Summing constraints in an across properties*

Wm. G. Bennett & Natalie DelBusso

December 2020

Abstract

Work in Optimality Theory examining the constraint set, Con, often considers the question of whether certain types of constraints are split into multiple specific versions or are single general constraint that effectively sums the violations of specific ones. Comparing and evaluating analyses differing in this way requires knowing the effect of such summing on the full typology. This depends on the relationship of summands in the full system, which can be difficult to ascertain from inspecting violation profiles alone. This paper uses Property Theory to analyze the systematic effects of summing constraints in two distinct kinds of relationships: i) across distinct properties, and ii) within a constraint class in a single property. The results show how these two types collapse the typology in different but predictable ways. Property Analysis provides a key to identifying constraint relationships and delineating the effect of summing.

1 Introduction

An area of theoretical interest in Optimality Theory (OT; Prince and Smolensky 2004) concerns the makeup of the constraint set, CON, and particularly whether various constraints exist as single general constraints, or multiple versions, specific to feature values, position, etc. For instance, is there a single Ident constraint that assesses violations for unfaithfulness to any feature, or a family of distinct Ident-[F] constraints that each assess faithfulness of just one feature (Ito et al. 1995; McCarthy and Prince 1995))? A candidate that violates three single-feature faithfulness constraints, Ident-[F], Ident-[G], and Ident-[H] will also violate an all-feature Ident constraint 3 times (once each for F, G, H, respectively). This kind of relationship is termed *summing*: the violation profiles of several constraints ('summands'), when pooled together, add up to the violation profile of the combined ('summed') constraint.

Comparing and evaluating analyses that differ in this way requires knowing the effect of such summing on the full typology. This, in turn, depends on the relationship of the summands in the full system, which can be difficult to ascertain from inspecting violation profiles in a violation tableau, where numerical quirks and other interactions can obscure the interactions. This paper uses Property Theory (Alber et al. 2016; Alber and Prince prep; DelBusso 2018) to show the effects of summing constraints in two distinct kinds of relationships. In explicating the predictable changes that occur in each case, the results allow for systematic comparison of related typologies.

Property Analyses provide a key to identifying the constraint relationships – and so too to delineating the effects of summing constraints. We show this by analyzing two cases where the summands stand in specific relations, called Summing Across (SA; §4) and Summing In (SI; §5). These names refer to how the summand constraints are related in a property analysis: distributed across different properties with parallel structures or found in the same property, respectively. The properties isolate the core constraint interactions that define the grammars of the typology. Both SA and SI collapse the typology systematically, by *eliminating* or *equalizing* properties and grammars, as defined in (1).

(1) Definitions

a. Eliminate: to lose properties (from an Property analysis) or grammars (from a typology).

b. Equalize: to make properties or grammars the same.

*The authors would like to thank Alan Prince and participants of the 4th meeting of the Society for Typological Analysis for discussion of the ideas in this paper.

SA equalizes properties and eliminates grammars: when properties become the same, some grammars are no longer possible (i.e., they are harmonically bounded, Samek-Lodovici and Prince (2005)). SI eliminates properties and equalizes grammars: when properties are lost, grammars differing in the values of the eliminated properties only become the same; their languages are co-optimal.

While the two cases are developed using abstract OT systems, both are manifested in concrete OT systems as well, including the typologies analyzed in Bennett and DelBusso (2018), which are based on the extant literature on Agreement By Correspondence (ABC; Rose and Walker 2004; Hansson 2010; Bennett 2015). The SA case connects to theories that posit families of parallel constraints, such as constraint schemata that make the same kind of choice independently available for different features, arrangements, directions, categories, etc. The SI case connects to the relationship between the two main ABC constraint types, Corr and CC.Id, which act together as a class to enforce harmony (or dissimilation).

2 Summing in Concrete OT: Agreement-by-Correspondence

Agreement-by-Correspondence (ABC)—and related theories examined in Bennett and DelBusso (2018); DelBusso and Bennett (2019)—provides examples of both types of summing analyzed here. In ABC theories, there are two main constraint types: Corr, which assigns violations when some designated segments (e.g. those that share a given feature) are not in surface correspondence with one another; and CC.Id, which evaluate feature agreement between segments that stand in correspondence with each other. Both types of constraints are satisfied when output segments either: (i) are in correspondence and in agreement on the feature values specified by CC.Id (producing assimilation, a.k.a. harmony); or (ii) are not in correspondence and disagree in feature values specified by Corr (producing dissimilation). Most ABC theories assume families of Corr and CC.Id constraints, with members of each family specified for different features or domains, etc., resulting in parallelisms among the properties of the typologies of such systems. (For example, the interaction between Corr.f and IO.Id.f occurs in a predictable way, regardless of which particular feature [f] represents.) There have also been proposals to combine constraints both within (i.e., Corr.v+Corr.c) and across (i.e., Corr.v+CC.Id.c) the two families, similar to the summing operations here. The Violation Tableau (VT) in (2) shows the violations of these summed constraints as well as the summands and IO.Id for the four non-harmonically bounded candidates in two csets.[1]

(2) ABC VT

Input	Output	Corr.c	**Corr.v+Corr.c**	Corr.v	**Corr.v+CC.Id.c**	CC.Id.c	CC.Id.v	IO.Id.v	IO.Id.c
td	t1d1						1		
	t1t1/d1d1							1	
	s1d2/t1z2								1
	t1d2	1	1						
dz	d1z1				1	1			
	d1d1/z1z1								1
	t1z2/d1s2							1	
	t1d2		1	1	1				

The ABC systems and sums thereon exemplify the typological effects of the two types, collapsing the typologies in different but predictable ways. These are discussed in further detail in the respective sections.

[1]Notations: numbers indicate correspondence indices; same = correspondence, different = non-correspondence; v and c abbreviate [voice] and [continuant], respectively. Co-optimal candidates are shown as cand1/cand2. Inputs represent strings of just consonants, leaving out vowels and other inert material.

3 Formal background

3.1 ERCs and fusion

Elementary Ranking Conditions (ERCs; Prince 2002) encode the rankings necessary for one candidate x to be optimal over another y. ERCs are vectors with a column for each constraint with one of three values: W (the constraint prefers x), L (the constraint prefers y) or e (the constraint does not distinguish the candidates). For example, the ERC `WeL` indicates that the first constraint must dominate the last for the chosen candidate x to be optimal over y. An ERC is *satisfied* when all Ls are preceded by a W. A non-trivial ERC contains at least one W and one L and is satisfied when at least one constraint with a W dominates all constraints with Ls in the grammar. A trivial ERC with only Ws and es (referred to as W*) is satisfied by any ranking; one with only Ls and es (referred to as L+) cannot be satisfied (Prince 2002). An OT grammar is defined by a set of ERCs (Merchant and Prince prep).

(3) Definition: ERC grammar: A set of ERCs delineating the rankings that give rise to the same language.

A fundamental operation on ERCs is fusion (∘; Prince 2002). Fusion takes a set of ERCs and produces an ERC entailed by the set. Column values are combined across the rows as follows: L∘X = L (L is dominant); W∘e = W (e acts as identity); X∘X = X (where X is any value). In (4), the C1 column contains a single W and two e's, fusing to W; all remaining columns contain at least one L, fusing to L. The fused ERC encodes that ranking information that C1>C3, entailed by the ERC set but not explicit in any individual ERC.

(4) Fusion

	C1	C2	C3	C4
ERC1	W	L	e	L
ERC2	W	e	L	W
ERC3	e	W	e	L
Fuse	W	L	L	L

The kind of constraint summing introduced in this paper reduces two or more constraints to one by fusing across the summand columns in an ERC set rather than across rows. The value in the summed column is determined by combining values just as in row fusion. In (5), C1 and C2 in ERC3 are fused into C1+2 in ERC3'. If a candidate violates either of C1 or C2, then is violates the summed C1+2.[2]

(5) ERC column fusion

	C1	C2	C3	C4
ERC3	e	W	e	L

	C1∘C2=C1+2	C3	C4
ERC3'	W	e	L

3.2 Property Theory

The set of ERCs delineating a grammar need not define a total—or partial—order over the entirety of Con: not all constraints are crucially ranked relative to one another in a given typology (see Prince (2017) on representing grammars). Identifying which rankings are crucial is the aim of Property Theory (see Alber et al. (2016); Alber and Prince (prep); DelBusso (2018) for more introductions to Property Theory).

A Property Analysis (PA) of a typology is a set of properties that, collectively, define all of the grammars in the typology, i.e., the set of choices sufficient to distinguish all languages in a typology. In a valid PA of a typology, the possible value combinations of the property set correspond one-to-one to the grammars.

A *property* is a binary choice between two mutually-exclusive ranking conditions. A property has two constraints (or groups of constraints) as *antagonists*, X<>Y, defining two *values*, α. X>Y and β. Y>X. Any grammar in which X and Y are crucially ranked has one value or the other. The values define an ERC (set) (α) and its negation (β); these are the value ERCs. A property grammar is defined as a set of values which generate the ERC set of the grammar.

[2]This is similar to constraint disjunction (Crowhurst and Hewitt 1997; Hewitt and Crowhurst 1996).

(6) Definition: Property grammar: A unique set of property values that defines an ERC set.

The antagonists in a property (X and Y) may be single constraints, or sets of (sets of) constraints. The latter case is a situation where crucial rankings are not strictly pairwise relationships between individual constraints but between groups of constraints that act together. Such groups form a constraint class, abbreviated κ, which is appended with an operator, *dom* or *sub*, designating the highest or lowest member of the class, respectively, in their linear ordering (Alber et al. 2016; Alber and Prince prep; DelBusso 2018). For example, {X,Y}.dom means that the dominant member of {X,Y} interacts with the antagonist. A κ.dom in a property generates an ERC with multiple Ws or Ls. When the class is dominant, the dominant constraint in it ranks above the antagonist, and when the class is subordinate, the antagonist ranks above the dominant constraint, thus transitively above all κ.dom members. A κ.sub, however, generates a set of ERCs. This set is conjunctive when the class is dominant; if the lowest-ranked member of κ.sub dominates the antagonist, then all other constraints in the class also dominate the antagonist (by dominating the lowest). When the class is subordinate, the set is disjunctive: the antagonist need only dominate the lowest-ranked member. If a PA contains a property ranking a κ.sub relative to an antagonist, it must also have some other property(ies) that antagonizes the members of κ.sub among each other, and so define which member is subordinate. The ERCs for a two-constraint class are shown in (7).

(7) Value ERCs for P: κ.op<>Z

κ.op	α. κ.op<>Z	β. Z<> κ.op
{X,Y}.dom	WWL	LLW
{X,Y}.sub	WeL & eWL	LeW | eLW

In a PA, properties can be related in complex ways, indicated by their *scope* (Alber et al. 2016; Alber and Prince prep).

(8) Definition: Scope: The set of grammars having a value, α or β, of a property. For grammars outside of the scope, the property is moot.

For a *wide-scope* property, every grammar in the typology has a value. For a *narrow-scope* property, only some grammars in the typology have a value. Narrow scope is defined by values of other properties: only if another specific ranking condition(s) occurs is the ranking in the narrow scope property needed. Thus, scope reflects dependencies between different properties. For a grammar outside the scope of a property, the choice made by that property is meaningless and/or inconsistent with the ranking conditions defining it.

4 Summing Across (SA): constraints in parallel properties

The SA case sums distinct constraints that play parallel roles in a typology, interacting with the same antagonists in the same ways, as, for example, the Corr constraints with the IO.Id constraints. This relationship is manifested in PAs by the presence of *parallel properties*, which have the same structure and differ minimally in a constraint or κ.op (9)[3]. Summing the differing constraints merges parallel properties, resulting in loss of those grammars differing in their values.

(9) Definition: Parallel Properties: A pair of properties, Px and Py, are *parallel properties* iff:

a. Px: {Cx...}.op<>Z & Py: {Cy...}.op<>Z, where '...' is the same.[4]

b. Their scopes are equivalent: either the scopes are defined by the same value of the same property, or by matching values of different properties that are also parallel to one another.

The SA case is first presented with a simple Base system consisting of three constraints (10). C1 and C2 are the prospective summands, and Z (either a single constraint or a κ.op) is the antagonist in parallel properties P1 and P2. These have the same form, differing in the left side (C1 or C2) and sharing the antagonist Z. Additionally, their scopes are equivalent since both are wide. The property values are freely

[3] McManus (2016) develops a similar concepts of parallel properties in studying related stress systems.

[4] If the antagonists are single constraints, these reduce to Cx<>Z and Cy<>Z.

combinable, as C1 and C2 can be ranked relative to Z independently of one another, generating the four grammars in the value table, which shows the value combinations that define them. The values generate ERCs, which combine to give the resulting grammars. ERCs are given in the order C1-C2-Z.

(10) SA Base: pre-sum PA

a. Properties

P	α	β
P1: C1<>Z	WeL	LeW
P2: C2<>Z	eWL	eLW

b. Value table

	P1	P2	Grammar
L1	α	α	WeL, eWL
L2	α	β	WLL, eLW
L3	β	α	LWL, LeW
L4	β	β	LeW, eLW (=LLW)

When C1 and C2 are summed, their ERC columns fuse (11), and P1 and P2 values thus become identical. When C1 and C2 are combined into a single constraint, there are only two possible rankings: C1+2>Z or Z>C1+2. As a result, any combination of α and β values for the parallel properties are contradictions, fusing to L+. Grammars with such mixed values rank the summands on either side of the shared antagonist Z, an impossibility when these are a single constraint, so the pre-sum grammars L2 and L3 are harmonically bounded (shaded in grey), leaving two grammars defined by the values of P1/2.

(11) SA Base: post-sum PA

a. Properties

P	α	β
P1: C1+2<>Z	WL	LW
P2: C1+2<>Z	WL	LW

b. Value table

	P1	P2	Grammar
L1	α	α	WL, WL
L2	α	β	WL$\circ$LW=LL
L3	β	α	LW$\circ$WL=LL
L4	β	β	LW, LW

These results generalize to the context of additional properties and constraints. Even when more than two constraints are summed across more than two properties, the consequence is that the parallel properties are merged into a single property. This eliminates from the typology the grammars that have mixed values on the parallel properties in the pre-sum system. The choice after summing is limited to just α or β: a mixture of the two values is impossible. Consequently, the grammars that were defined by mixtures of values on the parallel properties in the pre-sum system become impossible after summing. The SA result is spelled out in (12), along with its justification. To generalize across all property values, the following notations are used: v for a property value ERC, $\bar{v}$ for its reverse, and '...' to represent other constraints with the same values (W/L/e) in all ERCs.

(12) Proposition: SA equalizes properties and eliminates grammars.
Proof: Given a set of parallel Ps, P1,...,Pn differing in one antagonist, C1 to Cn:

a. For a Pi: Ci<>Z in the parallel property set, value ERCs v and $\bar{v}$ are:

	C1...Ci...Cn	Z
v	e...W...e	L
$\bar{v}$	e...L...e	W

b. Fusing columns C1 to Cn results in the same value ERCs for all parallel properties:

	$\circ$(C1,...,Cn)	Z
v	W	L
$\bar{v}$	L	W

c. For any two properties, Px and Py, if Px.v = Py.v then Px = Py and Px.v + Py.$\bar{v}$ is a logical contradiction (fusing to L+, as property values define an ERC (set) and its negation).

d. Therefore, a subset of any ERC set containing any α+β value combination fuses to L+ and is not a possible grammar. Any pre-sum grammar defined in part by such a combination is not a possible grammar in the post-summed system.

The effects of SA correspond to summing violations across summand columns in a Universal Violation Tableau (UVT; Prince 2015). A UVT is a VT that represents a full typology, where each row is a language. The ERCs resulting when each row is chosen as optimal and compared to every other row generate the grammar of the language. As constraints in parallel properties interact with the same antagonists independently of one another, there are four violation profile patterns in the pre-sum system (relative to each shared antagonist): i) both constraints have minimum violation value ($\alpha\alpha$);[5] ii) neither do ($\beta\beta$); iii) one and not the other does ($\alpha\beta$ or $\beta\alpha$). The SA Base pre- and post-sum systems' UVTs are shown in (13). Summing violations across the rows (post-sum system) results in the collective harmonic bounding of grammars that violate one of the summands and the antagonist (L2, L3) by those that violate either both summands or the antagonist (L1, L4).

(13) SA Base UVT

	pre-sum			post-sum	
	C1	C2	Z	C1+2	Z
L1: $\alpha\alpha$			2		2
L2: $\alpha\beta$		1	1	1	1
L3: $\beta\alpha$	1		1	1	1
L4: $\beta\beta$	1	1		2	

This pattern is iterated for multiple antagonist(s), as shown in the UVT with antagonist {Za,Zb}.sub. The seven grammars of the pre-sum UVT reduce to three post-sum.

(14) SA UVT: κ.sub antagonist

	pre-sum				post-sum		
	C1	C2	Za	Zb	C1+2	Za	Zb
L1				2			2
L2			2			2	
L3		1		1	1		1
L4		1	1		1	1	
L5	1			1	1		1
L6	1		1		1	1	
L7	1	1			2		

The formal changes from summing constraints in parallel properties manifest extensionally in tying together the treatment of different extensional traits-–traits which were determined by separate properties in the pre-sum system. In ABC typologies, a clear example is differences between features. In a system studied in Bennett and DelBusso (2018), there are two doubly-parallel properties, each antagonizing a class of one feature-specific Corr and one feature-specific CC.Id constraint against a class of IO.Id constraints (15).

(15) Doubly-parallel ABC properties

a. P1c: {Corr.v,CC.Id.c}.sub<>{IO.Id.c,IO.Id.v}.sub

b. P1v: {Corr.c,CC.Id.v}.sub<>{IO.Id.c,IO.Id.v}.sub

These properties highlight a parallelism in the extensional typology: languages may have harmony for neither, either, or both voicing (v) and continuacy (c) features, and the harmony/no harmony choice depends

[5] While the minimum value is not necessarily 0 (see Merchant and Prince (prep)), it can be reduced to 0 in the UVT. DelBusso (2018) defines a Minimal Violation Tableau, in which violations counts are reduced to the minimum value that preserves order and equivalence relations.

on P1v and P1c values, respectively. In segmental terms, the choice of whether an input with disharmonic stops like /td/ is faithful, or assimilates, or dissimilates, is a separate choice from whether inputs like /dz/ are faithful, or assimilate, or dissimilate. The two choices are intensionally parallel, as Corr and CC.Id constraints impart the same structure onto each ranking choice, abstracting away from differences in the particular features referred to. The choice about how to handle stricture disharmony works the same way as the choice about how to handle voicing disharmony.

Summing both Corr constraints into a single constraint and both CC.Id constraints into another constraint (16) collapses the two choices into one: harmony for both features or for neither (the summed constraints are violated if any of the summands are).

(16) Summed parallel ABC property
P1: {Corr.v|c,CC.Id.c|v}.sub<>{IO.Id.c,IO.Id.v}.sub

Grammars differing in values for the combined properties become harmonically bounded: the candidate languages that only have harmony for one feature violate both the IO faithfulness constraints and the ABC constraints, while the all-or-nothing harmony alternatives have violations of only one or the other constraint type. The summed constraints are more general in referring to larger sets of features. They are similar to completely general constraints proposed in the literature that lack any kind of feature reference: Corr (McCarthy 2010) and CC.Id (Gallagher and Coon 2009). The extensional result of summing across features is to yoke the two featural choices together.

5 Summing In (SI): constraints in a class

In many typologies, constraints act together in properties, forming a class, κ. For example, the pairs of Corr and CC.Id constraints form classes in P1c and P1v in (15). SI sums constraints within such a class, resulting in loss of other properties and equalization of grammars. These effects arise due to the structure of PA having κ.ops: the op indicates that the antagonist is ranked relative to the highest or lowest member of κ, but does not determine which constraint this is. Ordering among the κ.op members must occur in another property or set of properties. For example, in P1: {C1,C2}.sub<>Z, Z is ranked relative to the subordinate of C1 and C2, which are not themselves ordered in P1 values. As such, a second property, P2: C1<>C2, ranks the members of the class. For P2α, C1>C2, C2 is subordinate and thus Z is ranked relative to C2 in P1, and under P2β the reverse holds.[6] When the members of a κ.op are summed, properties antagonizing the summands are eliminated, as their values—attempting to rank a constraint relative to itself—become L+. Consequently, grammars of the pre-sum system differing only on the values of such properties become co-optimal. There is no distinction between optima differentiated only by the ranking of the summed constraints.

As with SA, SI is first shown with a simple three-constraint Base system (17). P1 has a two-constraint κ.op, {C1,C2}.sub and its values are a conjunctive (α) and disjunctive (β) ERC set (ERC order C1-C2-Z). P2 antagonizes the members of the P1 κ.op and is scoped under P1β: only in grammars with this value is ranking of C1 and C2 crucial.

(17) SI: pre-sum PA

a. Properties

P	α	β
P1: {C1,C2}.sub<>Z	WeL & eWL	LeW | eLW
P2: C2<>C2	WLe	LWe

b. Value table

	P1	P2	Grammar
L1	α		WeL, eWL
L2	β	α	eLW, WLe
L3	β	β	LeW, LWe

[6]See Alber and Prince (prep); DelBusso (2018) for more on classes.

Summing fuses the first two ERC columns, reducing the multi-ERC sets of P1 values to single ERCs (the conjuncts/disjunct become equivalent) (18). Additionally, P2 values become L+, as they cannot rank a single constraint relative to itself. Eliminating the illogical P2 makes L2 and L3 equivalent, as they are no longer distinguished by any property.

(18) SI Base: post-sum PA

a. Properties

P	α	β
P1: C1+2<>Z	WL	LW
P2: C1+2<>C1+2	Le	Le

b. Value table

	P1	Grammar
L1	α	WL
L2/3	β	LW

These typological effects of SI shown in the simple system generalize to more complex cases with larger summed κ.op's and consequently more complex property structures (19).

(19) Proposition: SI eliminates properties and equalizes grammars
Proof: Given a property P1: κ.op<>Z, where C1...Cn $\in$ κ.op:

a. Value ERCs for P1:

- P1.v: there is a least 1 W in columns C1-Cn and no Ls.
- P1.$\bar{v}$: there is at least 1 L in columns C1-Cn and no Ws.

b. Illustration of those value ERCs for both kinds of op:

- op = dom: there is a W (v) or L ($\bar{v}$) in all columns C1–Cn, fusing to W or L.

Pre-sum	C1...Cn	Z
v	W...W	L
$\bar{v}$	L...L	W
Post-sum	$\circ$(C1,...,Cn)	Z
v	W	L
$\bar{v}$	L	W

- op = sub: there is a conjunctive set with an ERC with a W for each C$\in$ κ.op and an e for all others, fusing to W (v); or a disjunctive set with an ERC with an L for each C$\in$ κ.op and an e for all others, fusing to L ($\bar{v}$).

Pre-sum	C1...Cn	Z
v	We... & ...eWe... & ...eW	L
$\bar{v}$	Le... $\mid$ ...eLe... $\mid$ ...eL	W
Post-sum	$\circ$(C1,...,Cn)	Z
v	W	L
$\bar{v}$	L	W

c. Properties antagonizing κ.op members are lost and grammars are equalized.

- Given any P2: Ci<>Cj $\in$ PA, either Ci = L or Cj = L under either P2 value.
- Fusion of anything with L yields L; therefore, the fused Ci+Cj column = L, eliminating the Ws.
- If P2 value ERCs = L+, then no grammar in the typology can have a value of P2, and P2 $\notin$ PA.
- Any pre-sum system pair of grammars, G1 and G2 sharing all property values except that G1 is P2.v and G2 is P2.$\bar{v}$, become equivalent when P2 is lost, as both are defined by the same set of values.

As with SA, PAs with SI-eligible κ.ops also have characteristic UVTs that show the summing effects. There are three violation patterns differing for a κ.sub and a κ.dom in whether a candidate violates neither or both summands, respectively (20). Summing results in identical violation profiles for L2 and L3.

(20) SI UVT

a. κ.sub

<table>
<tr><th rowspan="2"></th><th colspan="3">pre-sum</th><th colspan="2">post-sum</th></tr>
<tr><th>C1</th><th>C2</th><th>Z</th><th>C1+2</th><th>Z</th></tr>
<tr><td>L1</td><td></td><td></td><td>1</td><td></td><td>1</td></tr>
<tr><td>L2</td><td></td><td>1</td><td></td><td rowspan="2">1</td><td rowspan="2"></td></tr>
<tr><td>L3</td><td>1</td><td></td><td></td></tr>
</table>

b. κ.dom

<table>
<tr><th rowspan="2"></th><th colspan="3">pre-sum</th><th colspan="2">post-sum</th></tr>
<tr><th>C1</th><th>C2</th><th>Z</th><th>C1+2</th><th>Z</th></tr>
<tr><td>L1</td><td>1</td><td>1</td><td></td><td>2</td><td></td></tr>
<tr><td>L2</td><td>1</td><td></td><td>1</td><td rowspan="2">1</td><td rowspan="2">1</td></tr>
<tr><td>L3</td><td></td><td>1</td><td>1</td></tr>
</table>

The loss of properties and subsequent co-optimization of grammars manifests extensionally in the loss of a distinguishing trait in the pre-sum system, for example, the distinction between correspondence and non-correspondence in faithful grammars in ABC systems.

The SI case sums the classes (κ.ops) of Corrs and CC.Ids in the properties in (15). Such a combination is highly similar to Hansson's (2014) Agreement by Projection (ABP) theory, which combines Corr and CC.Id constraints into single constraints that simultaneously condition agreement on one feature (as in Corr) and enforce agreement on another (as in CC.Id), effectively transmuting a Corr and CC.Id pair into a single constraint that demands agreement (Agr). Such summing results in the properties in (21). Agr.c/v = Corr.v+CC.Id.c and Agr.v/c = Corr.c+CC.Id.v. This summing also eliminates from the PA additional properties antagonizing Corr and CC.Ids, and in so doing co-optimizes languages distinguished by only those values (Bennett and DelBusso 2020). Extensionally, the co-optimized languages have the same segmental surface forms and differ only in the presence/absence of correspondence indices (i.e., t1d1 or t1d2; indices are eliminated in Hansson's proposal). The summed system thus leave only a choice between faithful and unfaithful mappings.

(21) SI summing of ABC classes

a. P1c: Corr.v+CC.Id.c<>{IO.Id.c,IO.Id.v}.sub = Agr.c/v<>{IO.Id.c,IO.Id.v}.sub

b. P1v: Corr.c+CC.Id.v<>{IO.Id.c,IO.Id.v}.sub = Agr.v/c<>{IO.Id.c,IO.Id.v}.sub

6 Summary

A question in linguistic analysis is the effects of combining certain conceptual pieces (here, constraints) into a single mechanism, which could be argued to be preferable by Occam's Razor, all other things being equal. The converse is also of interest: the effects of splitting a single mechanism in an insufficiently rich theory to produce more degrees of freedom in a typology. Understanding the answers to these questions is fairly straightforward in any concrete case by defining both theories, producing their typologies, and comparing the results. But the possibility of a generalized solution holds the promise of obviating that extra legwork, and yielding greater insight into the structure of OT typologies.

The findings in this paper report a means of delineating systematic differences between typologies with related Cons. Using Property Theory, the typological effects of summing constraints in particular relations is predictable. SA applies to PAs with parallel properties, equalizing properties and eliminating grammars, while SI applies to PAs with constraint classes, equalizing grammars and eliminating properties. In addition to addressing the first question above on the effects of combining constraints, these results are a step towards an answer to the converse: the effects of splitting a constraint are hypothesized to be the reverse of summing. Together, these provide a means of understanding a lattice of typologies projected from any one typology that is fully analyzed and understood, related to the others through summing and unsumming.

References

Alber, B., DelBusso, N., and Prince, A. (2016). From intensional properties to Universal Support. *Language: Phonological Analysis*, 92.2:e88–e116.

Alber, B. and Prince, A. (in-prep). *Typologies.* Ms, Free University of Bozen-Bolzano & Rutgers University.

Bennett, W. G. (2015). *The phonology of consonants: Harmony, dissimilation and correspondence.* Cambridge University Press.

Bennett, W. G. and DelBusso, N. (2018). The typological effects of abc constraint definitions. *Phonology*, 35.1:1–37.

Bennett, W. G. and DelBusso, N. (2020). Summing within and across properties. Paper presented at SOTA IV, Eckerd College.

Crowhurst, M. and Hewitt, M. (1997). Boolean operations and constraint interactions in optimality theory. Technical report, Rutgers University. ROA 229.

DelBusso, N. (2018). *Typological structure and properties of Property Theory.* PhD diss, Rutgers University.

DelBusso, N. and Bennett, W. G. (2019). ABP and ABC: Agreement with/out correspondence. *Phonological Data and Analysis*, 1.3:1–25.

Gallagher, G. and Coon, J. (2009). Distinguishing total and partial identity: Evidence from Chol. *NLLT*, 27:545–582.

Hansson, G. O. (2010). *Consonant harmony: Long-distance interactions in phonology.* University of California Press.

Hansson, G. O. (2014). (Dis)agreement by (non)correspondence: inspecting the foundations. *UC Berkeley Phonology Lab Annual Report*, pages 3–62.

Hewitt, M. and Crowhurst, M. (1996). Conjunctive constraints and templates in Optimality Theory. In Beckman, J., editor, *Proceedings of NELS 26*, pages 101–116. GLSA Publications, Amherst.

Ito, J., Mester, A., and Padgett, J. (1995). Licensing and underspecification in optimality theory. *Linguistic Inquiry*, 26.4:571–613.

McCarthy, J. J. (2010). Agreement by correspondence without Corr constraints. Technical report, University of Massachusetts, Amherst. ROA 1089.

McCarthy, J. J. and Prince, A. (1995). Faithfulness and reduplicative identity. In Dickey, L., Beckman, J., and Urbanczyk, S., editors, *University of Massachusetts Occasional Papers in Linguistics.* GLSA Publications, Amherst.

McManus, H. (2016). *Stress parallels in modern OT.* PhD diss., Rutgers University.

Merchant, N. and Prince, A. (in-prep). *The Mother of all Tableaux.* Equinox.

Prince, A. (2002). Entailed Ranking Arguments. Technical report, Rutgers University. ROA 500.

Prince, A. (2015). One tableau suffices. Technical report, Rutgers University. ROA 1250.

Prince, A. (2017). Representing OT grammars. Technical report, Rutgers University. ROA 1309.

Prince, A. and Smolensky, P. (1993/2004). *Optimality Theory: Constraint interaction in generative grammar.* Wiley Blackwell, Malden, MA.

Rose, S. and Walker, R. (2004). A typology of consonant agreement as correspondence. *Language*, 80.3:475–531.

Samek-Lodovici, V. and Prince, A. (2005). Fundamental properties of Harmonic Bounding. Technical report, University College London and Rutgers University. ROA 785.