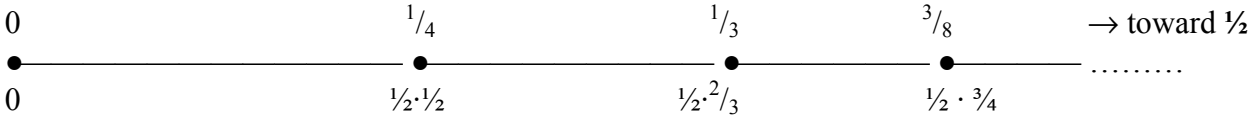
(Not drawn to scale.)



¶ Each subinterval hosts a class of violation profiles.
Every violation profile beginning with 0 goes into [0,½).
Every profile beginning with 1 goes into [$^1/_2$, $^2/_3$).
Every profile beginning with 2 goes into [$^2/_3$, $^3/_4$).
And so on, ad inf.



¶ To deal with the 2nd coordinate: using the same strategy, subdivide each of these subintervals. Consider for example [0, $^1/_2$). We shrink the sequence 0, $^1/_2$, $^2/_3$, $^3/_4$, … to fit into the subinterval.



¶ Each subinterval hosts a class of profiles.
Every profile beginning ⟨0,0,…⟩ fits into [0,$^1/_4$).
Every profile beginning ⟨0,1,…⟩ fits into [$^1/_4$, $^1/_3$).
Every profile beginning ⟨0,2,…⟩ fits into [$^1/_3$, $^3/_8$).
And so on, ad inf….. ❑

References

Prince, Alan. 2002. Anything Goes. In *A New Century of Phonology and Phonological Theory*, ed. T. Honma, M. Okazaki, T. Tabata, & S. Tanaka. Kaitakusha: Tokyo. 66-90. ROA-536.

Prince, Alan & Paul Smolensky. 1993/2002/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell. ROA-537.