The intensional and the extensional properties of Headed Agreement By Correspondence

Luca Iacoponi
Rutgers University

The goal of the paper is to analyze the relation between the *data* (the set of all possible input-output mapping generated by an OT system) and the *theory* (the ranking conditions that generate each grammar) for an *OT system* <GEN$_{HABC}$, CON$_{HABC}$> (abbreviated S$_{HABC}$) that includes crucial constraints and candidates from Headed Agreement By Correspondence (Iacoponi, 2015) using property theory (Alber and Prince, in prep., a.o.).

Such an analysis is important for three reasons. First, since the analysis concerns the structure of a basic HABC typology, it significantly facilitates the study of the differences between HABC and the theories which it extends, namely ABC, and ABCD (Hansson 2001/2010; Rose & Walker 2004; Bennett 2013). Second, as shown in Alber and Prince (in prep.) and in Bennett, DelBusso and Iacoponi (2016), the study of the properties of a system is useful when the system is extended to include more constraints or candidates. Finding the basic structure of a typology not only deepens our understanding of the theory, but it also significantly facilitates the analysis of its extended sub-systems, allowing us to rigorously study the effect the different components (such as classes of candidates or specific constraints) have on the theory. Finally, by using the property analysis of a typology, it is possible to validate the universality of the support used to obtain the typology (see Alber, DelBusso and Prince, 2015).

The paper is organized as follows. In section 1, I introduce the two core theories the paper builds on: Headed Agreement By Correspondence (HABC) and the formal properties of OT typologies. Section 2 contains the definitions of the constraints and of the candidate set. Section 3 discusses the typology, and the relation between its intensional and extensional properties.

1 Theoretical background

1.1 *HABC* Agreement By Correspondence (Walker 2000, 2001; Rose & Walker, 2004; Hansson 2001/2010, 2007; Rose 2000, 2011; McCarthy, 2007; Bennett 2013, 2014; and others) is a theory developed to account for harmony and dissimilation. In ABC, harmony is caused by two families of constraints: correspondence constraints that demand segments with a specific feature value to be in a correspondence relation, and Ident-CC/VV constraints, that ban segments in correspondence that have different feature values. Harmony occurs when at least one constraint that demands segments to correspond, and one constraint that demand segments in correspondence to be identical outrank a relevant set of Ident-IO constraints.

In ABC, consonants in correspondence have the same status (1b). The segment that determines the target of the process is no different from all other segments in correspondence. This formulation of surface correspondence is referred to as *CC-Correspondence*.

Headed Agreement By Correspondence is an extension of ABC (Iacoponi, 2015). HABC maintains the basic tenet of ABC that harmony is obtained via surface correspondence and feature identity. However, in HABC correspondence holds between Heads and Correspondents (**HD-Correspondence**), rather than between segments with the same status (1a). Similarly to some other phonological heads, such as onset heads (Smith 2002; Murray 2006), c-heads are not phonetically distinguishable from c-dependents, and the choice of the head consonant is mostly determined by rankable constraints. C-heads mainly determines the directionality of the harmony, and it has been used to explain Derived Environment Effect (ibid.).

---

* I am very grateful to Alan Prince for his essential guidance in this project. Akinbiyi Akinlabi, William Bennett, and Paul de Lacy also contributed with fundamental feedback and encouragements at various stages of the project. This work was presented at WoFST 2015 and at AMP 2015: thanks to all the participants for their comments.

(1) IO and surface correspondence relations

/**s**-wi-**s**-noxʃ-it/   /**s**-wi-**s**-noxʃ-it/

[ʃ-wi-ʃ-noxʃ-it]   [ʃ-wi-ʃ-noxʃ-it]

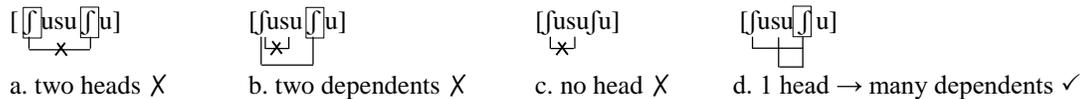a. HD-Correspondence[1]   b. CC-Correspondence

Heads are by definition a special constituents in a domain. For example, secondary stress is the head of a foot, and primary stress is the foot head of a prosodic word. The domain of c-heads is called an HD-Correspondence domain. The definition in (2) reflects the fact that correspondence domains are always headed.

(2) **Definition of correspondence domain**: A correspondence domain consists of a c-head and of its correspondents.

Well-formed correspondence domains have one and only one head (see Iacoponi, 2015 for a rationale). This means that no candidate contains two heads in correspondence, two dependents in correspondence with each other, or a correspondence domain without a head, as illustrated in (3).

(3) Possible and impossible correspondence relations

[ʃusuʃu]   [ʃusuʃu]   [ʃusuʃu]   [ʃusuʃu]

a. two heads ✗   b. two dependents ✗   c. no head ✗   d. 1 head → many dependents ✓

1.2 *HABC*   HD-Correspondence is governed by a family of constraints that penalize segments with the same feature value that do not correspond. The constraint Corr-HD(+sibilant), for example, assigns a violation for each [+sibilant] segment in the output that is not in the same correspondence domain of any other sibilant: Corr-HD(+sibilant) is violated twice in the output $\{ʃ_x…ʃ_y\}_{root}$ because neither sibilant in the domain root is in correspondence, while Corr-HD(+sibilant) is violated only once in $\{s_y…ʃ_x…ʃ_x\}_{root}$, because only one sibilant in the root is not in correspondence. Corr-HD is the family of constraints that determines the correspondence features in languages with harmonies.

(4) Definition of Corr-HD(αf, dom):
   Penalize segments with the same feature value αf that are not in the same correspondence domain.
   For each consonant in *dom*, assign a violation if:
      a. X is [αf], &
      b. There is a consonant Y [αf] in *dom,* &
      c. X is not in the same correspondence domain as Y.

Ident-HD constraints govern the selection of harmonizing features. This family of constraints demands that all segments in HD-Correspondence share the same value of a feature *f*. For example, the constraint Ident-HD(anterior) assigns a violation for each pair of segments in correspondence that has a different feature value for the feature [anterior].

---

[1] Outputs with HD-Correspondence relations are represented using the following notation: (i) heads are in bolds, and between parentheses; (ii) when using a linear notation (i.e. in the text or in tableaux) segments with the same index are in the same correspondence domain. For example, the output $[s_y…ʃ_x…(ʃ)_x]$ contains one c-head, indicated as $(ʃ)_x$. The two sibilants [ʃ] correspond—as indicated by the fact that they both have the same correspondence index *x*—and therefore constitute a correspondence domain. The [+anterior] sibilant [s] instead has a different, and unique correspondence index *y*. Therefore, it does not correspond with any other sibilants, and does not belong to any correspondence domain.

(5) Definition of Ident-HD(f):
 Penalize segments in HD-correspondence that have a different feature value f
 For each distinct pair of a consonant X [αf] & Y [βf] in HD-correspondence,
 assign a violation iff α≠β

C-heads are phonological constituents, just like feet and prosodic words. The c-head alignment constraint in (6) follows the Generalized Alignment schema (McCarthy and Prince, 1993; cf. McCarthy, 2003).

(6) Definition of Align(c-head, R) := Align(PrWrd, R, c-head, R):
 Penalize c-heads that are not aligned to the right edge of the prosodic word
 For each head H in the output,
 assign a violation for each segment C between H and the right edge of the prosodic ω,
 where ω is the prosodic word H is contained in, *and*
 C is a consonant in correspondence with H.
 *(H, D): {H…D}$_\omega$

For example, Align(c-head, R) is violated once in the output [(ʃ)$_\mathbf{x}$at$_y$as$_\mathbf{x}$]$_\omega$, since the head (ʃ) is separated from the right edge of the prosodic word by the one sibilant in correspondence [s]. Segments not in correspondence with the head are ignored by the alignment constraint.
 Even though the output [(ʃ)$_\mathbf{x}$at$_y$as$_z$]$_\omega$ is segmentally identical to [(ʃ)$_\mathbf{x}$at$_y$as$_\mathbf{x}$]$_\omega$, Align(c-head, R) is not violated. The sibilant [s] is not in correspondence with the head, and therefore it has the same status of any other segment.
 The last c-head constraint considered in the paper is the positional faithfulness constraint Ident-IO(c-head). Ident-IO(c-head) is violated when a consonant head is not faithful to its input correspondent, as per the definition in (7).

(7) Definition of Ident-IO(c-head):
 Penalize any unfaithful mapping of c-heads
 For each distinct pair of a consonant X in the input, and its correspondent X´ in the output, assign a violation if:
 a. if X is [αf], &
 b. X´ is not [αf], &
 c. X´ is a c-head

For example, Ident-IO(c-head) is violated by the mappings s → (ʃ), and ʃ → (s), but not by the mapping to a non-head segment s → ʃ.
 Positional faithfulness constraints are Ident-IO faithfulness constraints that refer to specific prosodic positions (e.g., Ident-IO(Head-σ), in Alderete 1995:14), morphological positions (e.g., Ident-IO(root), in McCarthy & Prince, 1995; Beckman, 2013) or phonological constituents (e.g., Ident-IO(onset) Lombardi 1999; Padgett 2002). Ident-IO(c-head) is then a positional faithfulness constraint that refers to a phonological constituent, of the same kind of Ident-IO(onset).

1.3 *Property theory*   In the rest of this section, I introduce three basic concepts of property theory (Alber and Prince, in prep., Prince and Merchant, 2015): intensional properties (1.3.1), mootness (1.3.2), and constraint classes (1.3.3). The theory is used to analyze the typology of HABC in section 3. All definitions are from Alber & Prince (2015).

1.3.1 *Intensional properties*   In property theory, a typology is defined by its *intensional properties*, which form the inventory from which all of its grammars can be obtained. Properties are set of mutually inconsistent *values*, where a value is a ranking condition that has scope over the set of all linear orders in a system. In their simplest form, properties consist of two constraints (or two singleton sets of constraints).

For example, the property DomType (3.2.4) is informally defined as follows[2]: DomType.al/faith := al <> id.hd. The notation indicates the name of the property (DomType), followed by its two possible values (DomType.al and DomType.faith) and the two ranking conditions it imposes (al ≫ id.hd and id.hd ≫ al). An analysis of the intensional property of a system S consists in finding the set of properties S_p = {P.a, P.b …}, so that every grammar in the typology is defined by the unique set of property values that it *satisfies*. The term satisfy is used following the definition in (8).

(8) **Definition.**
    A grammar G *satisfies* a property value iff every λ (linear order) ∈ G satisfies that value.
    Example:
        a grammar G *satisfies* DomType.al ⇔ al ≫ id.hd for each λ ∈ G
        a grammar G *satisfies* DomType.faith ⇔ id.hd ≫ al for each λ ∈ G

Property values apply to individual grammars. If all linear orders in a grammar satisfy one of the two property values expressed by a property, the grammar satisfies that property value. Grammars satisfying the same property value belongs to the same class: they are opposed to their antagonist class constituted by all grammars satisfying the opposite value, and constitute a homogenous group of languages with respect to that property. Properties are entirely defined in terms of ranking conditions (ERCs, in fact), but their effect is reflected on the extensional characteristics of the mappings they induce. Properties, then, concurrently group classes of grammars with similar extensional traits (e.g. grammars that have harmony, grammar with dissimilation etc.)

1.3.2 *Mootness* The scope of a property is the entire typology. When we define a property, we cannot restrict the set of languages that may or may not have one of the two possible property values. However, it is possible that some grammars do not satisfy either value. This occurs when neither ranking condition imposed by the property is satisfied by a grammar. In this case, the property is said to be *moot* for that grammar.

    Mootness is possible because grammars are collections of specific linear orders (those related by a single adjacent transposition of constraints), and in order for a grammar to have a property value, all linear orders must respect the ranking condition imposed by a property value, as per definition in (8). For example, the *Pure Dir* grammar discussed in (2.3) includes the following two linear orders:

(9) Two linear orders of the grammar *Pure Dir*

        a. f.-sib ≫ id.hd > corr ≫ **f.hd** ≫ **al** ≫ f.-ant ≫ f+ant
        b. f.-sib ≫ id.hd > corr ≫ **al** ≫ **f.hd** ≫ f.-ant ≫ f+ant

The definition in (8) requires that *all* linear orders in a grammar satisfy a value of the property. The property DomType has two values, evaluated by the conditions: al ≫ f.hd and f.hd ≫ al. The linear order in (9a) does not satisfy the condition al ≫ f.hd, because f.hd ≫ al. Hence, the grammar does not have the property value DomType.al. The linear order in (9b) does not have the ranking of the antagonist f.hd ≫ al, and so it does not satisfy the value DomType.faith. The grammar described by the two linear orders, then, does not satisfy any of the two possible property values, and so it is moot to the property DomType[3].

    Notice that mootness does not limit the analytic power of a property, but rather it adds a third element of distinction: it indirectly encodes the scope of a property, without explicitly indicating it in its definition. As an example, consider the property HType discussed in (3.2.2). Although binary, the property actually informs us of three classes of grammars: the one with directional or dominant harmony, where the two specific ranking conditions imposed by the property hold, and the class of grammar without a harmony type, where the property is moot.

---

[2] A property is formally defined as an expression of the form X(λ) <> Y(λ) where X and Y are functions that takes a linear order as they argument. X and Y are called antagonist of a property (Alber and Prince, 2015).
[3] It is in fact quite easy to prove that the property is moot for all the grammars (relative orders) where one of the two constraints of the property is unranked.

1.3.3 *Constraint classes*   Properties refer to specific sets of constraints defined by the analyst, called *constraint classes*. For example, the constraint class F (defined in 3.1), includes the constraints f.+ant and f.-ant. The important aspect of constraint classes is that they may constitute the argument of the two methods K.sub and K.dom, which permits the formulation of more general properties. The two methods are defined as follows.

(10) Definiton of K.dom and K.sub

   K.dom. For a constraint class K and a linear order λ,
   K.dom (λ) = χ ∈ K such that χ denotes the *highest ranked member* of K in λ. [*dom*inant]

   K.sub. For a constraint class K and a linear order λ,
   K.sub (λ) = X ∈ K such that X denotes the *lowest ranked member* of K in λ. [*sub*ordinate]

K.sub and K.dom are methods that take a linear order λ as their argument, and return the minimum (lowest ranked) and the maximum (highest ranked) element of the order λ. Because K.dom and K.sub operate on linear orders, we are guaranteed that there will be one and only one minimum and maximum for each order. An example of the application of F.dom and F.sub to F to a linear order $λ_i$ is given below. In (), we define most of the properties using constraint classes and the K.dom and K.sub methods.

(11) Example of K.sub and K.dom
       F := {f.+ant, f.-ant}
       $λ_i$ = corr ≫ f.+ant ≫ id.hd ≫ f.-ant ≫ al.
       F.dom($λ_i$) → f.+ant
       F.sub($λ_i$) → f.-ant

2 Typology

In this section, I define the parameters for the generation of the typology. In 2.1, I introduce the candidate set and the constraint set. In order to generate a sizeable typology, some simplifications have to be made; I discuss them in section 2.2. Finally, section 2.3 contains a description of the extensional typology. The following section of the paper contains the analysis of this typology using the concepts defined in section 1.

2.1 *GEN_{HABC} and CON_{HABC}*   A complete characterization of the candidates is provided in (12), using the notation in Iacoponi (2015). The input and the output consist of two segments with the correspondence feature [s, ʃ], and a non-correspondence segment [t]. Vowels are irrelevant, and they are thus always indicated as [a]. The structure of the input is /CVCV/. An output may have one and only one head, containing the same set of segment [s, ʃ, t] included in the input. For example, the algorithm generates the following candidate: /ʃasa/ → [s_xa(s)$_x$a].
    There is no deletion or epenthesis in the candidates, and so the input and the output always contain four segments (2 consonants and 2 vowels). The input contains the same set of segments as the output, but no correspondence indices or heads. The full list of candidates can be found in the supplement to this paper (Iacoponi, 2016b).

(12) GEN_{HABC}
    inputs
        [s, ʃ, t] a [s, ʃ, t]

    Mappings
        ʃ → s, ʃ, t
        s → s, ʃ, t
        * → faithful

Other

　　All combinations of surface correspondence among sibilants
　　one-to-one I/O mapping only

The constraint set CON$_{HABC}$ is reported in (13). Each row contains the full name, a short alias and an informal definition of the constraint (1.2). There are seven constraints in the system, which are necessary and sufficient to describe a typology with basic interactions.

　　Any basic analysis or typology of HABC (and *mutatis mutandis* of ABC) includes a constraint set similar to the one given below. Three Ident-IO constraints refer to the correspondence feature value [+sibilant] to control for dissimilatory effects, and the harmonizing feature [+anterior] and [-anterior] for the directionality of harmony (CON$_{m/f}$); the two constraints adapted from ABC that refer to HD-Correspondence are Corr-HD; Ident-HD; and two constraints on c-heads are Align(c-head, R) and Ident-IO(c-head).

(13) CON$_{HABC}$

　　CON$_{m/f}$

　　　　Ident-IO(+sib)/f.+sib: "*Penalize any unfaithful mapping of the feature value [+sibilant]*"
　　　　Ident-IO(+anterior)/f.+ant: "*Penalize any unfaithful mapping of the feature value [+anterior]*"
　　　　Ident-IO(-anterior)/f.-ant: "*Penalize any unfaithful mapping of the feature value [–anterior]*"

　　CON$_{HD-Corr}$

　　　　Corr-HD(+sib)/corr: "*Penalize [+sib] segments not in the same correspondence domain*"
　　　　Ident-HD(ant)/id.hd: "*Penalize correspondents with a different feature value [anterior]*"

　　CON$_{c-head}$

　　　　Align(c-head, R)/al: "*Penalize c-heads not aligned to the right edge of the prosodic word*"
　　　　Ident-IO(c-head)/f.hd "*Penalize any unfaithful mapping of c-heads*"

2.2　　*S$_{HABC}$*　　A Due to the large number of languages otherwise generated, and the impossibility to account for all the variables interacting with the basic process of harmony, the system is defined to generate a relatively small set of fundamental candidates and to include only basic constraints. Some of the processes observed in the empirical landscape of consonant harmony are therefore not captured. The rationale for delimiting the system is discussed in the rest of the section. S$_{HABC}$ is contrasted with possible expanded alternatives in (14).

　　The first simplification concerns the length of the mappings and the number of harmonies (correspondence domains) possible in an output. I only use candidates composed of two segments, with a single correspondence domain. Systems with more than one harmony are beyond the scope of this paper, as their empirical validation would depend on a careful examination of the empirical data, which in the domain of consonant harmony is scarce in this respect.

(14) Domain and simplifications for the typology

Component	S$_{HABC}$	Alternatives
GEN	2 segments	2 or more segments
GEN – Harmony	One harmony	Multiple harmonies
GEN – Morph	Root	Prefix(es), Suffixes, suffix(es) + prefix(es)
CON	Ident-IO(-ant) ⫸ Ident-IO(+ant)	Only Ident-IO(-ant), free ranking, stringency
CON	Align(c-head, R)	Align(c-head, R) and Align(c-head, L)

Directionality is in some cases determined by a specific morphological or prosodic position. The candidate sets does not include any indication of morphological or prosodic constituency. This simplification is justified, and the results are still significant for the following reasons.

6

First, root-control effects are obtained in HABC on the basis of the interaction of Ident-IO(root) with the constraint set discussed in this paper: the theory thus remains exactly the same.

Second, even in root-control harmonies there is a sub-system where the conflict between directional and dominant harmony has to be resolved. If a root is harmonic in the input, a choice has to be made within the root itself to determine which feature value is the harmonizing one. This choice is determined by either dominancy, alignment to an edge, or both.

Third, the resulting typology includes all the harmony types already generated in the simplified typology. In other words, the typology that includes root-control effects is just an extension of the one discussed earlier.

The last two items in (14) are assumptions. The first assumption reflects the generalization that in consonant harmony the harmonizing feature is always marked in dominant harmony, and always on the rightmost segment in directional harmonies. The fixed ranking relation between the Ident-IO(-ant) and Ident-IO(+ant) accounts for the former generalization, while the absence of a constraint Align(c-head, L) accounts for the latter. The discussion concerning why stringency or a model with only Ident-IO(-ant) do not make the right typological predictions, and concerning the empirical evidence for these generalization is out of the scope of this paper (but see Iacoponi, 2016b).

The other simplifications do not significantly change the structure of the typology. The language types discussed still belong to the typologies that include the two Ident-IO constraints freely ranked, and both Align(c-head, R) and Align(c-head, L).

To conclude, while important simplifications have been made for expository reasons, none of them significantly affects the structure of the typology generated by the theory.

2.3 *The extensional typology*   The universal support for the typology only includes the two inputs with disharmonic sibilants /ʃasa/ and /saʃa/. The table below contains the list of the grammars generated, called the *extensional typology* of S$_{HABC}$. The typology includes nine languages. For an empirical survey of the various grammars, and a more detailed description of each type, see Iacoponi (2015).

(15) The extensional typology

Inputs →	ʃasa	saʃa	Language description	Languages
Dom.Hright	(ʃ)$_x$aʃ$_x$a	ʃ$_x$a(ʃ)$_x$a	*Dominant harmony* Harmonize to the marked segment	Malto, Basque (Moroccan Arabic)
Dom.Hfaith	ʃ$_x$a(ʃ)$_x$a	ʃ$_x$a(ʃ)$_x$a		
Pure Dir	s$_x$a(s)$_x$a	ʃ$_x$a(ʃ)$_x$a	*Direction harmony* Harmonize to the rightmost segment	Tsilhqot'in, Chumash, Saisiyat, Thao
Dom-Dir.noCor	ʃ$_x$as$_y$a	ʃ$_x$a(ʃ)$_x$a	*Dominant-Directional harmony* Harm. only if rightmost segment is dominant	Ngizim, Pengo, Kera
Dom-Dir.Cor	ʃ$_x$a(s)$_x$a	ʃ$_x$a(ʃ)$_x$a		
Diss-Dir	ʃ$_x$at$_y$a/t$_x$as$_y$a	ʃ$_x$a(ʃ)$_x$a	*Dominant-Directional* dissimilation Harm if rightmost marked, diss. otherwise	(Javanese)/unattested?
Diss.	ʃ$_x$at$_y$a/t$_x$as$_y$a	ʃ$_x$at$_y$/t$_x$as$_y$a	*Dissimilation* Dissimilation for disharmonic inputs	Chol
Faith.noCor	ʃ$_x$as$_y$a	s$_x$aʃ$_y$a	*Faithful* No harmony or dissimilation	All languages
Faith.Cor	ʃ$_x$a(s)$_x$a	s$_x$a(ʃ)$_x$a		

3 *Analysis*

The typology generated in the previous section only contains the *mappings* that distinguish each language from all the others. In this section, I focus on the ranking information (the *intensional* typology), and on its relation with the extensional typology defined in 2.3. In 3.1, I introduce the constraint classes used for the analysis. The typology is then analyzed using property theory in 3.2.

3.1 *Constraint classes*   I start by defining the constraint classes referred to in the analysis. Each definition contains the name of the class (e.g. F.-sib), followed by the set of constraints it includes (e.g. {f.+sib}, and a description of the class and of its constraints.

FCorr := {f.+sib}. This class contains Ident-IO constraints that refer to the correspondence feature value. Typologies containing more than one class of segments in correspondence would include one Ident-IO constraint in the class for each potential correspondence feature in the system.

F := {f.-ant, f.+ant}. The class contains the Ident-IO constraints that refer to the harmonizing feature values. Note that correspondence is established among segment with a specific feature value (otherwise all segments would be in correspondence), while for the harmonizing feature, both values have to be specified to distinguish the directionality of harmony. Also notice that although the class contains both Ident-IO constraints, the two constraints are in a fixed ranking relation in the typology (**f.-ant ≫ f.+ant**).

Agr := {corr, id.hd}. In ABC harmony is obtained via correspondence and identity. This class contains the two constraints that favor them: **corr** demands segments with a specific feature value to be in correspondence, and **id.hd** demands that segments in correspondence are identical for a feature value.

HC := {al, f.hd}. HC stands for Head Constraints: these are the two constraints that refer specifically to heads. In particular, **al** refers to the position of the head, while **f.hd** is the positional faithfulness constraint that refers to its privileged status.

U := {Agr, F.dom, FCorr}. U is fundamental in defining harmonic from non-harmonic languages, as well as the type of directionality of the harmony. Notice that each constraint belonged to only one class and each class contains a set of constraints that only appear in that class. U, on the other hand, contains other constraint classes.

3.2 *Properties*   The typology generated by S$_{HABC}$ is analyzed using the properties listed in (16). Each row contains the definition of a property, its values, a brief description, and its actual scope (what languages are partitioned by it). Each property is discussed in the remainder of the section.

(16) Properties of S$_{HABC}$

Name	Values	Abbreviation	Scope
hasHarm.yes/no	U.sub <> F.sub	Has harmony?	All languages
HType.dom/dir	U.sub <> HP.sub	Harmony type	hasHarm.yes
DomType.al/faith	al <> f.hd	Dominancy type	HType.dom
IsSymm.yes/no	F.dom <> Agr.sub	Is symmetric?	All but HType.dom
DissRes.diss/faith	Agr.sub <> f.-sib	Disharmony resolution	isSymm.no
FCorr.yes/no	corr <> id.hd	Faithful correspond?	DissRes.faith

Recall that the scope of a property is the entire typology. This means that when we define a property, the specific ranking information it defines applies to all languages. It is thus useful to have all the rankings available during the analysis of all properties. The appendix inlcudes the Hasse diagrams of the languages generated by the typology, grouped by their property values. The details of the rankings are discussed in the rest of the section. To facilitate the reading of the diagrams, the constraints were colored differently whenever possible depending on their class, following the color scheme that appears in the list of constraint classes in adjacent to the rankings. Finally, the table in (17) contains a visual representation of the partitions created by property splits and of their scope, and it shows that each language is uniquely defined by its property values.

(17) Property values of the grammars in S$_{HABC}$

	hasHarm	HType	DomType	isSymm	disRes	FCorr
Dom.Hright	yes	dom	al			
Dom.Hfaith	yes	dom	faith			
Pure Dir	yes	dir		yes		
Diss-Dir	yes	dir		no	diss	
Dom-Dir.Corr	yes	dir		no	faith	yes
Dom-Dir.NoCorr	yes	dir		no	faith	no
Diss	no			no	diss	
Faith.Corr	no			no	faith	yes
Faith.NoCorr	no			no	faith	no

3.2.1   *HasHarm*   The property with the widest scope is hasHarm, which splits all languages in the typology into two partitions: languages that achieve harmony for at least one disharmonic input, and languages where disharmonic inputs do not harmonize. Recall that U is the macro-class that contains the constraints that demand correspondence and surface identity, as well as the two faithfulness constraints that refer to the correspondence and the dominant harmonizing feature. Harmony is obtained when all the constraints in U dominates the subordinate faithfulness constraint that refer to the harmonizing feature. If two segments have to correspond (corr), be identical (id.hd), and cannot dissimilate to escape the correspondence restriction (f.-sib), then they undergo harmony.

If we look at the rankings in the appendix, we can confirm that in all rankings in the partition hasHarm.yes, the lowest ranked constraint in U.sub dominates F.sub. For example, for HType.dom languages (Dom.HRight and Dom.HFaith) there are four possible linear orders of U and f.+ant (i.e. Fsub), listed below. In all cases, the lowest ranked of the U class (in bold) dominates f.+ant.

(18) Some linear orders of HType.dom languages
    id.hd ≫ f.-ant ≫ f.-sib ≫ **corr** ≫ f.+ant
    f.-ant ≫ f.-sib ≫ corr ≫ **id.hd** ≫ f.+ant
    f.-sib ≫ corr ≫ id.hd ≫ **f.-ant** ≫ f.+ant
    …

The property hasHarm defines the class of language that have the order relation U.sub > F.sub as having harmony, but also the antagonist languages F.sub ≫ U.sub that do not achieve harmony for disharmonic inputs. There are three such hasHarm.no languages in the typology for which the order F.sub ≫ U.sub have to hold.

Because of the limitation on the constraint set, F.sub is always f.+ant. Therefore, in the three languages hasHarm.no, f.+ant must dominate the lowest ranked constraint of the class U. The rankings in the appendix show that it is true for all the three languages left in the typology. In Faith.NC, f.+ant dominates corr, in Faith.C it dominates id.hd, and in Diss it dominates corr, in all linear orders of each grammar.

3.2.2   *HType*   The property HType distinguishes between two different types of harmony, known: dominant-recessive and directional. The type of harmony is determined by the interaction of the U constraints, with the other two constraints that are not in U: al and id.hd. These two constraints only have an effect in establishing the directionality of harmony[4].

---

[4] In this simplified typology. Derived Environment Effect as in Basque (Iacoponi, 2015), for example, and directionality in root-controlled harmony are also influenced by HP constraints.

The analysis also shows that directionality is not determined by the relative order of the two constraints with respect to each other. Directional harmony is obtained when HP.sub (i.e. both al and id.hd) dominates U.sub, dominant harmony obtains when U.sub dominates HP.sub.

The property HType splits hasHarm.yes languages into two partitions. If there is no harmony, we expect the type of harmony to be undefined. The property analysis formally confirms this intuition. We have observed that for a property value to be satisfied for a grammar, the ranking requirement has to hold for all its linear orders. HType is moot for the entire partition defined by hasHarm.no, meaning that every language hasHarm.no (i.e no harmony), is moot to the distinction (has no harmony type).

We can test this hypothesis by looking at three languages in the hasHarm.no group, and prove that neither U.sub ≫ HP.sub or HP.sub ≫ U.sub hold for any of the languages in the partition. In all three languages, both HP constraints are unranked. In terms of linear relations, an unranked constraints can appear in any position in the order. Both al and id.hd are unranked, and therefore for each language there is always a linear order where HP.sub dominates U.sub, and a linear order where U.sub dominates HP.sub. As an example, the two crucial orders for the language Faith.NC are given in (19). In (19a), corr dominates al (U.sub ≫ HPsub), while in (19b) al dominates corr (Hp.sub ≫ Usub). Both linear orders are part of the same grammar, and therefore the property HType is moot for it.

(19) HType in hasHarm.no languages
      a. f.hd ≫ f-ant ≫ f+ant ≫ f.-sib ≫ id.hd ≫ **al (HPsub)** ≫ **corr (Usub)**
      b. f.hd ≫ f-ant ≫ f+ant ≫ f.-sib ≫ id.hd ≫ **corr (Usub)** ≫ **al (HPsub)**

3.2.3 *HType and isSymm* In HType.dom languages, U.sub dominates HP.sub. Since there are four constraints in U and two constraints HP, there are 16 possible subsets of linear orders that could be potentially distinguished by the property. Nevertheless, there are only two dominant languages in the typology. If we look at the Hasse diagrams, we observe that only the constraints in HP have a distinctive relative order. The constraints in U.sub are unranked in HType.dom languages, and therefore, no matter what the specific constraint U.sub is, the language is entirely determined by the relative ranking of the constraints in the class HP. The order al ≫ f.hd thus gives Dom.HRight, while the opposite order f.hd ≫ al gives Dom.HFaith. The only languages where al and f.hd are ranked consistently are HType.dom languages; in all other languages the two constraints are unranked, and therefore the property is moot.

The property isSymm distinguishes pure directional languages (i.e. languages where there is always directional harmony), from languages where some mappings are not harmonic. Pure directional languages are obtained when Agr.sub dominates F.dom, which happens when all the constraints that demand harmony dominate all the faithfulness constraints that refer to the harmonizing feature. This causes the head to be always aligned to the right, and the output to be always harmonic.

In isSymm.no languages, instead, one of the faithfulness constraints in F dominates one of the constraints that demands agreement. In this case, harmony is either absent, or only enforced for some disharmonic inputs. Notice that whether or not there is harmony, is not determined by this isSymm, but by hasHarm. The property isSymm only defines whether the language has harmony everywhere, or only for some inputs. It is hasHarm that determines whether there is harmony or not.

Unlike HType, isSymm does not split a partition obtained from another property. If we look at all the rankings in the appendix, we observe that this ranking holds not only for the languages with harmony (hasHarmony.yes), but also for languages without harmony (hasHarmony.no). For example, in the faithful language Faith.NoCorr, f.+ant dominates corr, while in the language Diss, f.+ant dominates f.-sib. The property is pointing at a specific type of conflict. If there is a disharmonic pair of segments, you either get harmony, or something has to be changed. One among the following three option has to be chosen: (i) change the class of one of the segments so that it does need to correspond; (ii) map faithfully with the segments not in correspondence (iii) map faithfully with the segments in correspondence

This choice has to be made not only for languages where there is no harmony, but also for languages where harmony is restricted only to specific disharmonic inputs (i.e. grammars with hasHarm.yes and isSymm.yes). This is why the choice of how to resolve this conflict is moot for pure dominant and pure directional languages: the conflict never arises, because disharmonic inputs are always harmonized.

We can visualize the orders that determines this choice in the diagrams (in orange) in the appendix. Depending on which constraint (corr, f.-sib or id.hd) is dominated by the other two constraints (i.e. sub), we

obtain one of the three different types of mappings. This type of structure is called a bot cycle in Prince and Merchant (2015), and it can be analyzed with two properties. Notice that the partition is going to be arbitrary in terms of intensional properties. Any grouping of the three constraints (20) in a class of two members will do.

(20) Three possible classes for the analysis of the bot cycle
  C_1 = {corr, f.-sib}
  C_2 = {corr, id.hd}
  C_3 = {f.-sib, id.hd}

The three classes, however, can be distinguished by looking at the extensional property of the partition. If we want a property that groups together the languages that do not have correspondence, we define the class C_1; if we want to distinguish faithful from unfaithful languages, we pick C_2. The class C_2, however, is already defined, and corresponds to Agr. The property DissRes.diss/faith is then defined as Agr.sub $\diamond$ f.-sib. Dissimilation is obtained when the faithfulness constraint that refers to the correspondence feature is dominate by one of the two markedness constraints in the Agr class. If the faithfulness constraints dominates one of the two markedness constraints that demand harmony, the mappings for disharmonic input is always faithful. The last property FCorr := corr $\diamond$ id.hd distinguishes between the faithful disharmonic outputs: if corr $\gg$ id.hd, the faithfully mapped disharmonic segments are in correspondence; if id.hd $\gg$ corr, they do not correspond. This final split, concludes the analysis of the typology.

4 Appendix

Hasse diagrams for all rankings in the typology, and the color-coded constraint classes.

HP = al, id.HD
Agr = m.HD, f.-sib, corr
F = f.-ant, f.+ant
U = **Agr**, **F.dom**, f.-sib
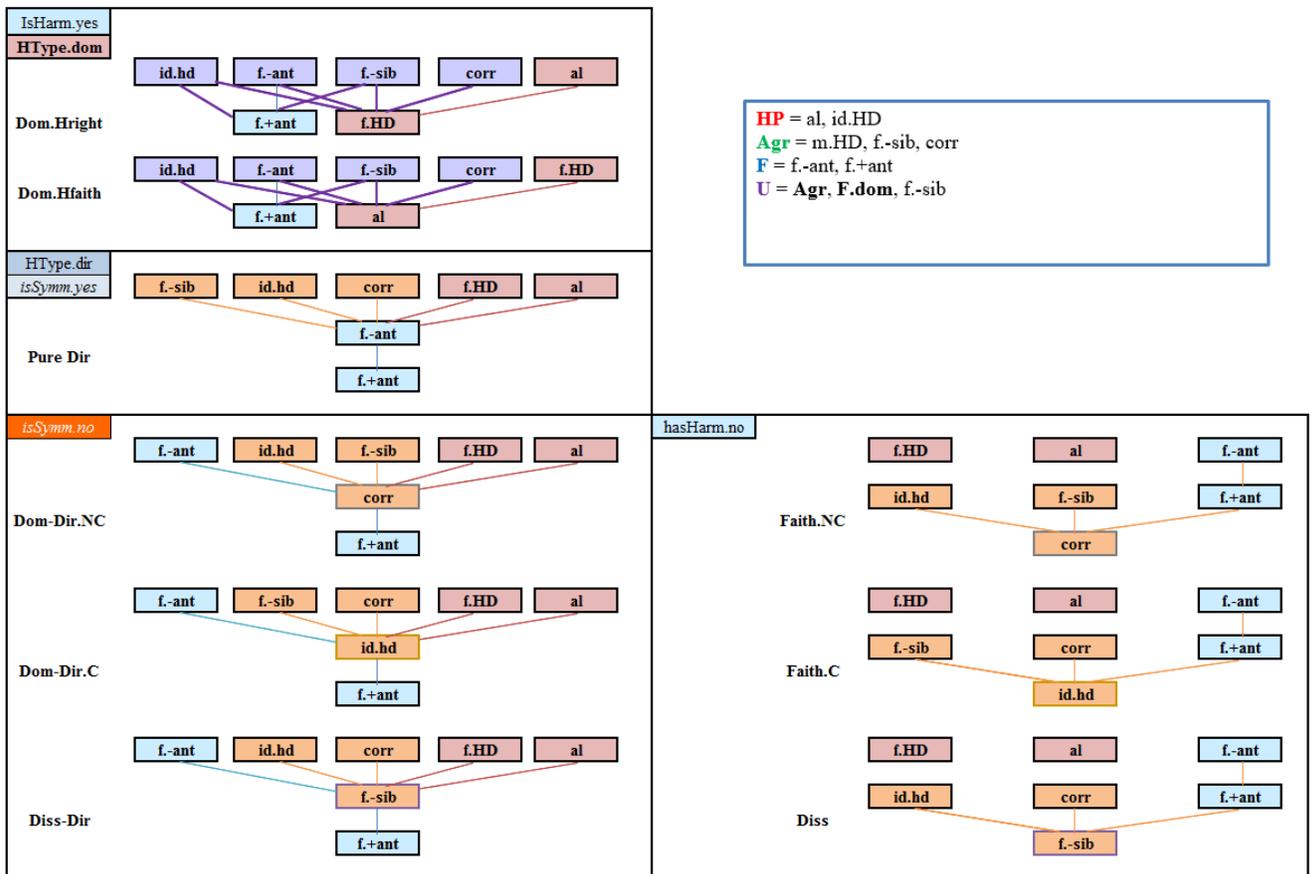
References

Alderete, D. John. (1995). Faithfulness to prosodic heads. Ms. University of Massachusetts, Amherst. [ROA-93].

Alber, Birgit & Prince, Alan. (2015). Outline of Property Theory (v.10). Ms. University of Verona & Rutgers University.

Alber, Birgit & Prince, Alan. (In prep). Typologies. Ms. University of Verona & Rutgers University.

Ananian, C. Scott, and Nevins, Andrew I. (2001). *Postvelar Harmonies: A Typological Odyssey*. Ms. MIT, Cambridge.

Applegate, Richard B. (1972). *Ineseño Chumash Grammar*. Ph.D. dissertation, University of California, Berkeley.

Bakovic, Eric (2000). *Harmony, Dominance and Control*. Ph.D. dissertation, Rutgers University.

Bakovic, Eric & Rose, Sharon. (2014). Directionality in distance effects. Paper presented at *ABC <-> Conference*, California, Berkeley.

Beckman, Jill N. (2013). Positional faithfulness: an optimality theoretic treatment of phonological asymmetries. Routledge.

Bennett, William G. (2013). *Dissimilation, Consonant Harmony, and Surface Correspondence*. Ph.D dissertation. Rutgers University.

Bennett, William G., DelBusso, Natalie, Iacoponi Luca. (2016). Formally mapping the typologies of interacting ABCD. *2015 Supplemental Proceedings of the Annual Meetings on Phonology*, Vancouver.

Brasoveanu, Adrian & Prince, Alan. (2005). Ranking and necessity. Part I: the fusional reduction algorithm. [ROA-794].

de Lacy, Paul. (2002). The interaction of tone and stress in Optimality Theory. Phonology 19.1: 1-32.

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

Iacoponi, Luca. (2015). Headed Agreement By Correspondence: Directionality in Consonant Harmony. Ms. Rutgers University. [ROA-1433].

Iacoponi, Luca. (2016a). Fixed ranking over stringency: Evidence from consonant harmony. Paper presented at PhoNE 2016, NYU, New York.

Iacoponi, Luca. (2016b). A supplement to 'The intensional and the extensional properties of Headed Agreement by Correspondence'. Ms. Rutgers University.

Lombardi, Linda. (1999). Positional faithfulness and voicing assimilation in Optimality Theory. *NLLT, 17*(2), 267–302.

Merchant, Nazarré & Prince, Alan. (2015). The mother of all tableaux. Ms. Eckerd College & Rutgers University.

McCarthy, John J. (2004). Headed spans and autosegmental spreading. Ms. University of Massachusetts, Amherst.

McCarthy, John J. (2007). Consonant harmony via correspondence: Evidence from Chumash. Ms. University of Massachusetts, Amherst.

McCarthy, John J. & Prince, Alan. (1993). Generalized alignment. In Yearbook of Morphology, 79-153. Dordrecht: Kluwer. ROA 7.

McCarthy, John J., and Prince, Alan (1995). Faithfulness and reduplicative identity. In: Suzanne Urbanczyk Beckman and Laura Walsh eds., *UMOP-18: Papers in Optimality Theory*.

Padgett, Jaye (2002). Feature classes in phonology. *Language*, 78, 81–110.

Prince, Alan, & Smolensky, Paul. (1993/2004). *Optimality Theory: Constraint Interaction in Generative Grammar*: Blackwell: Malden-Oxford-Carlton. [ROA-537].

Prince, Alan, Tesar, Bruce, Merchant, Nazarré. (2014). OTWorkplace. Software version 74.

Prince, Alan. (2002a). Entailed Ranking Arguments. ROA 500.

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

Prince, Alan. (2006). Implication and impossibility in grammatical systems. ROA 880.

Prince, Alan. (2015). One Tableau Suffices. Ms. Rutgers University, New Brunswick, NJ.

Rose, Sharon, and Walker, Rachel. (2004). A typology of consonant agreement as correspondence. *Language, 80*(3), 475–531.

Walker, Rachel. (2000). Long-distance consonantal identity effects. Paper presented at the Proceedings of the West Coast Conference on Formal Linguistics.

Walker, Rachel. (2001). Round Licensing, Harmony, and Bisyllabic Triggers in Altaic. *NLLT, 19*(4), 827–878.