The typological effects of ABC constraint definitions°

Wm. G. Bennett
University of Calgary and Rhodes University

Natalie DelBusso
Rutgers University

Abstract
The plethora of recent work under the theoretical banner of 'Agreement By Correspondence' (ABC) has produced a variety of different—and sometimes contradictory—formulations of the constraints central to this framework. In OT, the effects of such definitional choices come out in the factorial typologies they predict. Yet knowing *what* languages a theoretical system derives is insufficient without knowing *why* it does so. This requires analysis of the internal ranking structures of the typology itself. This paper compares the typologies produced under different proposed modifications to the main ABC constraints. We analyze the typologies in Property Theory, a theory of typological organization in OT. Our analyses show that all such ABC variations have a common core structure, and that differences in their factorial typologies reduce to differences in how this common structure expands and iterates for different features. This allows for precise delineation of how and why different definitions of ABC constraints affect typologies.

Keywords: typology, Agreement By Correspondence, Property Theory, Optimality Theory, harmony, dissimilation

1. Introduction
Much work in Optimality Theoretic (OT) phonology seeks to develop a theory of CON: what constraints are necessary to account for linguistic phenomena and how they should be formulated. Each proposed set of constraints, CON, together with a set of possible forms, GEN, generates a factorial typology of predicted languages, produced under distinct orderings of the constraints. The factorial typology gives us the languages as sets of their optimal forms. But knowing *what* a theory predicts is insufficient; only when we understand *why* these predictions arise can we precisely characterize the aspects of a constraint set necessary for predicting a given phenomenon. This requires analyzing the internal structure of the typology itself. Property Theory (Alber & Prince in prep.), a theory of the organization of this space, explicates how the constraint interactions generate the languages. It allows us to precisely define how and why systematic changes to constraints alter typological predictions and thus to compare distinct theories of CON.

° For valuable input on earlier stages of this work, we give thanks to: three anonymous reviewers and associate editor whose feedback greatly improved the exposition of the paper, as well as Alan Prince, Gunnar Hansson, Stephanie Shih, Eric Baković, Sharon Rose, Luca Iacoponi, Naz Merchant; and audiences at NELS, AMP, UC Santa Cruz, and Rutgers. Portions of the work were supported by grants from the American Philosophical Society and the National Research Foundation of South Africa (NRF grant number 104103. Any opinion, finding and conclusion or recommendation expressed in this material is that of the author(s) and the NRF does not accept any liability in this regard.)

In this paper, we examine a set of typologies of surface correspondence systems under the Agreement By Correspondence (ABC) theory, systematically varying CON. ABC advanced the study of segmental harmony with two core insights: (i) long-distance agreement between consonants is rooted in similarity; and (ii) such agreement can be explained in a principled way by positing a similarity-sensitive mechanism of correspondence, coupled with agreement constraints that hold only over correspondents (Walker 2000a, 2000b, 2001; Rose & Walker 2004, Hansson 2001/2010). Subsequent work further developed this idea, both by enriching formal details, and by using the same mechanism to analyze other phenomena like dissimilation, vowel harmony, consonant-tone interaction, and so forth (Bennett 2015; Hansson 2007; Inkelas & Shih 2014, Shih & Inkelas 2014; McCarthy 2007, 2010; Rhodes 2010; Walker 2009, 2014, 2016).[1]

ABC analyses quintessentially posit two general constraint types referring to a correspondence relationship between segments in an output string. CORR constraints are violated by non-correspondence; they demand correspondence between segments – typically based on similarity. CC.IDENT (CC.ID) constraints are violated by disagreement between correspondents; they demand feature agreement contingent on correspondence. CORR & CC.ID constraints together drive harmony: both are satisfied when two segments correspond and agree. This interaction is the intuitive basis of agreement by correspondence, but these two constraint types can have other consequences as well; for instance, CORR and CC.ID are satisfied equally well by dissimilation (Bennett 2015).

While CORR & CC.ID constraints are a central feature of ABC, previous literature offers numerous different theories of CON differing in the formulation and number of possible constraint of each type. Arguments for particular proposals are generally based on specific empirical case studies (i.e., Walker 2016, Gallagher & Coon 2009), or on theoretical parsimony, with the proposed CON claimed to have fewer/simpler/more intuitive constraints (i.e., McCarthy 2010, who seeks parallels between a general feature-blind CORR constraint and MAX). To fully understand the consequence of each proposed constraint, the *typological effects* must be analyzed: not only what the constraints are, but how they interact with the other constraints in CON over the space defined by GEN.

This paper has two goals. First, we show that there is a general shared structure of all ABC systems, arising from a basic interaction among the central constraint types. We analyze this structure within Property Theory (PT; see Alber & Prince, in prep., Alber, DelBusso & Prince 2016; see also Alber & Prince 2017, Bennett, DelBusso & Iacoponi 2016, McManus 2016 for work within this theory). PT provides the analytical framework needed to understand how the formal pieces—GEN and CON—give rise to a theory's typological predictions. Factorial typologies are analyzed into sets of *properties*, binary 'choices' about constraint rankings. The set of choices made by a language distinguishes it from other languages in the typology, and groups it with other languages that make some of the same choices. Property Analysis (PA)

[1] Other, more critical, responses to the ABC framework see the technical machinery as either not necessary or not sufficient (see, e.g., Jurgec 2014, Hansson 2014, McMullin & Hansson 2015, McMullin 2016).

provides new and deeper insight into the fundamental mechanism of ABC theory: it enables us to compare the typologies derived from related theories.

This insight facilitates our second goal: determining how definitional alterations of CORR or CC.ID change the typology both *extensionally* and *intensionally.* In keeping with standard logical use of these terms, the *extensional* definition of a typology is the set of languages it contains, understood in terms of the linguistic traits of the optimal mappings. The *intensional* definition of a typology is the formal constraint rankings that generate those languages. We define and analyze six systems realizing distinct combinations of how the key ABC constraint types relate to features. Increasing the feature specificity of these constraints narrows the set of candidates that violate them: for example, a general feature-blind CORR constraint is violated by non-correspondence between any consonants in the output, whereas a feature-restricted CORR[αF] constraint is violated only by non-correspondence between segments sharing the value α of a feature F. The systems we analyze have different combinations of the core constraint types, inspired by constraint proposals in the recent ABC literature (or combinations thereof).

Property analyses of these constraint systems show both the repetition of the basic ABC structure, and how this structure expands with certain definitions of constraints. These analyses allow us to precisely delineate the characteristics of CON necessary to produce languages with dissimilation as well as those with harmony. Dissimilation emerges when *either* of the CORR or CC.ID constraints are featurally restricted: if featural similarity is a precondition for incurring a violation, then the pair of constraints can be satisfied by changing the segmental features to be *less* similar. An output form does not violate CORR[αF] if the segments do not share [αF], a fundamental insight of Bennett (2015). Our results show that the same result is obtainable by similarly restricting CC.ID, in the spirit of Walker (2016), and furthermore that the resulting typologies are *intensionally* parallel. The different theories not only result in the same extensional predictions; they also generate them in identical ways.

After giving a brief overview of ABC theory and the languages such systems generate (§1.1), we define our theoretical objects. All systems share GEN and the basic constraint types (§2). We then turn to the analysis of a minimal ABC system, T_{Core}, with only a single constraint of each type (§3). We conduct a full Property Analysis of this system, showing how properties link specific *extensional* traits of languages to *intensional* rankings in the grammars. We then define and analyze a set of systems, each of which is characterized by systematically changing one or both the ABC constraint type(s) by adding featural specifications to their definitions (§4). While all share the common core structure, they differ in both empirical predictions—most notably the presence/absence of dissimilation—and the way in which the core structure expands and multiplies. In §5 we compare the systems on both extensional and intensional levels and explain how CON changes manifest in the typological predictions.

1.1. ABCs of ABC(D)

Agreement By Correspondence is a theory of segmental agreement that establishes a correspondence relationship over sets of segments in an output form. The existence of such a relation between segments in optima of a given language is controlled by CON, with GEN supplying both corresponding and non-corresponding forms (and mixes thereof). CORR constraints are violated by non-correspondence between segments defined by some natural class.

By referring to classes, CORR constraints require correspondence based on similarity: segments required to correspond share some set of feature values. Agreement occurs because of another type of constraints, CC.ID, that parallel IO.ID constraints in being violated by featural disagreement between the corresponding segments.

Bennett (2015; esp. §2.4) shows that the same core mechanisms that derive agreement also derive its complement, dissimilation. Segments are unfaithfully mapped to be *less* similar. This escapes violation of CORR constraints when the change results in the segments not sharing the natural class picked out by that constraint. By not corresponding, they vacuously satisfy any CC.ID constraints that require agreement between correspondents. Because of this result, we append (D) to ABC(D) when discussing a theory generating languages with dissimilation in some optima. Not all theories do so; a result of this study is a characterization of the necessary elements of CON to produce it.

There are four general types of optimal mappings that occur in simple ABC(D) systems. One, the original goal of the theory, is segmental agreement, called 'harmony'. In this type, abbreviated as har, some input segment(s) are unfaithfully mapped so that corresponding segments in the output forms agree in some feature(s). The complement, dissimilation (dis) involves unfaithful mapping to disagree. The other two types are faithful mappings, where input segment features are matched in the output. These differ in whether the output segments correspond (cor) or not (noc), with repercussions for their constraint violations. Note that these two types are overt-form indistinguishable, as correspondence indices are hidden structure; the segmental feature mappings in both types are equivalent.

The four types, which are the four possible optima, are schematized below for an input sequence /d z/ (we omit vowels from representations as the current focus is consonant relations). Subscripts indicate correspondence indices: matching for correspondence, non-matching for non-correspondence. Segments with a harmony mapping always have matching indices; ABC theories only produce harmony between correspondents. Segments with a dissimilatory mapping could potentially correspond, or not, depending on the constraints of a particular system. We represent indices as '#' for the dis form for this reason.

(1) Extensional mapping types

Input	*Output*[2]	*Comment*
/d z/	d_1 z_1	faithful correspondence
	d_1 z_2	faithful non-correspondence
	d_1 d_1	non-faithful harmony
	$d_\#$ $s_\#$,	non-faithful dissimilation

Languages are named by the extensional type(s) exhibited in their optima. Depending on CON, this need not be uniform across a language for every class of segments. For example, in the

[2]Here, and in subsequent tableaux and factorial typologies, additional co-optima are not shown. Certain types of co-optimality recur across all theories considered in this paper. For instance, with unfaithful mappings, either the first or second segment can change features, e.g. /t d/ → [t_1t_1] or [d_1d_1], as no constraint distinguishes between these. In general, we show the co-optimum in which the first segment is faithful.

majority of the systems analyzed here, a language may have harmony (har) between segments sharing [+voi] and faithful correspondence (cor) between those of the class [-cont].

2. Defining the theoretical objects of analysis

To analyze the effects of distinct variations of Con, we hold Gen constant, providing a fixed frame of reference within which we can rigorously and exhaustively examine the typologies. In defining Gen, we aim to have the minimum space of possibilities needed to show the basic mode of interaction of each theory. More specifically, the target interaction is (potential) similarity-based consonant harmony: "if two consonants share one feature, then they agree for another feature." We therefore consider only two-consonant strings, and two features, [±voice] ([±voi]) and [±continuant] ([±cont]), yielding the segmental inventory shown in (2).[3] These two features are used both for their simplicity and familiarity, and because they link the typological questions of this paper to a real-world point of contact in the form of Yabem, a language where voicing and stricture harmony co-occur (Hansson 2004, 2010, Hansson & Entwistle 2013).

(2) Segmental inventory

[cont] \ [voi]	+	-
+	z	s
-	d	t

As inputs, we admit all possible strings of any two consonants. Outputs consist of the same set of possible strings, with or without correspondence between the segments (the only two surface correspondence structures possible for two segments). Gen is defined below.

(3) Gen

Inputs: ∀x, y: x, y ∈ {t, d, s, z}, /xy/ ∈ In. (16 inputs)

Outputs: ∀x, y: x, y ∈ {t, d, s, z}, [x#y#] ∈ Out,
where # ∈ {1, 2} = correspondence indices. (32 outputs/input)

We follow Bennett (2015) in defining correspondence as *unitary, symmetric*, and *transitive*. *Unitary* means that there is a single correspondence relation, and outputs have a single structure of that relation; if x is in correspondence with y, then it cannot also be in *non*-correspondence with y.[4] *Symmetric* correspondence means that if x corresponds with y, then y also corresponds with x. Since there are only two segments, only two correspondence structures are possible: the output consonants either correspond, or do not. As we are considering only two-segment forms,

[3]While not all ABC interactions may be derivable in a 2×2 space, it is sufficient to distill the core structure of ABC(D) systems and differences between the Con proposals. Investigation of elaborations shows that both the general structure and the systematic differences persist under any (predictable) changes. See §5.

[4] This may seem trivial or obvious, but it sets correspondence apart from merely sharing features. Since segments have multiple features, two consonants can share the same value of one feature, while having different values of another feature. Accordingly, unitary correspondence is different from an alternative where each feature has its own correspondence relation (à la tiers in autosegmental phonology; see also Shih & Inkelas to appear).

transitivity—if x is in correspondence with y, and y in correspondence with z, then x is in correspondence with z—plays no part here.

All variations employ the same fundamental types of constraints to explain correspondence and consonant harmony, CORR and CC.ID[5], described informally in (4).

(4) Core constraint types of ABC
   i. **CORR**: violated by non-correspondence of output segments.
   ii. **CC.ID**: violated by non-agreement for some feature(s) between output correspondents.
   iii. **f.IO**: violated by non-agreement for some feature(s) between input-output correspondents.

The existence of—and distinction between—CORR and CC.ID constraints is a defining feature of ABC(D) theories, setting them apart from other approaches to long-distance segmental interactions, like those based on spreading (Mester 1986, Yip 1988, Jurgec 2011, Kimper 2011) or other mechanisms (Nevins 2004, Gallagher 2010, Walker 2011; also Hansson 2014). The basic idea of CORR and CC.ID constraints is rooted in earlier work, most notably Rose & Walker (2004) and Hansson (2001/2010).[6] Both CORR and CC.ID assess outputs only: their violations depend on the correspondence structure of the candidates, which only exists in outputs.

ABC(D) analyses derive consonant harmony through the interaction of these basic constraint types. Satisfaction of both CORR—segments are in correspondence—and CC.ID—corresponding segments agree—becomes possible for disagreeing input segments by changing features of one segment to match those of the other, violating f.IO.

Formal definitions for the core types are given below, adapted from Bennett (2015).

(5) CON of a simple ABCD theory: 'Core'

Constraint	*Definition*	*Prose: assign a violation for each:*
CORR	$*C_{\mathbf{x}}C_{\mathbf{y}} \in$ out	non-corresponding pair of output consonants.
CC.ID	$*C[\alpha F]_x.C[\beta F]_x \in$ out	pair of output correspondents that are not identical for all features.
f.IO	$*C[\alpha F] \in$ in & $C[\beta F] \in$ out for all features $F \in C$	pair of input-output correspondents that are not identical for all features.

3. T_{Core} and ABC(D) ranking structures

The Core system in (5) is the most basic instantiation of an ABC typology, using general feature-insensitive versions of all three basic constraint types. Though it does not predict the full desired range of languages, study of this system elucidates the central interactions of ABC, which we show to be the core defining structure of all subsequent systems.

---

[5] Bennett (2015) situates CC.ID in a larger class of CC.Limiter constraints, which include other kinds of constraints, such as those violated by correspondence across morphological or phonological boundaries.
[6] It also bears similarities to other related work that is not strictly in the ABCD tradition, such as Wayment (2009), Krämer (1998).

This section then develops the formal property analysis of this system, and its typology ($\underline{T}_{Core}$). We assume no prior knowledge of Property Theory (Alber & Prince, in prep., Alber, DelBusso & Prince 2016), defining the concepts and mechanisms as they are introduced. We do assume familiarity with basic tenets of OT logic, specifically Elementary Ranking Conditions (ERCs; Prince 2002a et seq.), as well as Violation and Comparative Tableau (VTs & CTs; see Prince & Smolensky 1993/2004, and Prince 2002b, respectively).

The core objects of analysis are defined below in (6). Following work in modern formal OT (Merchant & Prince 2016, Prince 2016a, 2016b), grammars are distinguished from languages. The former is a set of intensional rankings, characterized by an ERC set; the latter is the set of extensional forms that are optimal under these rankings. Typological analysis proceeds both *extensionally*, through the list of languages generated, and *intensionally*, through the rankings giving rise to those languages. The property analysis of a typology, PA(T), analyzes its intensional structure, decomposing it into the grammatical choices defining the system. By delineating the distinguishing rankings that determine the grammars of the typology and aligning these with the extensional linguistic structures, a PA leads to an understanding of how the constraints interact to produce the predicted languages in the typologies.

(6) Definitions: *Language, Grammar, Typology*
- a. *Language*: the set of optima under a given constraint hierarchy.
- b. *Grammar*: a set of linear orders on CON that produce the same language (select the same set of optima); characterized as an ERC set.
- c. *Typology (T_S)*: extensional: the languages of the system.
  intensional: the grammars of the system.

The violation tableau in (7) shows four inputs and the possible, non-harmonically bounded (Samek-Lodovici & Prince 2005), outputs for each in the Core constraint system.[7] For all systems analyzed in this paper, inputs (7)a–(7)b are a *Universal Support* (Alber, DelBusso & Prince 2016): a set of candidate sets (csets) necessary and sufficient to determine all possible grammars. In these candidates, input segments share one feature, but differ in the other. Either of these alone, or (7)d, is a universal support for T_{Core}, because no constraint refers to particular features. The violation profiles in (7)a, (7)b and (7)d are identical. Both inputs are necessary in subsequent systems, where some constraints are feature-specific, and their behavior can come apart. The choice of these particular combinations is justified in §4; to illustrate sufficiency here, consider inputs (7)c–(7)d. Input (7)c shows that for segments agreeing in *all* features—*more similar* than those in /td/ and /dz/; there is a single possible optimum satisfying all constraints. Consequently, such inputs always map faithfully and correspond. Input (7)d shows two segments differing in both features—*less similar* than /td/ and /dz/. Their treatment is predictable from that of /td/ and /dz/, entailed by the rankings deriving these.

For all systems analyzed here, order of the segments is unimportant; reverses of inputs have the same range of mappings. In T_{Core} there are three possible outputs: faithful correspondence (cor); faithful non-correspondence (noc); and unfaithful harmony (har), for which there are co-optima, changing a feature in either segment. Each of these violates a single constraint of the three types.

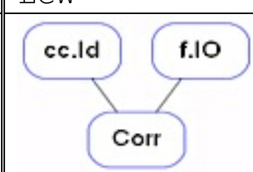
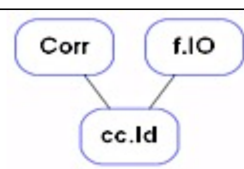
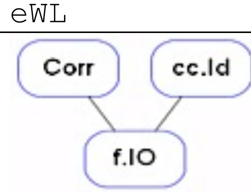
[7]Full typologies with full candidate sets for all systems were calculated and analyzed in OTWorkplace (Prince, Tesar & Merchant 2007-2017).

(7) T_{Core} Violation Tableau

Input	Output	Corr	CC.Id	f.IO
a. t d	t_1d_1		1	
	t_1d_2	1		
	t_1t_1			1
b. d z	d_1z_1		1	
	d_1z_2	1		
	d_1d_1			1
c. t t	t_1t_1			
d. t z	t_1z_1		1	
	t_1z_2	1		
	t_1t_1			1

The three languages and grammars of T_{Core} are shown in (15) below. Languages are named by the extensional types cor, noc and har, according to the optimal mapping for inputs /td/ and /dz/ in that language. In each grammar, one of the three constraints is dominated by both of the others, which are not crucially ranked relative to each other. Each grammar is named *C-bot*, where C is the bottom-ranked constraint in all hierarchies consistent with the grammar, following the terminology of Merchant & Prince (2016). This ranking structure was shown to be characteristic of grammars in ABC(D) systems in Bennett et al. (2016) (also Bennett 2015:75). In Corr-bot, all optima are faithful and do not correspond; they violate Corr but not f.IO or CC.Id. In CC.Id-bot, optima are also faithful, but segments *do* correspond; this violates CC.Id, because they differ in feature values, but satisfies Corr instead. Finally, in f.IO-bot, optima are unfaithful, because they correspond and *harmonize*; this violates only the bottom ranked f.IO constraint. The order of constraints in the ERCs follows that of the VT above: Corr-CC.Id-f.IO.

(8) Languages of T_{Core}

	no-c(orrespondence)	cor(respondence)	har(mony)
/t d/	t_1d_2	t_1d_1	t_1t_1
/d z/	d_1z_2	d_1z_1	d_1d_1
	faithful		*unfaithful*
Grammar	Corr-bot	CC.Id-bot	f.IO-bot
ERCs	LWe LeW	WLe eLW	WeL eWL
Hasse diagram			

To understand how the system defines and classifies the grammars, we turn to *Property Analysis*. A property analysis (PA) analyzes a typology into a set of properties. *Properties* are binary sets of ranking conditions, over two sets of constraints, X and Y. The values of the properties, α and β, are the rankings, as ERCs, that are generated by reading the property statement in either direction: α. X ≫Y and β. Y ≫ X. The property is written as X <> Y.

T_{Core} is a three-way partition. No single ERC-representable ranking is shared by any two grammars; instead, each grammar is defined by the constraint on the *bottom*, which can only be expressed as a conjunction of two ERCs, one for each crucial domination. This structure might at first seem un-amendable to a PA consisting of binary classifications. However, there is an intuitive binary distinction to be found between the grammars by grouping the faithful languages {cor, noc} together, as opposed to the unfaithful one, har. This extensional trait is captured in the Property Analysis by referring to a *class* of correspondence constraints, κ = {CORR, CC.ID}.[8] The grammars with faithful mappings share the characteristic that exactly *one* of these constraints is crucially dominated, while in their complement, har, *both* are dominated. These rankings are generated using the operator *sub* that picks out the *subordinate* member of a set of constraints in a linear order, λ (Alber & Prince in prep.).[9]

(9) *Def*. κ.sub(λ): returns the lowest ranked (*subordinate*) constraint in the class κ in λ.

For example, if in a linear order, CORR ≫ CC.ID, then CC.ID is the subordinate member, κ.sub. The operator has quantificational force through transitivity: if the subordinate member (CC.ID) dominates f.IO, then so too does the dominant member, CORR, by dominating CC.ID. Conversely, if the subordinate member is *dominated*, f.IO ≫ CC.ID, then no ranking is established between the dominant member, CORR, and f.IO.

The result of classifying the typology into the faithful vs. unfaithful languages is a nested property structure. The faithful languages are defined by the ranking of {CORR, CC.ID}.*sub* vs. f.IO – a binary choice. The distinction between the two faithful languages, cor and noc, is another choice, about the ranking between CORR and CC.ID. The three-way partition comes out as the result of two properties – two binary choices.

The first property, P1, is stated in (10), with the ERCs generated by its values. All ERCs are given in the order CORR-CC.ID-f.IO, following the same order as the VT above. This property classifies the grammars by faithfulness of input feature mapping: the grammar that admits unfaithful mappings has the ranking generated by the α value of the first property, P1.α. The two faithful grammars instead have the β value, P1.β. The bifurcation of T_{Core} by P1 is shown in the value table in (11), which lists the grammars in the left column, with their values on P1.

[8] Note that the symmetry of the ranking structures of bot systems allows for alternative groupings of the constraints into classes. In this paper, we use classes of correspondence constraints and of f.IO constraints for all systems. Different groupings produce the same set of grammars, but the properties differ in the extensional classifications they make. For example, classing together {CC.ID, f.IO} against CORR aligns with a correspondence {har, cor} versus non-correspondence {noc} partition. Property analysis finds such binary divisions even in the absence of an intuitive divide (DelBusso & Prince, in prep.).

[9] The operator *sub* has a dual in *dom* that picks out the dominant, top-ranked constraint in the set over a λ. This does not occur in PAs in this paper; see Alber, DelBusso & Prince (2016), Alber & Prince (2017). Prince & Smolensky (1993/2004, ch. 8) use similar operators called *min* and *max*.

(10) PA(T_{Core})

P1: {CORR, CC.ID}.sub <> f.IO

α. `WeL & eWL` CORR & CC.ID ≫ f.IO-bot

β. `LeW | eLW` f.IO ≫ CORR *or* f.IO ≫ CC.ID

(11) Property 1 Value Table

Grammar	P1
CORR-bot	β
CC.ID-bot	β
f.IO-bot	α

P1.α generates two ERCs: CORR ≫ f.IO, and CC.ID ≫ f.IO. These two ERCs are the ranking conditions that fully define the grammar f.IO-bot and under which unfaithful mappings are possible in the languages: when both CORR and CC.ID dominate f.IO. Property P1 links the *intensional* characteristic of having IO faithfulness as the bottom-ranked constraint, to the *extensional* trait of allowing unfaithful mappings – both of which distinguish f.IO-bot from the other two grammars in the typology.

P1 does not distinguish between the two grammars CORR-bot and CC.ID-bot. These both share the characteristic of having the IO faithfulness constraint ranked above one or the other of the correspondence constraints, so they share the extensional trait of having only segmentally faithful mappings. Segmentally faithful candidates differ in which correspondence constraint they violate based on their correspondence indices: disagreeing segments can either correspond (violating CC.ID), or not (violating CORR). The CC.ID-bot language chooses the former; the CORR-bot language chooses the latter. These grammars are characterized by P1.β, which generates a disjunction of two ERCs; in these grammars, *either* f.IO ≫ CORR *or* f.IO ≫ CC.ID. Determining which requires an ERC ranking CORR and CC.ID relative to each other, not just relative to f.IO—a second property. P2 orders CORR and CC.ID; this intensional choice correlates the extensional trait of the existence of correspondence between segments in optima.

(12) PA(T_{Core})

P2: CORR <> CC.ID

α. `WLe` CORR ≫ CC.ID (= CC.ID-bot)

β. `LWe` CC.ID ≫ CORR (= CORR-bot)

CC.ID-bot has the value P2.α, while CORR-bot has P2.β. These values, together with the P1.β value shared by both grammars, yield the ERC sets that fully define each grammar.

The grammar f.IO-bot has neither value of the property P2. The constraints ranked by its values are not crucially ordered in this grammar: when both correspondence constraints dominate f.IO, their relative ranking does not matter — either ordering results in the same forms being optimal. P2 is a *narrow-scope* property, one for which only a subset of grammars have values, where the subset being picked out by a shared value on another property. P2 is *moot* for all other grammars (Alber & Prince in prep.). A property has *wide-scope* if all grammars in the typology have a value. The scope of P2 is P1.β, which includes CC.ID-bot and CORR-bot.

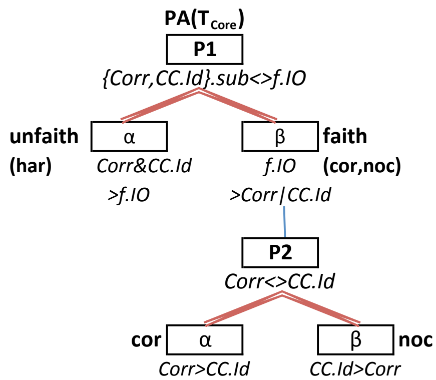
The full value table shows the possible value combinations of both properties (13). P1 distinguishes f.IO-bot from the other two grammars. P2 splits CORR-bot and CC.ID-bot, so all grammars are defined by distinct sets of values. The PA structure is represented in a *treeoid* (13b): a directed tree graph, augmented with different kinds of lines to encode kinds of domination among nodes (Alber & Prince in prep.). Double red lines connect a property to its values, representing a mutually-exclusive choice. Single blue lines indicate scope: a grammar has a value of that property if it also has the value dominating that property in the treeoid. The treeoid is annotated with the rankings defined by each property value, and the extensional traits that result from each choice.

(13) PA(T_{Core})

a. Value Table

Grammar	P1	P2
CORR-bot	β	β
CC.ID-bot	β	α
f.IO-bot	α	

b. Treeoid



P1 and P2 constitute a *full* property analysis of the Core system typology, PA(T_{Core}). This set of properties jointly generates all and only the grammars of the typology T_{Core}. It is a complete intensional and extensional classification of the typology. The T_{Core} languages are fully determined by two basic choices: a) are segments faithful to the input? and, b) if so, are segments in correspondence?

The simplicity of T_{Core} does not trivialize the insights of the property analysis. While extensional predictions are easily seen from the factorial typology, the properties link these to specific rankings, allowing us to understand how the constraints interact and which rankings are crucial. In essence, PT gives us not just a look at *what* constraints predict, but also *how* they make those predictions. In the following section we show that the T_{Core} structure and properties lie at the core of *all* ABC(D) systems. Larger and more complex typologies use the same basic units of analysis, expanded in principled ways.

4. The systems: the effects of varying CON

This section analyzes the typologies resulting from distinct formulations of the core constraint types, to address what effect such choices have on the typology. The results show where the systems come apart in their empirical predictions, and also crucially *why* they do—or do not—predict different typologies. We first review some of the motivation for the various formulations and combinations instantiated in the systems.

A central insight of ABC theory is that harmony is conditioned by segmental similarity, with more similar segments more likely to harmonize. The original work in this theory proposed families of CORR.αF and CC.ID.F constraints, often specified for particular features (F) or feature values (αF). These regulate correspondence between segments, or require agreement between correspondents for a single feature (rather than all features). Constraint sets of this type are proposed by Walker (2000a, 2000b, 2001), Rose & Walker (2004), Hansson (2001/2010), Bennett (2015), among others.

The similarity sensitivity characteristic to ABC theory does not arise in a system with only general, feature-blind constraints like T_{Core}. However, this system is grounded in extant proposals from the literature that merge together all members of one or the other of the essential ABC constraint families, CORR and CC.ID. McCarthy (2010) proposes a revision of ABC in which a family of feature-specific CC.ID constraints interacts with a single CORR constraint that requires correspondence between any two segments regardless of featural similarity. Gallagher & Coon (2009) propose an analogous unification of CC.ID constraints; in that proposal, a single constraint requiring total identity, interacting with a family of feature-specific CORR constraints.[10] Other versions of ABC build on these fundamental adjustments. Walker (2016) adopts McCarthy's unified CORR proposal, but builds additional feature restrictions into CC.ID constraints such that they penalize disagreement on one feature only between segments that share another feature (i.e. CC.ID constraints for [±F], that assign violations only to forms with [αG] segments). This feature restriction parallels that used on feature-restricted CORR constraints, in that agreement violations are contingent on similarity. These proposals form the basis for variations of ABC(D) theory analyzed here. We depart from the cited authors in some of the formal details of the constraint definitions and other assumptions about GEN, in the interest of holding these constant across theories.

The systems examined here are distinct combinations of varying the kinds of CORR and CC.ID constraints used, in how they relate to features. We recognize three distinct ways that previously-proposed constraints may be sensitive to features (14).

(14) Types of feature-sensitivity in constraint definitions

i. **General** (G): does not refer to features (all segments treated the same way; no feature sensitivity).
ii. **Specified** (S): picks out a particular feature as locus of violation (e.g. [±F] agreement).
iii. **Restricted** (R): only segments with a particular feature value can incur violations (e.g. assign a violation *only if* [αF]).

General constraints are feature-blind, as in T_{Core}'s unified versions of CORR or CC.ID. Feature-*specified* constraints assess violations for specific features: the CC.ID.F constraints that require agreement for F. Feature-*restricted* constraints assign violations only to segments that share a particular feature value specification, and also fail some other condition. These are constraints like the CORR.αF of Rose & Walker (2004), which require correspondence between two consonants only if both are [αF]. The specific versus restricted distinction is highly significant:

[10] We take some liberties in labeling here, in order to convey the fundamental relationships between different theories; these proposals use different labels for some of the constraints. McCarthy (2010) calls the unified CORR constraint 'MAX-CC', rather than CORR. Gallagher & Coon (2009) use the name 'LINK' instead of CORR.

as we show, the presence of feature-restricted constraints in a system is what gives rise to dissimilation.

We consider six ABC systems, characterized by different types of feature-sensitivity for the CORR and CC.ID constraints. Formal definitions of all constraints in the systems are in (15). (Recall from §2 that all systems share the same GEN.)

(15) CON (all systems)

Constraint	*Type*	*Definition (F,G = features)*	*Prose definitions: assign a violation for each…*
CORR	G	$*C_x\ C_y \in$ out	non-corresponding pair of output consonants.
CORR.αF	R	$*C[\alpha F]_x\ C[\alpha F]_y \in$ out	non-corresponding pair of [αF] output consonants.
CC.ID	G	$*C[\alpha F]_x\ C[\beta F]_x \in$ out	pair of output correspondents that are not identical for *all* features.
CC.ID.F	S	$*C[\alpha F]_x\ C[\beta F]_x \in$ out	pair of output correspondents disagreeing in [±F].
CC.ID.F/αG	R	$*C[\alpha G,\alpha F]_x\ C[\alpha G,\beta F]_x \in$ out	pair of [αG] output correspondents disagreeing in [±F].
f.IO.F	S	$*C[\alpha F] \in$ in & $C[\beta F] \in$ out	pair of input-output correspondents disagreeing in [±F].

In all our systems, feature-specified and feature-restricted constraints occur in pairs, one for each feature – e.g. a CC.ID.v with a CC.ID.c. Because feature-restricted constraints by nature target specific values of the relevant feature, we standardize feature values across the systems: CORR.αF and CC.ID.F/αG constraints target [+voi] and [-cont].[11] The combinations making up each system are in (16). Each system is named for the kind of CORR and CC.ID system used, G/S/R, in that order. The total number of constraints, including the two f.IOs, follows the name, in parentheses.

(16) CON for each system

	CORR	CORR.+v, CORR.-c
CC.ID	G.G (4)	R.G (5)
CC.ID.v, CC.ID.c	G.S (5)	R.S (6)
CC.ID.v/-c, CC.ID.c/+v	G.R (5)	R.R (6)

The system G.G differs from T_{Core} only in having two feature specific f.IO constraints instead of one unified faithfulness constraint; it retains the unified versions of both CORR and CC.ID. The effect of splitting f.IO is that harmony or its lack is determined on a feature-specific basis, rather than globally for all inputs. The uncontroversial featural split in the f.IO constraints is maintained in all other systems, as the current focus is the effect of varying the correspondence-sensitive constraints. G.S and G.R also retain the general CORR of T_{Core} but add specification and restriction to CC.ID, respectively. The G.S system most closely implements McCarthy's (2010)

[11] A feature-restricted constraint with no value reference would be like CORR.v, 'segments that have the feature [±voi] must correspond'. Since all segments are assumed to be specified for [+voi] or [-voi], this amounts to the same thing as feature-general CORR. Note also that R and S types of feature-sensitivity are not mutually exclusive: CC.Id.F/αG constraints are specified for one feature, and restricted based on another.

proposal to unify the CORR constraints; G.R is inspired by Walker's (2016) proposal to supplement this with feature restrictions on CC.ID. The remaining systems realize all combinations of the various CC.ID constraints with a restricted CORR. R.G approximates Gallagher & Coon's (2009) proposal to unify all CC.ID constraints into one requiring total identity. R.S represents the 'standard' model familiar from much ABC(D) work, particularly Rose & Walker (2004). Detailed analysis of the final system, R.R, is not shown in this paper. It is extensionally and intensionally equivalent to R.S – the unsurprising result of imposing the same preconditions on agreement redundantly by both constraint types.

To compare the different versions of the constraints, an outputs-only VT with all of them is shown in (17). CORR and CC.ID only assess output forms, so their violation profiles are the same for any input.

(17) Relations between Cs: VT[12]

		CORR	CORR.+v	CC.ID	CC.ID.c	CC.ID.c/+v
correspondence	t_1t_1					
	t_1d_1			1		
	t_1s_1			1	1	
	t_1z_1			1	1	
	d_1d_1					
	d_1s_1			1	1	
	d_1z_1			1	1	1
	s_1s_1					
	s_1z_1			1		
	z_1z_1					
non-correspondence	t_1t_2	1				
	t_1d_2	1				
	t_1s_2	1				
	t_1z_2	1				
	d_1d_2	1	1			
	d_1s_2	1				
	d_1z_2	1	1			
	s_1s_2	1				
	s_1z_2	1				
	z_1z_2	1	1			

General CORR requires correspondence irrespective of similarity; it assigns violations to all non-correspondent candidates, even maximally dissimilar ones like [t_1z_2]. Feature-restricted CORR.+v, assigns violations only to a subset of these: violations are incurred only if the two segments are both [+v]. So, d_1z_2 receives a violation, but t_1z_2 and d_1s_2 do not; correspondence is not required if two segment are not both [+v].

Similarly, general CC.ID assigns violations to pairs of correspondents that differ in any feature. Feature-specific CC.ID.c assigns violations only to the subset of those that differ specifically in

[12]Omitted are here violation-identical forms that differ from those shown only in segmental order.

the feature [±cont]: it penalizes t_1z_1 and d_1s_1, but not t_1d_1 or s_1z_1 (which share the same [±c] value but differ in [±v]). Finally, the violations of feature-restricted CC.ID.c/+v are a subset of those assigned by CC.ID.c: it penalizes only correspondents that disagree on [±c] *and* share [+v]. Thus, d_1z_1 receives a violation, but t_1z_1 does not; the constraint is vacuously satisfied whenever two correspondents fail to meet the similarity threshold required.

A central insight of work in the ABC(D) literature is that dissimilation offers an alternative to agreement to jointly satisfy CC.ID.F and CORR.αF constraints (observed by Walker 2000b, Gallagher & Coon 2009, formally developed in Bennett 2015). When segments do *not* share the feature specification picked out by a CORR.αF, their lack of correspondence does not incur a violation. This effectively permits those segments to disagree for another feature, since disagreement between non-correspondents does not violate CC.ID. Consequently, limits on correspondence (like agreement demanded by CC.ID) are incentives for non-correspondence – achievable by dissimilation. We show that the same result also obtains with feature-restricted CC.ID constraints: if correspondents differ in αG, then disagreement on F does not violate CC.ID.F/αG. Thus dissimilating candidates are possible optima in similar ways across different theories. The property analyses show that this has a direct intensional motif: T_{Core} grammars are bot structures over three types of constraints. Including a feature-restricted constraint changes these structures to involve *four* constraints, involving the f.IO constraints for both features.

In the following subsections we give the PAs of each of the ABC(D) systems defined in (16). The treatment is less in-depth than for PA(T_{Core}), as the property structures and scopes are familiar from that analysis. We focus on the differences that arise under each variation. In §5, we compare the systems both extensionally and intensionally.

4.1. G.G: unified CORR and CC.ID

The G.G system contains the same single generalized CORR and CC.ID constraints as Core, along with faithfulness constraints for each feature. The typology, T_{GG}, consists of 7 languages, defined by combinations of har, cor and noc, on a feature-specific basis. These are the same types possible in T_{Core}; however, with a non-unified f.IO, inputs differing in [±voi] can behave differently from those differing in [±cont]. The languages are shown in (18) as their optima for two inputs: /dz/ and /td/ (two csets that constitute a Universal Support, as noted in §3). The first, /dz/, determines the mapping for segments sharing [+voi] and differing on [±cont]. The second determines the mapping for segments sharing [-cont] and differing in [±voi].[13] Inputs where segments differ in both features have mappings predictable from the combination of those in the support. Inputs with two identical segments (i.e. no difference) are mapped the same way in every grammar.

[13] The two inputs here are not the only valid Universal Support for this system. The choice of these as opposed to /ts/ and /sz/ is based on the later systems, where [+voi] and [-cont] are the feature values targeted by restricted constraints.

(18) Languages in T_{GG}

Inputs / Lgs: c.v	/d z/	/t d/
har.har	d_1 d_1	t_1 t_1
har.cor	d_1 d_1	t_1 d_1
har.noc	d_1 d_1	t_1 d_2
cor.har	d_1 z_1	t_1 t_1
cor.cor	d_1 z_1	t_1 d_1
noc.har	d_1 z_2	t_1 t_1
noc.noc	d_1 z_2	t_1 d_2

The grid in (19) shows the possible combinations of mappings that constitute the typology. As in Core, no candidate with dissimilation (dis) can be optimal. Because both correspondence and agreement constraints are feature-blind, an unfaithful mapping that results in disagreement does not satisfy either: it is not an escape from correspondence or agreement in this theory. The typology also does not include languages realizing every possible combination of the three types: the two faithful types, cor and noc, are incompatible. If a language is faithful, all optima either have or lack correspondence, regardless of the features of the input.

(19) T_{GG}: combinations of mappings for subsystems

cont / voi	har	cor	noc
har	X	X	X
cor	X	X	
noc	X		X

Exactly as in PA(T_{Core}), the grammars are determined by a bot ranking structure between CORR, CC.ID and f.IO. Different from that system, however, is that this structure exists with *each* f.IO constraint. Thus, f.IO.c may be dominant (faithfulness when segments disagree in ±cont, /dz/), while f.IO.v is dominated by both CORR and CC.ID (harmony for segments disagreeing in ±voi, /td/). Each f.IO constraint defines a *subsystem* of the full PA. A 'subsystem' is a subset of the properties—involving the interaction of a subset of the constraints—that determines the optima for a subset of inputs. This formalizes the fact that only some of the constraints determine the choice of a given mapping. In G.G, the subsystem that includes f.IO.c determines the treatment of inputs with segments differing only in [±cont]; that with f.IO.v determines the mapping where input segments differ in [±voi]. In Core there is a single property for the ranking between the class of correspondence constraints and faithfulness, resulting in a single extensional choice of faithfulness or harmony. T_{GG} splits this into two properties, one for each of the feature-specific faithfulness constraints, allowing for distinct mappings depending on segmental features.

The full PA of the T_{GG} system is given below. P1c and P1v replicate P1 of PA(T_{Core}) in antagonizing the class of correspondence constraints with one of the faithfulness constraints. Each belongs to a distinct subsystem, named by the type of agreement it determines (i.e. the [voi] subsystem governs [voi] harmony). All grammars have values of both of these, aligning with the faithful/unfaithful extensional distinction, feature-specifically. P2 replicates P2 of PA(T_{Core}) exactly. This property is shared across both subsystems. If the faithful value is chosen on either

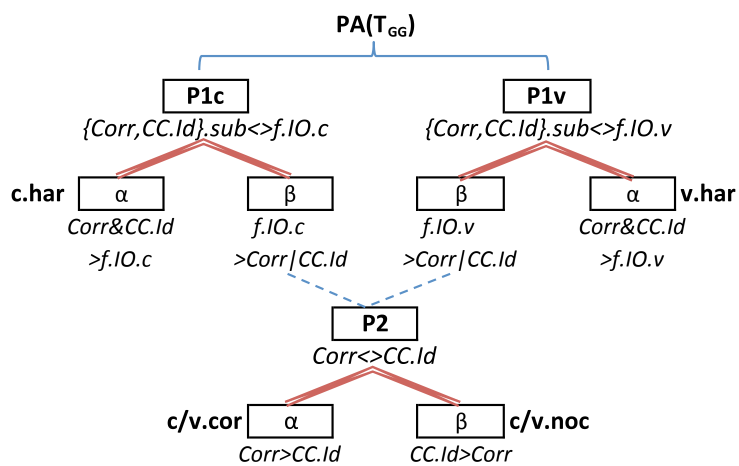
P1c or P1v, then the ranking between Corr and CC.Id is required to determine correspondence behavior of the faithful forms. P2's scope is the *union* of P1c.β and P1v.β, which jointly include six of the seven grammars in the typology - all except har.har, the grammar defined by choosing the unfaithful value on both P1c and P1v.

(20) PA(T_{GG})

Subsystem	*Property*	*Scope*
[cont]	P1c.$_{GG}$: {Corr, CC.Id}.sub <> f.IO.c	wide
[voi]	P1v.$_{GG}$: {Corr, CC.Id}.sub <> f.IO.v	wide
[cont] & [voi]	P2$_{GG}$: Corr <> CC.Id	P1c$_{GG}$.β ∪ P1v$_{GG}$.β

The difference between this system and T_{Core} is strikingly highlighted by the treeoid, shown in (21). The dotted lines in the treeoid indicate disjunctive scope. The nested structure of T_{Core} is replicated over two pairs of properties, {P1c, P2} and {P1v, P2}, P2 being the narrow scope member of both pairs.

(21) Annotated treeoid



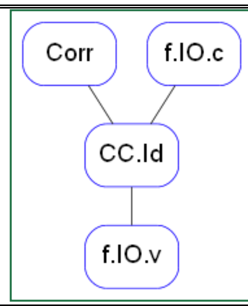
The PA of T_{GG} alters that of T_{Core}, by splitting the top level of the treeoid. The consequence is that the choice between faithful mappings and harmony is made separately for each feature. If f.IO.c is dominated by both Corr and CC.Id, it does not entail that f.IO.v is too. By contrast, P2 remains unsplit, in the scope of both P1v and P1c. This is why the typology has no grammars that combine both cor and noc mappings. The extensional choice of noc or cor depends on the ranking between Corr and CC.Id. Only a single ranking between them is possible in any one grammar. Consequently, the choice of correspondence cuts across both featural subsystems. Grammars with the value P2.α (Corr ≫ CC.Id) have correspondence in all faithful optima; those with P2.β (CC.Id ≫ Corr) have systematic non-correspondence. The value table (22) shows the possible value combinations that delineate the grammars. Only the grammar for language har.har falls outside the scope of P2; in this grammar, both Corr and CC.Id dominate both f.IO constraints, and all non-agreeing inputs harmonize.

(22) PA(T_{GG}) value table

	P1c	P1v	P2
har.har	α	α	
har.cor	α	β	α
har.noc	α	β	β
cor.har	β	α	α
noc.har	β	α	β
cor.cor	β	β	α
noc.noc	β	β	β

As an example of how the properties define the grammars of languages in the typology, consider language cor.har, defined by the value set (β.α.α) (ordered as in the value table). In this language, segments disagreeing in [±cont] are faithfully mapped and correspond: /dz/→$[d_1z_1]$ (similarly for /ts/→$[t_1s_1]$, etc.). This faithful mapping, with correspondence, is the extensional consequence of two rankings: f.IO.c ≫ CC.Id and Corr ≫ CC.Id. The former results from $P1_{GG}$.β; the latter from P2.α. This value combination sets CC.Id as the bot among the set {CC.Id, Corr, f.IO.c} – the constraints in the subsystem related to the disposition of [±cont]. In the same language, segments that disagree in voicing undergo harmony: /td/→$[d_1d_1]$ (also /tz/→$[d_1z_1]$/$[t_1s_1]$, etc.). These mappings are the extensional force of P1v.α. This value sets f.IO.v as the bottom constraint among {Corr, CC.Id, f.IO.v}, the constraints of the subsystem related to the disposition of [±voi]. These are configured for harmony rather than faithfulness. The combined property values and rankings are shown below in (23).

(23) T_{GG}: language cor.har (β.α.α)

Property values	*Hasse*
P1c.β: f.IO.c ≫ Corr | **CC.Id** `LeWe	eLWe` P2.α: Corr ≫ CC.Id `WLee` P1v.α: Corr & CC.Id ≫ f.IO.v `WeeL,eWeL`	

The ranking structure of the cor language in T_{Core}, seen in (8) above, is clearly recognizable in the top portion of the Hasse diagram: f.IO.c and Corr both dominate CC.Id. The bottom portion has the same structure as har in T_{Core}: f.IO.v is dominated by both Corr and CC.Id (though the familiar V-shape is distorted in the graph due to the ranking Corr ≫ CC.Id).

The typology of the G.G system is organized around the very same central set of choices—faithfulness and correspondence—as T_{Core}. The only additional complexity introduced by splitting f.IO into f.IO.v and f.IO.c is that faithfulness can be the bot constraint with respect to one feature, while also *not* being the bot with respect to another (because different kinds of faithfulness are different constraints). As such, this theory differs from the predictions of Core: languages can have harmony for some segments, but not all. It splits the simple bot interaction of

T_{Core} into two subsystems, one governing each feature. Within each, the extensional behavior — choice between har, cor, or noc — is determined exactly as in PA(T_{Core}).

4.2. G.S: unified CORR, feature-specific CC.ID.F

Like G.G and Core, G.S has a single CORR constraint; unlike the other two, it has two feature-specific CC.ID constraints rather than one general one. This system realizes a CON of the kind proposed by McCarthy (2010). The typology, T_{GS}, has 9 languages, two more than T_{GG} (24). As in T_{GG}, languages are combinations of the extensional types {har, cor, noc} on a feature-specific basis. Unlike T_{GG}, this typology permits all logical combinations of all three mapping types for each feature, generating the full 3x3 grid (25). The growth of the typology does not come with wider empirical coverage, however. The two additional languages possible in T_{GS} are cor.noc and noc.cor; these are identical to cor.cor and noc.noc in terms of overt forms, and differ only in the correspondence indices of faithful segments. Thus, the G.S system does not predict any new combinations of segmental mappings, only freer combinability of indices.

(24) Languages of T_{GS}

Inputs / Lgs: c.v	/d z/	/t d/
har.har	d_1 d_1	t_1 t_1
har.cor	d_1 d_1	t_1 d_1
har.noc	d_1 d_1	t_1 d_2
cor.har	d_1 z_1	t_1 t_1
cor.cor	d_1 z_1	t_1 d_1
cor.noc	d_1 z_1	t_1 d_2
noc.har	d_1 z_2	t_1 t_1
noc.cor	d_1 z_2	t_1 d_1
noc.noc	d_1 z_2	t_1 d_2

(25) T_{GS}: combinations of mapping types

cont / voi	har	cor	noc
har	X	X	X
cor	X	X	X
noc	X	X	X

As in T_{GG}, no languages have dissimilation. Changing an input feature to be *less* similar has no effect on CORR violations; the general CORR constraint requires correspondence irrespective of featural makeup of segments. Dissimilation can only make a candidate worse: it introduces more faithfulness violations relative to faithful alternatives, or adds new violations of CC.ID by introducing new disagreement. Consequently, dissimilating candidates are always harmonically bounded (Samek-Lodovici & Prince 2005) by faithful candidates. This is illustrated in (26) for input /dz/, for a non-corresponding dis candidate: it has the same violations as the noc candidate, plus a violation of f.IO.v (violation-equivalent candidates omitted).

(26) Harmonic Bounding of dissimilation

Input	Intended winner	Loser	CORR	CC.ID	f.IO.c	f.IO.v
/dz/	dis: d_1s_2	har: d_1d_1	L		W	L
		cor: d_1z_1	L	W		L
		noc: d_1z_2				**L**

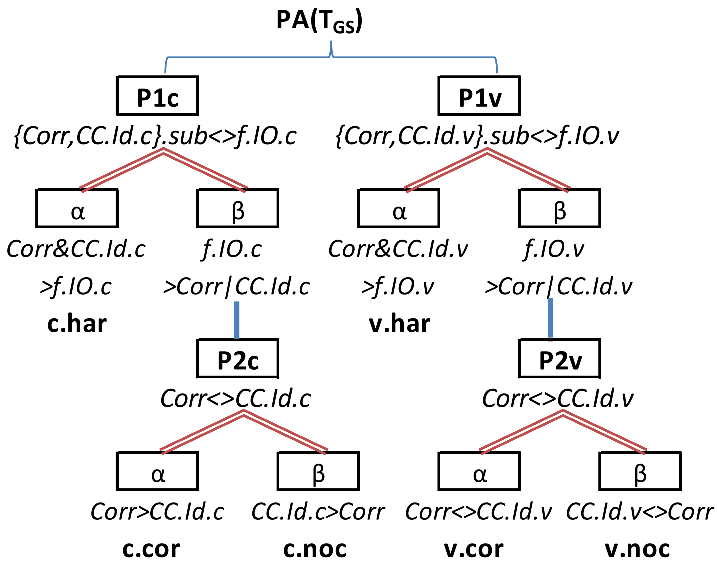
Intensionally, the structures familiar from T_{Core} and T_{GG} recur in T_{GS}. All grammars are defined by two bot-ranking structures, one for each feature. As in T_{GG}, the choice of faithfulness for one feature is separated from faithfulness for the other; the two subsystems involve different faithfulness constraints. Unlike T_{GG}, the subsystems in PA(T_{GS}) also involve different CC.ID constraints. This de-couples the ranking of CORR and CC.ID across the subsystems. One featural subsystem can have a CORR ≫ CC.ID.F structure, while the other has CC.ID.G ≫ CORR, since these ranking involve *different* CC.ID constraints. The result is the total free combination of types, rather than the dependency between cor and noc found in T_{GG}.

PA(T_{GS}) is given in (27). Its treeoid (28) is exactly two copies of that of PA(T_{Core}). P1c and P1v closely resemble the P1s in T_{GG}, but differ in that the class of constraints opposing faithfulness in each property puts CORR together with a different CC.ID constraint: CC.ID.c in the first and CC.ID.v in the second. P2 of T_{GG} is replaced by two properties, P2c and P2v, a split analogous to the split of P1 made in T_{GG}. Values of these two properties correlate with the extensional choice of correspondence in faithful forms; they are narrow scope choices that depend on the value of the two P1s. Unlike T_{GG}, a choice about correspondence made for one featural subsystem does not dictate the choice in the other. Grammars can choose faithful correspondence for /td/ (which disagree in [±voi]), and also non-correspondence for /dz/ (which disagree in [±cont]), and vice versa. The properties group together into subsystems of {P1c, P2c} and {P1v, P2v}, which make the familiar set of grammatical classifications, but independently for each feature.

(27) PA(T_{GS}): Properties

Subsystem	*Property*	*Scope*
[cont]	$P1c_{GS}$: {CORR, CC.ID.c}.sub <> f.IO.c	wide
	$P2c_{GS}$: CORR <> CC.ID.c	$P1c_{GS}.\beta$
[voi]	$P1v_{GS}$: {CORR, CC.ID.v}.sub <> f.IO.v	wide
	$P2v_{GS}$: CORR <> CC.ID.v	$P1v_{GS}.\beta$

(28) Annotated treeoid



The effect of feature-specific CC.ID.F constraints is clear in the progression from T_{Core} to T_{GG} to T_{GS}, which realize increasingly independent choices of three basic mapping types for different kinds of segments. In T_{Core}, the same three constraints determine mappings for both features. Therefore, they must be treated in the exact same way: if there is harmony for one feature, there must also be harmony for the other, etc. Splitting just the faithfulness constraints produces two subsystems of constraints, each internally like T_{Core}, but with partial overlap. This allows different mappings to combine for different subsets of inputs: a grammar can do two different things for the two different features. When CC.ID is also split into two CC.ID.F constraints, there is less overlap between the constraint subsystems, and consequently greater combinability. In T_{GS}, the unified CORR constraint is the only point of contact between the subsystems. This eliminates the potential for ranking contradictions: one constraint does not a ranking make. Therefore, the values of the subsystems permute freely.

4.3. G.R: unified CORR, feature-restricted CC.ID.F/αG

The final system using a unified CORR constraint is G.R. It differs from G.S in adding an additional featural-similarity condition to CC.ID constraints, in the form CC.ID.F/αG. These require agreement in [±F] *only if* segments share another feature specification [αG]. This combination of constraint types, and the idea for incorporating featural restriction into the CC.ID.F/αG constraints, is based on an analysis proposed by Walker (2016).[14]

The typology of the G.R system (T_{GR}) consists of 10 languages. Unlike the progression from T_{GG} to T_{GS}, this typology adds languages with overt form differences, not just differences in

[14] Hansson (2014) uses a similarly formulated agreement constraint but in a system without correspondence; segments are required to agree in F if they share αG. Shih & Inkelas (to appear) also develop constraints similar to Walker's, indexing agreement constraints to specific dimensions of similarity (each encoded by a distinct correspondence relation). These proposals resemble the OCP as formulated in autosegmental phonology (McCarthy 1986, Mester 1986, Yip 1988, etc.): a constraint assessing only those segments that are represented on a featurally-defined tier.

correspondence possibilities. This is because T_{GR} includes a fourth extensional type of mapping: dissimilation (dis).

Dissimilation is possible in T_{GR} because of the feature-restricted CC.Id.F/αG constraints. These are satisfied both by candidates in which corresponding segments agree on [±F], and those in which they *dis*agree on G. This is shown in (29), for input /dz/ undergoing [+voi] dissimilation to [t_1z_1] (or, equivalently, to [d_1s_1]; co-optima are omitted here). The dis candidate [d_1s_1] satisfies Corr: the two segments correspond. It also satisfies CC.Id.c/+v and CC.Id.v/-c, albeit in a less intuitive way. The only [+voi] segment is [d], and [d] does not disagree with itself for [±cont]; the only [-cont] segment is also [d], similarly not an instance of [±voi] disagreement. The dis candidate does have a f.IO.v violation not shared by cor or noc, but these alternatives violate either Corr or a CC.Id constraint. By eliminating the feature value picked out by αG in CC.Id.F/αG, two segments can correspond (satisfying general Corr) without incurring disagreement violations. The choice of harmony or dissimilation among correspondents falls to the faithfulness constraints.

(29) Potential optimality of dis with feature restricted CC.Id constraints

input	winner	loser	Corr	CC.Id.c/+v	CC.Id.v/-c	f.IO.c	f.IO.v
/dz/	dis: d_1s_1	har: d_1d_1				W	L
		cor: d_1z_1		W			L
		noc: d_1z_2	W				L

The 10 languages of T_{GR} are shown in (30). All languages are combinations of the one of the four types of possible mappings {dis, har, cor, noc} for each featural subsystem; these are shown in the grid in (31). Not all logical combinations of mappings are possible grammars: har and dis are incompatible with noc. Additionally, har.har and dis.dis languages are impossible. If a language has harmony for one feature, it can only have dis or cor for the other.

(30) Languages of T_{GR}

Inputs / Lgs: c.v	/d z/	/t d/
har.dis	d_1 d_1	t_1 z_1
har.cor	d_1 d_1	t_1 d_1
cor.har	d_1 z_1	t_1 t_1
cor.dis	d_1 z_1	t_1 z_1
cor.cor	d_1 z_1	t_1 d_1
cor.noc	d_1 z_1	t_1 d_2
noc.cor	d_1 z_2	t_1 d_1
noc.noc	d_1 z_2	t_1 d_2
dis.har	d_1 s_1	t_1 t_1
dis.cor	d_1 s_1	t_1 d_1

(31) T_{GR} mapping combinations

voi \ cont	har	cor	noc	dis
har		X		X
cor	X	X	X	X
noc		X	X	
dis	X	X		

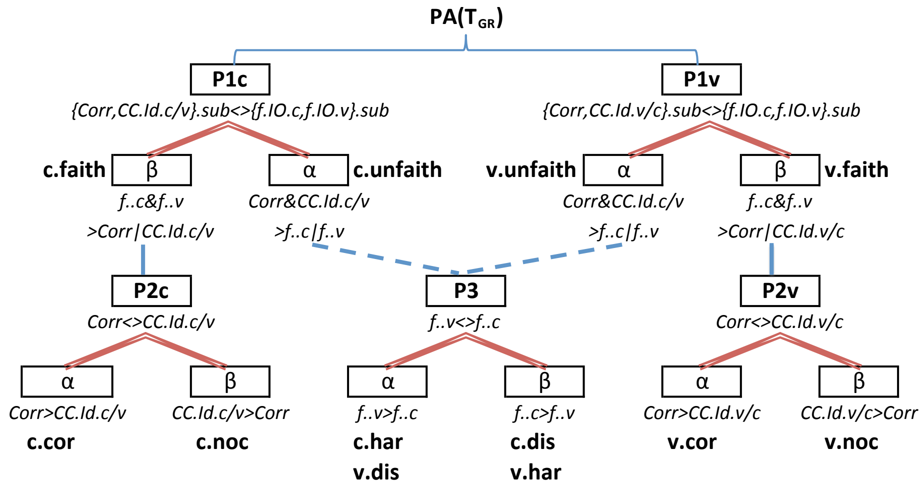
The gaps in the typology arise from interactions between overlapping subsystems of constraints. As in previous systems, the basic mapping types align with permutations of a bot ranking structure. But, crucially, in T_{GR} these bot structures occur over groups of *four* constraints not three. *Both* f.IO constraints occur in *both* subsystems. In PA(T_{GS}), the P1s (whose values determine faithfulness of mappings) antagonize a class of correspondence constraints against a single f.IO. But in PA(T_{GR}), the P1s are antagonized with another class, of both f.IO constraints. An unfaithful mapping, har or dis, is optimal when *either* f.IO constraint is dominated, the difference between them being determined by the relative ranking of the faithfulness constraints. This choice in encoded in the new property, P3. Since both f.IOs occur in both subsystems, this property has disjunctive scope under both P2v and P2c (32).

(32) PA(T_{GR})

Subsystem	*Property*	*Scope*
[cont]	P1c_{GR}: {CORR, CC.ID.c/+v}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P2c_{GR}: CORR <> CC.ID.c/+v	P1c_{GR}.β
[voi]	P1v_{GR}: {CORR, CC.ID.v/-c}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P2v_{GR}: CORR <> CC.ID.v/-c	P1v_{GR}.β
both	P3$_{GR}$: f.IO.c <> f.IO.v	P1c_{GR}.α ∪ P1v_{GR}.α

The typology is organized by the same core set of choices seen in previous systems, but adds one additional dimension of variation. The faithful/unfaithful distinction remains, as does the choice of correspondence for faithful. But, unlike in PA(T_{GS}), the unfaithful class of languages is also further distinguished, by choice on P3: harmony or dissimilation. The treeoid (33) shows the resulting structure: either choice of value on P1 necessitates a further choice on P2 or P3.

(33) Annotated treeoid



Intensionally, the distinction between dis and har is located in the ranking between f.IO.c and f.IO.v – the values of P3. Since this property is involved in both subsystems, either ranking of them (i.e. either value of P3) restricts the permutations available in both. In order to have [±voi] harmony, f.IO.v needs to be the bot constraint among {Corr, CC.Id.v/-c, f.IO.v, f.IO.c}. This entails value P3.α: f.IO.c ≫ f.IO.v. If that ranking holds in a grammar, it follows that f.IO.c *cannot* be the bot constraint among {Corr, CC.Id.c/+v, f.IO.v, f.IO.c}, rendering [±cont] harmony impossible. The same mechanism also explains why dis.dis is missing from the typology. Choosing dis as the way to resolve [±voi] disagreement is the same as choosing the value P3.β, f.IO.v ≫ f.IO.c: this choice means the subsystem that handles [±cont] disagreement cannot have f.IO.v as its bot.

The restricted distribution of noc is similarly explained by restrictions stemming from overlap between the subsystems. This arises not from a shared P, but from shared antagonists in multiple Ps. A noc mapping is optimal if Corr is the bot constraint in a subsystem. Such rankings in one subsystem entails that both f.IO constraints dominate Corr. This is inconsistent with a ranking in which Corr dominates either f.IO — so the bot constraint of the other subsystem *cannot* be a faithfulness constraint. Thus configuring one subsystem of constraints to produce noc precludes the other subsystem from meeting the ranking requirements for an unfaithful mapping. A β value on either P2, giving noc, entails also choosing β for both P1s.

The feature restriction on CC.Id changes the T_{Core} structure most significantly of the variations considered so far, but the fundamental structure is still recognizably related. All grammars of T_{GR} are determined by bot structures involving faithfulness, Corr, and CC.Id. The difference arises from the increase in the size of each subset from three constraints to four, to include both faithfulness constraints.

4.4. R.G: feature-restricted Corr.F, unified CC.Id

Departing from all previous systems, R.G has a feature-restricted Corr and a unified CC.Id. This combination instantiates the system proposed by Gallagher & Coon (2009). The typology, T_{RG}, has 10 languages – the same number as T_{GR} discussed above. Also like T_{GR}, all the languages draw from four mapping types, notably including dissimilation, though it arises in a slightly different way in T_{RG}. Shedding a shared feature allows two consonants to satisfy CORR.F without corresponding, rather than allowing them to correspond without violating CC.ID.F/αG. Thus dissimilation here entails *non*-correspondence.

(34) Potential optimality of dis with feature restricted CORR constraints

input	winner	loser	CORR.+v	CC.ID.c	CORR.-c	CC.ID.v	f.IO.c	f.IO.v
/dz/	dis: t_1z_2	har: d_1d_1					W	L
		cor: d_1z_1		W				L
		noc: d_1z_2	W					L

As a consequence of dissimilation coupling with a different correspondence structure, different correspondence structures are compatible with different mapping types. The languages of T_{RG} (35) are *exactly* the same segmental mapping possibilities as in T_{GR} (30), but differ in that in T_{RG}, unfaithful types har and dis, are compatible with *noc,* rather than cor. This is systematic: where T_{GR} has noc, T_{RG} has cor, and vice versa, clear from the grids of both typologies (36). Interestingly, even though the typologies are not the same, it is in principle impossible to tease them apart by looking at data, because correspondence structure is hidden, not apparent in overt forms.

(35) Languages of T_{RG}

Lgs: c.v \ Inputs	/d z/	/t d/
har.dis	d_1 d_1	t_1 z_2
har.cor	d_1 d_1	t_1 d_2
cor.cor	d_1 z_1	t_1 d_1
cor.noc	d_1 z_1	t_1 d_2
noc.har	d_1 z_2	t_1 t_1
noc.dis	d_1 z_2	t_1 z_2
noc.cor	d_1 z_2	t_1 d_1
noc.noc	d_1 z_2	t_1 d_2
dis.har	d_1 s_2	t_1 t_1
dis.cor	d_1 s_2	t_1 d_2

(36) Combinations in T_{RG} and T_{GR}

a. T_{RG}

voi \ cont	har	cor	noc	dis
har			**X**	X
cor		X	X	
noc	**X**	X	X	**X**
dis	X		**X**	

b. T_{GR} (from (30) above)

voi \ cont	har	cor	noc	dis
har		**X**		X
cor	**X**	X	X	**X**
noc		X	X	
dis	X	**X**		

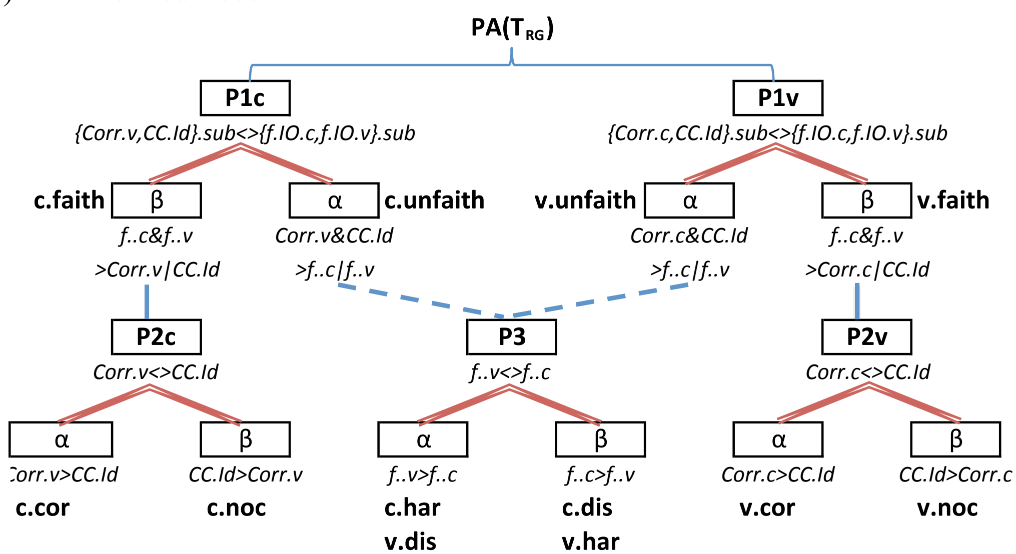
The extensional similarity between the typologies follows from a shared intensional structure. The same types of properties that define PA(T_{GR}) also define the grammars of T_{RG}, only switching the feature restriction from CC.ID.F/αG to CORR.αF. Both systems are the combination of two constraint subsystems, one governing each feature, and involving the interaction of four constraints. All properties of are given in (37).

(37) PA(T$_{RG}$)

Subsystem	*Property*	*Scope*
[cont]	P1c$_{RG}$: {CORR.+v, CC.ID}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P2c$_{RG}$: CORR.+v <> CC.ID	P1c$_{RG}$.β
[voi]	P1v$_{RG}$: {CORR.-c, CC.ID}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P2v$_{RG}$: CORR(-c) <> CC.ID	P1v$_{RG}$.β
both	P3$_{RG}$: f.IO.c <> f.IO.v	P2c$_{RG}$.α ∪ P2v$_{RG}$.α

An even more perspicuous view of the shared structure comes from scrutinizing the treeoid: the structures are completely isomorphic. The PAs classify the typologies along exactly the same set of choices: (P1) faithfulness?; (P2) if yes, correspondence?; (P3) if no, harmony or dissimilation?

(38) Annotated treeoid



The results of featurally-restricting either constraint type—holding other aspects of CON constant—are the same. They generate equivalent typologies. It is not crucial which constraint has the restriction, as long as it is present in the system. As long as there is some featurally-restricted ABC constraint, dissimilation is a possible mapping type, and the subsystems involve both f.IO constraints rather than just one.

4.5. R.S: feature-restricted CORR.F and feature-specific CC.ID.F

The final variation we analyze in detail has feature-restricted CORR constraints (like R.G), and feature-specified CC.ID constraints (like G.S). The combination of distinct families for both CORR and CC.ID constraints is common in many ABC(D) analyses (i.e., Rose & Walker 2004, Hansson 2001/2010, Bennett 2015). The 14 extensional languages (39) draw from the same four mapping types as T$_{GR}$ and T$_{RG}$. The locus of the featural restriction is CORR.F, so dissimilation entails non-correspondence, as in T$_{RG}$. The typology expands because unfaithful mappings are compatible with *both* cor and noc. However, these do not produce new combinations in terms of *overt* structure; the set of predicted languages in terms of non-hidden structure is the same.

(39) Languages of T_{RS}

Lgs: c.v \ Inputs	/d z/	/t d/
har.dis	d_1 d_1	t_1 z_2
har.cor	d_1 d_1	t_1 d_1
har.noc	d_1 d_1	t_1 d_2
cor.har	d_1 z_1	t_1 t_1
cor.dis	d_1 z_1	t_1 z_2
cor.cor	d_1 z_1	t_1 d_1
cor.noc	d_1 z_1	t_1 d_2
noc.har	d_1 z_2	t_1 t_1
noc.dis	d_1 z_2	t_1 z_2
noc.cor	d_1 z_2	t_1 d_1
noc.noc	d_1 z_2	t_1 d_2
dis.har	d_1 s_2	t_1 t_1
dis.cor	d_1 s_2	t_1 d_1
dis.noc	d_1 s_2	t_1 d_2

(40) Combinations of T_{RS}

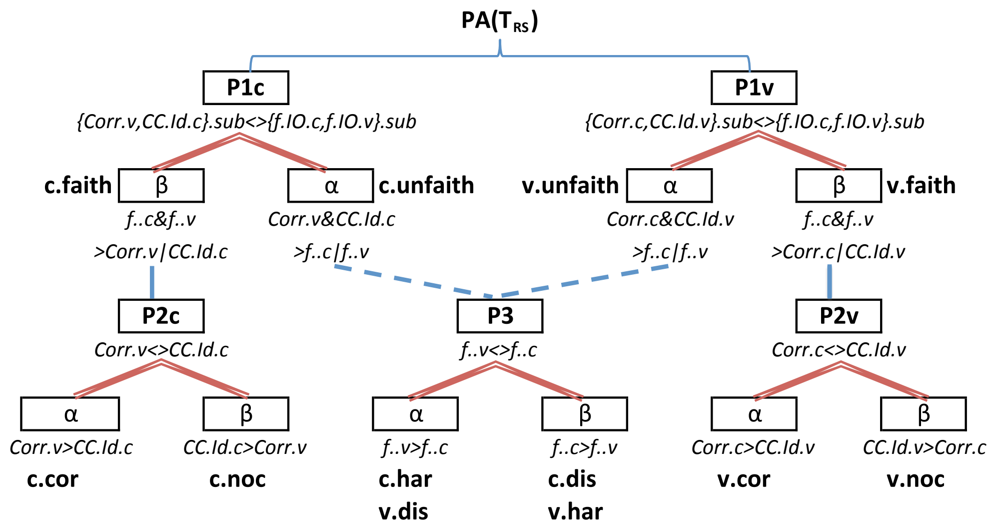
voi \ cont	har	cor	noc	dis
har		X	**X**	X
cor	X	X	X	X
noc	**X**	X	X	**X**
dis	X	X	**X**	

The similarity to both T_{RG} and T_{GR} results from a nearly identical intensional structure. As with the prior property analyses, PA(T_{RS}) consists of five properties that make the same familiar choices, organized into two subsystems (two P1s, and two P2s – one for each feature). The difference in the size of the typology arises from the degree of overlap between the antagonist sets of the two constraint subsystems. In T_{GR} and T_{RG}, feature-general unified constraints CORR or CC.ID interact in both subsystems, so ranking conditions in each restrict those in the other. In T_{RS}, each CORR and CC.ID constraint is in one subsystem. Thus one can have a CORR constraint as its bot, while the other has a CC.ID constraint as its bot; there is no ranking contradiction, since different CORR and CC.ID constraints are involved in each ranking. The only shared constraints are f.IO constraints. As before, their ranking (=P3) is the choice between har versus dis – a choice only made in the case of an unfaithful mapping (=P1c.α or P1v.α). Thus, the isomorphism between the treeoids of PA(T_{GR}) and PA(T_{RG}) also obtains here (42).

(41) PA(T$_{RS}$)

Subsystem	*Property*	*Scope*
[cont]	P1c$_{RS}$: {CORR.+v, CC.ID.c}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P1v$_{RS}$: CORR.+v <> CC.ID.c	P1c$_{RS}$.β
[voi]	P2c$_{RS}$: {CORR.-c, CC.ID.v}.sub <> {f.IO.c, f.IO.v}.sub	wide
	P2v$_{RS}$: CORR.-c <> CC.ID.v	P1v$_{RS}$.β
both	P3$_{RS}$: f.IO.c <> f.IO.v	P1c$_{RS}$.α ∪ P1v$_{RS}$.α

(42) Annotated treeoid



The final variation defined at the start of this section is R.R, where both CORR and CC.ID have feature restrictions. We do not include detailed analysis here. The typology is the same size as T$_{RS}$, but the languages have additional co-optima, due to a difference in how dissimilation arises. Dissimilation entails non-correspondence in the R.G and R.S systems, and correspondence in the R.G system. In the R.R system, it entails neither; both kinds of dissimilating candidates are co-optimal. Doubling the restriction does not change the set of possible languages. Building featural restrictions into both CORR and CC.ID constraints merely reinforces (redundantly) the same division between the constraint subsystems made in PA(T$_{RS}$). The f.IO constraints are still shared, and so har.har and dis.dis languages remain impossible.

5. Comparison

In this section, we bring together the results of the previous sections to compare the systems in terms of both their empirical predictions—the languages—and their intensional structures—the grammars. This allows us to discern the effects of certain CON changes.

All systems generate languages defined by (a subset of) the four extensional types, cor, noc, har and dis. The languages of all typologies are shown in (43). Three main distinctions exist in the systems' predictions. First, a choice of mapping type may be applied globally for all inputs, or featurally, to segments sharing a particular feature(s). This separates Core from all other systems. Second, some systems predict languages with dissimilation, separating {Core, G.G, G.S} from their complement set, {G.R, R.G, R.S, R.R}. Finally, the possible combinations of mapping types for each feature are subject to different degrees of limitation. In R.G and G.R, for example, unfaithful types, har and dis, are compatible with only one of the faithful types, cor *or* noc; in R.S & R.R, they are compatible with both kinds of faithfulness.

(43) Extensional languages in each system

cont	voi	Core	G.G	G.S	G.R	R.G	R.S	R.R
har	har	X	X	X				
	cor		X	X	X		X	X
	noc		X	X		X	X	X
	dis				X	X	X	X
cor	har		X	X	X		X	X
	cor	X	X	X	X	X	X	X
	noc			X	X	X	X	X
	dis				X		X	X
noc	har		X	X		X	X	X
	cor			X	X	X	X	X
	noc	X	X	X	X	X	X	X
	dis					X	X	X
dis	har				X	X	X	X
	cor				X		X	X
	noc					X	X	X
	dis							
Total		3	7	9	10	10	14	14

The same kinds of core intensional rankings derive all the mapping types across all systems. All grammars are defined by bot ranking structures over subsets of CON involving one CORR constraint, one CC.ID constraint, and one or both f.IO constraints. By examining the similarities and differences in the PAs of the systems, we find the intensional correlates aligning with the differences in extensional predictions, summarized in (44).

(44) PA features

T	*Subsystems*		
	#	*Overlap in constraint sets*	*# of Cs in each*
T_{Core}	1	--	3
T_{GG}	2	2 (CORR, CC.ID)	3
T_{GS}	2	1 (CORR)	3
T_{GR}	2	3 (CORR, f.IO.c, f.IO.v)	4
T_{RG}	2	3 (CC.ID, f.IO.c, f.IO.v)	4
T_{RS}	2	2 (f.IO.c, f.IO.v)	4
T_{RR}	2	2 (f.IO.c, f.IO.v)	4

T_{Core} differs from the others in having a single subsystem, consisting of the entirety of CON_{Core}.[15] This yokes together the extensional behavior of segments of different feature classes in that the

[15] Alternatively, T_{Core} consists of two subsystems that overlap in all constraints. The effect of yoking features together is analogous to the 'Basic Syllable Theory' in Prince & Smolensky (1993/2004, ch. 6),

same, single ranking decides the optima for all inputs. In every other system, at the least the f.IO constraints are split into two feature-specific counterparts, which can be ranked in distinct ways relative to the correspondence-sensitive constraints; this breaks the dependency up. In any S/R system, one or more correspondence constraint is also featurally defined, furthering the distinctness.

The overlap between the constraint subsets governing each subsystem dictates which combinations of types for each feature system are possible. The extent of the overlap is of course related in part to the size of CON, something brought out clearly by comparing the G.G system to G.S. The subsystems of G.G both involve the same two CORR and CC.ID constraints, which permits only one ranking between them; on the other hand, the subsystems of G.S. involve two distinct members of a CC.ID constraint family, permitting two independent pairwise choices about the ranking of the CC.ID and CORR that pertain to the two different features. This allows for four possible ranking relations among the correspondence constraints: CORR can dominate both CC.ID constraints, be subordinate to both, or be ranked above one and below the other, in either order. The difference comes out in the treeoids via the split of a single P2—$P2_{GG}$, shared by subsystems—into separate P2s: $P2c_{GS}$ & $P2v_{GS}$, whose values freely combine to produce four options. The same relationship exists between G.R & R.G versus R.S & R.R, though it is less obvious in the treeoids. It is due not to shared Ps, but to shared antagonists in different Ps. In the former two systems, a general CORR or CC.ID constraint occurs in both subsystems, while in the latter set, separate subsystems involve separate constraints for both types.

Finally, whether a typology generates dissimilation depends on how many constraints interact in the subsystems. Crucially, to get both faithfulness constraints in a four-constraint ranking structure, one of the correspondence-sensitive constraints, CC.ID or CORR, must be featurally restricted. It does not depend *which* is so defined, as long as one within the entire system is. The treeoidal isomorphism reflects the deep parallels between systems. This result underscores our central claim that understanding the full effects of CON proposals requires analysis of their intensional typologies. The question of whether a unified CORR or CC.ID is sufficient to capture a certain set of languages or patterns is cannot be answered without knowing the full set of constraints with which it interacts.

The insights accrued through our property analysis allow us to fully understand the basic interactions giving rise to ABC(D) typologies and how these change upon changes to CON. This extends beyond the simple two-segment, two-feature universe of this paper. A third feature merely creates additional subsystems, adding a third f.IO constraint for the added feature. The same intensional structures persist in all new subsystems, though: they can offer maximally the same three or four permutations seen in our analyses above (depending on how the CORR and CC.ID constraints are defined). The effect of constraint subset overlap likewise generalizes: the permutability of any additional subsystem(s) is restricted based on the ranking of constraints shared by the other subsystems.

Similarly, scaling up from forms with two segments to three or more does not fundamentally change how the constraints interact. If the constraints assess all segments as they do in the

---

where unified PARSE and FILL constraints couple together the repair strategies for ONSET and NOCODA violations.

systems analyzed above, then the typologies computed from three-consonant strings are not observably different from those analyzed above (a result confirmed through analysis of such systems in OTWorkplace[16]).

Such modifications to GEN only change the typology if accompanied by other theoretical changes, particularly other modifications of CON. But even these do not alter the essential subsystems organization—they just create more subsystems. For example, distinct CORR constraints can be defined according to proximity, following the idea that correspondence is sensitive to distance as well as featural similarity (see Hansson 2010:232 and citations there; also Suzuki 1998, Bennett 2015). The now-understood consequence of splitting CORR is that distinct CORR constraints can be ranked differently relative to f.IO and CC.ID constraints in separate subsystems—the same difference seen between G.S and R.S. Two proximity-based CORRs yield two overlapping subsystems: one for each CORR, interacting with the same subset of CC.ID and f.IO constraints. Similar reasoning applies to splitting CC.ID, to other non-featural divisions of the essential ABC constraint types, and likely to different variations on the range of correspondence possibilities offered by GEN.

6. Concluding remarks

Our aim in this paper has been to determine the typological consequences of different formulations of ABC(D) constraints. By defining and analyzing a set of systems realizing variations of these, we showed how they interact to give rise to particular types of extensional mappings, and combinations thereof.

The systems laid out above deviate from the original ABC(D) formulation in different—and sometimes opposite—ways, in unifying or restricting different constraint types. G.R and R.G are mirror images in this regard: one unifies CORR and has feature-restricted CC.ID while the other does the reverse. One might naïvely expect such opposite changes to yield quite different typologies. Instead, their extensional predictions are *the same*. Intensional analysis reveals why: it is not the formulation of individual *constraints* alone, but whether a *system* of constraints together creates the conditions under which dissimilation is possible. ABC theories without feature-restricted constraints do not generate dissimilation; those that have *either* feature-restricted CORR or CC.ID constraints produce dissimilation. This level of typological consequence cannot be observed on the basis of data from any one language or database of languages, but only from engaging with the theory on its own terms.

In closing, we note that the findings of this paper do not establish the superiority of any one ABC theory over the others; the sufficiency of any theory depends on the intended target of analysis. If our target is a theory of harmony and lack thereof, then unifying both CORR and CC.ID suffices, as in the system G.G. If the goal is to derive dissimilation from the same theory, then G.G is clearly inadequate; however, alternatives that do produce dissimilation—G.R, R.G, and R.S—are

[16]Forms with 3 segments invite a further choice about how to assess violations of CC.ID constraints: counted globally, or in local pairwise fashion (as proposed by Hansson 2007). Walker (2016) assumes the latter. Our findings show that this is not crucial: if CC.ID.F/αG constraints assess agreement as in §4, then the typology is the same regardless of whether violations are counted over all correspondents, or only local pairs. Local-counting reduces violation counts for some candidates, but in ways that do not affect the set of possible optima and factorial typology.

all plausible choices. There is no empirical justification for preferring any one of these: their typologies differ extensionally only in terms of non-observable correspondence relationships. Moreover, we have shown that their intensional structures are highly similar, yielding typologies with the same types of internal divisions.

References

Alber, Birgit and Alan Prince. 2015. Outline of Property Theory. Ms, University of Verona, Italy & Rutgers University, New Brunswick, NJ.

Alber, Birgit and Alan Prince. In prep. *Typologies*. Ms, University of Verona, Italy & Rutgers University, New Brunswick, NJ.

Alber, Birgit and Alan Prince. 2017. The book of nGX. Ms, University of Verona, Italy & Rutgers University, New Brunswick, NJ. ROA 1312.

Alber, Birgit, Natalie DelBusso, and Alan Prince. 2016. From intensional properties to universal support. *Language: Phonological Analysis* 92(2): e88-116. ROA 1235.

Bennett, Wm. G. 2015. *The phonology of consonants: harmony, dissimilation and correspondence*. Cambridge University Press. [Adapted from 2013 PhD thesis, Rutgers University. ROA 1173.]

Bennett, William G., Natalie DelBusso, and Luca Iacoponi. 2016. Formally mapping the typologies of interacting ABCD systems. *Supplemental Proceedings of the Annual Meeting of Phonology 2015 (AMP 2015)*, Washington DC: LSA. [ROA 1282.]

DelBusso, N. & Prince, A. In prep. Weak Order Typologies. Ms, Rutgers University, New Brunswick, NJ.

Gallagher, Gillian. 2010. *The perceptual basis of long-distance laryngeal restrictions*. PhD thesis, MIT.

Gallagher, Gillian and Jessica Coon. 2009. Distinguishing total and partial identity: Evidence from Chol. *Natural Language & Linguistic Theory* 27:545-582.

Hansson, Gunnar Ólafur. 2001. *Theoretical and typological issues in consonant harmony*. PhD thesis, University of California, Berkeley.

Hansson, Gunnar Ólafur. 2004. Tone and voicing agreement in Yabem. In B. Schmeiser, V. Chand, A. K. and Rodriguez, A. (eds.), *Proceedings of WCCFL 23*, 318–331, Somerville, MA. Cascadilla Press.

Hansson, Gunnar Ólafur. 2007. Blocking effects in agreement by correspondence. *Linguistic Inquiry*, 38.2:395–409.

Hansson, Gunnar Ólafur. 2010. *Consonant Harmony: Long-Distance Interaction in Phonology*. Berkeley, CA: University of California Press. Revised version of Hansson (2001).

Hansson, Gunnar Ólafur. 2014. (dis)agreement by (non)correspondence: Inspecting the foundations. Presented at the ABC↔C conference, UC Berkeley, 2014. In UC Berkeley Phonology Lab Annual Report, pp. 3-62. [http://linguistics.berkeley.edu/phonlab/documents/2014/ABCC/Hansson.pdf]

Hansson, Gunnar Ólafur and Allie Entwistle, A. 2013 Consonant stricture harmony in Yabem: manner assimilation at a distance. In S. Luo (ed.) *Actes du Congrès de l'ACL 2013 / 2013 CLA Conference Proceedings*. Association canadienne de linguistique/ Canadian Linguistic Association. [http://homes.chass.utoronto.ca/~cla-acl/actes2013].

Inkelas, Sharon and Stephanie S. Shih. 2014. Unstable surface correspondence as the source of local conspiracies. In J. Iyer & L. Kusmer (eds.) *Proceedings of NELS 44*, pp. 191-204.

Jurgec, Peter. 2011. *Feature Spreading 2.0: A Unified Theory of Assimilation*. PhD thesis, University of Tromsø.

Jurgec, Peter. 2014. Consonant harmony as feature spreading. Presented at the ABC↔C conference, UC Berkeley, May 19, 2014. In UC Berkeley Phonology Lab Annual Report, pp. 295-401. [http://linguistics.berkeley.edu/phonlab/documents/2014/ABCC/Jurgec.pdf]

Kimper, Wendell. 2011. *Competing Triggers: Transparency and opacity in vowel harmony*. PhD thesis, University of Massachusetts Amherst.

Krämer, Martin. 1998. *A correspondence Approach to Vowel Harmony and Disharmony*. SFB 282 Working Paper, number 107. Heinrich-Heine-Universität, Düsseldorf.

McCarthy, John J. 1986. OCP effects: Gemination and antigemination. *Linguistic Inquiry* 17: 207-263.

McCarthy, John J. 2007. Consonant harmony via correspondence: Evidence from Chumash. In L. Bateman, M. O'Keefe, E. Reilly, & A. Werle (eds.), *Papers in Optimality Theory 3*, pp. 223–237. Amherst, MA: GLSA.

McCarthy, John J. 2010. Agreement by correspondence without CORR constraints. Ms. University of Massachusetts, Amherst. ROA 1089.

McManus, Hope. 2016. Stress Parallels in Modern OT. PhD thesis, Rutgers University. ROA 1295.

McMullin, Kevin J. and Gunnar Ó. Hansson. 2015. Long-Distance Phonotactics as Tier-Based Strictly 2-Local Languages. *Proceedings of the Annual Meeting of Phonology 2014 (AMP 2014)*, Washington DC: LSA. [https://journals.linguisticsociety.org/proceedings/index.php/amphonology/article/download/3750/3468]

McMullin, Kevin J. 2016. *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. PhD thesis, University of British Columbia.

Merchant, Nazarré and Alan Prince. 2016. *The Mother of all Tableaux.* Technical Report, Eckerd College, St. Petersburg, FL & Rutgers University, New Brunswick, NJ. ROA 1285.

Mester, Armin. 1986. *Studies in Tier Structure*. PhD thesis, University of Massachusetts, Amherst.

Nevins, Andrew. I. 2004. *Conditions on (Dis)Harmony*. PhD thesis, MIT.

Prince, Alan. 2002a. *Entailed Ranking Arguments*. Technical Report, Rutgers University, New Brunswick, NJ. ROA 500.

Prince, Alan. 2002b. *Arguing Optimality*. Technical Report, Rutgers University, New Brunswick, NJ. ROA 562.

Prince, Alan. 2016a. One tableau suffices. Technical report, Rutgers University. ROA 1250.

Prince, Alan. 2016b. What is OT? Technical Report, Rutgers University, New Brunswick, NJ. ROA 1271.

Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Blackwell. ROA 537.

Prince, Alan, Bruce Tesar and Nazarré Merchant. 2007-2017. OTWorkplace. https://sites.google.com/site/otworkplace/.

Rhodes, Russell. 2010. Vowel harmony as agreement by correspondence. Ms. University of California, Berkeley.

Rose, Sharon Rose Walker. 2004. A typology of consonant agreement as correspondence. *Language* 80: 475-531.

Samek-Lodovici, Vieri and Prince, Alan. 2005. Fundamental properties of harmonic bounding. Technical Report, University College London and Rutgers University. ROA 785.
Shih, Stephanie S. and Sharon Inkelas. 2014. A subsegmental correspondence approach to contour tone (dis)harmony patterns. In J. Kingston, C. Moore-Cantwell, J. Pater, & R. Staubs (eds.). *Proceedings of AMP 2013*. Washington, DC: LSA.
Shih, Stephanie S. and Sharon Inkelas. to appear. Autosegmental aims in surface optimizing phonology. *Linguistic Inquiry*.
Suzuki, Keiichiro. 1998. A Typological Investigation of Dissimilation. PhD thesis, University of Arizona.
Walker, Rachel. 2000a. Long-distance consonantal identity effects. In R. Billerey & B. Lillehaugen (eds.) *Proceedings of WCCFL 19*, pp. 532–545. Somerville, MA: Cascadilla Press.
Walker, Rachel. 2000b. Yaka nasal harmony: Spreading or segmental correspondence? In L.J. Conathan, J. Good, D. Kavitskaya, A.B. Wulf, & A.C. Yu (eds.), *Proceedings of BLS 26,* pp. 321–332. Berkley Linguistics Society.
Walker, Rachel. 2001. Consonantal correspondence. In R. Kirchner, J. Pater, & W. Wikeley (eds.), *Proceedings of the Workshop on the Lexicon in Phonetics and Phonology*: *Papers in Experimental and Theoretical Linguistics*, vol. 6, pp. 73–84. Edmonton: University of Alberta.
Walker, Rachel. 2009. Similarity-sensitive blocking and transparency in Menominee. Paper presented at the Annual Meeting of the Linguistics Society of America (LSA), San Francisco, CA, Jan. 9, 2009.
Walker, Rachel. 2011. *Vowel Patterns in Language*. Cambridge University Press, Cambridge.
Walker, Rachel. 2014. Prominence-control and multiple triggers in vowel harmony: An ABC analysis. Presented at the ABC↔C conference, UC Berkeley, May 19, 2014. In UC Berkeley Phonology Lab Annual Report, pp. 202-213. [http://linguistics.berkeley.edu/phonlab/documents/2014/ABCC/Walker.pdf]
Walker, Rachel. 2016. Surface correspondence and discrete harmony triggers. In A. Albright & M. Fullwood (eds.), *Proceedings of the Annual Meeting of Phonology 2014 (AMP 2014)*. Washington, DC: LSA.
Wayment, Adam. 2009. *Assimilation as Attraction: Computing Distance, Similarity, and Locality in Phonology*. PhD thesis, Johns Hopkins University, Baltimore, MD.
Yip, Moira. 1988. The obligatory contour principle and phonological rules: A loss of identity. *Linguistic Inquiry*, 19:65–100.