One Tableau Suffices

Alan Prince

Abstract

The grammars of any OT typology can be derived from a single tableau in which each row, when asserted as optimal, delivers the grammar of a language in the typology. In the single-tableau representation, each candidate represents an entire language. When, as is typically the case, a typology is constructed from more than one candidate set, a single tableau representation may be built from the *Minkowski sum* of the whole collection.

Since all typologies are representable as a finite collection of finite candidate sets, it follows that a single tableau is always sufficient to represent the grammars of a typology. The study of formal typologies therefore reduces to the study of single tableaux, which are just matrices of nonnegative integers. The result shown here thus gives us a new point of entry into the study of typologies as abstract objects. It also allows us to move around easily in the set of all typologies on *n* constraints, shown here to be a lattice, because the meet of two typologies is the Minkowski sum of single tableaux that represent them. This is significant because, as a generalization structure, the lattice of typologies plays a role in classifying the grammars of a typology according to shared and contrasting ranking properties (Alber and Prince 2015, in prep.). Perhaps surprisingly, it also follows that an OT system imposes orders and equivalences not just on individual candidates but on the grammars of its typology, a matter explored in detail in Merchant & Prince 2016.

Acknowledgment. The original work on this project was supported by grants from the University of Verona, Italy, in academic year 2010-2011. Thanks to Birgit Alber, Nazarré Merchant, and Natalie DelBusso for discussion that has improved the paper throughout. The anxieties expressed by unidentified commenters (2016) have been addressed in a few places, though concerns about the dangers of the internet have not.

Version history. Posted to ROA November 10, 2015. Minor mods. Nov. 19, Nov. 26, 2015; Mar. 30, June 14, 2016.

1 Introduction

An OT Typology is typically obtained from a diverse collection of candidate sets ('csets'), each derived from a different input. The Basic Syllable Theory of Prince & Smolensky 1993/2004:111-16, for example, is explored in terms of two inputs, /V/ and /CVC/, and the full typology implied by their assumptions may be produced by adding /C/ to this list. A typology can sometimes be generated from a *single* omnibus input, one which provides all the configurations that distinguish optima. In the case of the Basic Syllable Theory, /CCVVC/ will work, as computed in Riggle 2004:109, the first report of the actual extent of the typology. Similarly, in some subtheories of stress, the five-syllable forms distinguish all languages, and one need look no further.

But it may happen that limitations inherent in $\textsc{Gen}_S$ for a system S, or in mathematics, simply do not allow this kind of reduction. In the stress theory of Alber 2005, for example, it is necessary to include words of both odd and even length in any collection of csets that yields the full typology. Some phenomena are restricted to odd lengths, others to even lengths, and they cross-classify the languages of the typology. But no form can be simultaneously 3 and 4 syllables long.

Contrary to the messy expectations created by such examples, we will see here that any OT typology whatever can always be produced from a single violation tableau or 'VT'. (We will use the term 'VT' to refer exclusively to the representation of *one* cset.) This result is consequential, because it means that the structure of any and all typologies is inherent in the structure of single VTs. This gives us a new handle on treating the OT typology as an abstract object (Merchant & Prince 2016), extending our established understanding of the OT grammar (Prince 2002a,b; Brasoveanu & Prince 2011, Merchant & Riggle 2016) as a certain kind of order structure. More surprisingly, perhaps, it means that an OT system, in providing the means to compare competing forms, also imposes relations of order and equivalence between grammars. When the relations shared across all the different single VTs that yield a given typology are taken into account, an entire *grammar* within a typology can, with complete accuracy, be said to perform better on a given constraint — to be less marked in that respect, for example, or to be more faithful — than another grammar in the same typology.

'Typology' has several related senses in OT, and it's necessary to tease them apart. The term may be construed as referring to a family of languages, where a language is conceived of as a set of linguistic objects: usually, mappings from an input to an output representation, though other conceptions may be easily accommodated, and the mechanics of choice do not depend on the nature of the objects chosen. Following the terminology of Alber & Prince 2015, in prep., we refer to this sense as 'extensional'. A syllable structure typology, in this sense, is a typology of syllable structure; a voicing typology is about voicing; a typology of do-support and subject-aux inversion is about subject-aux inversion and do-support; and that's that. In this extensional sense, you can have two typologies generable from exactly the same VT — taking a VT to be just an array of integers — and the typologies would be considered different, and incomparable, if they arise in different linguistic domains or from different linguistic assumptions. Since OT is at bottom a theory of selection from alternatives, the typologies could even be more distant, involving systems of choice and decision far from language.

A 'typology' may also be understood as a collection of *grammars* — sets of linear rankings, each delimited by an ERC set (Prince 2002a,b). In this sense, termed *intensional* by Alber & Prince, it doesn't matter what the candidates are, structurally, and, if we focus on the patterns of ranking rather than the names of constraints, it doesn't even matter what the constraints are. All that matters is that we have a collection of grammars generable from some VT — or set of VTs, which we show to be the same thing. In this sense, if two substantively distinct systems are derived from identical VTs, they can be understood as instantiations of the same *intensional typology* because they have isomorphic ranking structures.

We can distinguish between Abstract OT (AOT), which deals with the general properties of OT as a theory of choice, and Concrete OT (COT), in which the basic constructs of AOT are attached to particular constraints and structures, yielding a theory of some phenomenal domain. The study of AOT frees us from particular linguistic assumptions and allows us to focus on the properties of OT evaluation that are constant across any such assumptions. AOT is, then, the theory of EVAL; arriving at COT requires specification of GEN_S and CON_S, defining an OT system S. Any instance of COT inherits all of its ranking theory from AOT, which includes, prominently, the theory of intensional typologies.

The main result, then, is that every intensional typology, in the sense we have given to this term, is representable as a single VT, a 'unitary VT' (UVT). We produce UVTs constructively. Given any target set of VTs, whose intensional typology we wish to represent, we create a UVT from it in the following way. A row in the UVT is obtained by collecting one row from *each* of the VTs in the target set and then simply adding up all the rows we have collected. Row addition is done by adding the values in each constraint column separately. If, for example, the first row of the first VT is (0,1,2) and the first row of the second is (1,3,0), we can start the process by adding them to get (0+1, 1+3, 2+0) = (1, 4, 2). The rows are added like vectors, component by component, where each constraint defines a component.[1] Component-wise addition compresses a collection of rows to a single row. The UVT gathers all such sums, summing every possible choice of rows from the target set of VTs, with one row chosen from each. We might expect loss of information from compressing a set of rows into a single row by simply adding them all up. But no loss of ranking information occurs. The process of compression, we will show, leaves the intensional typology intact in every detail.

This result is perhaps surprising, but not inaccessible to intuition. OT depends on comparison of alternatives, not absolute values assigned to individual candidates. As ERC theory makes clear (Prince 2002a *et seq*.), comparison involves the subtraction of one violation profile from another,

[1] For our purposes, a vector is just a list of numbers. The elements in such a list may be called 'components', 'coordinates', or 'entries'. We'll settle on the first.

followed by normalization of the result to three values (positive, negative, zero = W, L, *e*). As we will see, subtraction cancels out irrelevancies, revealing the original. When exact cancellation does not occur in the derived UVT, the result is, we show, redundant, in that it is logically entailed by those cases where cancellation is perfect.

Adding up each possible selection from a family of sets, where a selection draws one element from each set, produces a collection known as the *Minkowski sum* of the set family. Given two sets of addable things, the Minkowski sum, often written ⊕, may be defined like this:

(1) **Minkowski sum**. $\mathbf{P \oplus Q = \{p+q \mid p \in P, q \in Q\}}$

Our principal finding, then, is that the Minkowski sum of a collection of VTs has the exactly the same intensional typology as the collection itself.

2 How it works

The sense of this result can be garnered from a pair of examples. From the second and more abstract of the two, it is possible to see how the proof of the general claim unfolds.

2.1 A Concrete Example

Let's consider first the version of Basic Syllable Theory (BST) treated in Prince & Smolensky, which we will call PST, for 'partial syllable theory'. Two inputs are considered (p.111-116): /V/ and /CVC/. This gives a typology of 9 languages, 3 short of the full 12 that are obtained when the input /C/ is included.

The PST is a coarsened version of the BST, obtained when an informative class of inputs is omitted: those which contain consonants that cannot be syllabified without a faithfulness violation. The input /C/, for example, must either be redeemed by an epenthetic vowel (violating f.depV) or omitted in the output (violating f.max). There is no candidate that simply syllabifies it as $[C]_\sigma$ without violation, because $\text{G}{\scriptsize\text{EN}}_{\text{BST}}$ defines its syllable object as having a vocalic nucleus. This is among the assumptions that make the BST 'basic'.

The PST is still an OT typology: it simply avoids some distinctions among languages that the BST enforces. We can think of $\text{G}{\scriptsize\text{EN}}_{\text{PST}}$ as being slightly different from $\text{G}{\scriptsize\text{EN}}_{\text{BST}}$ by virtue of tighter limits on what can be an *input*. GEN_{PST} excludes solo /C/ and, more generally, initial and final CC along with intervocalic CCC; that is, inputs which contain a C not adjacent to a V. In PST, the input possibilities are therefore more closely tailored to the permitted outputs than in BST. But it is not idle to study it: 6 of its grammars survive entirely intact in BST, and the 3 others split neatly in two when refined to include the requirements imposed by choosing an optimal output from the additional BST input /C/.[2]

Let's be clear about what we mean by PST and BST.

• **GEN**$_{BST}$

- a **candidate** is a structure ⟨in, out, ∼⟩ consisting of an *input*, an *output*, and a correspondence relation '∼' between them.

- an **output** is of the form $\{[(C)V(C)]_\sigma\}^*$, that is: fully syllabified strings as well as the empty string, where a syllable always contains a vowel and may contain a single consonant at either margin, as in the formula.

- an **input** is of the form $\{C,V\}^+$, that is, any nonempty string on C and V; and

- a **correspondence relation** holds between input and output, relating C-to-C and V-to-V in a way respects the order of elements in the input. An element of the input may lack an output correspondent, an element of the output may lack an input correspondent, signifying deletion and insertion respectively, in the familiar way (McCarthy & Prince 1995:16).

● **GEN**$_{PST}$. GEN$_{PST}$ differs only in that it excludes as *inputs* those forms of $\{C,V\}^+$ without vowels as well as those beginning #CC, ending CC#, and containing CCC anywhere. Thus, the input is defined as consisting of the strings $\{(C)V(C)\}^+$.

• **CON**$_{PST}$ = **CON**$_{BST}$. Here there is no difference between PST and BST. CON$_{PST}$ and CON$_{BST}$ are identical and include the 5 familiar constraints: m.Ons, m.NoCoda, f.max, f.depV, and f.depC. In the modern notation used here, constraints are explicitly typed by prefix: 'm' for markedness, 'f' for faithfulness. They are defined in the usual way, which we present concisely here for the reader's convenience, using the familiar OT *-operator, which denotes a function from candidates to nonegative integers {0, 1, 2,…} that returns the number of matches in a candidate to the pattern specified after the *. The expression 'x∼y' denotes that x corresponds to y.

---

[2] There are two possible optima from /C/: .C𝔸. with insertion, and ε, the empty string, with deletion. The faithful monoconsonantal output is not feasible, because every non-null output is required to consist of well-formed syllables, and both BST and PST define a syllable as having a nucleus. The BST input /C/ therefore gives rise to 2 ranking requirements, f:depV≫ f:max, and f:max≫f:depV, one of which must be present in every grammar of the typology because /C/ is present as an input in every language of the typology. These play out freely over the requirements shared with the PST languages. In those grammars where the f.depV/f.max ranking is already fixed, nothing happens. But PST grammars which contain both rankings among their orders are split into two distinct languages in the BST. These are the languages where {f.max, f.depV}≫m.NoCoda in PST, in which codas appear because only faithful IO maps are allowed. There are 3 of these, one for each of the 3 ways in which onsets are handled. Thus of the 9 grammars in PST, 6 survive intact into the BST, and 3 bifurcate, yielding a total of 12.

(2) $\textsc{Con}_{BST}$ = $\textsc{Con}_{PST}$

a) m.Ons $*[_\sigma V$

b) m.NoCoda $*C]_\sigma$

c) f.max $*x \in in$ s.t. $\nexists y \in out$ s.t. $x{\sim}y$

d) f.depC $*y \in out$ s.t. $y = C$ & $\nexists x \in in$ s.t. $x \sim y$

e) f.depV $*y \in out$ s.t $y = V$ & $\nexists x \in in$ s.t. $x \sim y$

The grammars of the PST can be derived from the following two csets, which therefore constitute a *universal support* for it. Terminology: a *support* for a grammar is a finite collection of csets from which its ranking requirements may be completely derived. *A universal support* for a typology is a finite collection of csets from which all the grammars of the typology may be derived. Recall that every grammar and every typology has finite support. Valid typological analysis begins with a valid universal support. See Alber, DelBusso, and Prince 2016 for a method of proving that a given collection of csets is universal for a specified typology.

(3) The P&S Universal Support for PST

Input	Output	Type	m.Ons	m.NoCoda	f.depC	f.depV	f.max
(A) V	a. .V.	v.F	1		0		0
	b. .𝕋V.	v.Ins	0		1		0
	c. ∅	v.Del	0		0		1
(B) CVC	i. .CVC.	cvc.F		1		0	0
	ii. .CV.C𝔸.	cvc.Ins		0		1	0
	iii. .CV.∅	cvc.Del		0		0	1

To enhance orthographic perceptibility, we indulge in a few departures from notational rectitude: correspondence indices are suppressed, the occurrence of deletion is marked with ∅ and epenthetic elements are spelled 𝔸 and 𝕋. Syllable brackets are condensed to a single '.' at each syllable edge. Blank cells are numerically 0; they are left blank only to highlight the contrastive aspects of the violation pattern. To emphasize the IO relations determined by the individual candidates, they are coded in the "Type" column: v.F indicates the faithful mapping from V, while cvc.Ins indicates the mapping from CVC to CV.C𝔸 , which includes an epenthetic vowel, and so on. The suffixes F, Ins, Del indicate how a particular grammar deals with markedness problems raised by faithful rendition of the input — onsetlessness, presence of coda.

To construct a single-input tableau that yields a typology equivalent to that of the two inputs in (3), we sum up every pair of rows, one from the first cset, the other from the second. The first row below in (4), for example, represents the sum of row (a) + row (i):

(a) (1, 0, 0, 0, 0)
(i) + <u>(0, 1, 0, 0, 0)</u>
(a+i) = (1, 1, 0, 0, 0).

In the following tableau, we write $m^{\oplus}$, $f^{\oplus}$ as constraint prefixes to indicate that the column records a summed value, not the direct evaluation by a candidate of a constraint of type m or type f. The column labeled "Source" mentions the rows from (3) which are summed to produce the rows of (4). Under σ, we indicate the syllable shapes allowed in the language.

(4) **A Minkowski UVT for PST**

Lg#	Source	Source types	σ	$M^{\oplus}$:Ons	$M^{\oplus}$:NoCoda	$f^{\oplus}$: depC	$f^{\oplus}$: depV	$f^{\oplus}$: max
1.	(a) + (i)	v.F & cvc.F	(C)V(C)	1	1	0	0	0
2.	(a) + (ii)	v.F & cvc.Ins	(C)V	1	0	0	1	0
3.	(a) + (iii)	v.F & cvc.Del	(C)V	1	0	0	0	1
4.	(b) + (i)	v.Ins & cvc.F	CV(C)	0	1	1	0	0
5.	(b) + (ii)	v.Ins & cvc.Ins	CV	0	0	1	1	0
6.	(b) + (iii)	v.Ins & cvc.Del	CV	0	0	1	0	1
7.	(c) + (i)	v.Del & cvc.F	CV(C)	0	1	0	0	1
8.	(c) + (ii)	v.Del & cvc.Ins	CV	0	0	0	1	1
9.	(c) + (iii)	v.Del & cvc.Del	CV	0	0	0	0	2

The structures and constraints of the PST are such that it is easy to construct an input with a candidate set matching the summed profiles of (4): /VCVC/. The output options involving the initial input vowel are completely independent of those of the final C, and the constraints evaluating those options function independently. Thus /VCVC/ mirrors within a single form the independence of the candidate sets from separate inputs /V/, /CVC/. This is by no means the general case, however. Nothing in the idea of GEN or CON compels the existence of a single complex form from which a typology devolves. There is in principle no limit to the number of inputs required to universally support a typology, except that the number will be finite in any particular case.

This example points to a way of using the Minkowski sum as a tool for establishing that a collection of csets actually provides a universal support for a typology. If the VT of possible optima associated with an input can be shown to be the Minkowski sum of csets in a claimed Universal Support, then that input adds nothing new to the typology. If we can show that every input other than those in the proposed support has this character, then we have established its universality. Note that we can repeat VTs from the support to achieve a weighting effect, since the repetition of an entire cset provides no new information. [3]

[3] Here's the bare bones of an argument for the validity of the offered support for PST using this idea. In PST, for any input we can distinguish just two kinds of violation-inducing landmarks: instances of V not preceded by C ('problematic V'), and instances of C not followed by V ('problematic C'). Each of the first may be dealt with in the three ways of cset (3A), which comes from /V/; each of the second may be dealt with exactly as in (1B), which comes from /CVC/. A generic form will have, say, p instances of bad V and q instances of bad C, with $p,q \geq 0$. The optima that it generates will be a subset of those generated by the Minkowski sum of a collection of csets consisting of p copies of (3A) and q copies of (3B). The ranking information provided by a generic input is then exactly the ranking information associated with that of the $p+q$ cset collection, which in turn is exactly that of the Universal Support in (3), since copies of the same cset have no effect on the typology. Since we're talking about a generic input, we can conclude that the two csets of (3) tell us all there is to know about the ranking structures of PST.

The success of Minkowski summation in representing the typology of an OT system S depends in no way on G_{EN_S}, and depends on C_{ON_S} only to the extent that a constraint is a function to the non-negative integers.[4] To see the real generality of the result, we need look no farther than the simplest nontrivial abstract system: two constraints, and two candidate sets, each with two candidates. This provides our second example. The argument requires nothing more than arithmetic, a minimum of patience with subscript-chasing, and acquaintance with the basics of ERC logic, such as may be obtained from casual examination of Prince 2002a or b.

2.2 An Abstract Example

Consider two sets **A** = {**a**,**b**} and **X** = {**x**,**y**}, each containing vectors of integers ('violation profiles'). Linguistically, the labels **a**,**b**,**x**,**y** would identify specific structures, 'candidates'; for us, they merely name the violation profiles.

From this assemblage, four extensional 'languages' may be constructed by choosing one desired optimum from each set. This yields (**a**,**x**), (**a**,**y**), (**b**,**x**), and (**b**,**y**) as our languages. These are the elements of **A**×**X**, the cartesian product of **A** and **X**.

The system can be represented in the usual VT format as follows, writing each violation vector of integers as a row, with one VT for **A** and another for **X**.

(5) VT components of **A**×**X**

A	C_1	C_2
a	a_1	a_2
b	b_1	b_2

X	C_1	C_2
x	x_1	x_2
y	y_1	y_2

The Minkowski sum of **A** and **X**, written **A**⊕**X**, comes out like this:

(6) **A**⊕**X**

A⊕X	$C_1^{\oplus}$	$C_2^{\oplus}$
a+x	$a_1 + x_1$	$a_2 + x_2$
a+y	$a_1 + y_1$	$a_2 + y_2$
b+x	$b_1 + x_1$	$b_2 + x_2$
b+y	$b_1 + y_1$	$b_2 + y_2$

[4] The argument depends on properties shared by all real numbers, including the non-negative integers.

In such a tableau, let's call the rows 'Minkowski candidates'. Recall that the Minkowski sum is defined as follows:

(7) **Minkowski sum**. **P**$\oplus$**Q** = {**p**+**q: p**$\in$**P, q**$\in$**Q**}

To investigate the relationship between the two systems, let's examine the language (**a**,**x**)$\in$**A**$\times$**X**. This means choosing **a**$\in$**A** and **x**$\in$**X** as desired optima in **A** and **X** respectively. These will be simultaneously optimal under a given ranking if and only if the ERCs [**a~b**] from **A** and [**x~y**] from **X** are both satisfied by that ranking. Because of the high symmetry of the example, we're not jeopardizing the generality of our conclusions by focusing on a particular case, since every case is structured the same way.

To see how the Minkowski candidates impose ranking conditions, we need to correlate ERCs in **A** and **X** with those in **A**$\oplus$**X**, which we'll refer to as 'Minkowski ERCs'.
The ERC [**a**$\sim$**b**] is obtained from the numerical values of **b** – **a**. The sign of each component $b_k – a_k$ will indicate the comparative relations between a_k and b_k, exactly as recorded in the ERC.

(8) **Comparative Values from the Sign of the Difference**

$b_k – a_k > 0 \Leftrightarrow$ [**a**~**b**]$_k$ = W   i.e., $a_k < b_k$

$b_k – a_k = 0 \Leftrightarrow$ [**a**~**b**]$_k$ = *e*   i.e., $a_k = b_k$

$b_k – a_k < 0 \Leftrightarrow$ [**a**~**b**]$_k$ = L   i.e., $a_k > b_k$

Call **b–a** the *value* of the ERC [**a**~**b**]. Let us calculate the values of all ERCs in **A**$\oplus$**X** with **a**+**x** taken as the desired optimum.

(9) **A**$\oplus$**X, a+x as desired optimum**

1. Raw ERC	2. Value	3. Equivalent ERC	4. Raw Calculation	5. Re-Organized
[a+x $\sim$ a+x]	**0**	***e***	(**a**+**x**) – (**a**+**x**)	(**a–a**) + (**x–x**)
[a+x $\sim$ a+y]	**y–x**	[**x**$\sim$**y**]	(**a**+**y**) – (**a**+**x**)	(**a–a**) + (**y–x**)
[a+x $\sim$ b+x]	**b–a**	[**a**$\sim$**b**]	(**b**+**x**) – (**a**+**x**)	(**b–a**) + (**x–x**)
[a+x $\sim$ b+y]	(**b–a**) + (**y–x**)	**φ**	(**b**+**y**) – (**a**+**x**)	(**b–a**) + (**y–x**)

The Minkowski candidate **a**+**x** will be optimal under a ranking if and only if all the listed ERCs (cols. 1 and 3) are satisfied by that ranking. Recall that a ranking (linear order) λ satisfies an ERC vector **α** if and only if every constraint assessing L in **α** is dominated in λ by some constraint assessing W.

The key analytical move consists of reorganizing the difference of sums (col. 4) into a sum of differences (col. 5). Through this rearrangement, the direct rendition of a Minkowski ERC in **A**$\oplus$**X** (col. 4) turns into a sum in which each parenthesized summand corresponds to an ERC within one of the original csets **A** or **X** (col. 5). Instead of an additive mash-up of violation profiles from both csets subtracted from another such collocation (col. 4), we deal with a neatly organized sum in which elements from **A** face off ERC-style in one parenthesized summand while elements from **X** face off in the other (col. 5).

Observe that ERCs **[a∼b]** and **[x∼y]**, native to **A** and **X** respectively, are both present undisguised in table (9), col. 3, appearing in their value-form as **y–x** and **b–a** respectively (col. 2). This already hints that a direct connection exists between (**a**,**x**)∈**A**×**X** and **a+x**∈**A**⊕**X**.

These particular ERCs are guaranteed to show up because the Minkowski sum consists of *all* combinations of choices from **A** and **X**. There will necessarily be pairs of sums which differ only in one place, like **a+x** and **b+x**, distinct only in the first position, which is drawn from **A**; and pairs like **a+x** and **a+y**, distinct only in the second position, which is drawn from **X**. Put aside ERCs of the form [**q**∼**q**], which compare a candidate with itself, as *noncontrastive*. Call a contrastive ERC *unconfounded* if it compares two Minkowski candidates differing in exactly one position, like **a+x** and **b+x**. Call all other contrastive ERCs *confounded*. The beauty of the unconfounded ERC is that by the subtractive method of ERC valuation, all summands vanish except the ones upon which they differ: (**b+x**) – (**a+x**) is simply (**b–a**), by arithmetic. From the liberality of the Minkowski sum, which combines each with each, we are assured that for every optimum-suboptimum pair within a participating cset, an unconfounded Minkowski ERC exists. If we want to find [**a**∼**b**] among the ERCs of **A**⊕**X**, we know that we can look to [**a+x** ∼ **b+x**], because we can be certain that candidates **a+x** and **b+x** exist in the Minkowski sum.

This observation implies more generally that the all the ERCs derivable within any member of an arbitrary collection of csets {**P**, **Q**, … , **Z**} will show up among the ERCs of **P**⊕**Q**⊕…⊕**Z**, via the guaranteed existence of unconfounded ERCs. All that's needed is the deployment of sufficient indices to keep track of what's going on. What happens in the general case is thus perfectly mirrored in our simple example.

What then of the *confounded* ERCs, which the Minkowski sum also yields? In the case at hand, there's just one: the ERC **φ** = [**a+x** ∼ **b+y**] corresponding to the value **(b–a)** + (**y–x)**. The summands of the ERC **φ** differ in both first and second positions.

The remarkable fact is that **φ** is *entailed* by the fusion [**a~b**]∘[**x~y**].[5] This means that the ranking requirements imposed by **φ** follow from the logical conjunction of the requirements imposed by [**a~b**] and by [**x~y**]. Since both [**a~b**] and [**x~y**] are present as unconfounded ERCs in the Minkowski sum, it follows that **φ** is redundant and places no further restrictions on the rankings that render **a** and **x** optimal. As we pursue the argument, it will become clear that this is true of all confounded Minkowski ERCs, no matter how many csets are involved.

Let's now turn to the entailment argument. Each parenthesized summand in the reorganized value expression **(b–a)** + (**y–x)** corresponds to an ERC that is native to one member of the basic two set family {**A**, **X**}, these being [**a~b**] and [**x~y**] respectively. To establish that **φ** is jointly entailed by **[a~b]** and **[x~y]**, it suffices to show that [**a~b**]∘[**x~y**] ⊨ [**a+x** ~ **b+y]**.

[5] The fusion α∘β of two ERCs is entailed by their logical conjunction. Thus if α∘β⊨γ, then the ranking conditions of {α,β} jointly entail those of γ. Recall that the fusion is calculated coordinatewise according the following scheme: for any V∈{W, *e*, L}, V∘V=V; L∘V=V∘L=L; *e*∘V=V∘*e*=V (Prince 2002a,b).

Of interest are the relations between *nontrivial* ERCs: those that contain both W and L, and therefore impose ranking restrictions. *Trivial* ERCs lack either W or L. ERCs without L are satisfied by any ranking whatever, because no constraint need be subordinated. ERCs with L but no W cannot be satisfied by any ranking, because the demand that all L's be dominated by W cannot be met, since there are no W's to dominate them.

Among nontrivial ERCs, entailment reduces to the requirement that certain relations between entailer (antecedent) and entailed (consequent) hold simultaneously in every component. Recall that entailment follows the scale $L < e < W$, where the entailer must be less than or equal to the entailed in that order. (Cf. $F < T$ in the Boolean world.) Consider an arbitrary component φ_k of $\boldsymbol{\varphi}$, $1 \leq k \leq n$, for n the number of constraints in the system, here just two. The value of φ_k is determined by the expression $(b_k - a_k) + (y_k - x_k)$. Each summand of φ_k is either positive, negative, or zero, yielding four cases with respect to the behavior of the entire sum.

Case 1. Both summands of φ_k are positive.

$(b_k - a_k) > 0$, $(y_k - x_k) > 0$. Thus the sum $(b_k - a_k) + (y_k - x_k) > 0$.

Therefore, $[\mathbf{a{+}x} \sim \mathbf{b{+}y}]_k = \varphi_k = W$. The behavior of component k of the ERC $[\mathbf{a{+}x} \sim \mathbf{b{+}y}]$ in this case is consistent with entailment of φ, since W, standing at the top of the order, is entailed by anything. In particular, no matter what value is assumed by the fusion $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim\mathbf{y}]_k$ in component k of φ, we have $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim \mathrm{b}]_k \vDash \varphi_k$.

Case 2. One summand of φ_k is zero and one is positive.

Then $(b_k - a_k) + (y_k - x_k) > 0$ so that $[\mathbf{a{+}x} \sim \mathbf{b{+}y}]_k = \varphi_k = W$. As with case 1, this is consistent with entailment of φ since anything entails W, and we have $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim\mathbf{y}]_k \vDash \varphi_k$.

Case 3. Both summands of φ_k are zero.

$$(b_k - a_k) = 0 \quad \Rightarrow \quad [\mathbf{a}\sim\mathbf{b}]_k = e$$

$$(y_k - x_k) = 0 \quad \Rightarrow \quad [\mathbf{x}\sim\mathbf{y}]_k = e$$

At φ_k, we find

$$(b_k - a_k) + (y_k - x_k) = 0 \quad \Rightarrow \quad \varphi_k = e$$

Since $e \circ e = e$, and every comparative value is self-entailing, this configuration is consistent with entailment as well. Here too, $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim\mathbf{y}]_k \vDash \varphi_k$

Case 4. At least one summand of φ_k is negative.

Any such summand corresponds to an ERC $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}_k = L$. Any fusion $\boldsymbol{\varphi} = \boldsymbol{\alpha} \circ \ldots$ involving $\boldsymbol{\alpha}$ will likewise have $\varphi_k = L$, because $L \circ V = L$, for any $V \in \{W, e, L\}$. Because one summand is negative in coordinate k, the fusion of ERCs corresponding to the summands in this case must be L at k. Since L entails anything, the behavior of the fusion $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim\mathbf{y}]_k$ is again consistent with entailment. Once again, we have $[\mathbf{a}\sim\mathbf{b}]_k \circ [\mathbf{x}\sim\mathbf{y}]_k \vDash \varphi_k$.

Observe that φ_k may assume any value in this case, since we cannot predict the sign of a sum from the knowledge that at least one summand is negative.

This exhausts the cases. We have established the desired entailment [**a**∼**b**]∘[**x**∼**y**] ⊨**φ** because at any index *k*, on a case-by-case basis:

1,2) φ_k = W, or

3) φ_k = [**a**∼**b**]∘[**x**∼**y**]$_k$ = *e*, or

4) [**a**∼**b**]∘[**x**∼**y**]$_k$ = L.

The confounded ERC [**a+x** ∼ **b+y**] is therefore entailed. We've looked at just one ERC from **A**⊕**X**, but by symmetry, any confounded ERC will go the same way. Therefore all confounded ERCs are redundant and the intensional typology of the Minkowski sum depends only on the ERCs that arise in the base system of VTs from which it is derived.

We've shown that the ERCs associated with any choice of desired optima in the base system will appear in the Minkowski sum. It may happen, of course, that a combination of choices is *inconsistent* in that there is no ranking, and therefore no grammar, that simultaneously picks all choices as optimal within their individual csets. This means that the choices involve a contradiction in ranking conditions across candidate sets. Suppose, for example, that (**a,x**) doesn't yield a language because its ERCs [**a**∼**b**] and [**x**∼**y**] impose contradictory ranking requirements. (In our example, a contradiction will be no more subtle than having one require $C_1 \gg C_2$ and the other $C_2 \gg C_1$.) What becomes of **a+x** in the Minkowski sum?

Since **a+x**, when selected as the desired optimum, generates an inconsistent ERC set, it follows that **a+x** cannot be optimal in **A**⊕**X**.

To see this, consider the ERC set Ω(**a,x**) that contains the ERCs associated with **a** as desired optimum in **A** and **x** as desired optimum in **X**. This set is, by assumption, inconsistent. But we've just seen that Ω(**a,x**) is a subset of Ω(**a+x**), the ERC set associated with the Minkowski candidate **a+x** as desired optimum in **A**⊕**X**. Any ERC set with an inconsistent subset is itself inconsistent.

This means that **a+x** is harmonically bounded within **A**⊕**X**. In particular, **a+x** will be bounded by the set of violation vectors that, when compared with it, yield the contradictory ERCs. In the present case, these will be **b+x**, yielding [**a**∼**b**], and [**a+y**], yielding [**x**∼**y**]. In short, a contradiction in ranking caused by choice of incompatible optima in different csets is transformed by Minkowski summation into a contradiction within the single cset: harmonic bounding.

Our example stands at the low end of complexity, but the proof based on it generalizes easily. We outline it here.

● In the general case, we have some finite number *n* of constraints. But there is no dependence on *n* in the proof. The argument runs component-wise, and entailment between ERCs holds when it holds in every component. It doesn't matter how many constraints there are.

● In the general case, we have some finite number *m* of VTs and their typology, which is the set of grammars that are produced by every choice of optimum, one from each participating VT. The proof establishes that the Minkowski sum of *two* VTs yields a VT with a typology that is identical to the typology obtained from the original two VTs. We can use this fact to roll up the set of m VTs in a 2-by-2 fashion. Let $\Sigma = \{VT_1,\ldots,VT_m\}$. Create Σ' by replacing VT_1 and VT_2 in Σ by $VT_1 \oplus VT_2$. Now $\Sigma' = \{VT_1 \oplus VT_2,\ldots\}$ has exactly the same typology as Σ, because as we've shown, the Minkowski sum $VT_1 \oplus VT_2$ has the same typology as $\{VT_1,VT_2\}$. If $m = 2$, we're done. If $m > 2$, simply repeat the process on Σ' to create $\Sigma'' = \{ (VT_1 \oplus VT_2) \oplus VT_3,\ldots\}$, and so on until the entirety of Σ has been summed to $\oplus\Sigma$, preserving equivalence at each step. Thus for any collection Σ of VTs, its Minkowski sum $\oplus\Sigma$ has exactly the same intensional typology as Σ.

3 Perspectives

We conclude with four observations.

1. **Summing optima only**. Every typology is generable from a finite collection of csets, a *universal support* for the typology, where each cset is represented by a single VT. In constructing a Unitary VT from a set of VTs, it is not necessary to produce the complete Minkowski sum of the original set, running through every possible choice of candidates, one per cset. We need only attend to each language of the typology as it is represented in the support, summing up the violation profiles of its optima. A candidate that is harmonically bounded in its own cset does not belong to any language of the extensional typology: the Minkowski candidates it participates in will be harmonically bounded in the UVT and therefore need to be included in the first place. Similarly, Minkowski candidates which contain incompatible optima from different csets will not correspond to languages and will also therefore be harmonically bounded in the UVT. In OTWorkplace (Prince, Tesar, and Merchant 2007-2015), the Minkowski sum method of creating a UVT (Factorial Typology>Unitary VT (Minkowski), alternatively ctrl+U) follows this strategy.

2. **Order among grammars**. Stripped of redundant and harmonically bounded rows, a Minkowski sum tableau will contain one row for each language in the typology. Each row in the reduced tableau supplies a single violation profile that generates an entire grammar when asserted as the optimal. The numerical values assigned by any $C^{\oplus}$ in a UVT impose an order structure on the grammars of the typology with respect to C. Entire grammars may then be said, accurately, to be evaluated by the constraints of the system in that UVT. Thus, one grammar is rated as better or worse than another, or identically the same, with respect to a given constraint and a given support. We can then meaningfully say that one language is more marked, or less faithful in a certain way than another, based on these order relations. In fact, we *must* say this, since these relations are not artificially imposed but follow from the way Gen_S and Con_S are defined for any system S.

These numerically-based relationships will typically vary in certain ways from support to support and from UVT to UVT among the (large) set of UVTs that generate a given typology T. Of particular interest, then, are those relations between grammars are *the same* in every UVT yielding T. These do not depend on the particularities of any UVT for T, and are independent of any of its UVT representations, constituting therefore an intrinsic property of the typology itself.

Merchant & Prince 2016 construct an order and equivalence structure that they show to encapsulate the invariant relations, the "Mother of All Tableaux" or MOAT. From MOAT(T) for a given typology T, one may derive every UVT that yields T. In addition one may determine whether a given collection of grammars constitutes a typology, and, most usefully, how the grammars of a typology may be amalgamated to form classes of grammars which are themselves the grammars of another more abstract typology, a basic step in the analysis of typological structures (Alber & Prince, 2015, in prep., Alber, Delbusso, and Prince 2016).

3. **The typology as an abstract object**. An OT grammar is known to be a certain kind of abstract object: a set of linear orders delimitable by an ERC set, and therefore an *antimatroid* (Merchant & Riggle 2016). This means that we can, and perhaps must, pursue linguistics under OT using the techniques of the theoretical sciences, in which the objects of a theory are a focus of analysis, building systematically upon the theory of consistent ERC sets. (On the matter of obligation, see Prince 2007.) As for *grammar,* so for *typology*. We can ask for analytically useful characterizations of the kind of object an (intensional) OT typology is, just as we asked the same question about grammars. Recall that it is typical for the objects and relations of a theory to have multiple characterizations, providing different avenues of analysis and generalization, and the search for these is an essential part of formal methodology.

With the toehold supplied by the UVT, we can define a typology as a certain kind of partition[6] of the set of all orders on $\text{C}\textsc{on}_S$: a partition derivable from a UVT via the standard definition of optimality in OT, where a UVT is just a matrix of nonnegative integers with columns associated with constraints and rows with grammars (Merchant & Prince 2016). From this we can advance to full abstractness, identifying a typology as partition of the set of all orders on $\text{C}\textsc{on}_S$ with an acyclic MOAT, as shown by Merchant & Prince (§3.3.3), and no numbers in sight. Both grammar and typology, then, are specific types of acyclic order structures, equivalently specific types of consistent logical systems, and can be studied as such, with concrete instantiations inheriting the formal properties discerned at the abstract level.

4. **The lattice of typologies**. In the lattice of all typologies, the Minkowski sum allows us to reason about and compute the meet of two elements. But what is *the lattice of all typologies*? Here we will briefly characterize it and indicate how it is shown to be a lattice, pointing as well to the reasons why it is of interest.

With an OT typology understood as a certain kind of partition, we can develop a coherent notion of the *set* of all typologies that have the same constraints. A 'constraint' is now simply a member of the set of opaque objects whose linear orders are gathered into the sets, abstract grammars, that form the blocks of a partition. For example, if we denominate the objects of a four-element set as S = {x, y, z ,w}, we can study the partitions of a set that we will denote as S!, which

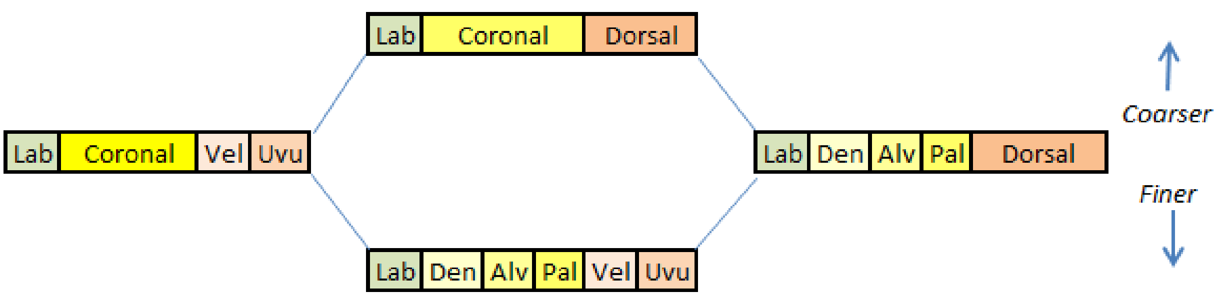
[6] A a partition of a set S is a collection of nonempty pairwise disjoint subsets X⊆S which are such that their union is S. The elements of a partition are known as 'parts' or 'blocks'.

contains the permutations of S — xyzw, yxzw, and so on. For any set A, let's denote by $\mathbb{P}$(A) the set of all partitions of A. Using the criteria established by Merchant & Prince, we can determine which of the members of $\mathbb{P}$(S!) qualify as OT typologies, giving us the set of all typologies on the constraints of S, namely $\mathbb{T}(S!) \subseteq \mathbb{P}(S!)$.

Any set of partitions of a given set can be equipped with the partial order of *refinement*. A partition $\pi_1 \in \mathbb{P}(A)$ for some set *A* is 'finer' than another partition $\pi_2 \in \mathbb{P}(A)$, written $\pi_1 \leq \pi_2$, iff every block of π_1 is a subset of a block of π_2. The other way around, we can say that if $\pi_1 \leq \pi_2$, then π_2 is 'coarser' than, or 'coarsens', π_1. The blocks of the coarser subsume the blocks of the finer; the blocks of the finer split up blocks of the coarser. What this means, qualititatively, is that a refinement of a given partition respects all its distinctions, but may impose more.

For example, let the places of articulation {labial, dental, alveolar, palatal, velar, uvular} partition the set of suprapharyngeal consonants into 6 blocks. We may think of each as the name of a set, e.g. labial = {p, b, f, v, φ, β, m}, and so on. The place categories {labial, *coronal*, velar, uvular} also partition the same set, but lump dental, alveolar, and palatal stops into one category, 'coronal'. The first partition is strictly finer than the second: the dental stops are a subset of the coronal group, as are the alveolars and palatals. The set {labial, coronal, *dorsal*}, with dorsal = velar ∪ uvular, coarsens the second partition yet further. And if we put each consonant in it own set, we have a maximally fine partition from which all the others may derived by coarsening. As this example suggests, coarsening represents generalization, and refinement its opposite. For discussion, see Prince 2013.)

These effects can be seen in the following diagram.



Under the refinement/coarsening order, the set of all partititions of S! forms a *lattice* $\langle \mathbb{P}(S!), \leq \rangle$. A lattice is a partially ordered set in which every pair of elements has a unique meet (greatest lower bound, infimum) and a unique join (least upper bound, supremum).[7] These restrictions endow the lattice object with a startling amount of useful structure.

Because $\mathbb{T}(S!) \subseteq \mathbb{P}(S!)$, we have immediately that $\mathbb{T}$(S!) is also partially ordered by refinement. In this case, the blocks are grammars, and coarsening is the amalgamation of grammars into larger sets that are also, formally, grammars. For example, the typology of the stress system nGX

[7] Observe that the notions upper/lower bound, and the greatest and least of these, are defined with respect to the partial order on the set under discussion. Refinement of partitions gives us the partial order we need to construct the lattice of partititions, and, as we show immediately below, the lattice of typologies.

(Alber & Prince 2015; Alber, DelBusso, & Prince 2015), splits exactly in two along the lambic vs. trochaic distinction. This split corresponds to an abstract typology $\mathcal{F}$ on $\textsc{Con}_{nGX}$! with just two blocks {Iambic, Trochaic}. $\mathcal{F}$ is coarser than nGX itself, so that we may write nGX$\leq\mathcal{F}$. Every grammar of nGX, construed as a set of linear orders, is a subset of one and only one of the two blocks in $\mathcal{F}$, satisfying the definition of coarsening. The typology nGX refines the (abstract) typology $\mathcal{F}$, maintaining the iambic/trochaic distinction and imposing others, which are concerned with the positioning, unarity, and density of feet in output forms.

$\mathbb{T}$(S!) is also a lattice. This is not an immediate consequence of the fact that $\mathbb{T}$(S!)$\subseteq\mathbb{P}$(S!), because to qualify as a lattice, $\mathbb{T}$(S!) must have unique meets and joins within it for each pair of elements $T_i,T_j\in\mathbb{T}$(S!). To show latticehood requires proving the existence and uniqueness of these objects.

As a step in this direction, we first observe that $\mathbb{T}$(S!) is a *meet semilattice*: each pair of elements T_i, $T_j\in\mathbb{T}$(S!) has a unique meet $T_i\wedge T_j\in\mathbb{T}$(S!). To see why, note that the greatest lower bound (meet) of any two elements in the full partition lattice $\mathbb{P}$(S!) under the refinement order is known to be equal to the set of all nonempty intersections of pairs of blocks, one from each participant in the meet.

(10) **Fact. Meet in a Partition Lattice**. For partitions $\pi_1, \pi_2\in\mathbb{P}(A)$, the set of partitions of some set A, the meet $\pi_1\wedge\pi_2$ is given by:

$$\pi_1\wedge\pi_2 = \{B\cap D|\ B\in\pi_1,\ D\in\pi_2,\ B\cap D\neq\varnothing\}.$$

Now consider two typologies $T_i,T_j\in\mathbb{T}$(S!)$\subseteq\mathbb{P}$(S!). Let $T_i\wedge T_j$ be as in (10). We need to show that:

(1) $T_i\wedge T_j$ *is* a typology, and therefore a member of $\mathbb{T}$(S!), and

(2) $T_i\wedge T_j$ is the greatest lower bound of $\{T_i, T_j\}$ in $\mathbb{T}$(S!).

As for (1), we know that T_i can be associated with a UVT_i that delimits it; similarly for T_j. Now consider $UVT_i\oplus UVT_j$. Its elements are of the form **a**+**x**, for **a**$\in UVT_i$, **x**$\in UVT_j$. From UVT_i the element **a+x** inherits all the ERCs expressing the conditions required for **a** to be optimal in T_i, and all the ERCs associated with the optimality of **x** in T_j, exactly as we saw above. The optimality of **a+x** therefore requires the conjunction of these conditions, equivalently the union of the two ERC sets. Let **a** be associated with the grammar $\hat{\mathbf{a}}\in T_i$ and **x** with the grammar $\hat{\mathbf{x}}\in T_j$. Then **a+x** is associated with $\hat{\mathbf{a}}\cap\hat{\mathbf{x}}$, the grammar simultaneously meeting both sets of conditions. The typology of $UVT_i\oplus UVT_j$ therefore contains as its grammars exactly the nonempty intersections of one grammar from T_i and another from T_j. This is $T_i\wedge T_j$, establishing $T_i\wedge T_j\in\mathbb{T}$(S!).

Assertion (2) follows because $T_i\wedge T_j$ is also the meet of $\{T_i, T_j\}$ in the full partition lattice $\mathbb{P}(S^!)$, which includes all of the elements of $\mathbb{T}(S^!)$.

To complete the proof, we call on a general result that relates meet semilattices to lattices: *if a finite ordered set P has a greatest element and every pair of elements has a meet, then P is a lattice* (Nation, p. 17). The greatest element of $\mathbb{T}$(S!) is its coarsest: the typology that every other typology refines. This typology has only one block, the trivial grammar with no ranking restrictions at all, which consists of S! in toto. It follows that the set of all typologies is a lattice under refinement.

A lattice describes patterns of generalization that hold between its elements (as discussed for example in Prince 2013). Understanding the intrinsic strucrure of OT typologies, the *classification program* of Alber & Prince (2015, in prep.), has its natural starting place in the lattice of OT typologies. From this perspective, a set of grammars in a typology can be classed together if their union forms a single grammar in a generalized, coarser typology. The UVT gives us a formal tool for defining and dealing with the typology as the central object of OT.

■

References

Alber, B. 2005. Clash, lapse and directionality. *Natural Language* & *Linguistic Theory* 23, 485-542.

Alber, B. N. DelBusso and A. Prince. 2015. From Intensional Properties to Universal Support. LANGUAGE: Phonological Analysis, 92.2. ROA-1235.

Alber, B. and A. Prince. 2015. Outline of Property Theory. (Entwurf einer verallgemeinerten Eigenschaftentheorie). Ms. Rutgers University and Università degli Studi di Verona.

Alber, B. and A. Prince. In prep. *Typologies*. Ms., University of Verona and Rutgers University, New Brunswick.

Brasoveanu, A. & A. Prince. 2005/2011. Ranking and Necessity: the Fusional Reduction Algorithm. ROA-794.

McCarthy, J. and A. Prince. 1995. Faithfulness and Reduplicative Identity. ROA-60.

Merchant, N. & Prince, A. 2016. *The Mother of All Tableaux.* Ms. Eckerd College and Rutgers University, New Brunswick.

Merchant, N. and J. Riggle. 2016. Erc sets and Antimatroids. *Natural Language* & *Linguistic Theory*: 34.1, 241-269. ROA-1158.

Nation, J.B. undated. *Notes on Lattice Theory*. http://math.hawaii.edu/~jb/math618/LTNotes.pdf

Prince, A. 2002a. *Entailed Ranking Arguments*. ROA 500.

Prince, A. 2002b. Arguing Optimality. In A. Coetzee, A. Carpenter, & P. de Lacy (eds.) *Papers in Optimality Theory II.* Amherst, MA. ROA 562.

Prince, A. 2007. The Pursuit of Theory. In The *Cambridge Handook of Phonology*, p. 22-46.

Prince, A. 2013. Metrical Theory as a Portal on Theory. YouTube Video.

Prince, A. and P. Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell and ROA-537.

Prince, A., B. Tesar, and N. Merchant. 2015. OTWorkplace. Installer Package. Updates.

Riggle, J. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory. PhD dissertation, UCLA.