

The Holographic Principle: Typological Analysis Using Lower Dimensions

Nazarré Merchant and Martin Krämer

Eckerd College and University of Tromsø

1 Introduction

A moderately complex factorial typology may consist of tens or hundreds of languages which can opaquely encode linguistically salient categories and generalizations. We propose in this paper that these complex typologies can be decomposed and understood using what we call the *holographic principle*: a large typology can be projected onto simplified versions of itself which can be completely understood using Property Theory (Alber & Prince 2016). The simplified versions can then be re-incorporated into the original in such a way that the properties of the simple are maintained and provide a framework for analyzing the full system.

In this paper, we demonstrate this technique using two systems, a *basic stringency system* (BSS), and a *coda stringency system* (CSS). We show how a complete analysis of BSS, using Property Theory, provides fundamental insights into the more complicated CSS which BSS is a simplification of. A *property analysis* is a set of *properties* that divide the languages of the typology in such a way that each language and its grammar can be identified uniquely by its property values. Such an analysis identifies the crucial rankings among constraints that distinguish all grammars of the typology so that languages that share property values share extensional traits.

BSS generalizes systems in which there is one stringency hierarchy (see Prince 1997 for a comprehensive discussion, de Lacy's (2006) typology of sonority-driven unstressed vowel reduction or Alderete's (2008) analysis of stress in the Pama-Nyungan language family for further examples and DelBusso (2018) for an insightful perspective relating stringency hierarchies and EPO¹ structures). The constraints of BSS consist of four markedness constraints and one faithfulness constraint. The markedness constraints form a stringency hierarchy in which each markedness constraint is in a stringency relationship with every other markedness constraint. For constraints X and Y to be in a stringency relationship we mean $X(\alpha) \leq Y(\alpha)$ for all candidates α of the system (Prince 1997), with the additional requirement of 'transference' (discussed in §2). This stringency hierarchy imposes a markedness hierarchy on the forms of the system in which every form of the system has a unique position on the markedness hierarchy. This yields a total order on the forms. We then show, using Property Analysis, that each grammar in the typology is completely determined by the lowest unfaithfully mapped form on the markedness hierarchy. This result applies to all stringency systems in which there is one stringency hierarchy.

CSS is an analysis of the system presented in Krämer & Zec (2017)'s typology of manners in the syllable coda. There are seven constraints in the system, one faithfulness constraint and two stringently ordered sets of markedness constraints, an F-scale set and a P-scale set, each comprised of three constraints. The F-scale consists of a constraint against fricatives, one against fricatives and liquids, and one against fricatives, liquids and nasals. The P-scale follows the same building principle based on the category of stops. Each of the stringency hierarchies imposes an independent markedness hierarchy on the forms of the system. We give a property analysis of CSS in which the properties are organized in a parallel manner to the properties of BSS. The basic system embeds in CSS in that each stringency hierarchy in CSS has a set of properties associated with it that are structurally identical to the properties of BSS. As in BSS, a grammar's mappings in CSS are determined by where on each of the markedness hierarchies the language is first unfaithful. This shared extensional trait in BSS and CSS manifests as structurally identical properties.

Stringency systems vary in their complexity from the number of classes they refer to, to how they interact, either with another orthogonal and conflicting stringency set (e.g., Alber 2001's analysis of

* Thanks to Birgit Alber, Nalatie DelBusso, Alan Prince, Draga Zec, and audiences at UiT, UCSC, and the Manchester Phonology Meeting.

¹ Equivalence-augmented privileged orders (Merchant and Prince 2018).

regional variation in glottal stop insertion in German), with one conflicting constraint (e.g., the vowel reduction patterns alluded to above) or another parallel stringency set and a conflicting constraint (e.g., the coda manner typology). In this paper we show how the structure of a maximally reduced stringency system is reproduced using the holographic principle in the more complex system via its properties. Understanding the relations that inhere between the simple and the complex is central to explicating larger typologies that defy easy analysis.

2 Stringency

Here we define what we mean by a stringency relation between constraints (our definition is the same as Prince 1997 with the addition of the ‘transference’ assumption). A constraint C is said to be *less stringent* than a constraint D exactly when it satisfies the following three criteria.

- (1) Constraint C is *less stringent than* D when
- a. For each candidate of the system, α , $C(\alpha) \leq D(\alpha)$ (Ordered)
 - b. For each pair of candidates in a candidate set, α and β ,
if $C(\alpha) \leq C(\beta)$, then $D(\alpha) \leq D(\beta)$ (Transference)
 - c. There is some candidate β such that $C(\beta) < D(\beta)$ (Distinction)

The first property says that two constraints in a stringency relationship have an inherent numerical valuation order. A violation count of n on the less stringent constraint on a candidate ensures that the more stringent constraint on that candidate will incur n or more violations. Transference allows cross-candidate reasoning: a candidate having fewer violations than another candidate on constraint C will also have fewer violations than that candidate on constraint D . The final criterion ensures that the constraints are distinct on at least one candidate.

A stringency hierarchy, $S = \{C_1, \dots, C_n\}$, then is a set of constraints in which each sequential pair of constraints, C_i and C_{i+1} , has the property that C_i is less stringent than C_{i+1} . Note that nothing in the definition requires that the constraints be markedness constraints, even though that is the usual construal. If the constraints are all markedness constraints we say that the stringency hierarchy is a *markedness stringency hierarchy*. Merchant and Krämer 2018 show that a markedness stringency hierarchy imposes a total order on the classes of outputs that the constraints of the markedness hierarchy distinguish. Merchant and Krämer call this the *markedness hierarchy relative to S*. This total order imposed by the stringency hierarchy is defined by the relation $\alpha <^S \beta$ if there is some constraint $C \in S$ such that $C(\alpha) < C(\beta)$. We signify that this relation is relative to the stringency hierarchy S by superscripting the less than symbol, $<^S$. It is crucial to do so, because a given system may have multiple stringency hierarchies, each of which imposes a distinct order relation on the outputs of the system, as is the case in the CSS discussed in §3 and, e.g., Alber (2001). And each stringency hierarchy imposes a distinct relative markedness hierarchy.

To see this in action, consider the following system, which we call the *Basic Stringency System* (BSS). GEN of BSS, denoted GEN_{BSS} , consists of inputs, outputs, and sets of candidates, one set for each input. Each candidate is an ordered pair, (in, out) , where in is an element of the input set and out is an element of the output set. The input set, In_{BSS} , is comprised of four elements: A, B, C, and D. The output set, Out_{BSS} , is identical to the input set. There are no restrictions on which output is associated with which input in a candidate, representing that any input can map to any output in BSS. These facts are summarized in (2).

- (2) GEN_{BSS}
- | | |
|------------------------------|--|
| In_{BSS} | $\{A, B, C, D\}$ |
| Out_{BSS} | $\{A, B, C, D\}$ |
| $\text{GEN}_{\text{BSS}}(x)$ | $\{(x, A), (x, B), (x, C), (x, D) \mid x \in \text{In}_{\text{BSS}}\}$ |

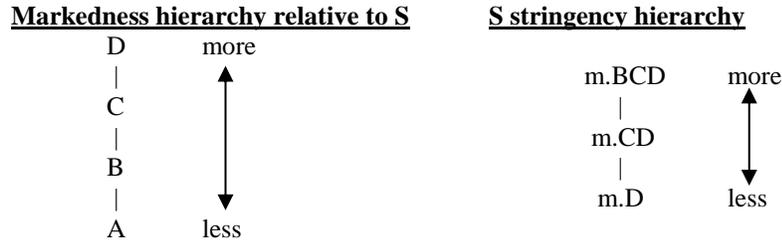
There are four constraints in BSS. We prefix markedness constraints with ‘m.’ to distinguish them from faithfulness constraints. Three of the constraints are markedness constraints: m.BCD, m.CD, and m.D. The constraint m.BCD incurs a violation when the output of a candidate is either B, C, or D; m.CD incurs a violation for an output of C or D; and m.D for an output D. There is one faithfulness constraint, Faith, which is violated once for any candidate in which the input is not identical to the output. The constraints

are listed in (3).

- (3) CON_{BSS}
- | | | |
|-------|--------------------------|--|
| m.BCD | *B, C, or D | 'Do not have B, C or D' |
| m.CD | *C or D | 'Do not have C or D' |
| m.D | *D | 'Do not have D' |
| Faith | *(X, Y) where $X \neq Y$ | 'Do not have the input differ from the output' |

It is straightforward to see that the markedness constraints $S = \{m.D, m.CD, m.BCD\}$ form a markedness stringency hierarchy by satisfying the three conditions in (1). The order relation imposed on the outputs of the system are $A <^S B <^S C <^S D$ which is, by definition, the markedness hierarchy relative to S. This markedness hierarchy, represented graphically in (4) (along with the stringency hierarchy S), provides the central organizing structure for understanding the grammars of BSS. As shown in the next section, a language's mappings are entirely determined by the lowest position on the markedness hierarchy (least marked relative to S) that the language is unfaithful at. As we will also show, this markedness structure manifests directly in the property analysis of BSS.

- (4) Markedness hierarchy of BSS relative to $S = \{m.D, m.CD, m.BCD\}$ and the stringency hierarchy S



3 Analysis of Basic Stringency System (BSS)

The BSS system admits four languages (verified using OTWorkplace). Extensionally, it is straightforward to understand the mappings that obtain. The first language, fA, is faithful at A and unfaithfully maps all other inputs to A. Language fB faithfully maps inputs A and B while mapping C and D to A. Language fC faithfully maps input A, B, and C while mapping D to A. Finally, language fD maps each input to itself.

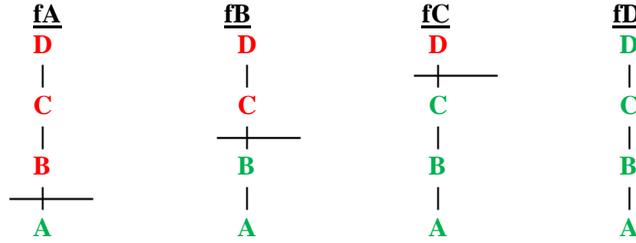
- (5) Basic Stringency System (BSS) Factorial Typology

fA	fB	fC	fD
D → A	D → A	D → A	D → D
C → A	C → A	C → C	C → C
B → A	B → B	B → B	B → B
A → A	A → A	A → A	A → A

3.a Extensional Patterns of BSS

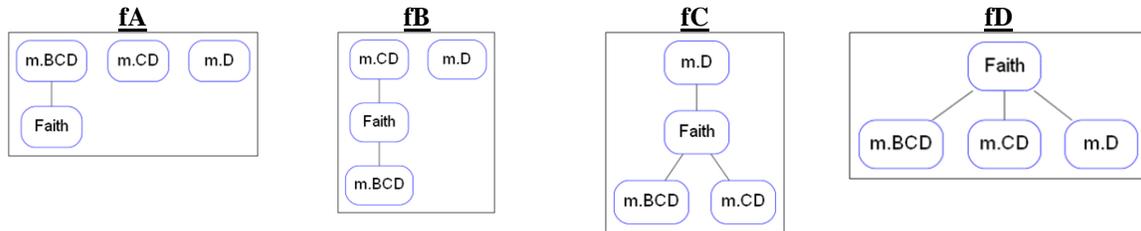
The central organizing structure of the BSS typology is the markedness hierarchy $A <^S B <^S C <^S D$ with respect to the stringency hierarchy $S = \{m.D, m.CD, m.BCD\}$. We arrange the output forms vertically with the least marked form at the bottom and the most marked at the top. A language's mappings then are entirely determined by the last faithfully mapped form on this scale. All unfaithfully mapped forms map to A. So, language fA maps $A \rightarrow A$ but maps all other forms unfaithfully (and therefore to A). The dividing line between faithful and unfaithful is between A and B on the markedness scale. Each subsequent language, as shown in (6), draws its faithfulness dividing line one step higher on the markedness scale. The dividing line separates the faithful (here rendered in green) below from the unfaithful (rendered in red) above.

(6) Patterns of the BSS languages



This hierarchy completely answers the extensional pattern question. A language is determined by which two bordering forms differ in faithfulness, if there are any. At this stage, there is no connection to the grammars and the extensional pattern appears almost as epiphenomenal. We will see in §4 that our property analysis directly ties this pattern to the grammars of the typology. To this end, the grammars of BSS are given in (7).

(7) Grammars of BSS



4 Property Analysis

A property analysis of an OT typology consists of a set of *properties*, binary choices that some, possibly all, languages of the typology must decide between (if every language of the typology has a value for a given property, the property is called a *wide-scope* property; the property is called *narrow-scope* if a strict subset of the languages of the typology choose). A given property, P , has two values, $P.a$ and $P.b$, representing the two choices the property encodes. Each value has a set of mutually exclusive ranking conditions that languages having a respective value must satisfy. The grammar of a language, in a property analysis, must be exactly the ranking conditions determined by the property choices the language makes.

Property values enforce ranking conditions on the total orders of a grammar via the *sub* and *dom* operators. A typical property value is of the form $K.op_1 \gg J.op_2$ where $op_1, op_2 \in \{\text{sub}, \text{dom}\}$ and K and J are classes of constraints. The sub operator selects, for a total order λ and constraint class K , the lowest ranked (subordinate) constraint from K in λ . The dom operator selects, for a total order λ and class K , the highest ranked (dominant) constraint from K in λ . A single property, P , represents its two values by $P:K.op_1 \langle \rangle J.op_2$, where the $P.a$ value represents the ranking condition $K.op_1 \gg J.op_2$ while the $P.b$ value represents the ranking condition $J.op_2 \gg K.op_1$. If the class K , for example, consists of a single constraint then $K.sub$ is the same as $K.dom$, which again is identical to K . In this case we will omit the operator from the representation of the property. For more explicit and elaborate discussions and applications of PT see Alber & Prince (2016, 2018), Alber, DelBusso & Prince (2016), McManus (2017), and DelBusso (2018).

4.a Property analysis of BSS

We propose a property analysis of BSS consisting of three properties $P_D:m.BCD \langle \rangle \text{Faith}$, $P_C:m.BC \langle \rangle \text{Faith}$, and $P_B:m.B \langle \rangle \text{Faith}$. The naming of the properties encodes the extensional characterization the property represents. For property P_X , languages having the $P_X.a$ value (having the markedness constraint dominate Faith) is unfaithful on form X , while the $P_X.b$ value languages are faithful on form X . The properties and

their values along with representative ERCs² (the constraint order of the ERCs is m.BCD, m.CD, m.D, and Faith) are given in (8).

(8) Properties of BSS

Property	Property Value	Ranking Requirement	ERC
P _B : m.BCD<>Faith	P _B .un	m.BCD≫Faith	WeeL
	P _B .f	Faith≫m.BCD	LeeW
P _C : m.CD<>Faith	P _C .un	m.CD≫Faith	eWeL
	P _C .f	Faith≫m.CD	eLeW
P _D : m.D<>Faith	P _D .un	m.D≫Faith	eeWL
	P _D .f	Faith≫m.D	eeLW

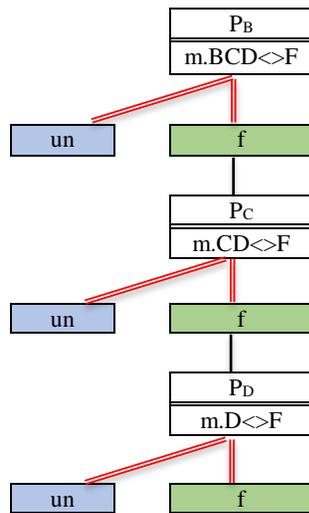
The property P_B has wide scope: all languages of the BSS typology select either the value P_B.un or P_B.f. The property P_C is a narrow scope property, all languages except for fA select a value of P_C. Finally, property P_D is narrow scope, only applying to languages fC and fD.

The telescoping nature of the three properties (all languages must select a P_B value, three of the languages select a P_C value, and two of those three select P_D) exactly mirrors the grammatical choices made when interrogating the markedness hierarchy, A<^SB<^SC<^SD, starting at B and moving up the hierarchy. All languages must decide whether they are faithful or not at B. If unfaithful, then all choices are made, everything maps to A. This choice is exactly P_B.un. If B is faithfully mapped, then the languages face the same choice at C on the hierarchy. If unfaithful at C, then C and D map to A (B is faithfully mapped because of the P_B.f choice) and all mappings are determined. This is P_C.un. The final possibility for those languages having chosen P_B.f and P_C.f is whether to faithfully map D or not. The values of P_D represent those two options. These series of choices are represented in a treeoid (Alber & Prince 2018), a graphical representation of the dependencies among the property values. The treeoid for this PA of BSS is given in (9).

The treeoid of BSS has the same structure as the markedness hierarchy imposed on the forms of BSS, A<^SB<^SC<^SD only with the order inverted. Choice of where to be unfaithful on the markedness hierarchy is precisely represented in the treeoid. This is represented in the grammars of BSS also, each of which is determined by the Faith constraint's position relative to the markedness constraints. Language fA, which is faithful only to A, has its grammar entirely delimited by the property value P_B.a which yields the ERC WeeL (m.BCD ≫ Faith). Language fB is comprised of P_C.un and P_B.f which together yield the ranking requirements of m.BC≫Faith and Faith≫m.BCD, which together give fB's grammar. The grammars of fC and fD follow similarly. The property values of these grammars are shown in (10).

² ERC = Elementary Ranking Condition. See Prince 2002.

(9) Treeoid for PA of BSS (constraint Faith is abbreviated F)



(10) Property values of the languages of BSS

	P _B	P _C	P _D
fA	un		
fB	f	un	
fC	f	f	un
fD	f	f	f

The property analysis of the BSS typology straightforwardly encodes the mappings seen in the markedness hierarchy. A surprising result is that this nearly maximally simple system is reproduced in more complicated systems that have multiple markedness hierarchies. This is shown in the CSS system in §5.

5 The Coda Stringency System (CSS)

Krämer & Zec (2017, henceforth K&Z) compiled a database of over 200 languages with wide genetic and geographic distribution to investigate markedness of different consonant classes in coda position defined by manner of articulation. They ignore laryngeal segments, i.e., languages that allow only [ʔ] and [h] in the coda are not counted as allowing fricatives and stops. Glides were not counted either. With these restrictions they found that gross coda patterns can be described by considering each segment as belonging to the distinct classes P, F, N, L, which describe stops, fricatives, nasal consonants (often also referred to as nasal stops) and liquids (lateral and rhotic), respectively, which accords with the classification proposed by the authors in the literature consulted. In some cases, segments were reanalyzed because there was disagreement in the literature and a position had to be taken, or the description didn't match the analysis.

(11) Main consonantal coda categories

- P (oral stops)
- F (fricatives)
- N (nasals)
- L (liquids, approximants or laterals, and rhotics)

Codas were divided into two types, word-internal codas and word-final codas. Given the data situation and the large number of languages, it was not possible to make further distinctions, e.g., codas in stressed syllables versus those in unstressed syllables etc. For word-internal codas, special care was taken to detect restrictions that should not be considered pure coda restrictions but rather syntagmatic relations, such as the Syllable Contact Law (Vennemann 1988), which restricts a preconsonantal inventory in relation to the sonority of the following onsets in that the coda has to at least match the sonority of the following onset or

be higher on the scale. Sometimes, only individual members of a class were systematically missing from the coda position (e.g., the rhotic but not the lateral or vice versa) or only a subset was excluded, as by final devoicing or a coda condition on place of articulation. In such cases the manner class was assumed to be present.

The prevailing opinion about manner neutralization in the coda is that it is dependent on the sonority hierarchy. Lower sonority is more marked in the coda and languages draw a line on which classes they permit in the coda, with the classes below that line on the sonority hierarchy being banned (Vennemann 1988, Prince & Smolensky 1993, Baertsch 2002, Green et al. 2007, Baertsch & Davis 2001, Bye & de Lacy 2008). K&Z did not corroborate these claims on high sonority but rather find a dominance of N. The patterns they found are summarized in the following table.

(12) Surface systems

1 class	P	F	N
2 classes	PN	FN	NL
3 classes	PNL	FNL	PFN
4 classes	PFNL		

Following the logic of OT to posit negative markedness constraints, it is potentially more instructive to look at the banned classes. I.e., if only P is allowed, FLN are absent. If only F is present, PNL are absent.

(13) Banned classes

1 class	P	F	L
2 classes	PL	FL	PF
3 classes	PFL	FLN	PNL
4 classes	PFNL		

From this one can draw several crucial observations. First, nasals are only banned if other classes are banned too. The two-class systems PF, PL, and FL are unattested, as is the three-class system PFL. While this points at an implicational hierarchy, the rest of the generalizations don't straightforwardly point that way. There are languages that ban each of the other classes individually or in any combination. Furthermore, while P and F can both be excluded, leaving only L and N, this is not the case for the sonorants, since we don't find the inventory PF. On the basis of these observations, K&Z develop two stringently organized sets of markedness constraints on which we base our property analysis.³ These sets will be discussed in more detail in the next subsection.

5.a The CSS and its Property Analysis

The Coda Stringency System (CSS) we introduce here abstracts away from onsets, nuclei, deletion and insertion to distill the patterns present in the languages described in §3.a. See K&Z for discussion of such issues. The system represents possible patterns in a single coda. There are four inputs to GEN_{CSS} : P, F, N, L, representing plosives, fricatives, nasals, and liquids respectively. The possible outputs for a given input are taken from the same collection as the inputs: P, F, N and L. Any input can map to any output. We represent a candidate by the pair (x, y) where x is an element of the $In_{CSS} = \{P, F, N, L\}$ and y is an element of $Out_{CSS} = \{P, F, N, L\}$. These facts about GEN_{CSS} are summarized in (14).

(14) GEN_{CSS}

In_{CSS}	$\{P, F, N, L\}$
Out_{CSS}	$\{P, F, N, L\}$
$GEN_{CSS}(x)$	$\{(x, P), (x, F), (x, N), (x, L)\}$

Note that if P, F, N, and L were replaced throughout with A, B, C, and D respectively, GEN_{CSS} would be

³ The pattern PFN requires an additional markedness constraint, m.L. For reasons of space we exclude this pattern from our analysis here, see Merchant & Krämer for discussion.

identical to GEN_{BSS} .

The constraints of CSS, CON_{CSS} , are given in (15).

(15) CON_{CSS}			
m.FLN	*F, L, or N		'Do not have F, L or N'
m.FL	*F or L		'Do not have F or L'
m.F	*F		'Do not have F'
m.PLN	*P, L, or N		'Do not have P, L or N'
m.PL	*P or L		'Do not have P or L'
m.P	*P		'Do not have P'
Faith	*(X, Y) where $X \neq Y$		'Do not differ from the input'

A surprising result is that this system produces the target coda categories described in §3.a, so that the factorial typology of CSS, comprised of the thirteen languages listed in (16) verified using OTWorkplace (Prince, et al.), align with M&K's coda classes.

(16) Factorial typology of CSS

Coda Inputs->	P	N	L	F
1.PPPP	P	P	P	P
2.PNPP	P	N	P	P
3.PNLP	P	N	L	P
4.PNLF	P	N	L	F
5.PNNN	P	N	N	N
6.PNLN	P	N	L	N
7.FFFF	F	F	F	F
8.FNFF	F	N	F	F
9.FNLF	F	N	L	F
10.NNNF	N	N	N	F
11.NNLF	N	N	L	F
12.NNNN	N	N	N	N
13.NNLN	N	N	L	N

To understand the languages of CSS we engage in *projection onto a lower dimension* of CSS. We define the dimension of system to be the number of constraints in the system, so the dimension of CSS is 7. We further define that an n -dimensional typology, T^n , is projected onto a k -dimensional system, S^k , when Gen of S^k is identical to Gen of T^n and when the constraint set of S^k is a subset of the constraint set of T^n . One can think of the projection of T^n onto S^k as simply removing $n-k$ constraints from T^n – this produces exactly the system S^k .

Here we project CSS onto the lower-dimensional system we call F-Sys by removing the constraints {m.P, m.PL, m.PLN}. This leaves F-Sys with four constraints and a GEN identical to CSS; these are shown in (17).

(17) GEN and CON of F-Sys

GEN_{F-Sys} (= GEN_{CSS})		CON_{F-Sys}
In_{F-Sys}	{P, F, N, L}	m.FLN
Out_{F-Sys}	{P, F, N, L}	m.FL
$GEN_{F-Sys}(X)$	{(x, P), (x, F), (x, N), (x, L)}	m.F
		Faith

A crucial fact of F-Sys is that it is structurally identical to the BSS system presented in §2. This identity is obtained by changing P for A, N for B, L for C, and F for D. This entails that the previous analysis of BSS can be brought to bear on F-Sys, immediately recognizing its stringency hierarchy $sF = \{m.F, m.FL,$

$m.FLN\}$ and the induced markedness hierarchy relative to sF : $P \prec^{sF} N \prec^{sF} L \prec^{sF} F$. Furthermore, the property analysis of BSS can be adopted (with constraints appropriated swapped) without modification. The factorial typology and property analysis of F-Sys are given in (18).

(18) Factorial typology and property analysis of F-Sys

Fs-P	Fs-N	Fs-L	Fs-F	Properties of F-Sys	
$F \rightarrow P$	$F \rightarrow P$	$F \rightarrow P$	$F \rightarrow F$		$P.F_N: m.NLF \langle \rangle \text{Faith}$
$L \rightarrow P$	$L \rightarrow P$	$L \rightarrow L$	$L \rightarrow L$		$P.F_L: m.LF \langle \rangle \text{Faith}$
$N \rightarrow P$	$N \rightarrow N$	$N \rightarrow N$	$N \rightarrow N$		$P.F_F: m.F \langle \rangle \text{Faith}$
$P \rightarrow P$	$P \rightarrow P$	$P \rightarrow P$	$P \rightarrow P$		

Interpreting the properties and languages follows directly from our understanding of BSS. A language's mappings are determined by the lowest position in the markedness hierarchy relative to sF the language is unfaithful, and all unfaithful forms map to P. Of note is that viewed as languages shorn of their grammars, and so only considered purely as their mappings from inputs to outputs, these four languages manifest in the CSS. We will see below that the CSS property analysis incorporates exactly the properties of F-Sys. This is possible because the constraints of F-Sys are a subset of the constraints of CSS and so each property of F-Sys is a candidate property of CSS. This, of course, is true in general for two systems one of which is a projection of the other.

We now project CSS onto a different lower dimensional space we call the P-Sys. We do so by removing the constraints $\{m.F, m.FL, m.FLN\}$ yielding a system that again, is structurally identical to BSS (and also to F-Sys). The constraints of P-Sys are shown in (19). As always with project, $\text{Gen}_{P\text{-Sys}}$ is identical to Gen_{CSS} (and therefore to $\text{Gen}_{F\text{-Sys}}$).

(19) Constraints of P-Sys

$$\text{CON}_{P\text{-Sys}} = \{ m.PLN, m.PL, m.P, \text{Faith} \}$$

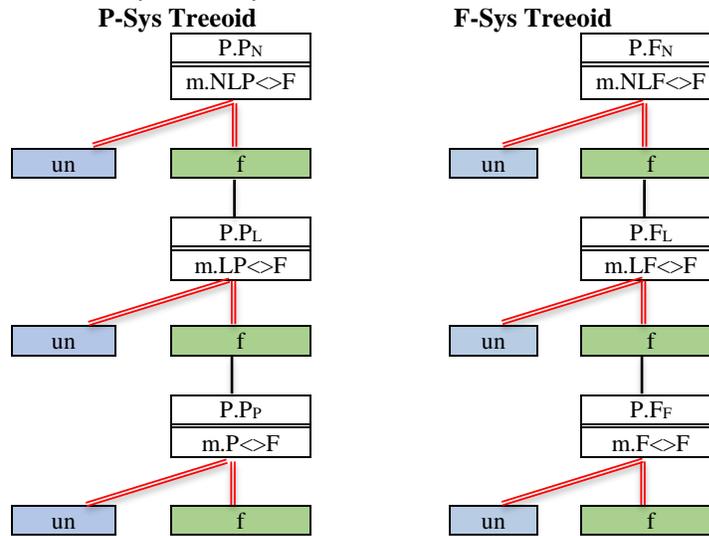
Given the structural identity of P-Sys, F-Sys, and BSS we immediately can produce the factorial typology and property analysis of P-Sys. This is given in (20).

(20) Factorial typology and property analysis of P-Sys

Ps-F	Ps-N	Ps-L	Ps-P	Properties of P-Sys	
$P \rightarrow F$	$P \rightarrow F$	$P \rightarrow F$	$P \rightarrow P$		$P.P_N: m.NLP \langle \rangle \text{Faith}$
$L \rightarrow F$	$L \rightarrow F$	$L \rightarrow L$	$L \rightarrow L$		$P.P_L: m.LP \langle \rangle \text{Faith}$
$N \rightarrow F$	$N \rightarrow N$	$N \rightarrow N$	$N \rightarrow N$		$P.P_P: m.P \langle \rangle \text{Faith}$
$F \rightarrow F$	$F \rightarrow F$	$F \rightarrow F$	$F \rightarrow F$		

Both the P-Sys and the F-Sys have structurally identical treeoids given below in (21).

(21) Treeoids of the P-Sys and F-Sys



The utility of projecting CSS onto the F-Sys and the P-Sys is that the stringency hierarchies present in the two lower-dimensional systems are also present in the CSS. This yields two different markedness hierarchies in the CSS, one for the stringency hierarchy $sF = \{m.F, m.FL, m.FLN\}$, yielding $P \langle^{sFN} \langle^{sFL} \langle^{sFF}$ and one for the hierarchy $pF = \{m.P, m.PL, m.PLN\}$, yielding $F \langle^{sPN} \langle^{sPL} \langle^{sPP}$. These two markedness hierarchies then induce identical mappings for a subset of the languages of CSS. Surprisingly, the properties of F-Sys and P-Sys can be directly incorporated into a property analysis of CSS. Of course, further properties are needed in the full CSS system. This need can be seen in the first language of CSS among others.

The first language of CSS listed above in (16) is 1.PPPP. We note that as a language (viewed only as mappings) it is identical to the first language in the F-Sys since both the CSS and F-Sys share a GEN and the mappings are the same. In fact, the first four languages of CSS are exactly the languages of F-Sys. Crucially, even though the first language of CSS is identical to the first language of F-Sys (and the second to the second, and so forth), their grammars are necessarily different. No knowledge of their respective rankings need even be known to deduce this difference: CON_{CSS} is not identical to CON_{F-Sys} , and so no grammar for any language could be the same even if they share every optima for every input, as they do here.

We briefly note that the grammar for 1.PPPP's correlate language in F-Sys, Fs-P, is $m.FLN \gg \text{Faith}$ and is completely determined by single the property value $P.F_N:m.FLN \gg \text{Faith}$. The grammar for 1.PPPP in CSS is notably more complex. Its ERC-representation is given in (22).

(22) Grammar of 1.PPPP in CSS

m.PLN	m.PL	m.P	m.FLN	m.FL	m.F	Faith
L	<i>e</i>	<i>e</i>	W	W	W	<i>e</i>
<i>e</i>	L	L	W	<i>e</i>	<i>e</i>	L

Obscured in the ERC representation of 1.PPPP is that its grammar is determined by three extensional questions that manifest in the property analysis of CSS. The first is, *which stringency hierarchy is dominant in this language, the sP or the sF?* We represent this choice with the property D-Sys, shown in (23).

(23) D-Sys: $\{m.FLN, m.FL, m.F\}.dom \diamond \{m.PLN, m.PL, m.P\}.dom$
 ERC representation: $LLL.WWW.e \diamond WWW.LLL.e^4$

The D-Sys.F value represents languages in which the sF is dominant and the D-Sys.P value represents

⁴ The constraint order for ERCs of the CSS is (m.PLN, m.PL, m.P, m.FLN, m.FL, m.F, Faith).

languages in which the sP is dominant. ‘Dominance’ of sF (respectively sP) is exhibited in a language by mapping the most marked form of the F-Sys, F, to the least marked form, P (and P to F in the the P-Sys). The 1.PPPP language co-resides in the F-Sys and so the language, in CSS, chooses the D-Sys.F value.

All languages in which one stringency hierarchy is dominant over the other then must answer where on the markedness hierarchy relative to the dominant stringency hierarchy they are first unfaithful. This question is answered exactly in CSS as it is in F-Sys and P-Sys. Here is where the importation of the lower-dimensional properties bears fruit. So, for example, the 1.PPPP language has property value D-Sys.F, and so we import the three properties of the F-Sys, repeated below, but now part of the property analysis of CSS. The grammar of 1.PPPP has the same values on these properties as its corresponding language’s grammar does in F-Sys.

(24) F-Sys incorporated properties of CSS

P.F_N: m.NLF <> Faith

P.F_L: m.LF <> Faith

P.F_F: m.F <> Faith

The language 1.PPPP must choose the unfaithful property value P.F_N.un, and it is moot on the other two properties.

The final property does not have a correspondent in either the F-Sys or the P-Sys. For a language like 1.PPPP, with dominant pF, the constraints that comprise the sF are still active in the CSS. This activity can manifest in the mappings of N and L. The extensional question is, if L maps unfaithfully, does it map to the least marked form of the dominant stringency hierarchy or does it map to N, the form that is second least marked in the markedness hierarchies relative to both pF and sF? Effectively, this is asking, *does the subordinate stringency hierarchy exert any force in the language?* In 1.PPPP, the answer is no, since the unfaithful N and L both map to P, the *most* marked form on the sP markedness hierarchy. This question has two property variants corresponding to the two questions, does the sP stringency hierarchy have any residue in a dominant sF language? And, does the sF stringency hierarchy have any residue in a dominant sP language? These properties are given in (25) and (26).

(25) FRes: m.FLN <> {m.PL, m.P}.dom

ERCs: eLL . Wee . e <> eWW . Lee . e

(26) PRes: m.PLN <> {m.FL, m.F}.dom

ERCs: Wee . eLL . e <> Lee . eWW . e

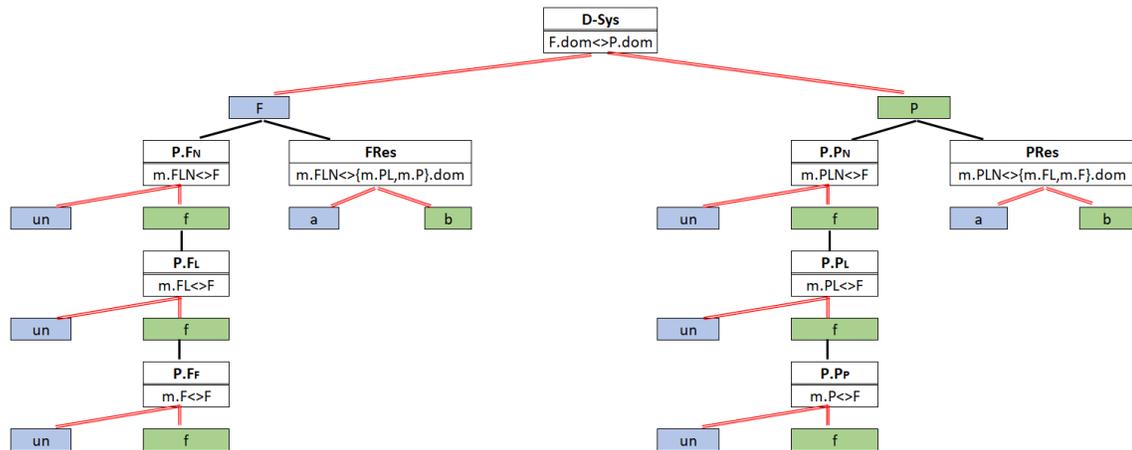
Putting these properties together we give a partial property analysis of CSS.

(27) Partial PA table for CSS

PA table	D-Sys	PRes	P.F _N	P.F _L	P.F _F	FRes	P.P _N	P.P _L	P.P _P
1.PPPP	F	a	un						
2.PNPP	F	a	f	un					
3.PNLP	F	a	f	f	un				
7.FFFF	P					a	un		
8.FNFF	P					a	f	un	
9.FNLF	P					a	f	f	un

A partial treeoid is given in (28). Of note is that the two treeoids of P-Sys and F-Sys are reproduced exactly in this partial CSS treeoid and that further, they serve the same purpose in the CSS, namely determining where on their respective markedness hierarchies the languages are first unfaithful.

(28) Treeoid of partial property analysis for CSS



Finally, we note that the property analysis for the full typology can be completed with three more properties, namely $P_{minF}:m.FLN \langle \rangle \{m.P, m.PL\}.dom$, $P_{minP}:m.PLN \langle \rangle \{m.F, m.FL\}.dom$ and the property $P_u: \{ \{m.FL, m.PL\}.dom, \{m.P, m.PL\}.dom, \{m.F, m.FL\}.dom \}.sub \langle \rangle Faith$. See Merchant and Krämer 2018 for a full discussion.

6 Conclusion

A moderately complex system can be projected onto lower dimensions which exhibit extreme simplicity. Analyses of the lower-dimensional systems can be incorporated into the higher-dimensional yielding a fruitful perspective on the more complicated system. Moreover, markedness hierarchies emerging from sets of stringently related constraints can exhibit the same behavior across typologies and the mappings that inhere in one system can be exploited in analyzing another with a structurally similar stringency hierarchy.

References

- Alber, Birgit 2001. Regional variation at edges: glottal stop epenthesis and dissimilation in Standard and Southern varieties of German. *Zeitschrift für Sprachwissenschaft*. Volume 20, Issue 1, Pages 3–41
- Alber, Birgit and Alan Prince. 2018. *Property Analysis*. Ms.
- Baertsch, Karen 2002. *An Optimality Theoretic Approach to Syllable Structure: The Split Margin Hierarchy*. PhD Dissertation. Indiana University, Bloomington.
- Baertsch, Karen & Davis, Stuart. (2001). Turkic C+/l/(uster) phonology. *CLS*, 37(1), 29-44.
- De Lacy, Paul 2006. *Markedness Reduction and Preservation in Phonology*. Cambridge University Press.
- DelBusso, Natalie 2018. *Typological Structure and Properties of Property Theory*. Ph.D. Dissertation. Rutgers University.
- Green, Christopher, Stuart Davis, Doubacar Diakite, & Karen Baertsch. 2007. On the role of margin phonotactics in Colloquial Bamana complex syllables. *Natural Language and Linguistic Theory*.
- McManus, Hope Eliza 2016. *Stress Parallels in Modern OT*. Ph.D. Dissertation. Rutgers University.
- Merchant, Nazarré and Martin Krämer. 2018. “Stringency hierarchies and the holographic principle in property theory”. Ms., Eckerd College and UiT The Arctic University of Norway.
- Merchant, Nazarré and Alan Prince. 2018. *The Mother of All Tableaux*. Equinox Publishing. ROA-1285.
- Prince, Alan and Paul Smolensky. 1993/2004 *Optimality Theory. Constraint Interaction in Generative Grammar*. Ms. Rutgers University and University of Colorado, Boulder./Blackwell.
- Prince, Alan, Nazarré Merchant, and Bruce Tesar. 2017. OTWorkplace. <https://sites.google.com/site/otworkplace/>.
- Prince, Alan 1997. “Endogenous Constraints on Optimality Theory.” LSA Linguistic Institute.
- Prince, Alan. 2002. Entailed Ranking Arguments. ROA-500.
- Vennemann, Theo 1988. Preference Laws for Syllable Structure and the Explanation of Sound Change: With Special Reference to German, Germanic, Italian, and Latin. Berlin: Mouton de Gruyter.