Phonology **37** (2020) 367–381. © The Author(s), 2020. Published by Cambridge University Press
doi:10.1017/S0952675720000172

*Summing constraints in and across properties**

Wm. G. Bennett
Rhodes University

Natalie DelBusso
Wayne State College

Work in Optimality Theory on the constraint set, Con, has often raised the question of whether certain types of constraints have multiple specific versions or are single general constraints that effectively sum the violations of specific variants. Comparing and evaluating analyses that differ in this way requires knowing the effect of this kind of summing on the full typology, which itself depends on the relationship of summands in the full system. Such relationships can be difficult to ascertain from inspecting violation profiles alone. This paper uses Property Theory to analyse 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, yet predictable, ways. Property Analysis provides a key to identifying constraint relationships and so to delineating the effect of summing.

1 Introduction

An area of theoretical interest in Optimality Theory (OT; Prince & Smolensky 1993) is the make-up of the constraint set, Con, and particularly whether various constraints exist as single general constraints, or are split into multiple versions, specific to feature values, position, etc. For instance, is there a single Ident constraint that assesses violations for all features, as opposed to a family of distinct Ident[F] constraints that each assess just one feature (Itô *et al*. 1995: 586, McCarthy & Prince 1995: 16)? A candidate that violates three single-feature faithfulness constraints, Ident[F], Ident[G] and Ident[H], will also violate an all-feature Ident constraint three times (once each for [F], [G] and [H]). 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.

* E-mail: w.bennett@ru.ac.za, n.delbusso@rutgers.edu.
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.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

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, DelBusso 2018, Alber & Prince in preparation) to show the effects of summing constraints in two distinct kinds of relationships. The results show the predictable changes, and allow for the systematic comparison of related typologies.

Property Analysis provides a key to identifying the constraint relationships – and so too to delineating the effects of summing constraints. We show this by analysing two cases where the summands stand in specific relations, which we call Summing Across (SA; §4) and Summing In (SI; §5). These names refer to how the summand constraints are related in a property analysis: either distributed across different properties with parallel structures or found in the same property. 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 equalising properties and grammars; see (1).

(1) *Definitions*

a. *Eliminate*
Lose grammars or properties (from the typology or the Property Analysis respectively).

b. *Equalise*
Make grammars or properties the same.

SA equalises properties and eliminates grammars: when properties become the same, some grammars are no longer possible (i.e. they are harmonically bounded; Samek-Lodovici & Prince 2005). SI eliminates properties and equalises grammars: when properties are lost, grammars differing only in the values of these become the same, and 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 analysed in Bennett & DelBusso (2018) and DelBusso & Bennett (2019), which are based on the literature on Agreement by Correspondence (Rose & 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 Agreement by Correspondence constraint types, Corr and CC.Id, which act together as a class to enforce harmony (or dissimilation).

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

2 Summing in concrete OT systems: Agreement by Correspondence

Agreement by Correspondence (ABC) – and related theories examined in Bennett & DelBusso (2018) and DelBusso & Bennett (2019) – provides examples of both types of summing analysed 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 evaluates 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 for the feature values specified by CC.ID (producing assimilation), 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 and across the two families, similar to the summing operations here. The VIOLATION TABLEAU in (2) shows the violations of both the unsummed and summed constraints (the latter outlined in bold) for two candidate sets; only non-harmonically bounded candidates are shown.[1]

(2)

input	output	CORR [vce]	CORR [cont]	CC. ID [vce]	CC. ID [cont]	**CORR[vce] + CORR [cont]**	**CORR[vce] + CC.ID [cont]**	IO. ID [vce]	IO. ID [cont]
td	t_1d_1			1					
	t_1t_1/d_1d_1							1	
	s_1d_2/t_1z_2								1
	t_1d_2		1			1			
dz	d_1z_1				1		1		
	d_1d_1/z_1z_1								1
	t_1z_2/d_1s_2							1	
	d_1z_2	1				1	1		

The ABC systems and sums thereon exemplify the typological effects of the two types, collapsing the typologies in different but predictable

[1] The following notational conventions are used: numbers indicate correspondence indices (same numbers = correspondence, different = non-correspondence). In Cand1/Cand2, '/' indicates pairs of co-optimal candidates. Inputs are represented as strings of consonants, leaving out vowels and other inert material: any string of the form [… t_1 … d_2 …] (e.g. [t_1ad_2a], [it_1ud_2a]) is represented as [t_1d_2]. [cont] = [continuant], [vce] = [voice].

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

ways. CORR[vce] + CORR[cont] is an example of Summing Across (SA); CORR[vce] + CC.ID[cont] is an example of Summing In (SI). These are discussed in further detail in the respective sections.

3 Formal background

3.1 Elementary Ranking Conditions and fusion

Elementary Ranking Conditions (ERCs; Prince 2002) encode the rankings necessary for one candidate x to be more optimal than y. ERCs are vectors with a column for each constraint with one of three values: W (prefers x), L (prefers y) or e (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. An ERC is SATISFIED when all Ls are preceded by a W. The ranking encoded in the ERC makes x optimal over y. 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 e's is satisfied by any ranking; one with no Ws (only Ls and e's) cannot be satisfied (Prince 2002). The latter is referred to as $\mathcal{L}^+$. An OT grammar is defined by a set of ERCs (Merchant & Prince to appear), as in (3).

(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 the ERC entailed by the set. Column values are combined across the rows as follows: L∘X = L (i.e. L is dominant); W∘e = W (i.e. e acts as identity); X∘X = X (where X is any value). In (4), the C1 column contains a single W and 2 e's, fusing to W; remaining columns contain at least one L, fusing to L. The fused ERC encodes the ranking information that C1≫C3, entailed by the fused 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. The value in the summed column is determined just as in fusion. In (5), C1 and C2 in ERC3 are fused into C1+2 in ERC3′. If a

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

candidate violates either of C1 or C2, then it violates the summed C1+2.[2] Such fusion can also produce trivial $\mathcal{L}^+$ ERCs, as is the case if C1 and C2 in ERC1 are fused, yielding L*e*L.

(5) *ERC column fusing*

	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, DelBusso 2018 and Alber & Prince in preparation for introductions to Property Theory).

A Property Analysis 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 Property Analysis 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$\lll\!\ggg$Y, defining two mutually exclusive values, α. X$\gg$Y and β. Y$\gg$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 that generates the ERC set of the grammar.

(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, DelBusso 2018 and Alber & Prince in preparation). For example, {X,Y}*.dom* means that the dominant member of {X,Y} interacts with the antagonist. A κ*.dom* in a property

2 This is similar to constraint disjunction (Hewitt & Crowhurst 1996, Crowhurst & Hewitt 1997).

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

generates an ERC with multiple Ws or Ls; when class κ dominates the antagonist, the dominant constraint in the class ranks above the antagonist; when κ is subordinate, the antagonist ranks above the dominant constraint in κ, and 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; i.e. 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 does not need to dominate all members of the class in order to dominate the lowest-ranked member. If a Property Analysis contains a property ranking a κ*.sub* relative to an antagonist, it must also have some other property that antagonises the members of κ*.sub* among each other, and so defines which member is subordinate. The ERCs for a class with two constraints are shown in (7).

(7) *Value ERCs for* P: $\kappa.op \lll\ggg$ Z

$\kappa.op$	α. $\kappa.op \gg$ Z	β. Z $\gg \kappa.op$
{X,Y}.*dom*	WWL	LLW
{X,Y}.*sub*	{WeL, eWL}	LeW | eLW

In a Property Analysis, properties can be related in complex ways, indicated by their SCOPE (Alber *et al.* 2016, Alber & Prince in preparation), defined in (8).

(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 the values of other properties: only if another specific ranking condition occurs is the ranking in the narrow-scope property needed. Thus scope reflects dependency 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: 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[F] constraints. This relationship is manifested in Property Analyses by the presence of PARALLEL PROPERTIES, defined in (9), which have the same structure, but differ minimally in a constraint or a κ*.op*. Summing the differing constraints

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

merges the parallel properties, resulting in loss of those grammars differing in their values.[3]

(9) *Definition: Parallel properties*

A pair of properties, Px and Px, are PARALLEL PROPERTIES iff:

a. Px: {Cx…}.op≪≫Z[4]
   Py: {Cy…}.op≪≫Z, where '…' is the same.

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 in (10) with a simple system called 'SA Base', consisting of three constraints. C1 and C2 are the prospective summands, and X (either a constraint or a $\kappa.op$) is the antagonist in parallel properties P1 and P2. These have the same form, differing in the lefthand side (C1 or C2) and sharing the antagonist X. Additionally, their scopes are equivalent (both are wide-scope). The property values are freely combinable, as C1 and C2 can be ranked relative to X independently of one another, generating the four grammars in the value table in (10b), 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-X.

(10) *SA Base: pre-sum system Property Analysis*

a. *Properties*

	α	β
P1: C1≪≫X	WeL	LeW
P2: C2≪≫X	eWL	eLW

b. *Value table*

	P1	P2	grammar
Lg1	α	α	WeL, eWL
Lg2	α	β	WLL, eLW
Lg3	β	α	LWL, LeW
Lg4	β	β	LeW, eLW (=LLW)

When C1 and C2 are summed, their ERC columns fuse, as in (11), and the P1 and P2 values thus become identical. When C1 and C2 are combined in a single constraint, there are only two possible rankings: C1+2≫X and X≫C1+2. As a result, any combination of α and β values for the parallel Ps are contradictions, fusing to $\mathcal{L}^+$. Grammars with mixed values of this sort rank the summands on either side of the shared antagonist X, an impossibility when these are a single constraint, so pre-sum grammars Lg2 and Lg3 are harmonically bounded (shaded in (11)), leaving two grammars, defined by the values of P1 and P2.

[3] McManus (2016) develops a similar concept of parallel properties.

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

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

(11) *SA Base: post-sum system Property Analysis*

a. *Properties*

	α	β
P1: C1+2⋘⋙X	WL	LW
P2: C1+2⋘⋙X	WL	LW

b. *Value table*

	P1	P2	grammar
Lg1	α	α	WL
Lg2	α	β	WL∘LW = LL
Lg3	β	α	LW∘WL = LL
Lg4	β	β	LW

These results generalise 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 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 α and β 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 formal result is spelled out in (12), along with its justification. To generalise across all property values, the following notations are used: 'v' for a property value ERC, '$\bar{v}$' for its reverse and '…' for other constraints with the same values (W/L/*e*) in all ERCs.

(12) *Proposition:* SA equalises properties and eliminates grammars.

Proof: Given a set of parallel properties, P1, … , P*n*, differing in antagonist, C1 to C*n*:

a. For a P*i*: C*i*⋘⋙X in the parallel properties set, value ERCs v and $\bar{v}$ are:

	C1…C*i*…C*n*	X
v	e…W…e	L
$\bar{v}$	e…L…e	W

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

	∘(C1–C*n*)	X
v	W	L
$\bar{v}$	L	W

c. For any two properties, P*x* and P*y*, if P*x*.v = P*y*.v, then P*x* = P*y*. If P*x* = P*y*, then P*x*.v + P*y*.$\bar{v}$ is a logical contradiction, fusing to $\mathcal{L}^{+}$ (as property values are mutually exclusive, defining an ERC (set) and its negation).

d. Therefore, a subset of any ERC set containing any $\alpha + \beta$ value combination fuses to $\mathcal{L}^{+}$, 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-sum system.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

The effects of SA are manifested as summing violations across summand columns in a UNIVERSAL VIOLATION TABLEAU (Prince 2015). A universal violation tableau is a violation tableau that represents a full typology, where each row is a language: the ERCs that result when each row is chosen as optimal and compared to every other row generate the grammar of the language (Prince 2015). 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 does, but not the other ($\alpha\beta$ or $\beta\alpha$). The universal violation tableaux for the SA Base pre- and post-sum systems are shown in (13). Summing violations across the rows (the 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 universal violation tableau*

	pre-sum			post-sum	
	C1	C2	X	C1+2	X
Lg1: α+α			2		2
Lg2: α+β		1	1	1	1
Lg3: β+α	1		1	1	1
Lg4: β+β	1	1		2	

This pattern is iterated for multiple antagonists, as shown in the universal violation tableau with antagonist {Xa, Xb}.*sub* in (14). The seven grammars of the pre-sum universal violation tableau reduce to three post-sum grammars.

(14) *SA universal violation tableau: κ.sub antagonist*

	pre-sum				post-sum		
	C1	C2	Xa	Xb	C1+2	Xa	Xb
Lg1				2			2
Lg2			2			2	
Lg3		1		1	1		1
Lg4		1	1		1	1	
Lg5	1			1	1		1
Lg6	1		1		1	1	
Lg7	1	1			2		

[5] While the minimum value is not necessarily 0 (see Merchant & Prince to appear), it can be reduced to 0 in the universal violation tableau. DelBusso (2018) defines a MINIMAL UNIVERSAL VIOLATION TABLEAU, in which violation counts are reduced to the minimum value that preserves order and equivalence relations.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

The formal changes from summing constraints in parallel properties are manifested extensionally in the parallel 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 by Bennett & DelBusso (2018), there are two doubly parallel properties, each of which has a class of one feature-specific CORR constraint and one feature-specific CC.ID constraint as one antagonist, and a class of IO.ID constraints as the other, as in (15).

(15) *Doubly parallel ABC properties*

P1[cont]	{CORR[vce],CC.ID[cont]}.*sub*≪≫{IO.ID[cont],IO.ID[vce]}.*sub*
P1[vce]	{CORR[cont],CC.ID[vce]}.*sub*≪≫{IO.ID[cont],IO.ID[vce]}.*sub*

These properties highlight a parallelism in the extensional typology: languages may have harmony for neither, either or both of the features [voice] and [continuant], and the harmony/no harmony choice depends on P1[vce] and P1[cont] values respectively. In segmental terms, the choice of whether an input with disharmonic stops like /t d/ surfaces faithfully, or assimilates or dissimilates, is a separate choice from whether inputs like /d z/ 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 in the same way as the choice about how to handle voicing disharmony.

Summing all of the CORR constraints into one constraint and all of the CC.ID constraints into another, as in (16), collapses the two choices into a single one: harmony for *both* features or for *neither* (as the summed constraint is violated).

(16) *Summed ABC parallel properties*

P1
{CORR[vce]|[cont],CC.ID[cont]|[vce]}.*sub*≪≫{IO.ID[cont],IO.ID[vce]}.*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. The summed constraints are more general, referring to larger sets of features. They are similar to completely general constraints proposed in the literature that lack any kind of feature reference; see McCarthy (2010) on CORR and Gallagher & Coon (2009) on CC.ID. The extensional result of summing across features is to yoke the two featural choices together.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

5 Summing In: 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 P1 [cont] and P1[vce] in (15). SI sums constraints within such a class, resulting in elimination of other properties and equalisation of grammars. These effects arise due to the structure of Property Analyses with $\kappa.op$'s: 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 $\kappa.op$ members must occur in another property or set of properties. For example, in P1: {A,B}.*sub*≪≫X, X is ranked relative to the subordinate of A and B, which are not themselves ordered in P1 values. As such, a second property, P2: A≪≫B, ranks the members of the class. For P2α, A≫B, B is subordinate and thus X is ranked relative to B in P1, and for P2β the reverse holds.[6] When the members of a $\kappa.op$ are summed, properties antagonising the summands are eliminated, as their values – attempting to rank a constraint relative to itself – become $\mathcal{L}^{+}$. Consequently, grammars of the pre-sum system differing only in the values of such properties become co-optimal (i.e. both are optimal under the same rankings). There is no distinction between optima which are differentiated only by the ranking of the summed constraints.

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

(17) *SI Base: pre-sum system Property Analysis*

a. *Properties*

	α	β
P1: {C1,C2}.*sub*≪≫X	WeL, eWL	LeW|eLW
P2: C1≪≫C2	WLe	LWe

b. *Value table*

	P1	P2	grammar
Lg1	α		WeL, eWL
Lg2	β	α	eLW, WLe
Lg3	β	β	LeW, LWe

Summing fuses the first two ERC columns, reducing the multi-ERC sets of P1 values to single ERCs (the conjuncts and disjunct become equivalent), as in (18). Additionally, P2 values become $\mathcal{L}^{+}$, as they cannot rank a single constraint relative to itself. Eliminating the illogical P2 makes

[6] See DelBusso (2018) and Alber & Prince (in preparation) for more on classes.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

Lg2 and Lg3 equivalent/co-optimal, as they are no longer distinguished by any property.

(18) *SI Base: post-sum system Property Analysis*

a. *Properties*

	α	β
P1: {C1+2}.*sub*≪≫X = C1+2≪≫X	WL	LW
P2: C1+2≪≫C1+2	Le	Le

b. *Value table*

	P1	grammar
Lg1	α	WL
Lg2/Lg3	β	LW

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

(19) *Proposition:* SI eliminates properties and equalises grammars.
Proof: Given a property P1: *κ.op*≪≫X, where C1…C*n* ∈ *κ.op*:

a. Value ERCs:
P1.v: there is at least 1 W in columns C1–C*n* and no Ls.
P1.$\bar{v}$: there is at least 1 L in columns C1–C*n* and no Ws.

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

i. *op* = *dom*: there is a W (v) or L ($\bar{v}$) in all columns C1…C*n*, fusing to W or L.

pre-sum	C1…C*n*	X
v	W…W	L
$\bar{v}$	L…L	W

post-sum	∘(C1-C*n*)	X
v	W	L
$\bar{v}$	L	W

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

pre-sum	C1…C*n*	X
v	{We…, eW…, …eW}	L
$\bar{v}$	Le…|eL…|…eL	W

post-sum	∘(C1-C*n*)	X
v	W	L
$\bar{v}$	L	W

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

c. Properties antagonising *κ.op* members are lost and grammars become equal.
   i. Given any P2: Y⋘⋙Z ∈ Property Analysis s.t. Y, Z ∈ *κ.op*.
   ii. Either Y = L or Z = L under either P2 value. Fusion of anything with L yields L; therefore, the fused column = L, eliminating the Ws.
   iii. If P2 value ERCs = $\mathcal{L}^+$, then no grammar in the typology can have a value of P2, and P2 ∉ Property Analysis.
   iv. Any pre-sum system pair of grammars, G1 and G2, s.t. the property values defining G1 and G2 are the same 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 property values.

Property Analyses with SI-eligible *κ.op*'s also have characteristic universal violation tableaux that show the summing effects. There are three violation patterns for which a *κ.sub* and a *κ.dom* differ in whether a candidate violates neither or both summands respectively, as in (20). Summing results in identical violation profiles for Lg2 and Lg3.

(20) *SI universal violation tableau*

a. *κ.sub*

	pre-sum			post-sum	
	C1	C2	X	C1+2	X
Lg1			1		1
Lg2		1		1	
Lg3	1			1	

b. *κ.dom*

	pre-sum			post-sum	
	C1	C2	X	C1+2	X
Lg1		1	1	1	1
Lg2	1		1	1	1
Lg3	1	1		2	

The loss of properties and subsequent co-optimisation of grammars are manifested 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.op*'s) of Corrs and CC.Ids in the properties above. 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 pair of Corr and CC.Id constraints into a single constraint that demands agreement (Agr). Summing the *κ. op*'s into Agr constraints results in the Ps in (21): Agr[cont]/[vce] = Corr[vce] + CC.Id[cont] and Agr[vce]/[cont] = Corr[cont] + CC.Id[vce]. This summing also eliminates from the Property Analysis additional properties antagonising Corr and CC.Id constraints, and in so doing results in languages becoming co-optimal, distinguished by only those values (Bennett & DelBusso 2020). Extensionally, the co-optimised

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

languages have the same segmental surface forms (i.e. [t_1d_1] or [t_1d_2]), and differ only in the presence or absence of correspondence indices (which 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*

P1[cont]	Agr[cont]/[vce]≪≫{IO.Id[cont],IO.Id[vce]}.*sub*
P1[vce]	Agr[vce]/[cont]≪≫{IO.Id[cont],IO.Id[vce]}.*sub*

6 Summary

A question in linguistic analysis concerns 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: what are the effects of splitting a single mechanism in an insufficiently rich theory in order to produce more degrees of freedom in a typology? Understanding the answers to these questions is fairly straightforward in any concrete case if we define both theories, produce their typologies and compare the results. But the possibility of a generalised 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 Property Analyses with parallel properties, equalising properties and eliminating grammars, while SI applies to Property Analyses with constraint classes, equalising 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 hypothesised 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 analysed and understood, related to the others through summing and unsumming (splitting).

REFERENCES

Alber, Birgit, Natalie DelBusso & Alan Prince (2016). From intensional properties to universal support. *Lg* **92**. e88–e116.

Alber, Birgit & Alan Prince (in preparation). *Typologies*. Ms, Free University of Bozen-Bolzano & Rutgers University.

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

Bennett, Wm. G. & Natalie DelBusso (2018). The typological effects of ABC constraint definitions. *Phonology* **35**. 1–37.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press

Bennett, Wm. G. & Natalie DelBusso (2020). Summing within and across properties. Paper presented at the SOTA IV workshop, Eckerd College, St Petersburg, Fl.

Crowhurst, Megan J. & Mark Hewitt (1997). *Boolean operations and constraint interactions in Optimality Theory*. Ms, University of North Carolina at Chapel Hill & Brandeis University. Available as ROA-229 from the Rutgers Optimality Archive.

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

DelBusso, Natalie & Wm. G. Bennett (2019). ABP and ABC: agreement with/out correspondence. *Phonological Data and Analysis* **1.3**. 1–25. https://doi.org/10.3765/pda.v1art3.15.

Gallagher, Gillian & Jessica Coon (2009). Distinguishing total and partial identity: evidence from Chol. *NLLT* **27**. 545–582.

Hansson, Gunnar Ólafur (2010). *Consonant harmony: long-distance interaction in phonology*. Berkeley: University of California Press.

Hansson, Gunnar Ólafur (2014). (Dis)agreement by (non)correspondence: inspecting the foundations. *UC Berkeley Phonology Lab Annual Report: ABC Conference Archive*. Slides available (July 2020) at http://linguistics.berkeley.edu/phonlab/documents/2014/ABCC/Hansson.pdf.

Hewitt, Mark & Megan Crowhurst (1996). Conjunctive constraints and templates in Optimality Theory. *NELS* **26**. 101–116.

Itô, Junko, Armin Mester & Jaye Padgett (1995). Licensing and underspecification in Optimality Theory. *LI* **26**. 571–613.

McCarthy, John J. (2010). Agreement by correspondence without Corr constraints. Ms, University of Massachusetts, Amherst. Available as ROA-1089 from the Rutgers Optimality Archive.

McCarthy, John J. & Alan Prince (1995). Faithfulness and reduplicative identity. In Jill N. Beckman, Laura Walsh Dickey & Suzanne Urbanczyk (eds.) *Papers in Optimality Theory*. Amherst: GLSA. 249–384.

McManus, Hope (2016). *Stress parallels in modern OT*. PhD thesis, Rutgers University.

Merchant, Nazarré & Alan Prince (to appear). *The mother of all tableaux*. London: Equinox. Available as ROA-1285 from the Rutgers Optimality Archive.

Prince, Alan (2002). Entailed ranking arguments. Ms, Rutgers University. Available as ROA-500 from the Rutgers Optimality Archive.

Prince, Alan (2015). One tableau suffices. Ms, Rutgers University. Available as ROA-1250 from the Rutgers Optimality Archive.

Prince, Alan (2017). Representing OT grammars. Available as ROA-1309 from the Rutgers Optimality Archive.

Prince, Alan & Paul Smolensky (1993). *Optimality Theory: constraint interaction in generative grammar*. Ms, Rutgers University & University of Colorado, Boulder. Published 2004, Malden, Mass. & Oxford: Blackwell.

Rose, Sharon & Rachel Walker (2004). A typology of consonant agreement as correspondence. *Lg* **80**. 475–531.

Samek-Lodovici, Vieri & Alan Prince (2005). Fundamental properties of harmonic bounding. Ms, University College London & Rutgers University. Available as ROA-785 from the Rutgers Optimality Archive.

https://doi.org/10.1017/S0952675720000172 Published online by Cambridge University Press