The Mother of All Tableaux

Nazarré Merchant
Alan Prince

Silent, upon a peak in Darien

Acknowledgments
The authors' names are arranged in alphabetical order. We would like to thank Birgit Alber and Natalie DelBusso for detailed helpful comments on the manuscript, and Nick Danis, Luca Iacoponi, Hope McManus, Kevin Thielen, Tonia Krueger, and Jane Grimshaw for much discussion of related issues. We have profited from interaction with audiences at Rutgers, UCSC, MIT, Stanford, OCP 2014 (Leiden), and the Workshop on the Formal Structure of OT Typologies (2015), where aspects of this work were presented. Prince would like to thank Rutgers, The State University of New Jersey and especially Dean Douglas Greenberg for support during the period in which this manuscript was prepared.

Contact
Merchant: Eckerd College, Department of Mathematics: merchann@eckerd.edu
Prince: The ThOT Lab: alan.s.prince@gmail.com

Last mod: July 14, 2016

Preface

OT grammars arise from the comparison of candidates over a set of constraints. An OT *typology*, we show, implicitly compares entire grammars over the same set of constraints. From the details of this comparison, each constraint can be seen in its essential form as an order and equivalence structure on grammars. At this level, a constraint is no longer a function penalizing concrete linguistic structures and mappings, but a more abstract order and equivalence structure that we call an EPO, an 'Equivalence-augmented Privileged Order'. The collection of the EPOs, each one representing a single constraint, forms the MOAT, the 'Mother of All Tableaux'. The unique MOAT of a typology is instantiated in every violation tableau that gives rise to that typology.

With this new characterization of 'typology' in hand, we can pose and answer fundamental questions about the structure imposed by OT on its grammars.

(1) **Typological Status**. Since a typology must have a well-formed MOAT, we can assess whether a given collection of grammars constitutes an OT typology. Simply dividing the set of all rankings into individually well-formed grammars is not guaranteed to produce a legitimate typology. Failures are detected by the appearance of cycles in the EPO graphs of the MOAT. Cycles indicate that it is impossible to realize the EPOs as OT constraints assigning violations in a consistent manner. Concomitantly, we can determine which VT representations are equivalent in the sense that they yield the same typology.

(2) **Classification**. Within a typology, MOAT structure determines whether a collection of grammars can be classified together as a kind of super-grammar, one that retains their shared linguistic patterns while abstracting away from their differences. This contributes to the foundations of the Classification Program of Alber & Prince (2015, in prep.).

(3) **Representation**. The MOAT arises from a notion of adjacency between constraint orders, which has a natural geometric interpretation. Each typology is associated with a geometric figure that represents the relations between its grammars: the *typohedron*. Super-grammars appear as regions on the typohedron. The MOAT brings out symmetries between constraints, and these appear on the typohedron as symmetries between super-grammar regions.

The argument proceeds in both concrete and abstract terms. We pursue the main line of analysis by examining the Elementary Syllable Theory (EST) of Prince & Smolensky, which presents the basic issues accessibly and allows for thorough application of the ideas and techniqes developed here. We also look at instructive typologies that are not as obviously rooted in language-based issues. Proceeding more abstractly, we provide formal analysis and proofs of assertions. In investigations of this nature, where broad claims are advanced, it is not possible to rest on examples, and we have introduced formal apparatus and methods of proof that allow us to state and establish claimed results. Not every reader will wish to work through every proof, but the leading ideas are built from the common vocabulary of linguistic analysis and worked out through concrete examples, so that they should be accessible to interested readers in essence and in detail.

The flow of discussion runs like this. We begin by fixing the fundamental notions of OT that play a role throughout, and from that base, move on to review the three key problems we address, enumerated above. We sketch their solutions in terms of the MOAT (§0).

The familiar typology of Elementary Syllable Theory (EST) is used as our touchstone and stalking horse, so we take some care in laying it out (§1). The concept of the MOAT is then developed through analysis of the EST (§2).

The key notion of the *Border Point Pair* is introduced with reference to the grammars of the EST (§2.2). A Border Point Pair consists of two linear orders that belong to different grammars and differ minimally: they are the same except for a single local transposition of two constraints. From its simple internal structure, order and equivalence relations between grammars emerge, populating the EPOs of the MOAT, which completely determine the possible numerical representations of a typology in VTs.

With the fundamental concepts in place, formal analysis begins (§3). The MOAT is related to another structure derived from the set of all VTs giving rise to a typology. From this relation, we deduce the main result: that the MOAT uniquely characterizes a typology (§3.2-3). We go on to examine the logical structure that inheres in Border Point Pairs, introducing the *ERCoid*. The ERCoid contrasts with the proper ERC in that it contains a fourth comparative value signifying complete lack of relational information (§3.4), providing the hidden microstructure from which three-valued OT emerges.

With the main results established, we return to the concrete via scrutiny of a self-contained subsystem of the EST, deriving its MOAT and showing how node merger in the MOAT parallels grammar union in the typology, illustrating how the MOAT functions as tool of typological analysis (§4).

Discussion then takes a mildly abstract turn toward the issue of coexistence of grammars within a single typology. Examples are presented which illustrate how the MOAT determines the typological compatibility of grammars, and how MOAT structure relates to the join of grammars into super-grammars (§5).

We conclude with a discussion of the remarkably well-behaved geometry of OT typologies, which makes graphically concrete the minimal-change relation inside the Border Point Pair. We show how a typology is represented on the permutohedron, a figure with a linear order at each vertex, and how this representation reduces to the *typohedron*, a structure in which each vertex represents an entire grammar. We complete the discussion by offering proofs of several striking results announced by Jason Riggle, showing that when distance is defined between between linear orders, grammars are convex regions analogous to the disks and balls of the familiar Euclidean world (§6).

The narrative arc thus proceeds from the essential premises of OT to an examination of its central object, the typology. We aim to find and relate ways of characterizing the typology that open up its properties to analysis. Multiple perspectives are explored, tied together by the the properties and uses of the MOAT.

Contents

0 Introduction & Overview

A *FACTORIAL TYPOLOGY* COMES INTO EXISTENCE whenever we specify an OT system as a set of constraints and the candidates they evaluate. The content of the typology follows from these assumptions, independent of whether we endeavor to familiarize ourselves with it. Experience tells us that a typology comes with ways of classifying its languages in terms of linguistic patterns and their correlated ranking relations. Again, this is a fact of logic, reflecting structures that inhere in forms and rankings.

The Basic Syllable Theory of Prince & Smolensky (P&S:1993/2004:p.105-118), for example, defines a certain set of admitted inputs, outputs, candidates sets, and constraints. The resulting factorial typology of the system distinguishes languages in terms of output types like 'onset required' and 'coda allowed' as well as by characteristic input-output relations of epenthesis, deletion, and faithfulness, each of which follows from specific ranking conditions playing out over the admitted forms (P&S:116). In the same way, a stress typology may predict dozens or even hundreds of languages, but still categorize them neatly into 'iambic' and 'trochaic', 'left-aligned' and 'right-aligned', or into more complex classes that devolve from its defining assumptions. With this in mind, Alber & Prince (2015) launch the Classification Program for typologies, studying a family of related stress typologies, developing a systematic analysis of how ranking relations determine the distribution of structural categories. Resolving these relations is essential if we wish to understand how the fundamental assumptions of OT, through constraint interaction, organize data into principled classes.

Here we identify and study key formal properties that underlie the linguistic macro-structure imposed by OT. We identify an invariant that is shared by all violation-tableau representations of the same typology: the MOAT ('Mother of all Tableaux'), and which therefore gives status to the typology as a formal object independent of its implementation in linguistic substance. The MOAT contains the linguistically relevant order and equivalence information that must be present in *every* violation-tableau (VT) that generates the same typology. Composed of numerical penalties assigned by the constraints to particular candidates, a given VT contains values that may or may not determine crucial ranking relations. The MOAT generalizes away from relational artifacts introduced by the use of numbers. It represents each constraint more abstractly, in a way that records only the information relevant to ranking in the grammars of the typology. Each such representation of a constraint we call an EPO ('Equivalence-augmented Privileged Order'). The MOAT of a typology, composed of EPOs, one for each constraint, is the abstract analog of a concrete VT composed of constraint columns containing numbers. From the MOAT, we can determine when two different representations of a typology are equivalent, whether a given collection of grammars forms a typology, and how the grammars of a typology can be classified together in all the ways that respect their typological status.

We aim to keep both the concrete and the abstract in view as we proceed: the MOAT concept is developed through particular typologies with a firm and familiar linguistic basis. The principal burden will be borne by Elementary Syllable Theory (EST), the version of the Basic Syllable Theory (BST) which unifies the two anti-insertion faithfulness constraints f.depV and f.depC into a single constraint f.dep (P&S:106). A simplified stress typology from Alber & Prince (2015) will also be called on.

We begin by laying out and refining the key concepts of OT (§§0.1-2). With the foundation in place, we state the three fundamental questions that the MOAT answers (§0.3). We then see in overview how the MOAT answers those questions (§§0.4-6).

The body of the work begins with a specification of the EST (§1). The principles of MOAT construction are then developed concretely in the EST context (§2) and the properties of the MOAT are established through formal analysis (§3). We then revisit the EST to analyze in systematic detail the subtypology that deals with the fate of consonants (§4). Turning to a pair of somewhat more abstract cases, we examine how MOAT structure restricts the cooccurence of grammars within a typology (§5). We conclude by developing a geometrical perspective that illuminates aspects of the MOAT concept and the intrinsic organization of OT typologies (§6), introducing an object, the *typohedron*, that carries the geometric relations between grammars.

0.1 Languages, Grammars, Typologies

In linguistic analysis, a factorial typology is calculated from a collection of candidate sets, each from a different input, each giving information about the treatment of the various structural issues implicit in the languages of the typology. Let us clarify the concepts that underlie this practice (see Prince 2015a for development).

First, all typological discussion takes place within the context of a fully-specified OT system *S*. This requires definitions of $\textsc{Gen}_S$ and $\textsc{Con}_S$, sufficiently precise to delimit these objects without ambiguity. The most familiar example may be the Basic Syllable Theory (BST, P&S:115ff). $\textsc{Gen}_{BST}$ and $\textsc{Con}_{BST}$ are specified in P&S:106-111. Observe that $\textsc{Gen}_S$ can be given by any mode of definition and need not involve a procedure of some sort.

In spelling out what we mean by an OT *system*, we cast the net wide, in anticipation of casting it yet wider. We count as a system $S = \langle \textsc{Gen}_S, \textsc{Con}_S \rangle$ any specification whatever of candidates and candidate sets ($\textsc{Gen}_S$), and any specification of a constraint set ($\textsc{Con}_S$) which characterizes each constraint as a function from candidates to {0, 1, 2,… }, the set of nonnegative integers. This way of defining a system we will call 'Concrete OT'

(COT), because it deals with candidates and the constraints that evaluate them. We diverge somewhat from one form of familiar usage, in that we regard such an *S* as a closed, fully articulated formal system and not as a piece of a larger something that is only partially defined (e.g. 'human language'). This allows us to say true things about *S* that are not contingent upon other things not specified or specifiable. The Basic Syllable Theory, for example, stands on its own and is not to be understood as a fragment of something else, a brick in a monumental edifice perpetually under construction. To advance beyond it, we propose other systems which may differ from it in various ways. On this view, the project of developing a theory is carried out by analyzing a growing body of limited, inter-related, well-defined *systems* rather than by programmatic conjecture in which large-scale ambitions dominate and obscure the discourse.

0.1.1 Concrete OT

The conceptual infrastructure we need involves the notions *ranking*, *optimality*, *language*, *grammar*, *typology*. It will prove worth our while to be clear about these five basic notions. We'll spend the most time here on the central notion of a *grammar*, which has perhaps received the least attention in the literature.

☐ By ***a ranking*** we will always mean a single linear order on the entire set $\textsc{Con}_S$. As usual, the notation $C_j \gg C_k$ means that C_j *dominates* or *is ranked above* C_k. To specify a set of linear orders G of which this holds, we may write $C_j \gg_G C_k$. The subscript G is customarily omitted when its reference is clear. The global structure of the system is determined by the set of all the linear orders it provides, which we denote Ord($\textsc{Con}_S$).

☐ The notion of ***optimality*** distinguishes certain candidates from others, given a ranking. This is defined within a *candidate set* ('cset'), a collection of competing candidates. Each candidate of S is associated by $\textsc{Gen}_S$ with a candidate set.

Optimality can be defined concisely this way: a candidate *q* is ***optimal*** in its candidate set K with respect to a ranking λ, if and only if for every candidate *z* in K, the highest-ranked constraint C in λ which distinguishes *q* and *z* by assigning different values to them is one for which the value assigned to *q* is less than the value assigned to *z*. For a more leisurely and analytical approach taking three steps, see Prince 2015a.

Optimality depends only on the values assigned by constraints and is blind to all structural differences that the constraints do not evaluate. Let's use the term *violation profile* to refer to the entire collection of values assigned to a candidate by the constraints of $\textsc{Con}_S$ (Samek-Lodovici & Prince 2005:1). If two candidates have identical violation profiles, there is no constraint on which they differ and they are indistinguishable with respect to every ranking: if one is optimal, so is the other. Similarly, if two candidates have *distinct* violation profiles and one is optimal under a given ranking, the other cannot be optimal on that ranking, because there must be at least one constraint on which they differ, and one of those must be highest-ranked, because a ranking is a linear order.

☐ A ***language*** of *S* is the collection of all optima for some ranking λ, drawn from every candidate set admitted by $\textsc{Gen}_S$. When we wish to emphasize its status as a collection of linguistic objects, we will use the term *extensional language*. Each linear order on $\textsc{Con}_S$ is thus associated with an extensional language. As is well known from descriptive practice, and as will be abundantly exemplified below, it often happens that more than one linear ranking delivers the same optima: in such a case, multiple (linear) rankings each yield the same extensional language.

☐ This leads directly to the notion of a ***ranking grammar***: the collection of all rankings that produce the same extensional language. A *ranking grammar* is characterized by an ERC set, yielding an ***ERC grammar***.

☐ A ***typology***, construed extensionally, is the collection of all the languages of a system. Since each language has a unique grammar associated with it, a typology may also be understood as the collection of all grammars of a system. This is the conception that we regard as truly fundamental. Since grammars are sets of rankings, a typology in the grammatical sense is a collection of disjoint sets of rankings which exhaust the set of all rankings: a partition of Ord($\textsc{Con}_S$). If the grammars are viewed as set of ERCs, then a grammatical typology is a collection of mutually inconsistent ERC sets.

Recall that an *Elementary Ranking Condition* or ERC (Prince 2002a,b) is an expression derived from the comparison of two candidates, typically denoted $[q{\sim}z]$. The ERC gives the ranking requirements that must prevail for the first candidate, *q*, to best the second, *z*, in their competition. The ranking condition associated with such a vector of W, L, and *e*'s — the 'Elementary Ranking Condition' proper — requires that *some* constraint assessing W of the pair dominates *every* constraint assessing L of the pair. Under any ranking satisfying this condition, the highest-ranked constraint that distinguishes the two competitors will favor the first, in the sense that it will assign fewer violations to the first than to the second. An ERC is represented by a sequence or 'vector' of characters W, L, *e*, one for each constraint in $\textsc{Con}_S$. The character W indicates that the constraint favors the first of the competitors; L indicates that the constraint favors the second; and *e* indicates that the constraint does not distinguish them by virtue of assigning both the same value.

Here's an example of pairwise competition from the syllable theory investigated below. The order of columns in a tableau is *not* presumed here or elsewhere in this paper to reflect a ranking order. Here and throughout, the brackets […] indicated that the enclosed string forms a syllable. Indices mark input-output correspondence. For visibility, an epenthetic segment is shown unsubscripted, underlined, and blue.

(1) Faithful vs. Epenthetic mappings in Elementary Syllable Theory: VT.

Input	Output	m.Ons	m.NoCod	f.dep	f.max	Remarks
V_1	a. $[V_1]$	1	0	0	0	*faithful*
	b. $[\underline{C}V_1]$	0	0	1	0	*epenthetic*

If the faithful map (1a) is desired to be better than the epenthetic map (1b), the following ERC emerges, presented as a 'comparative tableau' (CT).

(2) When Faithful bests Epenthetic: CT

a ~ b	m.Ons	m.NoCod	f.dep	f.max	Remarks
$\langle V_1 \rightarrow [V_1]\rangle \sim \langle V_1 \rightarrow [\underline{C}V_1]\rangle$	**L**	*e*	**W**	*e*	*faithf. ~ epen.*

In this simple case, with one W and one L, the interpretation of "some W dominates every L" is just f.dep$\gg$ m.Ons. In each of the 12 rankings that meet this requirement, the faithful candidate (1a) bests the epenthetic candidate (1b), as desired.

If we swap desired winner and desired loser, we obtain the following:

(3) Epenthetic bests Faithful: CT

b ~ a	m.Ons	m.NoCod	f.dep	f.max	Remarks
$\langle V_1 \rightarrow [\underline{C}V_1]\rangle \sim \langle V_1 \rightarrow [V_1]\rangle$	**W**	*e*	**L**	*e*	*epen. ~ faithf.*

Ecologically, as any practitioner can verify, ERCs with many Ws and Ls are common, reflecting the complexities of explanation when many factors are involved.

Optimality may also be defined in terms of ERCs. A candidate *q* is optimal under some ranking λ if and only if λ satisfies every ERC that compares *q* with another candidate in its cset.[1] The ERC notion supports a full theory of ranking in OT and the logic of ranking plays out in manipulation of ERC vectors. See Prince 2002ab, 2006, 2008, 2009, as well as Brasoveanu & Prince 2005/2011, for discussion.

An OT grammar as defined here is a formal object known an *antimatroid*, a type of order structure that generalizes the more familiar partial order.[2] Just as we speak of the set of

[1] This alternate definition of optimality is equivalent to the one given above. For discussion, see "RCD -- the Movie." Follow the "Optimality" link on the TOC, continuing to the worksheet "Challenge".

[2] On the equivalence of ERC sets and antimatroids, see Riggle 2010:p.13, and Merchant & Riggle (2016) for proof. So what is an *antimatroid*? The most accessible characterization may well be this: a set of linear orders delimited by an ERC set.All partial orders are antimatroids; some antimatroids are not partial orders. If the Skeletal Basis or Most Informative Basis of the grammar (Brasoveanu &Prince 2005/11) contains an ERC with two or more W's, the grammar is properly antimatroidal and not representable as a partial order.

linear orders consistent with a given partial order as the *linear extensions* of that order, so may we speak of a ranking grammar G_R as the set of linear extensions of an ERC grammar G_E. With this in mind, we introduce the acronym *leg* (‘***l***inear ***e***xtension of a ***g***rammar’) to refer to a ranking, and we will speak of the ***legs*** or ***leg set*** of an ERC grammar to refer to the rankings associated with it. The notions ‘ranking grammar’ and ‘ERC grammar’ are equivalent in the following sense: every leg set of an ERC grammar is a ranking grammar; every ranking grammar is the leg set of an ERC grammar.

A point of usage: the term *grammar* is sometimes used in the literature to refer to a single ranking. For us, *grammar* always refers to either a *ranking grammar* — a *set* of rankings that in Concrete OT yields the same extensional language — or to an *ERC grammar*, its characterization by a set of ERCs. These are the linguistically significant objects. The single ranking is a poor candidate for special recognition, since it almost always contains artifacts: ranking relations that arise not from the data, but from the formal requirement of linearity.

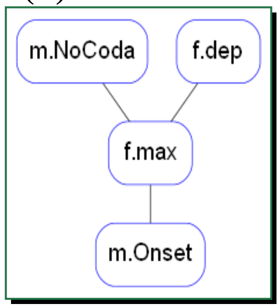
0.1.2 Abstract OT

Toto, I have a feeling we’re not in Kansas anymore.

Exactly as elsewhere in generative linguistics, the notion of a *grammar* disconnects us from the concrete. By shifting focus from the extensional language composed of concrete optimal candidates to the set of rankings that yield those optima, we characterizee the extensional language *intensionally*, in the terminology of Alber & Prince 2015. This marks a consequential step into abstraction.

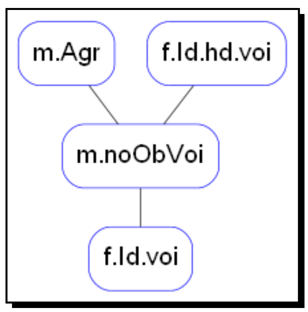
Two languages in completely unrelated Concrete OT systems S_1 and S_2 may have structurally identical grammars, in the sense that there’s a 1:1 mapping between their constraint rankings, even though their concrete content diverges wildly. For example, in the Elementary Syllable Theory examined below, there is a ranking grammar that consists of the two linear extensions, legs, of the following partial order:

(4) Grammar of (C)V.del from EST



Given GEN$_{EST}$, this yields the extensional language (C)V.del, in which all outputs consist of open syllables that may or may not have onsets, and in which the output pattern is achieved through deletion of refractory underlying material. Compare this with the grammar of voicing systems like that of Polish, from Lombardi's voicing typology (Lombardi 1999). The linguistic generalization is that obstruents in clusters take on the underlying voicing value of a cluster's head, if it has one, and are otherwise voiceless.[3] This is a substantively different pattern, but the ranking structure is identical.

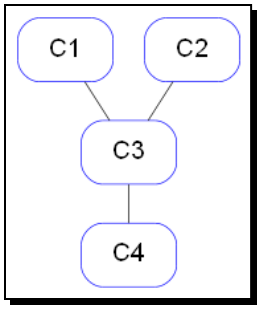
(5) Polish in the typology of Lombardi 1999.



Both exemplify the following Y-shaped ranking structure, appropriately identifying constraints across systems:

[3] To cover all the cases, think of a 'cluster' as consisting of *one or more* obstruents. A cluster is 'headed' if it precedes a vowel; the prevocalic obstruent is its head.

(6) Y-ranking



An ERC Grammar of (6), exactly mirroring those of (4) and (5), is given below as a CT.

(7) ERC Grammar of Y

C_1	C_2	C_3	C_4
W	*e*	**L**	*e*
e	**W**	**L**	*e*
e	e	**W**	**L**

Grammars like the three just cited must share all properties that follow from their identical ranking structure. This is true even if they are relevant to entirely incommensurable domains — one may be about stress, the other about consonant voicing, as here, or their domains may even be further apart: managing an ecosystem, making a legal decision, or choosing dessert. This realization leads to the notion of **Abstract OT** (AOT): the study of ranking structures in themselves, without calling on concrete candidates and constraints to generate them.

□ As with *grammar*, so with *typology*. The ***extensional typology*** XT(*S*) of a given concrete *S* is the collection of its extensional languages. The ***intensional typology*** IT(*S*) is the set of grammars of the extensional languages of XT(*S*). The intensional typology provides the natural setting for the study of Abstract OT. AOT is possible because both grammars and typologies are definable as formal objects. In AOT, any ERC set is an ERC grammar, and any set of rankings that can be exactly delimited by an ERC set is a ranking grammar. A typology, as we will see, is a set of grammars with a MOAT.

Concrete OT relies on GEN$_S$ and CON$_S$ to produce the violation tableaux (VTs) that determine the intensional grammars. Abstract OT starts with the VTs, without regard to possible origins, and explicates their properties by virtue of the way EVAL works on VTs. EVAL sees only violation profiles. Abstract OT therefore needs only arrays of integers (or entities ordered like them) — VTs — to generate the grammars that are its province. From this point of view, a Concrete OT system S = ⟨Gen$_S$, Con$_S$⟩ gives rise to XT(S) and IT(S), which always instantiate an Abstract OT typology τ. All concrete T that instantiate an abstract τ will be intensionally equivalent**,** in the sense that their ranking grammars are isomorphic**,** whether they are about prosody, conflict of laws, or puff pastry. Everything that is true of τ at the abstract level must also be true of any of its instantiations. We can access the properties of Abstract OT at the level of formal structure, or through its instantiations; we'll do both here.

This distinction between a concrete system and the abstract principles that inform it pervades all forms of what Chomsky has called 'rational inquiry' (see Chomsky 1988:41, for example). Perhaps mere ubiquity has diminished its role in familiar linguistic discourse: what we are doing here amounts to little more than insisting on it. To encourage a sense of the relation, we cite a couple of striking examples. The principles behind the efficient stacking of oranges at the produce stand are the same as those behind designing an error-free code for transmitting pictures of distant planets.[4] Closer to home, the analyst Nate Silver was 15-minutes-of- famously able to jump from baseball to politics in time for the 2008 presidential election, vanquishing those political observers who rely on intuition shaped by lengthy experience, and repeating the feat in 2012. Why was this possible? Because the relevant abstract principles of statistics are instantiated in both, inescapably. To understand either baseball or politics, one must turn away from the balls, bats, and ballot boxes to grasp the disembodied abstractions that shape the outcomes.[5] There is, therefore, every reason to welcome the distinction between abstract and concrete.

The grammars that make up an intensional typology may be understood as either ranking grammars or ERC grammars. The ranking grammars of a typology *partition* the set of all possible rankings: they divide the set of all rankings into non-overlapping subsets.[6]

A ranking grammar in Concrete OT is the set of all linear orders, legs, that give the same optima — those of the extensional language for which it is the grammar. Each leg determines a language in its entirety. From this, it follows that a given leg can belong to only one grammar. To see this, notice that if grammars share a leg, they share the extensional language, and they are therefore the same grammar, sharing *all* legs. Furthermore, every ranking must belong to some grammar. This means that the study of typologies is the study of certain kinds of partitions.

On the ERC side, the grammars of a typology are pairwise inconsistent: for any two grammars, each satisfies some ranking condition that contradicts a condition on the other, distinguishing their languages and guaranteeing that no leg can satisfy the ERC sets of both. Two grammars may share many ERCs, as well — for example, one language may be iambic, the other trochaic, while they are identical in every other respect. This means

[4] Discussed in Wolchover 2013.

[5] When other abstract principles are crucially involved, as in economics and climatology or even politics, basic statistics cannot expected to suffice. On these points see e.g. Krugman 2014 and Atkin 2014. And, of course, Silver 2016.

[6] A partition of a set S is a collection of pairwise disjoint nonempty subsets of S that unions to S.

that the study of typologies is the study of certain families of ERC sets that cannot be conjoined together without incurring contradiction.

Just as with individual grammars, the notion of an intensional typology sets us loose from the *materia linguistica* that it aims to explicate. Our focus is to understand central properties that all concrete typologies must have by virtue of their being instantiations of abstract typologies, revealing the large-scale consequences of the way optimality is defined in OT.

0.2 One Tableau Suffices

An extensional language is *finitely determined*: even when it contains an infinite number of optima, a well-chosen finite sample will fix all further choices. There may, of course, be many such determinative finite samples. Even when a single input yields, under $\text{G}\textsc{en}_S$, an infinite number of violation profiles, only a finite number of them can be optimal. Finite determination follows from the finitude of the number of rankings, which provides a sharp upper bound for the number of grammars. Any finite collection of finite candidate sets that determines an extensional language will also determine the grammar of that language. Grammars are therefore also finitely determined. The study of grammar is the study of a finite sets of rankings, or of finite sets of the ERCs that define them, deriving from a finite set of finite violation tableaux.

So: finitude everywhere outside $\text{G}\textsc{en}_S$. But we can go further. From these observations, as well as from the empirical practice of Concrete OT, one might gather the impression that a number of different candidate sets, perhaps large (if finite), will be generally required to determine a factorial typology. In some cases we can come up with a single input of sufficient complexity to generate the entire typology, one that manages to concatenate or otherwise contain all the relevant configurations. But there's no guarantee in Concrete OT that this must happen. For example, in stress theory it is typically the case that inputs of different lengths must be examined — odd and even, or perhaps a single syllable as well as longer lengths. But we will never be able to construct a string that is both odd and even in length, or both monosyllabic and polysyllabic.

If we attend to the *grammars* rather than to the languages, positioning ourselves within Abstract OT, the need for multiplicity of candidate sets disappears. Prince 2015b shows that any intensional typology — a collection of grammars, not languages — can always be characterized by a single violation tableau, which is of course finite.

Let's call such a typology-generating tableau a 'Unitary Violation Tableau' or UVT. We require that the rows of a UVT be distinct and that each give rise to a distinct grammar of the typology. Because the rows of a UVT represent languages in the typology, none can be harmonically bounded. We may think of a UVT as containing a single abstract candidate set. Each candidate — each violation profile occupying a row of the UVT —

is associated with an entire grammar. Comparing that row to the other rows yields the grammar of the associated abstract candidate. In Abstract OT, an abstract typology is any collection of grammars produced from a UVT. Any VT in which all the rows are distinct possible optima competing against each other can be interpreted as a UVT. If the rows of a UVT U give rise to the grammars of a typology T, we say that U *instantiates* T, and we will write T_U to denote it.

(8) **Definition**. **Unitary Violation Tableau** (UVT). A Unitary Violation Tableau, abbreviated UVT, is a violation tableau in which each row, when taken as the designated optimum competing against the other rows, gives rise to a distinct grammar.

Here's an example, with arbitrarily numbered constraint columns and candidate rows.

(9) **Specimen UVT**

U	C_1	C_2	C_3
L_1	2	0	0
L_2	1	0	1
L_3	0	1	1

There is no implication as to what structures might be involved in instantiations of the typology; they needn't even be linguistic. And there is no sense of structure-detecting constraints that might produce the numbers, which are *just there*. Nevertheless, grammars are generated by the usual definition of optimality. If we wish for example to assert L_3 as optimal, the result is the following collection of ERCs, presented as a comparative tableau (CT).

(10) **CT** from (9) with L_3 asserted as optimal

CT:L_3	C_1	C_2	C_3
$L_3 \sim L_1$	**W**	**L**	**L**
$L_3 \sim L_2$	**W**	**L**	*e*

This CT tells us that one of the grammars of the intensional typology, the ranking grammar of L_3, has the form $G_R(L_3) = \{C_1 \gg C_2 \gg C_3, C_1 \gg C_3 \gg C_2\}$, which is exactly the set of two rankings that satisfy the requirement "C_1 dominates both C_2 and C_3." The concise ranking condition is given by an ERC grammar G_E = {WLL}, which contains the non-redundant content of the CT (10).[7]

[7] The original set {WLL, WLe} is also an ERC grammar of the same system. The ERC WLe is entailed by WLL by virtue of "L-retraction," the ERC manipulation that mirrors the inference rule "and-out", p&q$\models$ p. See Prince 2002:7.

To bring this example closer to the common experience of dealing with many candidate sets at once, we note that the very same grammar is produced by the following pair of csets, which indeed deliver the entire typology.

(11) **Two abstract csets yielding the typology T**.

T	cand	C_1	C_2	C_3
cset A	a_1	1	0	0
	a_2	0	1	0
cset B	b_1	1	0	0
	b_2	0	1	1

The grammar of L_3 is obtained by choosing $a_2 \in A$ and $b_2 \in B$ as optima. As the reader may verify, the comparison $[a_2 \sim a_1]$ delivers the ERC WLe, and the comparison $[b_2 \sim b_1]$ delivers WLL, exactly recapitulating the ERCs of the UVT (10). Numerous other equivalents exist as well: for example, if C_2 is modified by changing the value 1 to 0 in cset B, the same grammars will result.

This simple example illustrates the much broader fact that any intensional typology whatever, constructed from no matter how many distinct csets, can always be exactly represented as a single VT, with just one abstract cset (Prince 2015b). Any row of such a UVT, when chosen as the asserted optimum, will generate the grammar of a language in the typology. **Therefore, understanding the grammatical structure of typologies reduces entirely to the study of typologies generated by a single UVT.**

The preceding notions allow us to define precisely what we mean by 'typology'. Given the entire set of linear orders on a constraint set $\text{C}_{\text{ON}_\text{S}}$, denoted Ord($\text{C}_{\text{ON}_\text{S}}$), a typology is a partition of Ord($\text{C}_{\text{ON}_\text{S}}$). But not every such partition qualifies as a typology, either because some block of the partition is not a grammar, or because the grammars can't coexist in the same typology (§5 below). We don't need to clarify these failures before we define our object of study. A typology, in our sense, is a partition of Ord($\text{C}_{\text{ON}_\text{S}}$) which can be derived from a UVT. This gives us a place to stand, which we will leverage to unfold the structures that inhere in the definition of optimality.

(12) **Definition. Typology**. Given a set of constraints $\text{C}_{\text{ON}_\text{S}}$, a partition of the set of all orders on $\text{C}_{\text{ON}_\text{S}}$, is a typology iff there is a UVT U, with columns that correspond 1:1 to the constraints of $\text{C}_{\text{ON}_\text{S}}$ and rows that correspond 1:1 to the grammars of T, such that each block in the partition T is the ranking grammar of a row in U.

0.3 Problems, problems, problems

Three basic structural questions arise from the abstract characterization of a typology as a set of grammars and from the availability of UVT representations. Here we outline the questions and indicate the answers. We then give an overview of the MOAT concept and how it resolves them.

Problem 1. **Typological equivalence of UVTs**.

Any intensional typology can be represented as a UVT. But optimality depends on the relations between the entries, not their numerical values. Consequently, it will always be the case that many UVTs, with different numbers in them, produce the same intensional typology. They are *typologically equivalent*. What do these UVTs have in common?

Status: solved by the MOAT. The MOAT of a typology T identifies exactly those order and equivalence relations that each constraint must impose to produce the grammars of T. Every intensional typology has one and only one MOAT. And every MOAT is associated with one and only one intensional typology. This is the force of Theorem (160), §3.2. The order and equivalence relations in a typology's unique MOAT determine all of its possible numerical representations as a UVT.

Problem 2. **Compatibility of Grammars within a Typology**

What conditions on mutual compatibility are imposed by a typology on its constituent grammars? An intensional typology turns out to be more than a set of pairwise disjoint ranking grammars, more than (equivalently) a set of pairwise inconsistent ERC grammars, exhausting Ord($\text{C}\small{\text{ON}}_S$). We must therefore ask: what conditions must a typology meet that a general partition of the ranking set doesn't have to?

Status: solved by the MOAT. A partition of the ranking set is a typology if and only if it has a well-formed MOAT (§3.3). Partitions that fail this condition may consist of grammars (§5). Abstractly, this gives a formal characterization of the notion 'typology', paralleling the way that the notion 'grammar' is definable as an antimatroid, or as the set of linear extensions of an ERC set. Concretely, this finding provides a valuable tool which we will immediately make use of in dealing with the third problem.

Problem 3. **Classification of Languages**

Classification and ranking. ERC Grammars in a typology are pairwise inconsistent because they contradict one other on *some* ranking requirement. But even as grammars differ, they may also share other requirements, leading to groupings of languages that have ranking restrictions in common. The nature of these groupings is the central focus of Alber & Prince 2015, under the Classification Program articulated there. In Elementary Syllable Theory (EST), for example, which will be studied in some detail below, $\text{C}\small{\text{ON}}_{EST}$ contains the markedness constraints m.Ons, m.NoCoda and the faithfulness constraints f.max and f.dep. One language may require f.dep≫f.max, another f.max≫f.dep. Of the 8 languages in the typology, four share the first requirement, and four share the second. This kind of intensional patterning is native to Abstract OT; it can be discerned without any grasp of concrete particulars, and will of course be inherited in the concrete instantiations of an abstract typology. Considered intensionally, EST is an instance of an abstract 4-constraint typology that has certain patterns of ranking relations defining its grammars.

On the extensional side, the *languages* of a typology will share and be distinguished by patterns of linguistic structure. In Elementary Syllable Theory, for example, one may observe that some languages require onsets in every syllable, and that others allow onsetless syllables under certain conditions; that some admit deletion and others epenthesis in the input-output mapping, and so on. In a typical stress typology, languages will differ in the size and shape of admissible feet: of the various types iambic, trochaic, unary, binary, some may be disallowed, some limited to certain positions; and so on. The Classification Program aims to explicate the ways that intensional groupings, based on shared and contrasting ranking patterns, impose a classification on the languages of a typology into extensional types.

Consider the intensional contrast in EST between languages requiring f.dep≫f.max and those requiring f.max≫f.dep. Concretely, the first ranking condition identifies the languages where deletion rather than insertion appears in optima to avoid certain structural configurations; the second identifies those with insertion, not deletion. A complete classification in the Alber-Prince sense will reconstruct the entire typology in terms of the interactions of the ranking conditions that define its classes.

Such a ranking-based classification not only gives insight into the functioning of the system $S = \langle Gen_S, Con_S \rangle$; it also resolves ambiguities that the plethora of extensional correlations may leave open. Going the other way, the possible extensional classes in an instantiation, determined by structural considerations, limit ambiguities in intensional analysis. A complete classification mates intensional with extensional, giving an account of the way that ranking structure classifies the languages of the typology XT(S), relating theory (ranking patterns) to the data (patterns of linguistic structure). As Alber and Prince observe, it *is* the “OT analysis” of the languages of a typology.

A class of grammars is associated with the union of the sets of rankings that constitute the individual ranking grammars of the class. From this perspective, a *class* of grammars is viewed as just another set of rankings.

Our goal in approaching Problem 3 is to resolve a fundamental formal question that underlies the Classification Program. In Abstract OT, any ERC set defines a grammar. Therefore any *class* of grammars which can be delimited by an ERC set is itself a *grammar*. In our EST example, the ranking condition ‘f.dep≫f.max’ is encodable as the

ERC set {WL*ee*}, taking the constraints f.dep and f.max as the first two in the listing of CON_{EST}. Ranking-wise, this abstract grammar — this *class* — contains 12 of the 24 linear orders on CON_{EST}. Every ranking belongs to the grammar that shares its defining ERC set and to no others. There is no guarantee that abstract grammars of this sort will have a concrete instantiation in terms of concrete candidate sets made available by Gen_S for any particular system S. *But they are grammars nonetheless*. When a class of grammars is characterizable by a set of ERCs, we will call it a ***grammatical class***, meaning that it has formal status as a grammar. We can think of such a grammar as a generalization of the grammars that it is constructed from.

In EST, the insertion class is likewise given by an ERC set, namely {LW*ee*}, representing the condition 'f.max≫f.dep'. It is therefore also a grammatical class. The two classes taken together have an additional property: not only do they partition the set of rankings of CON_S into two disjoint subsets, each containing 12 rankings; it also happens that the partition they impose is abstractly a *typology*. Recall that an abstract typology is any collection of grammars derivable from a UVT. These two classes of EST, which we may call 'Ins' and 'Del', are easily shown to meet the definition (12), which requires the existence of a witnessing UVT.

(13) Insertion / Deletion in the EST: the 2 language Abstract Typology "Ins/Del"

T:Ins/Del	f.dep	f.max	m.Ons	m.NoCoda	Remarks
L_1	0	1	0	0	L_1 opt ⇔ f.dep ≫ f.max
L_2	1	0	0	0	L_2 opt ⇔ f.max ≫ f.dep

It may be directly verified that choice of L_1 (Del) as optimal yields the deletional grammar{WL*ee*}, while choice of L_2 (Ins) yields the insertional grammar{LWee}, as promised. Within EST, the classes consist of all those grammars that delete problematic C and all those that epenthesize to support it syllabically.[8]

This example illustrates an entirely general phenomenon. Just as the amalgamation of several grammars can lead to an abstract *grammar* that represents their shared properties, so too can the amalgamation of grammars within a specific typology lead to another abstract *typology* which represents not just the properties of various sets of grammars, but also the ways that they are distinct from each other. In example (13), we have constructed the abstract typology "Ins/Del" which generalizes EST by amalgamating all inserting

[8] In this simple case we can find an input from GEN_{EST} that produces the result, namely /C/. If we restrict GEN_S to provide just the one input /C/ while setting $CON_S = CON_{EST}$, then S will have exactly the typology (13), which divides the set of rankings into two classes. In this case, the cset derived from the input /C/ has only two possible optimal outputs, $[CV]_\sigma$ with epenthesis, and ε the empty string, arrived at by deletion. This concretely instantiates the abstract typology Ins/Del. In this fortuitous case, the concrete leads directly to the abstract. Generically, more subtle tools are needed.

languages (pooling their legs) into one super-grammar and all deleting languages into the other. In cases like this, where amalgamations result in a well-formed typology, we will call any resulting generalized grammar a ***typological class***. What our example shows, in these terms, is that in the EST, there are typological classes ‘insertor’ and ‘deletor’, each of which generalizes over a subset of the languages in the typology of EST, producing a generalized typology that expresses this classification. The generalized typology (13) classifies concrete EST along one of its structural dimensions.

With the notion of a typological class in place, we stand on the threshold of the first formal step in the Classification Program: obtaining the typological classes of a typology. To advance, we need to know when a collection of languages within a typology constitutes a typological class.

Our specific goal, then, is to answer the following question: under what conditions can several grammars in a given typology be amalgamated into a single more general grammar within a generalized typology?

Status: solved by the MOAT. Since every typology has a MOAT, amalgamation must produce a new MOAT. When it produces a structure that does not qualify as a MOAT, the resulting classes do not constitute a (generalized) typology.[9] When it does, they do.

With the three motivating questions now on display, we proceed to an overview of their treatment in the following sections (§0.4, §0.5, §0.6), introducing the MOAT concept in terms of a simple example. Our aim is to show how the MOAT is used to resolve the three questions. Details of MOAT construction will be pursued in the sections following, through a more complete contemplation of the EST.

[9] The details can be given in the following concise form. The MOAT is a set of order-and-equivalence structures, EPOs, each of which represents the essential content of a single constraint. A UVT relates language to language, grammar to grammar. The MOAT delimits every UVT that yields the same typology. An EPO is representable by a kind of augmented Hasse-like diagram which marks equivalence as well as order but, unlike Hasse diagrams, is not subject to transitive reduction among the orders. Whether or not a set of languages in a typology can be amalgamated into a typological class is determined by the structure of the EPO diagrams in its MOAT. The union of grammars into a class corresponds to merging their nodes in the MOAT to produce a modified graph of their relations. If merger produces a well-formed MOAT, then a generalized typology results, corresponding to that MOAT and consisting of valid typological classes that analyze the original typology. If not, then not. Graphically, as we will see shortly, this boils down to whether node merger introduces order-involving directed cycles (fatal) or retains the acyclic character of the EPOs in a legitimate MOAT.

0.3.1 Typological Equivalence

Problem 1. ***E Pluribus Unum***. Every grammatical typology comes from a UVT. But many numerically distinct UVTs deliver the same typology. There is an algorithm ('Minkowski summation', as shown in Prince 2015b) that produces a single UVT from any set of VTs. That UVT is provably equivalent to the entire original VT collection, in the sense that they have exactly the same grammars. We refer to UVTs that produce the same typology as 'typologically equivalent'.

(14) **Definition**. Typologically Equivalent. Two UVTs based on the same constraint set set are *typologically equivalent* if the typologies associated with each are identical.

A typology can typically be derived from many different collections of candidate sets, and the Minkowski sum algorithm will produce a ranking-equivalent UVT for each of them. Thus, from concrete considerations alone, given a typical unbounded linguistic object as our target, we are already guaranteed an unlimited number of UVTs grounded in linguistic fact. These differ numerically yet produce the same intensional typology. If we step away from concrete linguistic analysis, we will find many more UVTs — taken as arrays of integers — which produce any given typology. What is it that all these UVTs have in common which ties them to the same typology? The MOAT gives the answer.

Abstractly put, but with the specificity of an example, let's return to the specimen UVT of example (9) above. We ask which other UVTs produce the same ranking typology, and then see how their shared patterns of equivalence and order are represented in a MOAT.

(15) **Specimen UVT**

U	C_1	C_2	C_3
L_1	2	0	0
L_2	1	0	1
L_3	0	1	1

This yields a 3-grammar typology which we'll call 'T'. Certain numerical properties are accessible from experience with OT analysis: for example, we *cannot* change any of the 0's here, leaving the other numbers in place, without altering the ranking structure.[10] But further questions arise:

[10] Consider the total order $C_3 \gg C_1 \gg C_2$. This selects language L_1. If we modify the C_3 value of L_1 to 1, then this total order now selects L_3, which is no longer ejected at the first step of filtration. In the case at hand, as the reader may verify, none of the 0's may be altered without crucially altering the filtration patterns, and therefore the grammars of the typology. See Lemma (123) in §3.2.1, p. 56.

Q1. Consider the numerically-based order relations between languages in column C_1. Using familiar functional notation to denote the values assigned by the constraint C_1, we have $C_1(L_1) > C_1(L_2) > C_1(L_3)$. These relations arise because $2>1>0$. Any UVT with the same order relations in C_1 will yield the same treatment of candidates by C_1, and therefore the same ranking information. Must these order relations be respected in *every* equivalent UVT? Answer: **Yes**.

Q2. Consider the relation between L_2 and L_3 in column C_3: L_2 is treated as equivalent to L_3 with respect to C_3, because $C_3(L_2) = C_3(L_3)$. Any UVT in which L_2 and L_3 assume identical values, so long as they are greater than zero, will yield the same treatment of candidates, and the same rankings. But must this equality be respected in *every* typologically equivalent UVT? Answer: **No**.

We sketch here how all such questions can be comprehensively settled.

To begin with, and to dispel the air of pure formality, let's concretize within an OT system built from recognizable linguistic predicates.

Suppose, following Alber & Prince 2015, we simplify a familiar type of metrical theory of stress patterns by these steps: omit one foot-type constraint, eliminating the contrast between iambic and trochaic; allow only one direction of alignment, eliminating certain contrasts in foot-positioning; and flatten the prominence scale to distinguish only stressed from unstressed, eliminating the distinction between primary and secondary stress. We arrive at an OT system, which Alber & Prince call nGX.IL.[11] The 3 constraints in the system are Parse-σ, All-Feet-Left (AFL), and Iamb. The system may be defined concisely as follows:

(16) **GEN$_{\mathbf{nGX.IL}}$**. A candidate consists of an input and an output. Inputs are sequence of syllables σ, taken to be primitive units: σ^n, $i\geq 1$. GEN$_{\mathrm{nGX.IL}}$ pairs each input σ^n with all prosodic parses of *n* syllables. A prosodic parse of σ^n consists of a single Prosodic Word embracing the whole string, with the PrWd node dominating *at least one foot* and perhaps many feet (F). A foot contains one or two syllables, and has one head. There is no deletion or insertion.

NB: Syllables in words of length ≥ 2 may therefore be structured as unfooted children of PrWd, as long as there is a foot in the PrWd, but the monosyllabic input may not be footless.

[11] In the acronym, 'n' denotes that the definition of Iamb and Trochee are 'new' in that they each penalize not just the binary foot of opposite headedness but also the unary foot, which is therefore neither 'iambic' nor 'trochaic', as opposed to the paleoclassical conception in which the unary foot is both. G indicates that all foot positioning is controlled by Generalized Alignment. X indicates that all outputs must contain at least one foot. The IL suffix indicates the subvariety of nGX in which, of the foot type constraints, only Iamb appears, and of the alignment constraints only All-Feet-Left.

The three constraints of nGX.IL can be specified as follows:

(17) **CON$_{\text{nGX.IL}}$**

Parse-σ	*o	=	card{σ∈out(κ)| σ∉F}
Iamb	*[$_F$ σ′	=	card{ [$_F$ σ′ ∈out(κ)}
AFL	*(σ,F): σ…F	=	card{(σ,F)∈out(κ)| σ precedes F}

Here we represent the head of a foot as σ′. We use the familiar OT * operator to define constraints: *x:P(x) takes a candidate κ as its argument (left implicit) and returns the number of matches to the pattern P(x), running over all occurrences of P(x) in the candidate κ. We recruit '∈' to denote any relevant version of 'belongs to', and we write '…' to express linear precedence. The formulation of the alignment constraint All-Feet-Left, abbreviated AFL, adapts Hyde 2012.

In representing full prosodic parses, we use the convenient Alber & Prince string-based abbreviations employed in OTWorkplace (Prince, Tesar, and Merchant 2015). 'X' denotes the head of a foot,'u' denotes a nonhead child of F, and 'o' denotes a 'unparsed syllable', a child of PrWd not F. Hyphens demarcate feet and unparsed sylls.

Thus defined, nGX.IL is a Concrete OT system. Its typology is provably determined by the 5 syllable candidate set, which has 3 optima, listed here. (See Alber & Prince 2015 for the richer system nGX, which contains the symmetrical defined constraints Trochee and AFR. See Alber, DeBusso, and Prince 2015 for general characterization of the candidate sets that suffice to generate the whole typology.)

(18) **5σ cset nGX.IL**

nGX.IL					
Input	output	Parse-σ	Iamb	AFL	Class name
σσσσσ	-uX-o-o-o-	3	0	0	sparse
	-uX-uX-o-	1	0	2	weakly dense
	-X-uX-uX-	0	1	4	strongly dense

Comparison with the abstract specimen UVT (15) shows some numerical divergences. These do not affect the intensional typology generated, as may be readily calculated. Thus, nGX.IL provides a concrete instantiation of T.

Each of these optima corresponds to a structural class, labeled as follows:

- **Sparse** (sp), taking the form Fo^n, with one foot per word,
- **Weakly Dense** (WD), taking the from F^n(o),
- **Strongly Dense** (SD), taking the form F^n.

Optima in the Weakly Dense language are parsed to the fullest extent possible when feet are strictly binary. In a Strongly Dense language, all optima are fully parsed into feet, so that they display unary feet in odd-length words. This suggests, correctly, that we are looking at a useful analysis of the more complicated system in which the neutralized distinctions in foot type and directionality are reinstated. The 5 syllable candidate set does the work of finding a single, typology-generating VT, without further calculation. We may simply re-label it to produce a UVT for the system. We prefix the constraint names with ‘u’ for ‘unitary’, to emphasize that they no longer refer to the familiar functions defined above, as they now evaluate abstract candidates and not linguistic forms.

(19) UVT of nGX.IL with category names

U.nGX.IL	u.Parse-σ	u.Iamb	u.AFL
sp	3	0	0
WD	1	0	2
SD	0	1	4

Now we step away from the specifics of $\text{G}\textsc{en}_{\text{nGX.IL}}$ and $\text{C}\textsc{on}_{\text{nGX.IL}}$ and look for VTs that produce exactly the same set of grammars. The following VTs will be included among them. Note especially the differences in the u.AFL column, boxed, where the relationship between the non-zero values runs through all possibilities.

(20) Another UVT for T

U′	u.Parse-σ	u.Iamb	u.AFL
L_1'	3	0	0
L_2'	1	0	4
L_3'	0	1	2

(21) And another UVT for T

U″	u.Parse-σ	u.Iamb	u.AFL
L_1''	3	0	0
L_2''	1	0	2
L_3''	0	2	4

(22) And another UVT for T

U‴	u.Parse-σ	u.Iamb	u.AFL
L_1'''	3	7	0
L_2'''	1	7	2
L_3'''	0	18	2

We can sensibly compare the typologies of these UVTs because they all have the same constraints, applying to different but correlated objects, namely sp – L_1 – L_1' – L_1'' – L_1''' and so on. To see that U, U′, U″, and U‴, and U.nGX.IL all produce the same grammars, note first that u.Iamb in UVTs (20)-(22) imposes the same numerical relations on cognate languages as in UVTs (19) and (20). Since OT comparison works on order, not quantity, the relevant relations between the candidates are identical in each UVT. By contrast, the extensions of u.AFL all differ in U, U′, and U″. Were the differences typologically significant, they would distinguish the second and third languages from each other in some filtration. But this does not happen: no matter what the order, u.AFL never decides the pair L_2, L_3 or any of its primed cognates: the members of the pairs are either rejected together or separated by the top-ranked constraint in every leg of every grammar, as may easily be seen by inspection.

What, then, do all these typologically-equivalent VTs have in common?

To obtain the answer, we attend to those relations that have impact on selecting optima. Since the numerical values are integers, there are only two kinds of relations that can hold between them: equivalence (equality) and order (greater than, less than).

In u.AFL, concrete WD and its abstract cognates L_2, L_2', L_2'' stand variously in every possible relation with their competitors SD, L_3, L_3', L_3'' .

(23) **Numerical Relations within u.AFL**

U	(19)	u.AFL(WD) $<$ u.AFL(SD)
U′	(20)	u.AFL(L_2') $>$ u.AFL(L_3')
U″	(21)	u.AFL(L_2'') $<$ u.AFL(L_3'')
U‴	(22)	u.AFL(L_2''') $=$ u.AFL(L_3''')

Because all 4 UVTs are typologically equivalent, this unstable relationship must have no impact on the filtration of the candidate set.

To find stability, let’s examine the relations of sp and WD (first two rows) over the entire constraint set. The UVT of ex (19), is repeated below in ex. (24):

(24) UVT of nGX.IL with category names

U (nGX.IL)	u.Parse-σ	u.Iamb	u.AFL
sp	**3**	**0**	**0**
WD	**1**	**0**	**2**
SD	0	1	4

Between sp and WD, we find the following numerical relations. In this compressed representation, we use the language label to refer the value obtained by applying the constraint to it.

(25) Orders and equivalences in U: WD vs. sp

u.Parse-σ	**u.Iamb**	**u.AFL**
WD < sp	WD = sp	sp < WD

Do these hold in *every* UVT that has the grammar structure of nGX.IL? There's a lot of them, but we can at least observe that the relations hold between the first two rows of all the ranking-equivalent tableaux we have collected,

(26) Orders and Equivalences in the 4 UVTs: WD vs. sp

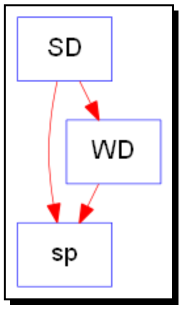
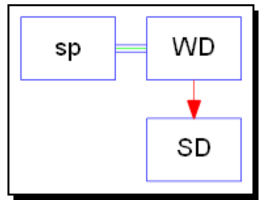
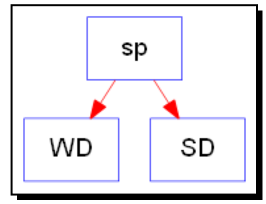
u.Parse-σ	**u.Iamb**	**u.AFL**
WD < sp	WD = sp	sp < WD
$L_2' < L_1'$	$L_2' = L_1'$	$L_1' < L_2'$
$L_2'' < L_1''$	$L_2'' = L_1''$	$L_1'' < L_2''$
$L_2''' < L_1'''$	$L_2''' = L_1'''$	$L_1''' < L_2'''$

If the given sample of equivalent tableaux contains all possibilities, then these relations are crucially determinative of the rankings, because any variation would lead outside the class of equivalents. The MOAT, in which each constraint is represented by its EPO (an "Equivalence-augmented Privileged Order"), will validate this hope.

Thinking globally, we want to obtain the relations that hold in *every* UVT that generates the grammars of the typology, putting aside those that vary from UVT to UVT. In §3.2 below, we will see how these global properties can be derived locally from an analysis based on the leg content of the typology, which may be determined from any one of its UVTs. This method of obtaining the crucial invariants turns on relations between pairs of grammars that are 'adjacent' in this sense: one contains a leg P<u>XY</u>Q and the other a leg P<u>YX</u>Q, where the only difference between the legs lies in the order of the underlined pair. The EPO is built from the equivalences and 'privileged' relations which are embodied in these transitions between 'adjacent' grammars. Here and in the following two introductory sections, we present EPOs as given objects, illustrating how their structure enables solutions to the three problems. The notions of privilege and adjacency are developed in detail in §2.2 and §2.3 below.

From among the relations imposed by each constraint in a UVT, the crucial grammar-determining subset is isolated in its EPO. Two UVTs are typologically equivalent if and only if they have structurally identical MOATs, so that each EPO in the one is isomorphic to an EPO in the other. An EPO will be portrayed as a graphical object with directed and undirected edges. In presenting an EPO, we mark the privileged order relations (directed) with single-headed red arrows and the equivalence relations (undirected) with a double blue lines. Under these conventions, the MOAT for the simplified stress theory nGX.IL looks like this:

(27) **MOAT**(nGX.IL)

EPO**(Parse-σ)**	EPO**(Iamb)**	EPO**(AFL)**
		

The EPO for Parse-σ indicates, exactly as we've claimed, that the order relations implicit in the violation values assigned by Parse-σ in the VT (19) are universally required. The Parse-σ EPO indicates that SD is better parsed than sp andWD, and that WD is in turn better parsed than sp.

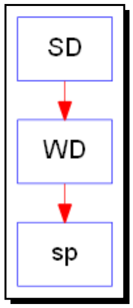
Although these judgments pertain to entire grammars rather than to linguistic forms, they are not shocking. In concrete Strongly Dense, with optima F^n, *every* output is fully parsed, with no violations of Parse-σ in any form. In Weakly Dense, with optima F^n(o), all *even* forms are fully parsed, but odd forms of length 3 or greater have one unparsed syllable, earning one violation of Parse-σ. And in Sparse, with optima Fo^n, all forms 3 syllables or longer have 1,2, … many unparsed syllables, with length *n* having *n*–2 of them, earning *n*–2 violations of Parse-σ.

This pattern reflects the general situation. If two grammars are ordered in the EPO for a constraint, that ordering cannot be numerically reversed in any cset of their associated languages, where optima from the corresponding languages are compared (Theorem (162) , §3.2.4). If two grammars are equivalent in an EPO for a constraint C, not only must the entire corresponding languages be equal on C in each UVT; the *optima* of associated concrete languages must be evaluated as equal by C within their csets in any concrete instantiation of the typology, indeed in any collection of csets that delivers the typology, whether or not the csets have a concrete interpretation. (Theorem (163), §3.2.4.) EPO relations hold in every UVT that yields a given typology, and they impose structure on multiple candidate set decompositions of the typology as well. This gives local content to the notion that one grammar can, in its entirety, be better than or equivalent to another.[12]

[12] To spell this out further, enumerate the candidate sets of nGX.IL in some way. The language we have called "SD" can be thought of as a (longish) vector **SD** with the optimum of candidate set *k* sitting in component *k* of **SD**, and similarly for the other languages. Let a given constraint, for example Parse-σ (Ps), evaluate the vector of optima **SD** at each coordinate, producing a numerical vector, call it Ps(**SD**), and so on for the other constraints. These numerical vectors C(**SD**) can be ordered coordinatewise. In this order, which we'll notate as $<_{coord}$, we have in the case of nGX.IL that Ps(**SD**)$<_{CO}$ Ps(**WD**), because at each

The Parse-σ EPO explicitly indicates that there is a *privileged relation* between SD and sp, even though that relation follows by transitivity. If we remove transitively derivable information for misguided esthetic reasons, we obtain a standard Hasse diagram, which is by definition transitively reduced.[13]

(28) Transitive Reduction (*deprecated*) of Parse-σ EPO



This supports exactly the same inferences about orders, but we eschew the formal simplification in order to retain sensitivity to privilege, which proves crucial to identifying the relations between grammars that determine their typological compatibility, as examined in §5.

With the MOAT (27) lodged in mind, let's go all the way back to our starting point in Concrete OT, the 5σ cset of nGX.IL. We can now disentangle the necessary relations from the artifacts of concreteness.

(29) **UVT from a single input for nGX.IL**

nGX.IL	Output	Parse	Iamb	AFL
σσσσσ	-uX-o-o-o- (**sp**)	3	0	0
	-uX-uX-o- (**WD**)	1	0	2
	-X-uX-uX- (**SD**)	0	1	4

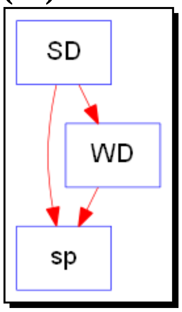
Let's compare these values with the demands imposed by each EPO.

---

coordinate k, we have numerically Ps(**SD**)[k]≤Ps(**SD**)[k] and at some (k odd and greater than 1), we have Ps(**SD**)[k]<Ps(**WD**)[k]. This is the sense in which the language SD is better parsed than the language WD. In general, with respect to the order $<_C$ from the EPO of C, which holds between grammars, we have it that $\Gamma_1 <_C \Gamma_2 \Rightarrow C(L_1) <_{coord} C(L_2)$. EPO equivalence implies equality, so that $\Gamma_1 \sim_C \Gamma_2 \Rightarrow C(\mathbf{L_1}) = C(\mathbf{L_2})$**.** Here we write Γ_i for the grammar, and $\mathbf{L}_i$ for its vector of optima. Observe, however, the converse does not hold. For example, it's true that for every input I, the SD optimum is equally or worse *left-aligned* that the WD optimum — the odd lengths of SD have that extra unary foot, the even lengths are parsed identically. Thus, AFL(**WD**) $<_{coord}$ AFL(**SD**). But, as we've seen, the numerical relations between WD and SD on AFL play no role in filtration and therefore no role in determining the ranking structure of the grammars. Consequently, WD and SD are not ordered in the AFL EPO. OT filtration works by lexicogaphic order, and this has effects beyond those of componentwise order.

[13] See e.g. the article Hasse Diagram in Wikipedia.

(30) Parse-σ EPO of nGX.IL

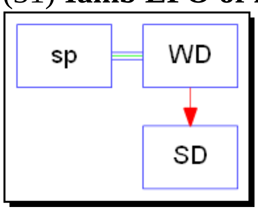


These EPO requirements are instantiated in the values in VT (29) as follows:[14]

- SD $<_{\text{Parse-}\sigma}$ WD $0 < 1$
- WD $<_{\text{Parse-}\sigma}$ sp $1 < 3$

The cited numerical order relations are all necessary, though of course the values in (29) are determined by concrete considerations: any order-respecting values will produce the same rankings.

(31) **Iamb EPO of nGX.IL**



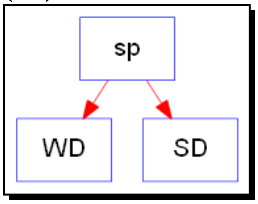
The EPO requirements are instantiated inVT (29) as follows:

- sp $\sim_{\text{Iamb}}$ WD $0 = 0$
- WD $<_{\text{Iamb}}$ SD. $0 < 1$

In terms of the cset decomposition of the concrete grammars, it follows that that competing Sparse and Weakly Dense optima must always have the same value on Iamb. In concrete reality, Sparse and Weakly Dense optima as shaped Fo^n and F^no, where F is a binary iambic foot. These forms never have feet that are non-iambic, always being evaluated to 0 by the constraint Iamb. Furthermore, in any UVT the shared value of their cognates must be strictly less that that of the cognate of Strongly Dense. In the concrete world of csets, even-length Strongly Dense optima are identical to even-length Weakly Dense optima, with both shaped F^n, weighing in iambically at 0. But odd-length Strongly Dense forms X-F^n always display a single monosyllabic foot, defined as non-iambic, earning the evaluation of 1. This exemplifies the fact that an EPO order relation can never be reversed in any cset that is part of a multi-cset rendering of a typology.

[14] An account of the details of relational instantiation is provided in definitions (132) and (133), p. 90.

(32) **AFL EPO of nGX.IL**



The EPO requirements are instantiated inVT (29) as follows:

- $\text{sp} <_{\text{AFL}} \text{WD}$ $\quad 0 < 2$
- $\text{sp} <_{\text{AFL}} \text{SD}$ $\quad 0 < 4$

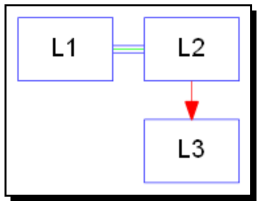
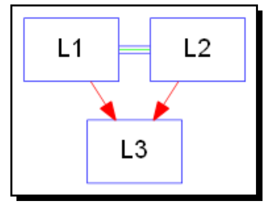
Because evaluation lives among the integers, a further relation is forced on us: $2<4$. This exceeds the requirements of the AFL EPO. The appearance of order between WD and SD languages in the values of VT (29) is artifactual from the grammatical point of view. It holds in some of the concrete instantiations of the intensional typology, of which nGX.IL is just one. As noted at the outset of this discussion, AFL values are never called on to choose between WD and SD. In the concrete world of nGX.IL, the WD violations of AFL incurred by optima are always equal to (even forms) or less than (odd forms) those of SD, but this has — perhaps unexpectedly — no effect on the functioning of the grammar. Consequently, as we've seen in UVTs (19)-(22), there exist renderings of the typology in which the cognates of WD and SD stand in any relation whatever with respect to AFL.

We conclude with a note on the nature of privilege. From numerical relations of the form $\text{sp}=\text{WD}$ and $\text{WD}<\text{SD}$, which are imposed by the Iamb EPO, it follows that $\text{sp}<\text{SD}$ — in all UVTs of nGX.IL. But $\text{sp}<\text{SD}$ is not included the EPO, as we've just found. Only 'privileged' orders appear in an EPO, and privileged status depends on a pairwise relation between legs of different grammars (§2.2). Continuing our focus on the properties of the EPO object, we observe that different privileged relations may define the very same numerical instantiations for a single constraint. For example, the following valid EPO structures sponsor the same numerical realizations, but will appear in different typologies.[15]

[15] EPO 1 is the Iamb EPO of nGX.IL. The following is a UVT with a MOAT different from that of nGX.IL, which gives rise to EPO 2 for column 1, as may be determined by applying the techniques of §2.2.

0	1	1
0	2	0
1	0	0

(33) **Distinct EPOs**

EPO-A	EPO-B
	

As we will see in §0.3.3, this distinction in privilege has important consequences for the classification of the participating languages. In particular, EPO-A and EPO-B both obstruct the amalgamation of L_1 and L_3 as a typological class contrasting with L_2, while EPO-B additionally obstructs the amalgamation of L_2 and L_3 as a class contrasting with L_1 . This effect is examined in the discussion of ex. (36) below.

0.3.2 Compatibility of Grammars within a Typology

> Johnson could see no bicycle would go.
> "You bear yourself, and the machine as well."
> – Empson

Problem 2. ***Why can't we all just get along*?** An intensional typology is a collection of grammars. But not every collection of grammars is a typology. We know that any typology must be generable by a UVT, so that lack of a generating UVT is fatal. The issue then becomes: what properties of a collection of ranking grammars will prevent it from having a UVT?

Recall that no two grammars in a typology can share a leg, because each leg of a grammar delivers all of its optima. Nor can there be a ranking that is not assigned to *some* grammar in the typology. A typology, as we have noted, is therefore minimally a *partition* of the set of all rankings, which divides the set into non-overlapping subsets. These are called 'parts' or 'blocks'. Each ranking grammar in a typology is a block in a partition of Ord(Con$_S$), the entire set of possible rankings.

A partition can fail to be a typology for two reasons. First, a block of the partition may not be a grammar. This happens when the set of linear orders constituting the block is not characterizable by an ERC set.

To see how this can come about, consider a simple Abstract OT system with three constraints X, Y, Z. Partition the set of 6 rankings into two blocks. Let the first block be {XYZ, YXZ}, where ranking order is notated by sequencing. The legs of this block are delimited by the requirement that both X and Y dominate Z. For this reason, we can call it ZBot, indicating that Z is at the bottom of the ranking and no other conditions apply. Its complement 'co-ZBot' contains all the other rankings. It is delimited by the condition '$Z \gg X$ *or* $Z \gg Y$', as may be ascertained through logic or through inspection of its contents: co-ZBot = {ZXY, ZYX, YZX, XZY}.

The ZBot block has a familiar ranking pattern given by the ERC set {WeL, eWL}. Grammars of this form were originally noticed in the subsystems of Elementary Syllable Theory (see P&S:112ff). In the form of {f.max, f.dep}≫m.Ons, 'Ons-Bot' in our terms, the grammar yields the no-deletion, no-insertion languages in which onsetless syllables are allowed because no breach of faithfulness can be called on to avoid them. Symmetrically, the grammar {f.max, m.Ons}≫f.dep, 'Dep-Bot', yields 'Onset Required' languages where insertion eliminates the possibility of onsetless syllables. And {f.dep, m.Ons}≫ f.max, 'Max-Bot', yields 'Onset Required' languages where deletion is called on to deliver onsetted syllables.

VTs like those that involve m.Ons (or m.NoCoda) contending with f.dep and f.max assume this kind of shape:

(34) **An Irreducible VT**

	X	Y	Z
k_1	0	0	1
k_2	0	1	0
k_3	1	0	0

The grammar selecting k_1 is ZBot, and so on for the others. The crisis comes when we try to implant ZBot in a typology with just one other grammar, co-ZBot, which is the union of the legs associated with k_2 and k_3. No ERC or set of ERCs can represent the condition '$Z \gg X$ *or* $Z \gg Y$'.[16] The block co-ZBot is not ERC-characterizable and hence not a grammar. The two-block partition {ZBot, co-ZBot} is not a typology because it does not consist of grammars.

Second, and perhaps in defiance of naïve intuition, a partition may consist of well-formed grammars but still fail to be a typology.[17] There will be no UVT that generates it, even

[16] It can't be {LLW} — 'Z dominates *both* X and Y'. It can't be {LeW, eLW}, which is equivalent to {LLW}. Observe that no one constraint is subordinated in all the ranking of co-ZBOT: X,Y,Z all appear top-ranked in some leg. Therefore, no ERC that describes the typology can contain an L in any constraint, which would require that it be subordinated in every leg of the grammar. The only grammar that this set belongs to is the trivial grammar that includes all rankings. See §5 for calculation of this fact via the *join*, an ERC logic-based operation introduced in Merchant 2008.

[17] Every grammar belongs to *some* typology. We can take an ERC grammar represented in a CT and mechanically convert it to a VT: insert a row containing only 1, representing the target grammar, and then set W=2, L=0, *e*=1. Comparing the inserted row in this VT against the others reproduces the target grammar. The typology of the constructed VT contains the target grammar.

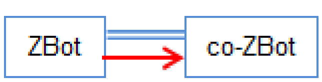
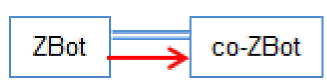
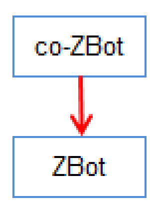
though each block is a grammar and can appear in some other typology. The simplest cases show up in systems with 4 constraints and will be examined below in §5.2. Being together in a typology thus requires a certain kind of compatibility between grammars. Two fundamental questions then arise: suppose $\mathcal{P}$ is a collection of grammars that partitions the set of all rankings of some constraint set.

(1) Is $\mathcal{P}$ a typology?
(2) If $\mathcal{P}$ is a typology, which UVTs produce it?

Both are resolved by the MOAT. The principles of MOAT construction (§2.2) can be applied to any partition of the ranking set. If a well-formed MOAT results, we have not just a partition composed of grammars, but a partition that is guaranteed to be a typology. (§3.3). Furthermore, as we have emphasized, *all* UVTs that produce the typology are determined by the order and equivalence relations represented in the EPOs of the MOAT (Instantiation Theorem (159), §3.2.4). Conversely, failure to produce a well-formed MOAT signals that the collection of blocks is not a typology.

To see how this works, consider first the Bot/coBot partition on three constraints. Using the techniques of §2.2, we find that it sponsors the relations portrayed in (35) below. Because they take us outside of what a UVT can produce, we must leave the formal class of EPOs to represent them. Objects that involve a set with an equivalence relation on it are known as "setoids" or "E-sets." An EPO is a setoid (E-set) that also carries a partial order. The more general structure that arises from partitions like Bot/coBot is also a setoid, but its second relation needn't be a partial order; and even if it is one, it needn't behave well in combination with the first. To represent EPOs and the more general setoids that arise in partitioning the set of all rankings, we introduce a diagram that we will call a 'bigraph'. A bigraph will have two kinds of arcs, distinguishing the two relations of the EPO or setoid it represents. We write bigraph(X) to label a representation of the EPO or setoid associated with constraint X. Here we collect bigraphs for each constraint in the 3-constraint partition Bot/co-ZBot.

(35) **Bigraphs of the Bot/coBot Typology**

Bigraph(X)	Bigraph(Y)	Bigraph(Z)
		

The bigraphs of constraints X and Y fail to be EPOs, indicating that they cannot be instantiated in a UVT. Each bigraph incoherently demands that the value assigned to ZBot be *equal* to that assigned to co-ZBot, shown by the blue double line, and at the same time strictly *less than* that assigned co-ZBot, shown by the red arrow. You can do many things with numbers, but not this. By contrast, the bigraph of Z is a proper EPO, but the damage has been done.

Graphically, the fatal configuration is a *cycle*. In a undirected graph, a *path* is a just sequence of unordered pairs of nodes, called 'arcs' or 'edges'. In a directed graph, the pairs are ordered, and a directed edge is therefore represented as an arrow pointing from one node to another. A *directed path* is a path that may be traversed by proceeding in the indicated direction. A *cycle* is a path which leads from a node back to itself.

A bigraph has the further complexity of including both directed arcs (arrows) and undirected arcs (double lines). The relevant notion of path in the bigraph respects the directional status of the arcs: in proceeding from node to node along a path, the direction of the arrows must be followed, but the double lines may be traversed either way. As always, any path that proceeds from a node to itself counts as a cycle. In a bigraph, we defined a *directed cycle* to be a cycle that *contains* a directed arc. Since directed cycles in this sense are the critical structures in OT theory, we will simply refer to them here as *cycles*. A bigraph that contains a (directed) cycle will be termed *cyclic*; a bigraph that does not is *acyclic*.[18]

The cycles in the X and Y bigraphs of (35) mix arrows and double lines: order and equivalence. A cycle may consist entirely of arrows, arising without need for equivalences between any of its members. One such case derived from nGX.IL is examined below in ex. (40). Another ('the Contradictory Snake') is examined in §5.2.2.

A nongrammar block in a partition is detected by the same kind of analysis that produces the EPOs of the MOAT for a valid typology. When legitimate grammars can't coexist in the same typology (§5.2), they reveal their incompatibility through participation in cycles.

In sum**:** cycles in a bigraph prevent it from being an EPO. Any ranking partition with a cyclic bigraph isn't a typology.

[18] To explicate this further, we can define the notion 'cycle' by taking the formal step of regarding equivalent nodes as being literally the same node – "identifying equivalents." This maneuver yields a structure $B^{\sim}$ from a bigraph B, in which every set of nodes in B connected by double lines is represented in $B^{\sim}$ as a single node. $B^{\sim}$ is just an ordinary directed graph, and we can apply familiar definitions to it. If there is a directed cycle in $B^{\sim}$, we say that B is cyclic; if not, acyclic. In our example, bigraph(X)$^{\sim}$ and bigraph(Y)$^{\sim}$ will each have the single node obtained by identifying the ZBot and coZBot nodes. In these derived structures, there's a loop — the smallest kind of cycle — connecting this single node to itself.

0.3.3 Classification

Problem 3. ***Us vs. Them***. A typology implicitly classifies its languages both intensionally and extensionally. Grammars are intensionally groupable by shared ranking relations; languages are extensionally groupable by shared structural traits. Understanding how the theory characterizes data requires aligning these conceptually distinct modes of categorization (Alber & Prince 2015; Alber, DelBusso, & Prince 2016).

This is not just a matter of annotating some prior correlational analysis of data. Extensional grouping may be non-unique in purely extensional terms, so that reference to ranking structure is required to decide the validity of proposed classes. Under Generalized Alignment, for example, X-F and F-o are regarded as better *left*-aligned than their respective competitors F-X and o-F; but within iterative theories, X-F parallels o-F. (Crowhurst & Hewitt 1995; Alber 2005:491).[19] Alber & Prince observe that extensionally, we have a free choice between positing data classes {F-o, X-F}, predicted by Generalized Alignment, and {o-F, X-F}, predicted by directional iteration. Which classification is deemed grammatically meaningful depends on the theory that generates them.

The structure of typologies, construed in this way, does not appear to be entirely trivial. Therefore, an attack on the problem requires levels of development. We aim here to establish the groundwork for a basic mode of intensional classification. Following §0.3 above, we take this to be founded on the notion of a typological class. Within a target typology, a *typological class* of grammars is set of grammars that is itself not only a grammar, but also a grammar within an abstract typology that generalizes the original target typology. One typology *generalizes* a target typology if its grammars each consist of unions of grammars from the target.[20] In this case, we will follow the terminology of partition theory and say that the more general typology *coarsens* the target typology, and that the target *refines* the more general typology.[21]

---

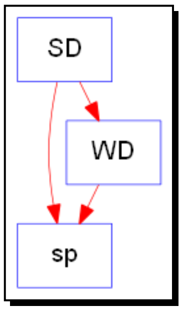
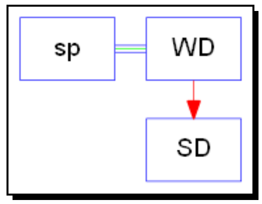
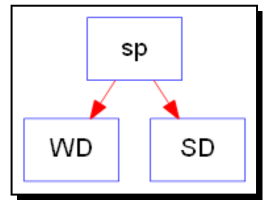
[19] Absent from SPE, iterative rules were introduced into generative phonology by C. Douglas Johnson 1972 and Irwin Howard 1972. After a period of stout resistance, they made a quick transition from abominable to obvious. *We've always been at war with Eastasia*.

[20] Prince 2015b shows that the set of all typologies on *n* constraints forms a lattice, a subset of the partition lattice based on a sets of *n* objects. It is not a sublattice because joins in the typology lattice needn't be the same as in the general lattice. The order relation in the lattice is *coarsening* (dually *refinement*). See Prince 2013 [Youtube] for discussion.

[21] The qualitive sense of the terminology is this: refinement breaks up a block of a partition into smaller pieces, introducing further distinctions within that block. Coarsening goes in the opposite direction, amalgamating parts into larger pieces, losing distinctions. For example, the grades A,B,C,D,F distinguish academic performance, as do Pass and Fail. Pass/Fail coarsens the grading partition by amalgamating A through D. Similarly, the A,B,C,D,F system *refines* the Pass/Fail system by dividing the Pass category into four subcategories, while identifying the Fail category with F.

Let's apply this mode of analysis to nGX.IL. To discover its typological classes, we consult its MOAT, reproduced below.

(36) MOAT(nGX.IL)

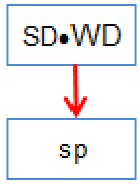
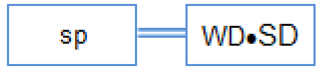
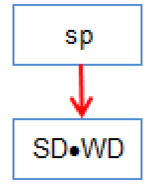
<table>
<tr><th>EPO(Parse-σ)</th><th>EPO(Iamb)</th><th>EPO(AFL)</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

Amalgamating a set of grammars in a typology — forming the union of their legs — is equivalent to merging their nodes in each EPO of the MOAT. We'll write the result of the merger of nodes A and B as A•B.

Let us first ask if there is a typological class 'dense' in nGX.IL, consisting of WD and SD taken together. Such a class would include the languages that have any multi-foot optima, abstracting away from the weak/strong difference in the treatment of syllables that lie outside binary feet in odd-length forms like F-F-o , o-F-F, X-F-F, and F-F-X. The dense class would stand in opposition to *sparse*, whose members have optima that contain one foot per word regardless of length. *Dense* is extensionally definable; is it also a typological class?

To determine the answer, we examine the effect of merging the nodes in the MOAT. With the relevant nodes labeled 'SD' and 'WD', their merger SD•WD corresponds to the union of their legs **SD** ∪ **WD** into a new ranking grammar. We use boldface here to distinguish the name of a leg set from the name of its corresponding node, to emphasize the distinction, just as we take care to distinguish node merger '•' from set union '∪'.

(37) MOAT(nGX.IL)/D

EPO(P-σ)	EPO(Iamb)	EPO(AFL)
		

All external relations of the merging grammars are preserved. These are the equivalence and privileged order relations that hold between any participant in the merger and any node not participating in the merger. In the case of the Iamb EPO, for example, the equivalence between sp and WD is retained by the merged node WD•SD. But there is no relation between sp and SD in the Iamb EPO of nGX.IL, either of equivalence or of privileged order, so there is nothing to preserve. All internal relations — those between nodes WD and SD — are lost. This explains the structure of the MOAT (37) which represents the result of merging WD and SD in MOAT (36).

The results can be tabulated as follows:

(38) Inheritance of component relations in EPO mergers

Constraint	Grammar relations	Inherited Relations from Unmerged Bigraph	
• **Parse-σ**:	SD•WD → sp	SD,WD → sp	in Parse-σ bigraph
• **Iamb**:	SD•WD ═ sp	WD ═ sp	in Iamb bigraph
• **AFL**:	**sp → SD•WD**	sp→ WD, SD	in AFL bigraph

These graphical effects run in perfect parallel to relations determined by examination of the legs. Anticipating §2.2ff, we use the symbols $<_E$ and $\sim_E$ to indicate the inter-grammar relations which make up the content of the EPO setoids. As before, we bold the grammar names to emphasize that we are referring to leg sets rather than graphical nodes.

(39) Reasoning from MOAT(nGX.IL): EPO by EPO

Constraint	Grammar relations	Component relations
• **Parse-σ**:	$\mathbf{SD}\cup\mathbf{WD} <_{\text{Parse-}\sigma} \mathbf{sp}$	$\mathbf{SD} <_{\text{Parse-}\sigma} \mathbf{sp}$ and $\mathbf{WD} <_{\text{Parse-}\sigma} \mathbf{sp}$
• **Iamb**:	$\mathbf{SD}\cup\mathbf{WD} \sim_{\text{Iamb}} \mathbf{sp}$	$\mathbf{WD} \sim_{\text{Iamb}} \mathbf{sp}$
• **AFL**:	$\mathbf{sp} <_{\text{AFL}} \mathbf{SD}\cup\mathbf{WD}$	$\mathbf{sp} <_{\text{AFL}} \mathbf{WD}$ and $\mathbf{sp} <_{\text{AFL}} \mathbf{SD}$

In each case, the analysis based on union of leg sets accords with the results of the graphical operation of merging nodes. MOAT (37) is also identical to what would be obtained directly from analyzing the partition of the ranking set into the two blocks **sp** and **D** = **WD**∪**SD**. (§2.3-5). However derived, the partition {Sparse, Dense} has a well-formed MOAT: there are no cycles in any EPO. We are therefore licensed to conclude that the partition {Sparse, Dense} is a typology — an intensional typology of Abstract OT, strictly coarser tha nGX.IL. It follows that Dense, the union of **WD** and **SD**, is a *typological class*.

There's another way of classifying the three grammars of nGX.IL: merge Sparse and Strongly Dense, contrasting them jointly with Weakly Dense. The informal names of the grammars prejudice us against this breakdown, but words may fail us. There's even an extensional rationale for the classification: WD forms have exactly one unparsed syllable in odd-length forms; the others do not. Does this hypothesized generalization yield a typological class {sp, SD} in nGX.IL? We do not need to enumerate and analyze the leg sets. With the MOAT of nGX.IL in hand, we can test the status of the proposed generalization through graphical merger. The result is a collection of bigraphs: but only one of of them is a valid EPO.

(40) Partition {**WD**, **SD**∪**sp**}

bigraph(Parse-σ)	bigraph(Iamb)	bigraph(AFL)
$\begin{matrix} \text{SD•sp} \\ \downarrow\ \uparrow \\ \text{WD} \end{matrix}$	$\text{SD•sp} \overset{=}{\leftarrow} \text{WD}$	$\begin{matrix} \text{SD•sp} \\ \downarrow \\ \text{WD} \end{matrix}$

This bigraph set fails to attain MOAT status: the first two show cycles, implying contradictory requirements that cannot be realized in any structure ordered like the integers. We may parse the components as follows, showing how the relations in the amalgamated structure follow from the equivalence and privileged order relations in the unamalgamated EPOs of nGX.IL.

(41) Partition {**WD**, **SD**∪**sp**}, bigraph by bigraph

Constraint	Merger Relations in (40)	Component relations in nGX.IL (36)
● **Parse-σ**:	SD•sp → WD	SD → WD
	SD•sp ← WD	WD → sp
● **Iamb**:	SD•sp ═ WD	sp ═ WD
	SD•sp → WD	WD → SD
● **AFL**	SD•sp → WD	sp → WD

We conclude that {SD, sp} is not a *typological class* of nGX.IL. In this case, the the failure comes about because **SD**∪**sp** is not a grammar. In §5.2 we look at two subtler examples, in which legitimate grammars cannot coexist within a typology because the local relations between them cannot be realized as a globally consistent orders and equivalences.

Main classification result. A partition of the set of all rankings is a typology if and only if it has a MOAT (Corollary (190), §3.3). The MOAT is determined from the partition. Union of blocks in a partition corresponds exactly to merging the corresponding nodes in the EPO bigraphs. The typological status of a partition $\mathcal{P}$ that is arrived at by the union of some grammars of a typology T can be evaluated by an effective calculation on the MOAT. It is only necessary to check that each of the bigraphs associated with $\mathcal{P}$ has no cycles, and is therefore a well-formed EPO. Any hypothesis that a set of grammars is a typological class can therefore be verified or falsified by direct calculation.

The MOAT solves the typological class problem, and it solves the grammatical class problem as well whenever the two notions coincide. When they do not, other techniques are available, using the join of ERC sets, discussed in §5.1 (Merchant 2008, 2011).

The notions *typological class* and *grammatical class* initiate the enterprise of classification but do not end it. Consider the 'Onset Required' set of languages within Elementary Syllable Theory. As discussed above in relation to the bigraph example, this class has no ERC set characterization, since it involves an irreducible disjunction among the necessarily dominated constraints. ERCs require conjunction among do`minated constraints. 'Onset Required' comes from either m.Ons≫f.dep *or* m.Ons≫f.max: therefore, no grammar. The two subcases 'enforced by deletion' and 'enforced by insertion' cannot be merged into a grammatical class, and so a fortiori cannot be merged into a typological class.

To recognize 'Onset Required' as coherent, ranking-defined class requires further analytical apparatus, developed in Alber & Prince, which can take account of symmetries in the behavior of disparate constraints, recognizing classes of constraints as well as classes of grammars. In this case, the class of faithfulness constraints turns out to be structurally significant. We are assured, then, that there is structure in typologies that lies beyond the immediate reach of the concepts developed here. But we can see it clearly, and extend analysis to it, only when we understand the groundwork. Our strategy, then, is to focus on the groundwork, so that further structure can be soundly built upon it.[22]

[22] καὶ πᾶς ὁ ἀκούων μου τοὺς λόγους τούτους καὶ μὴ ποιῶν αὐτοὺς ὁμοιωθήσεται ἀνδρὶ μωρῷ ὅστις ᾠκοδόμησεν αὐτοῦ τὴν οἰκίαν ἐπὶ τὴν ἄμμον…

1 The EST Typology

Let's develop the ideas in the context of an example: Elementary Syllable Theory (EST), and its reduced cousin the C-System of EST, which deals only with the disposition of consonants, to be explored in §4 below.

1.1 Definition of EST

In the first version of syllable theory laid out in P&S:104-115, epenthetic differences between V and C are conflated, being handled by a single constraint, FILL. This system we will call *Elementary Syllable Theory* (EST), as distinct from Basic Syllable Theory (BST: P&S: 115ff), in which two faithfulness constraints (FILL^{Nuc}, FILL^{Ons}) treat epenthesis of vowels distinctly from epenthesis of consonants. We adapt the containment-based system of P&S to correspondence theory (McCarthy & Prince 1995).

A Concrete OT system S requires articulation of GEN_{S} and CON_{S}. Since predictive consequences, reified in the system's typology, follow necessarily, it will be worth our while to be clear at the definitional stage. We first present the definitions descriptively and then conclude with a compact formal statement of their contents.

GEN_{S} for any system S spells out what a candidate of S is, and what the candidate set is, within which comparison takes place. For GEN_{EST}, as is typical in phonological analysis, a candidate consists of an input, an output, and a correspondence relation between them.

Input. GEN_{EST} accepts as inputs non-empty strings of arbitrary length composed of the characters C and V.

Output. GEN_{EST} accepts as the outputs for any input all arbitrary-length strings of C and V, including the empty string. Nonempty strings are fully syllabified, so that each segment belongs to a syllable. A *syllable* has exactly one V; at most one C preceding the vowel; and at most one C following the vowel. Syllable boundaries will be denoted by square brackets. A prevocalic C is termed the *onset* of the syllable; a postvocalic C is termed the *coda* of a syllable. These terms are descriptive of string position and do not name constituents in syllable structure, as they do in P&S:110.[23]

[23] P&S allow for unsyllabified segments in the output, phonetically interpreted as deletion, and empty structural nodes, phonetically interpreted as insertion, following earlier works such as Steriade 1982, Ito 1986, 1989. Correspondence theory represents deletion and insertion in the phonology itself, GEN_{EST} imposes on phonology the output conditions that are met only after phonetic interpretation in P&S: exhaustive syllabification, and segmental saturation of all higher-order structural nodes (here just σ).

Correspondence. Segments in the input may be associated with segments in the output in a *correspondence* relation. We limit this in GEN_{EST} so that an input segment has at most one correspondent output segment, and vice versa. In addition, C may correspond only to C; V only to V. The linear order of segments in the input is maintained in the order of their correspondents in the output. Crucially, a segment in the input need not have a correspondent in the output, a state of affairs representing deletion; and a segment in the output need not have an input correspondent, representing epenthesis.

These considerations may be spelled exactly in the following terms. The notation S*, where S is a set of strings, denotes the set of all strings concatenating members of S any number of times, including none. S^+ is defined similarly, but omits the empty string.

(42) $\mathbf{GEN}_{\mathbf{EST}}$

IN $=\{C,V\}^+$

OUT = $\{[(C)V(C)]\}^*$

Correspondence: Each input-output pair (*in*, *out*), where length(*in*) = *n* and length(*out*) = *m*, comes with a set of partial functions CORR.IO(*in*,*out*) defined as follows. First, the partial function and the conditions on it:

a) ${}_{\mathrm{p}}\mathrm{f}_{nm}$: $\mathbf{n} \rightarrow \mathbf{m}$, where $\mathbf{n} = \{1,2, \ldots,n\}$, $\mathbf{m}=\{1,2, \ldots,m\}$, for **n** and **m** the ordinal positions of the characters in strings $in\in$INand $out\in$OUT, with ${}_{\mathrm{p}}\mathrm{f}_{nm}$ a partial function on **n**.

b) LIN: $i < j \Rightarrow {}_{\mathrm{p}}\mathrm{f}_{nm}(i) < {}_{\mathrm{p}}\mathrm{f}_{nm}(j)$, for $i,j\in\mathbf{n}$, whenever both ${}_{\mathrm{p}}\mathrm{f}_{\mathrm{nm}}(i)$ and ${}_{\mathrm{p}}\mathrm{f}_{\mathrm{nm}}(j)$ are defined.

c) TYPE: ${}_{\mathrm{p}}\mathrm{f}_{nm}(i) = k \Rightarrow \mathrm{in}[i] = \mathrm{out}[k]$, ensuring that in$[i]$ and out$[k]$ have the same value. Here this means they are both C or both V.

Now, the set of all admitted correspondence relations between an IO pair:

CORR.IO(*in*,*out*) = $\{{}_{\mathrm{p}}\mathrm{f}_{nm} \mid {}_{\mathrm{p}}\mathrm{f}_{nm}$ satisfies LIN & TYPE, n=length(in), m=length(out), $in\in$IN, $out\in$OUT$\}$.

The set of candidates admitted by GEN_{EST}, denoted CAND_{EST}, is then the following:

(43) $\mathbf{CAND}_{\mathbf{EST}} = \{\langle in, out, \mathbf{c}\rangle \mid in\in\text{IN}, out\in\text{OUT}, \mathbf{c}\in\text{Corr.IO}(in, out)\}$

The correspondence relation is interpreted as a partial function from a set of indices on the segments of the input to a set of output indices. The indices simply give the ordinal position of characters in the strings. A *partial* function need not map every member of its domain to its co-domain. Here proper partiality represents deletion; epenthesis obtains when the co-domain contains indices not in the range of the function.

We define four constraints in CON_{EST}, each of which is a function from CAND_{EST} to $\mathbb{N}$. Though modified in light of correspondence theory, these operate very much as they do in P&S:106. Following OT best practices, we mark the type of each constraint by a prefix: ‘m’ for markedness, indicating the constraint only evaluates the output, and ‘f’ for faithfulness, indicating it requires the correspondence relation for its evaluation of disparities between input and output.

The constraints of Con_{EST} are four in number:

m.Ons	returns the number of onsetless syllables in a candidate's output
m.NoCoda	returns the number of syllables that have a coda in a candidate's output
f.max	returns the number of input segments that lack output correspondents
f.dep	returns the number of output segments that lack input correspondents

To spell this out, we adopt a short-hand for substring containment, defining '$x \sqsubseteq s$' = 'x is a contiguous substring of the string s'. The notation 'card S' denotes the cardinality of the set S. The (partial) correspondence function is denoted c.

(44) **$\textbf{Con}_{\textbf{EST}}$**

$$\text{m.Ons}(\langle in, out, \mathsf{c}\rangle) = \text{card}\{\text{“[V”} \sqsubseteq out\}$$
$$\text{m.NoCoda}(\langle in, out, \mathsf{c}\rangle) = \text{card}\{\text{“C]”} \sqsubseteq out\}$$
$$\text{f.max}(\langle in, out, \mathsf{c}\rangle) = \text{card}\{x \sqsubseteq in \mid x \in \{C, V\} \text{ and } \neg\exists y \sqsubseteq out, y = \mathsf{c}(x)\}$$
$$\text{f.dep}(\langle in, out, \mathsf{c}\rangle) = \text{card}\{y \sqsubseteq out \mid y \in \{C, V\} \text{ and } \neg\exists x \sqsubseteq in, y = \mathsf{c}(x)\}$$

1.2 Three Representations of a Typology

Any well-defined theory will invariably give rise to multiple characterizations, each equally valid, each shining different light on the theory. OT is no different. In the typological realm we will be concerned with the extensional typology, the intensional ranking typology, and the intensional ERC typology. Each is determinable from a well-chosen set of candidates.

1.2.1. A Universal Support for EST

Having articulated Gen_{EST} and Con_{EST}, the next typological question is this: which candidate sets determine the typology? A set of candidate sets that makes all possible distinctions for all languages in a typology we call a *universal support*. (See e.g. Prince 2015a). Given any concrete instantiation of OT, typological claims can only be justified after a universal support has been identified (see Bane & Riggle 2012 for the unwelcome consequences of candidate omission). Below is a universal support for EST consisting of three candidate sets.[24] We write ε for the empty string, and indicate correspondence with subscripts. Epenthetic segment, shown without indices, are also marked typographically.

[24] P&S:112-114 deploy the first and third in their analysis of BST, delivering a typology that is coarser than the typology of the target system. Riggle 2004:108ff recognizes that a telling candidate with a C that must be unfaithfully syllabified (for him, CCVVC) will refine the P&S typology to the BST as defined.

(45) A Universal Support for EST

input	output	m.Ons	m.NoCoda	f.dep	f.max	Type
V_1	$[V_1]$	1	0	0	0	F
	ε	0	0	0	1	del
	$[\underline{\mathbf{C}}\ V_1]$	0	0	1	0	ins
C_1	ε	0	0	0	1	del
	$[C_1\underline{\mathbf{V}}]$	0	0	1	0	ins
$C_1\ V_2\ C_3$	$[C_1V_2C_3]$	0	1	0	0	F
	$[C_1V_2]$	0	0	0	1	del
	$[C_1V_2][C_3\underline{\mathbf{V}}]$	0	0	1	0	ins

Notation in Type column: F = "faithful", del = 'deletional', ins = 'insertional'.

Each cset in this universal support contains all and only the non-harmonically bounded outputs for each of their respective inputs. For example, in the second cset, the input $/C_1/$ has only two possibly optimal outputs. One lacks the C, resulting in the empty string as the output. The other contains an epenthetic V, forming a valid syllable. Further deletions can't happen: we've reached the zero lower bound. Further insertions merely worsen performance on f.dep and perhaps m.Ons and m.NoCoda.[25] Since further unfaithfulness is either impossible or lacking in benefit, no other candidate can displace either of the two listed. Similar arguments hold for the sufficiency of the other candidate sets.

Observe that /C/ must be unfaithfully mapped, as [C] is not a licit parse in EST. For outputs of /C/, GEN_{EST} only admits unfaithful candidates.

These three candidate sets provide a universal support for this typology since no additional candidate sets can induce further ranking distinctions in any language of the EST, as may be shown by working through the predicted grammars with GEN_{EST} in hand.

There are 8 languages in the typology of the EST corresponding to the 8 viable selections of optima, one from each of the three candidate sets above. Even though there are 18 candidate languages — 3×2×3 combinations of candidates — 10 of these involve inconsistent rankings. For example, no ranking can deletionally map $/V_1/$ to ε while

[25] See P&S:116-118 for the theory of optimal epenthesis sites in BST. The same mode of analysis applies to the EST studied here.

insertionally mapping /C_1/ to [C_1 $\underline{V}$]. The first map requires f.dep ≫ f.max and the second f.max ≫ f.dep, an impossible requirement on any total order on the constraints.

1.2.2 Extensional Languages of EST

The eight languages of EST are characterized extensionally in terms of three distinctions. The determinative differences are these:

1. Onset Required (**OR**), with all syllables [C…]
   vs. Onset Lack Allowed (**OLA**), permitting [V…]
2. Coda Prohibited (**CP**), requiring […V]
   vs. Coda Allowed (**CA**), allowing […C]
3. Insertion vs. Deletion.

As each language in EST must select one choice from these three binary distinctions, we can label the eight languages perspicuously by their selections, adapting the naming style of P&S:116. Below is a chart giving our labels of the eight, their possible outputs across the range of all inputs, and how input-output disparities in syllable structure are managed. Recall that the notation {*string*}* indicates the set of all strings formed by concatentating repetitions of *string*, including none, while $\{string\}^+$ has at least one instance of *string*.

(46) **Languages of EST**

Name	**Outputs**	**IO disparities**	**Output Type**
1:CV.del	{ [CV] }*	deletion	CP, OR
2:(C)V.del	{ [(C)V] }*	deletion	CP, OLA
3:CV.ins	$\{ [CV] \}^+$	insertion	CP, OR
4:(C)V.ins	$\{ [(C)V] \}^+$	insertion	CP, OLA
5:CV(C).del	{ [CV(C)] }*	deletion	CA, OR
6:(C)V(C).del	{ [(C)V(C)] }*	deletion	CA, OLA
7:CV(C).ins	$\{ [CV(C)] \}^+$	insertion	CA, OR
8:(C)V(C).ins	$\{ [(C)V(C)] \}^+$	insertion	CA, OLA

Even the languages 6 and 8, which allow every admitted syllable structure form, are distinguished by deletion vs. insertion. This is because inputs like /C/, which as noted cannot be mapped faithfully, will nevertheless have an output and must obtain it through either insertion or deletion.

The inputs of the universal support map to their respective outputs as follows for each of the 8 languages in the EST.

(47) Input-Output map of the universal support for the EST

	/V_1/	**/C_1/**	**/$C_1V_2C_3$/**
1:CV.del	ε	ε	[C_1V_2]
2:(C)V.del	[V_1]	ε	[C_1V_2]
3:CV.ins	[C V_1]	[C_1V]	[C_1V_2][C_3V]
4:(C)V.ins	[V_1]	[C_1V]	[C_1V_2][C_3V]
5:CV(C).del	ε	ε	[$C_1V_2C_3$]
6:(C)V(C).del	[V_1]	ε	[$C_1V_2C_3$]
7:CV(C).ins	[C V_1]	[C_1V]	[$C_1V_2C_3$]
8:(C)V(C).ins	[V_1]	[C_1V]	[$C_1V_2C_3$]

In sum, the three-fold binary nature of the eight languages leads to a natural grouping which is reflected in the universal support and in our naming practice.

(1) **OR/OLA**. Onsets are either required or allowed to be absent.
(2) **CP, CA**. Codas are forbidden or allowed.
(3**) ins/del**. Compliance with these and with GEN-imposed syllabification requirements is achieved either by insertion or by deletion.

Compare the classification of BST in P&S:116.

(48) Extensional categorization of EST languages

		Onset	
		Required	Lack Allowed
Coda	Prohibited	1:CV.del 3:CV.ins	2:(C)V.del 4:(C)V.ins
	Allowed	5:CV(C).del 7:CV(C).ins	6:(C)V(C).del 8:(C)V(C).ins

1.2.3 Representations of the Grammars of 1:CV.del and 2:(C)V.del

Every extensional language is characterized intensionally by its ranking grammar, the set of total orders that select its optima. The ranking grammar of language 1:CV.del contains the following 6 legs:

(49) G_R(1:CV.del), the ranking grammar of 1:CV.del

m.Ons	≫	m.NoCod	≫	f.dep	≫	f.max
m.Ons	≫	f.dep	≫	m.NoCod	≫	f.max
m.NoCod	≫	m.Ons	≫	f.dep	≫	f.max
m.NoCod	≫	f.dep	≫	m.Ons	≫	f.max
f.dep	≫	m.Ons	≫	m.NoCod	≫	f.max
f.dep	≫	m.NoCod	≫	m.Ons	≫	f.max

Gross enumerations like this are not linguistically informative. Inspection reveals that the set G_R(1:CV.del) consists of those rankings in which f.max is dominated by the other three constraints, which may occur in any order among themselves, hence the 6 = 3! legs. This generalization is represented directly in an ERC grammar G_E(1:CV.del).

(50) G_E(1:CV.del), ERC grammar of Lg. 1:CV.del

1:m.Ons	2:m.NoCoda	3:f.dep	4:f.max
W			**L**
	W		**L**
		W	**L**

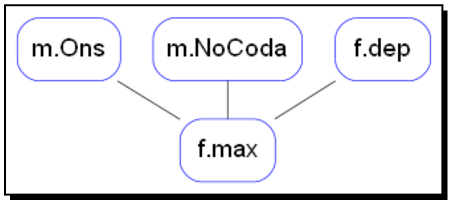
This tells us directly that m.Ons, m.NoCoda are maximally respected, leading to all syllables being onsetted and codaless, and that any and all compliance with the demands of GEN_{EST} and markedness is obtained by deletion (violating f.max), rather than by insertion (violating f.dep).[26]

There is a precise relationship between the ranking grammar and the ERC grammar. The ERC grammar delimits a set of total orders that are consistent with it, its linear extensions or legs. Each total order of the ranking grammar must also be a leg of the ERC grammar, and there can be no legs of the ERC grammar that are not total orders of the ranking grammar. The legs of the ERC grammar are thus exactly the total orders of the ranking grammar.

The ranking grammar in the case of G_R(1:CV.del) consists of the linear extensions of a partial order. This partial order can be perspicuously displayed as a Hasse Diagram.

[26] In this simple case, each ERC derives from a single winner-loser pair coming from a different input, as may be seen from analyzing the universal support VT (45).

(51) 1:CV.del



The ranking grammar of Language 2:(C)V.del contains only two legs:

(52) Ranking grammar of 2:(C)V.del

m.NoCod	≫	f.dep	≫	f.max	≫	m.Ons
f.dep	≫	m.NoCod	≫	f.max	≫	m.Ons

The linguistic generalizations inherent in this collection are represented in an ERC grammar below.

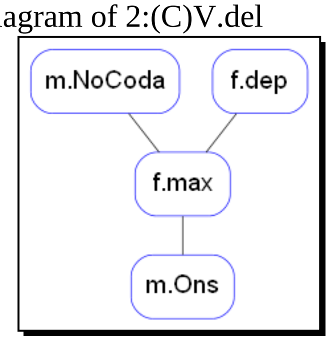
(53) ERC grammar for 2:(C)V.del

ERC#	m.NoCoda	f.dep	f.max	m.Ons
1	**W**		**L**	
2		**W**	**L**	
3			**W**	**L**

Observe that f.dep≫ f.max (ERC#2) secures deletion as the mode of unfaithfulness. ERC#1 requires m.NoCoda≫f.max, ensuring that all optimal syllables are codaless. ERC#3 positions f.max with respect to m.Ons, leading to onsetless syllables via faithful reproduction of the input sequences like /V/ and /CVV/.

Here too the legs linearly extend a partial order, which can be displayed as follows:

(54) Hasse Diagram of 2:(C)V.del



1.2.4 A Unitary VT for EST

The preceding exemplification of the grammars of EST can be fully projected from the three candidate sets given above in (45). Even so, this concrete instantiation of EST will not serve us well when we ask further questions about the typology. We must step away from the concrete and begin to operate in Abstract OT. From Prince 2015, we know that the Minkowski sum[27] of these three candidate sets provides a single unitary VT (UVT), from which all 8 grammars can be produced. In this UVT, shown below, each row represents an entire language, and its grammar is determined by competing for optimality against the other languages.

(55) A Unitary VT for EST

U	**m.Ons**	**m.NoCoda**	**f.dep**	**f.max**
1:CV.del	0	0	0	3
2:(C)V.del	1	0	0	2
3:CV.ins	0	0	3	0
4:(C)V.ins	1	0	2	0
5:CV(C).del	0	1	0	2
6:(C)V(C).del	1	1	0	1
7:CV(C).ins	0	1	2	0
8:(C)V(C).ins	1	1	1	0

With the UVT U in hand, we have transitioned to the abstract. Candidates no longer have the structure of ⟨input, output, correspondence-relation⟩ triples, but are violation profiles labeled with language names. Each constraint is newly reformulated as a function from abstract candidates to $\mathbb{N}$, the set of nonnegative integers {0, 1, 2,…}. In the interests of simplicity, we continue with the same constraint names.

This UVT was produced from the three candidate sets above in (45), but other universal support candidate sets could have been chosen, resulting in different UVTs. The Minkowski sum algorithm, from various universal supports, will produce typologically equivalent UVTs, even if numerical equality is not guaranteed.

We are brought back to the first question raised above in §1, asked there of nGX.IL, now asked of EST. Which of the numerical relations in the UVT are linguistically meaningful, in that they recur in every UVT for the typology? Equivalently, which relations play a crucial role in the process of selecting optima? The MOAT provides the answer.

---

[27] The Minkowski sum of two sets, often denoted by ⊕, is a set consisting of the sum of every pair of elements from the two sets, one from the first, the other from the second. Vectors sum coordinatewise: here adding the violation values in each constraint. In short, $A \oplus B = \{\mathbf{a}+\mathbf{b} | \mathbf{a} \in A, \mathbf{b} \in B\}$, with $(\mathbf{a}+\mathbf{b})[i] = \mathbf{a}[i] + \mathbf{b}[i]$.

2 EST: The MOAT and its EPOs

SubTOC

THREE NOTIONS ARE CENTRAL to construction of the MOAT: order, equivalence, and privilege. Order and equivalence are singled out because of the way that OT chooses optima, privilege because of the way languages class together. Every grammatical typology comes from many typologically-equivalent UVTs, which differ numerically but give rise to identical grammars. The MOAT tells us exactly which order and equivalence relations must hold within any UVT that realizes the typology. The EPOs of a MOAT retain structural information about grammar adjacency that determines the possibilities of typological classification.

This section contains example-based exposition of the key ideas and procedures associated with the MOAT, developing only the basic technical apparatus required for clarity (§2.1-3). In §3, we move on to a more detailed formal analysis in the course of demonstrating the claims advanced here.

We begin with the order and equivalence relations that are shared across all UVTs of the same typology (§2.1). We then show how these relations can be obtained from the grammars of the typology, construed as sets of legs (§2.2). The procedure relies on *border point pairs*, two maximally similar legs that belong to different grammars. These identify the privileged order relations and the equivalences. This information gives a complete account of the shared ranking and order relations. The privileged order and equivalence relations for a single constraint are contained in the EPO ('*E*quivalence-augmented *P*rivileged *O*rder') of that constraint. The collection of EPOs, one for each constraint, is the MOAT of the typology. The MOAT is unique and independent of the peculiarities of any individual UVT. In addition to delimiting the numerical possibilities of the UVTs, the MOAT determines the range of typological classifications based on unioning grammars to form an abstract typology coarser than the original.

2.1 From Order and Equivalence to the EPO

OT filtration is famously sensitive only to the relations of equality and order between violation values, not to the magnitudes of the values themselves. If a given number is minimal for the set of candidates evaluated by a constraint in a ranking, then all candidates *not* assigned that value will be rejected.

Though the decision made by a constraint is binary — in / out — each constraint establishes order and equivalence relations among *all* competing candidates by virtue of its numerical assignments, simply because the integers have their own inescapable logic. In Concrete OT, candidates are linguistic forms, typically input-ouput-correspondence triples, and languages are collections of optimal candidates. In Abstract OT, the candidates are violation profiles, and there is no notion of "language" in the concrete sense. When a UVT is constructed by Minkowski summation of multiple concrete VTs, each of its rows — each abstract candidate — represents an entire language of the concrete typology by virtue of delivering the grammar of that language when chosen as optimal. We label the rows of the UVT in this situation with the names of the corresponding languages. For convenience, at minor risk of ambiguity, we refer to the UVT rows as "languages" rather than "abstract candidates with language-name labels."

Consider the languages of the EST as evaluated in a UVT by the constraint m.Ons. The relevant column from UVT (55) is repeated here, with the rows rearranged to emphasize the behavior of interest. To indicate that the row labels deliver the grammar of the indicated language in the context of this particular UVT, we use a superscript to index them to the UVT they live in.

(56) m.Ons in EST UVT (55)

U, in part	**m.Ons**	Output Type in EST
1:CV.del^{U}	0	Onset Required: OR
3:CV.ins^{U}	0	
5:CV(C).del^{U}	0	
7:CV(C).ins^{U}	0	
2:(C)V.del^{U}	1	Onset Lack Allowed: OLA
4:(C)V.ins^{U}	1	
6:(C)V(C).del^{U}	1	
8:(C)V(C).ins^{U}	1	

Four languages receive '0'. In their concrete extensional versions, every syllable in optimal outputs has an onset; we call these 'Onset Required (OR)'. The other four receive '1'. Concretely, some syllables in some of their optimal outputs are onsetless; in IO terms, neither epenthesis nor deletion is employed in the syllabification of input sequences VV or #V – we call these languages 'Onset Lack Allowed' (OLA). In this particular UVT, all languages of the set OR are given the same value on m.Ons, as are the members of the set OLA. Neither the languages nor their associated grammars are thereby rendered 'equal', because two things that are *equal* are just one thing. The values assigned to the languages define an equivalence relation on grammars.

It is worth our while to spell this out. We notate this relation with '$\approx_{\mathrm{C:U}}$', indicating that it responds to the values assigned by C in the UVT U. Since grammars are the objects that typologies consist of, and the languages or language labels merely players in the drama that leads to them, we want the relation to hold of grammars, not languages. We notate the grammar corresponding to $\mathrm{L}_i^{\mathrm{U}}$ as Γ_i.

(57) UVT-induced Equivalence Relation. For UVT U, and constraint C:

$$\Gamma_j \approx_{\mathrm{C:U}} \Gamma_k \text{ iff } \mathrm{C}(\mathrm{L}_j^{\mathrm{U}}) = \mathrm{C}(\mathrm{L}_k^{\mathrm{U}})$$

The constraint m.Ons induces two equivalence classes of EST grammars, OR and OLA.

(58) Equivalence classes of m.Ons in UVT U (56)

OR: 1:**CV.del** $\approx_{\mathrm{m.Ons:U}}$ 3:**CV.ins** $\approx_{\mathrm{m.Ons:U}}$ 5:**CV(C).del** $\approx_{\mathrm{m.Ons:U}}$ 7:**CV(C).ins**

OLA: 2:**(C)V.del** $\approx_{\mathrm{m.Ons:U}}$ 4:**(C)V.ins** $\approx_{\mathrm{m.Ons:U}}$ 6:**(C)V(C).del** $\approx_{\mathrm{m.Ons:U}}$ 8:**(C)V(C).ins**

For conciseness, we will refer to languages by their position in the enumeration, bolded. Further simplifying by dropping the subscript, we get the following equivalences.

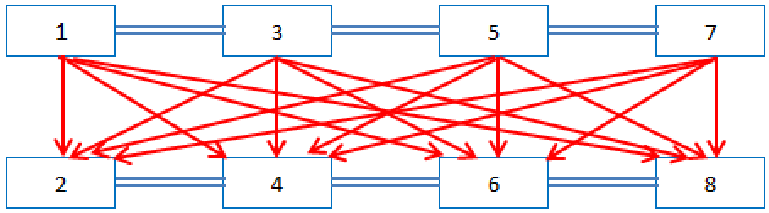
(59) Equivalence classes of m.Ons in UVT U (56), concisely

OR: $\mathbf{1} \approx \mathbf{3} \approx \mathbf{5} \approx \mathbf{7}$

OLA: $\mathbf{2} \approx \mathbf{4} \approx \mathbf{6} \approx \mathbf{8}$

Along the same lines, the purely numerical relation $0<1$ leads to an abstract order relation '$\lessdot_{\mathrm{m.Ons:U}}$' holding between the grammars of OR and those of OLA in UVT U. Both order and equivalence can be portrayed in a bigraph, which here amounts to a Hasse diagram augmented to show equivalence. As always, double blue lines indicate equivalence; red arrows, order.

(60) Bigraph induced by m.Ons in UVT (55)



We may now ask which of these relations are necessary to produce the EST typology. For example, may we disrupt the equivalences of the bottom-dwelling OLA set {**2, 4, 6, 8**}? Suppose a different Universal Support had been used to construct a UVT for EST, or that we had simply managed to write down a UVT with a typology identical to that of EST. Could such a UVT U′ contain the following assignment of values by m.Ons?

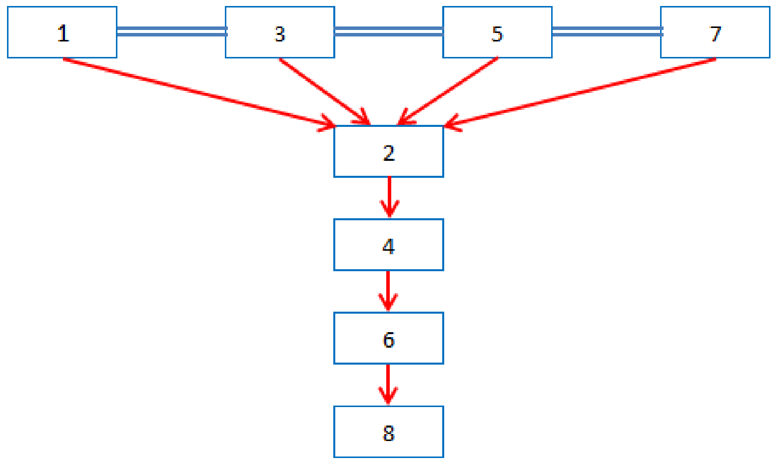
(61) Attempted alternative version of **U**, modified from EST UVT (55)

U′	**m.Ons**	Output Type
1:CV.del$^{U'}$	0	Onset Required: OR
3:CV.ins$^{U'}$	0	
5:CV(C).del$^{U'}$	0	
7:CV(C).ins$^{U'}$	0	
2:(C)V.del$^{U'}$	1	Onset Lack Allowed: OLA
4:(C)V.ins$^{U'}$	2	
6:(C)V(C).del$^{U'}$	3	
8:(C)V(C).ins$^{U'}$	4	

Our notational effusion bears fruit here, allowing us to speak coherently of the one m.Ons while the corresponding languages receive different values from it. This practice clarifies the intuitive idea that different UVTs can deliver the same grammars.

The numerical assignments in U′ represent one of the many possible ways of putting an order on the OLA class while retaining both the equivalences within the OR class (all evaluate to 0) as well as the order relations between the members of the two classes (0 being less than the positive integers assigned to OLA languages). This contemplated rendition of m.Ons gives rise to the following bigraph:

(62) Alternative m.Ons bigraph from U′, with equivalence abandoned in OLA



The shock is that this still works. In any UVT for the EST, we can replace the entries in the m.Ons column with any numerical values that follow the order and equivalence relation in the bigraph (62), including of course those of ex. (61). No filtration ever depends upon distinguishing the members of the OLA class from each other by virtue of their performance on m.Ons.[28]

May one similarly disrupt the equivalences in the OR class {**1, 3, 5, 7**}? Here's one way to do it, among many.

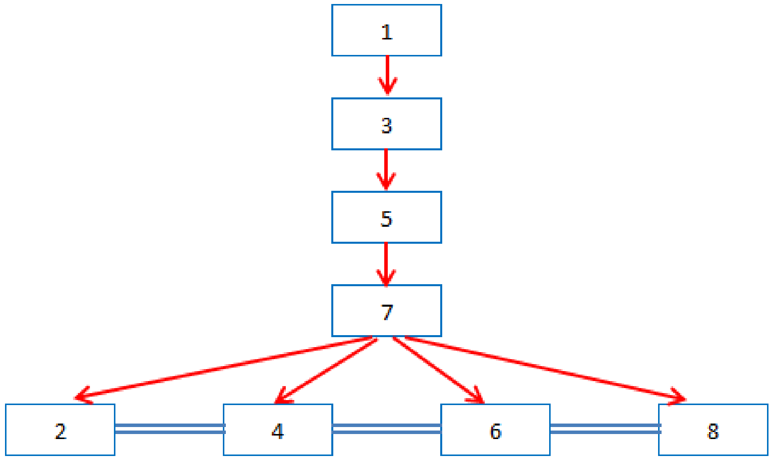
(63) Attempted alternative version of m.Ons (55)

U″	**m.Ons**	Output Type
1:CV.del$^{U''}$	0	Onset Required: OR
3:CV.ins $^{U''}$	1	
5:CV(C).del$^{U''}$	2	
7:CV(C).ins$^{U''}$	3	
2:(C)V.del$^{U''}$	4	Onset LackAllowed: OLA
4:(C)V.ins$^{U''}$	4	
6:(C)V(C).del$^{U''}$	4	
8:(C)V(C).ins$^{U''}$	4	

These assignments, as well as any that have the same order properties, induce the following bigraph:

[28] To see this, return to UVT (55) and contemplate any total order. Follow its filtration pattern. You will find that when an OLA member is optimal the decision between the OLA class members will be made elsewhere, before m.Ons has a chance to distinguish them.

(64) Attempted alternative bigraph for m.Ons in the EST, abandoning OR equivalence



This fails radically and obviously. Suppose m.Ons is at the top of a total order. The only survivor of filtration by m.Ons as rendered in (64) would be 1:CV.del. But all 4 OR grammars include legs with m.Ons undominated, and so all OR languages must pass through m.Ons together when it's the first constrant in a linear order. Thus, in every UVT for EST, the m.Ons column must assign the same numerical value to all OR languages — and that value must be minimal in the column.

To understand the linguistically significant order and equivalence relations inherent in a UVT — those which determine the functioning of the grammar and therefore cannot be altered — we need to know which of them hold not in some single specimen UVT but in *every* UVT that yields the typology under scrutiny. Consider the set of all UVTs which yield the EST typology. The rows are labeled with language names indexed to indicate the UVT they're in, allowing us to correlate their behaviors; the columns are labeled with unadorned constraint names, allowing us to compare grammars directly. Collect from this (infinite) set of UVTs exactly those relations between grammars that appear in each and every one. These relations will be both necessary and sufficient to delimit the numerical values of any UVT that generates the EST.

Formally, the notion of shared relations is delivered by intersection. This is intuitively clear, since intersection is all about sharing. A little development will make this precise. Recall that a relation on a set X is a subset $R \subseteq X \times X$, where we say aRb iff $(a,b) \in R$. The order relation *less than* '<' on $\mathbb{N}$ is subset of $\mathbb{N} \times \mathbb{N}$, which contains pairs such as (0, 1), (57, 300), and so on.

The numerical values in a UVT column induce the order relation $\lessdot_{C:U}$ and the equivalence relation $\approx_{C:U}$ on the grammars of its typology. To be painfully explicit, we'd have to say that the ordered pair of grammars (CV.del, CV.del) *belongs to* the relation $\lessdot_{m.Ons:U,}$ which is a subset of $T \times T$, where T = {CV.del, (C)V.del, …}, the EST typology.

This conceptual stance allows us to talk about the relations that are shared across the entire set of UVTs. To obtain the set of m.Ons order relations that appear in *every* UVT, we ask for the ordered pairs that appear in every relation associated with m.Ons in every UVT for EST. This is the *intersection* of the orders, conceived of as sets of pairs. Similarly for the equivalence relations.

To present the idea concisely, we write it out. Let $\mathcal{U}$(T) be the set of UVTs that yield typology T. For every U∈$\mathcal{U}$(T), let C be a variable ranging over the names of constraint columns. As above, we write $\lessdot_{C:U}$ for the partial order on grammars induced by the numerical values in column C of U. We are interested in the partial order $\lessdot_{C}$ obtained by intersecting the relations $\lessdot_{C:U}$ for every U∈$\mathcal{U}$(T).

(65) Intersection of Order Relations

$$\lessdot_{C} = \bigcap_{U \in \mathcal{U}(T)} \lessdot_{C:U}$$

We gather the equivalence relations that occur in every UVT in the same way, writing as above $\approx_{C:U}$ for the equivalence relation local to one UVT U and $\approx_{C}$ for the intersection.

(66) Intersection of equivalence relations

$$\approx_{C} = \bigcap_{U \in \mathcal{U}(T)} \approx_{C:U}$$

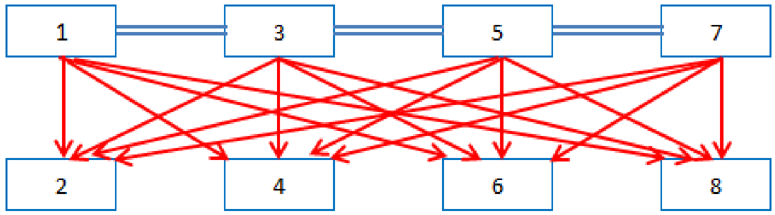
We have then a pair $\langle \lessdot_{C}, \approx_{C} \rangle$ for each constraint column C in a given $\mathcal{U}$(T). We call $\langle \lessdot_{C}, \approx_{C} \rangle$ the "*i*EQO for C", for "***I***ntersection of all ***EQ***uivalence and ***O***rder relations for column C." Collecting together each such object for the columns of $\mathcal{U}$, we have the "*i*OAT" of $\mathcal{U}$(T) — "***I***ntersection ***O***f ***A***ll ***T***ableaux." For the first we will write *i*EQO(C) when we wish to be concise, and *i*EQO_T(C) when it is necessary to draw attention to the typology it lives on. For the second, we will write *i*OAT($\mathcal{U}$(T)).

In our two successful UVT columns (56) and (61) above, the OR grammars are all equivalent on m.Ons. As we argued, this must hold in every UVT in $\mathcal{U}$(EST). Therefore the *i*EQO for m.Ons will declare the OR languages equivalent.

By contrast, our examples suffice to show that the OLA languages share no relations at all on m.Ons: in the lingo of order, they are 'noncomparable'. In bigraph (60), derived from the numbers in (56), all the OLA grammars are equivalent. In bigraph (62), derived from the numbers in (61), they are linearly ordered. No pair of numbers can be both equal and strictly ordered. The intersection of all shared pairwise order relations $\lessdot_{m.Ons}$ across the UVTs of EST therefore contains none that relate any OLA language to any other. The same is true for the intersection of all equivalences: none relate OLA languages.

From these considerations, we deduce that the bigraph showing the order and equivalence relations present in every m.Ons column of the EST UVTs will take the form of (67) below. This accurately represents the *i*EQO for m.Ons. But it is not an EPO because privilege has not been enforced.

(67) $i\text{EQO}_{\text{EST}}$(m.Ons)



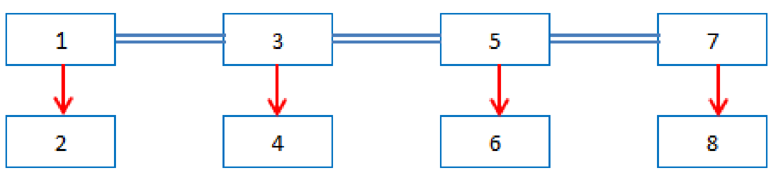
This representation of the *i*EQO records the fact that the members of OLA (2nd row) are neither equivalent nor ordered with respect to each other. As with EPO bigraphs, we do not explicitly show every pairwise equivalence relation.

We've now got a crucial object: the *i*OAT and its constituent *i*EQOs, one for each constraint in the OT system at hand. This object answers the first question posed above: any UVT satisfying its relations will derive the typology; and only those (proved as (136) and (140) in §3.2.3).

To characterize the typological classes of the system, in answer to Problem 3, we must go one step beyond the *i*OAT. We must determine which of the order relations in the *i*EQOs are to be included as essential for classification and which are to be omitted as not only inessential but inimical to it. We will construct the MOAT itself from the typology as delivered by *any* UVT, giving an effective procedure for calculating the relations of the *i*OAT, which are defined nonconstructively as those holding in every member of an infinite class. The next section develops this procedure in detail.

We obtain this EPO for m.Ons:

(68) EPO_{EST}(m.Ons)



The equivalence relations of the *i*EQO are identical to those implied by the EPO. The EPO provides only a proper subset of the *i*EQO order relations. However, the *i*EQO relations derive from those in the EPO through the way that the order and equivalence relations comport with themselves and with each other. The interactions go beyond the familiar transitivity of order, which may be summarized as a<b & b<c ⇒ a<c, writing '<' for any order relation. But because we are in a setoid, we must acknowledge an interaction with equivalence, ensuring that equivalent items have the same order relations: if a∼b and b<c, we want to be able to deduce that a<c. This we call 'hypertransitivity'. The hypertransitive closure of the relations in the EPO, combining order and equivalence, will yield the order relation of the *i*EQO (Theorem (158), §3.2.3).

The effects can be seen in the contrast between the *i*EQO for m.Ons (67) and its EPO (68). Both show the equivalence of the OR languages and the lack of relation amongst the OLA languages. The more profuse order relations in the *i*EQO (67), represented by its 16 arrows, follow from the fact that equivalent items must share order properties if we are to map to the integers to produce a UVT. In a UVT, EPO-equivalent items bear the same numerical value (Lemma (110), §3.1). From the sparser m.Ons EPO, with its 4 relational arrows, it follows by combination with the equivalences that each OR language is ordered above each OLA language.

The EPO of a constraint is thus a kind of skeletal version of its *i*EQO. The *i*EQO contains every relation shared across the entirety of the UVTs that give a typology. The EPO holds the core subset that determines the *i*EQO and limits the joining of grammars into typological classes.

2.2 Building the MOAT & the EPO

Abstract relations of order and equivalence between grammars arise from those numerical assignments to languages that influence the course of filtration in UVTs. These are collected in the *i*OAT, where each constraint of the system is represented by an *i*EQO distilled from the entirety of its numerical possibilities. The MOAT, by contrast, is built not from the arithmetic of UVTs, but from the the grammars themselves, construed as collections of linear orders.

The distribution of legs among the grammars of a typology determines its MOAT. Certain legs stand out as particularly informative: those that differ minimally in constraint ranking and belong to different grammars. Even more strikingly, it turns out that to obtain the privileged relations, we must attend to the sequential structure of such legs. These we call *border points*, and they come in pairs. Here's a typical *border point pair*, with the zone of minimal difference, in which two adjacent constraints transpose, highlighted typographically. To keep track of the relevant memberships as we explore the cases, a leg of grammar *N* will be designated as ⟨*N*.*x*⟩, where *x* is an arbitrary alphabetical or numerical identifier distinguishing the legs of grammar *N*.

(69) **A Border Point Pair in the EST**

Leg	Border Point Pair	Language	Type
⟨**7**.a⟩	f.max ≫ ***m.Ons*** **≫ f.dep** ≫ m.NoCoda	**7**:CV(C).ins	OR, CA
⟨**8**.a⟩	f.max ≫ **f.dep ≫ *m.Ons*** ≫ m.NoCoda	**8**:(C)V(C).ins	OLA, CA

Leg ⟨**7**.a⟩ belongs to the grammar of **7**:CV(C).ins, which requires onsets (OR) in every syllable. Leg ⟨**8**.a⟩ belongs to grammar of **8**:(C)V(C).ins, which allows syllables to lack onsets (OLA) when they arise from input vowels not preceded by a consonant. Otherwise both grammars handle problematic input configurations, those requiring breach of faithfulness in optima, identically. Extensional considerations like these only hint at the true generality of the relationship between these legs, which is revealed by attention to their form. The key is the notion of smallest possible significant difference as it manifests in the world of order and permutation.

A *border point pair* consists of two legs that are identical except for a single adjacent transposition of constraints, where each member of the pair belongs to a different grammar. A pair takes the form {P$\underline{\text{XY}}$Q, P$\underline{\text{YX}}$Q}, where X, Y are individual constraints and P, Q are sequences of constraints, possibly empty. We call the shared sequences P and Q the *prefix* and the *suffix* of the pair, respectively.The sequence $\underline{\text{XY}}$ we call the *transposition* (as we do $\underline{\text{YX}}$), underlining its participating constraints for visibility. We will abbreviate the name to BPP for occasional conciseness.

The members of a border point pair can be thought of as sitting next to each other if we construe the legs of Ord(CON$_S$) as corresponding to points in a space of ranking possibilities, in which an adjacent transposition maps a point into a neighboring point.[29] On this view, grammars are sets of points and they may also be construed as adjacent, by virtue of the adjacency of points they contain. It is natural therefore to speak of 'border points' which inhabit the border region or boundary of a grammar, where a well-chosen minimal local change in constraint order translates into a higher-level change from one grammar to another. The term loses its metaphorical sheen completely in §6, where we examine the geometry behind it. As we will see, a grammar is entirely determined by its border; a typology is determined by the relations established at the borders of its constituent grammars. The notion of the border point pair may be spelled out as follows:

(70) **Definition**. **Border Point Pair**. Let T = $\{\Gamma_1, \ldots, \Gamma_n\}$ be a typology on a set of constraints CON$_T$, given as a set of ranking grammars. Let $\lambda_1 = \mathrm{P}\underline{\mathrm{XY}}\mathrm{Q}$ and $\lambda_2 = \mathrm{P}\underline{\mathrm{YX}}\mathrm{Q}$ be legs over CON$_T$, with P,Q sequences of constraints from CON$_T$ and X,Y$\in$CON$_T$. Then $\{\lambda_1, \lambda_2\}$ is a *border point pair* for $\Gamma_j, \Gamma_k \in$ T, $\Gamma_j \neq \Gamma_k$, iff $\lambda_1 \in \Gamma_j$ and $\lambda_2 \in \Gamma_k$.

Such pairs are unordered: transposition, like adjacency, has no inherent directional bias. A border point pair licenses EPO relations. In the example above, repeated below, relations between grammars **7** and **8** are determined in three different EPOs:

(71) **A Border Point Pair in the EST**

Leg	Border Point Pair
⟨**7**.a⟩	f.max ≫ ***m.Ons* ≫ f.dep** ≫ m.NoCoda
⟨**8**.a⟩	f.max ≫ **f.dep ≫ *m.Ons*** ≫ m.NoCoda

The structure of this pair is interpreted in EPO terms as follows:

- The *prefix* of the pair is **f.max**: prefix constraints impose EPO equivalences ($\sim_{\text{f.max}}$).
- The constraints in the *transposition* impose EPO order relations ($<_{\text{m.Ons}}$, $<_{\text{f.dep}}$).
- The constraints in the *suffix* determine no relations.

(72) **Relations from BPP (71)**

EPO	Relation	Relevant segment	Status	Relation type
f.max	**7** $\sim_{\text{f.max}}$ **8**	f.max ≫ …	prefix	equivalence
m.Ons	**7** $<_{\text{m.Ons}}$ **8**	… *m.Ons* ≫ f.dep…	transposition	order
f.dep	**8** $<_{\text{f.dep}}$ **7**	… f.dep ≫ *m.Ons*…	transposition	order
m.NoCoda	*none*	… ≫ m.NoCoda	suffix	*none*

[29] The use of transposition in this way is taken from the analysis of the 'symmetric group' S_n, the group of all permutations of a set S of n objects. Adjacent permutation is a generator for S_n; its Cayley graph is embodied in the permutohedron (§6).

For a transposition $\underline{\text{XY}}/\underline{\text{YX}}$, the leg P$\underline{\text{XY}}$Q sponsors an order relation in EPO(X) that sets its grammar as '$<_{\text{X}}$' with respect to the grammar of its P$\underline{\text{YX}}$Q-containing neighbor. Similarly for P$\underline{\text{YX}}$Q in EPO(Y). For every constraint Z in P, the grammars are deemed equivalent in EPO(Z), standing in the relation '$\sim_{\text{Z}}$'.

These are abstract ascriptions, reflecting the order of constraints in legs at the border. But they are echoed in the way that candidates are filtered by the numerics of the UVTs: this gives them their interest and their potency. To see how it works, it is instructive to pursue the details of filtration by legs $\langle$**7**.a$\rangle$ and $\langle$**8**.a$\rangle$. Filtration of UVTs proceeds in exactly the same way as in familiar concrete OT. The top-ranked constraints accepts those candidates to which it gives the minimal value and disregards the rest. And so on down the hierarchy, at each step dealing in the same way with the survivors of the previous step. Since the candidates are languages, filtration by a leg chooses a language as optimal and therefore assigns the filtering leg to the grammar of the language it selects.

First up: f.max. The legs in the border point pair start off the same way:

$\langle$**7**.a$\rangle$ = **f.max**$\gg$… $\in$ **7**
$\langle$**8**.a$\rangle$ = **f.max**$\gg$… $\in$ **8**

Languages **7** and **8**, whatever their numerical values, must both pass through f.max in order to show up as optimal in the final reckonings. This happens in UVT (55) because f.max(**7**) = f.max(**8**) = 0, the minimal value.

This is no quirk: equality must hold in *all* UVTs that deliver the EST. To test this claim, suppose that in some putative UVT, we found instead f.max(**7**) $<$ f.max(**8**). In this case, language **8** would be ejected right away by any leg beginning with f.max, because **8** would bear a nonminimal value. But we know from the concrete EST that leg $\langle$**8**a$\rangle$ belongs to the grammar of **8**, so this inequality cannot be tolerated. Similarly, an inequality in the other direction ejects $\langle$**7**a$\rangle$from grammar **7**. With neither inequality allowed, equality is the only remaining option.

When we add the abstract equivalence **7** $\sim_{\text{f.max}}$ **8** to EPO(f.max), we may therefore rest secure in the knowledge that this relation will be instantiated as numerical equality in every UVT for EST, although we have looked at none of them.

Next up: m.Ons and f.dep. Here the two legs differ.

$\langle$**7**.a$\rangle$ = f.max $\gg$ <u>**m.Ons**$\gg$ f.dep</u> $\gg$ … $\in$ **7**
$\langle$**8**.a$\rangle$ = f.max $\gg$ <u>**f.dep**$\gg$ m.Ons</u> $\gg$ … $\in$ **8**

MOATwise, we add to EPO(m.Ons) the relation: **7** $<_{\text{m.Ons}}$ **8** and to EPO(f.dep) the relation **8** $<_{\text{f.dep}}$ **7**. Filtrationwise, these abstract relations correlate with the following actions, which take place at the leading constraint in the transposition:

- On$\langle$**7**.a$\rangle$, m.Ons selects language **7** from the set {**7**, **8**} and ejects **8**.
- On $\langle$**8**.a$\rangle$, f.dep selects language **8** from the set {**7**, **8**} and ejects **7**.

Looking back to the numbers in UVT (55), we see that selection of **7** / rejection of **8** by $\langle$**7**a$\rangle$happens because m.Ons(7) $<$ m.Ons(8). Selection of **8** / rejection of **7** by $\langle$**8**a$\rangle$ is similarly due to the fact that f.dep(**8**) $<$ f.dep(**7**) numerically in (55). Let's confirm that these relations must occur in every UVT for EST by running through — and dismissing — the alternatives.

Suppose first that the relative magnitudes were reversed: that m.Ons(**8)** $<$ m.Ons(**7**) in some claimed UVT. Then **7** is going to be ejected by $\langle$**7**a$\rangle$ when m.Ons is reached in the filtration, because its value is nonminimal. According to this VT, $\langle$**7**.a$\rangle\notin$**7**, contrary to the facts of the EST. Fail! So this order can't be reversed.

Experimenting further, suppose that we were to set m.Ons(**7**) = m.Ons(**8**). Now m.Ons doesn't distinguish **7** from **8**. But this means that the entire transposition zone can't distinguish them either: in selecting from {**7**, **8**}, the ranking m.Ons $\gg$ f.dep gives the same results as f.dep $\gg$ m.Ons when m.Ons treats **7** and **8** as equals. Therefore, under this hopeless assumption, $\langle$**7**.a$\rangle$ and $\langle$**8**.a$\rangle$would yield the same optimum, since the only difference between them has been erased. But in reality they give different results: a fact we cannot escape. Equality is out, and we must have everywhere m.Ons(**7**) $<$ m.Ons(**8**).

For parallel reasons, the f.dep EPO relation **8**$<_{\text{f.dep}}$**7** derived from border point analysis will correlate in every UVT with numerical f.dep(**8**)$<$f.dep(**7**).

Last up: m.NoCoda.

$\langle$**7**.a$\rangle$ = f.max$\gg$ <u>m.Ons$\gg$ f.dep</u> $\gg$ **m.NoCoda** $\in$ **7**

$\langle$**8**.a$\rangle$ = f.max $\gg$ <u>f.dep$\gg$ m.Ons</u> $\gg$ **m.NoCoda** $\in$ **8**

No further consequences follow, since all decisions have been made for the set {**7**,**8**}. Here there is only one constraint in the suffix, but in the general case, we know that any permutation of the constraints in a suffix will produce the same outcome, yielding no ranking information, because the decision between the competing pair of languages has been made at the transposition. This completes the analysis of the BPP $\langle$**7**.a$\rangle$,$\langle$**8**.a$\rangle$.

We have conducted this investigation among the particulars of UVT (55), but in every case we were able to argue that the results apply generally across the entire set of UVTs. We know that each UVT for the EST assigns the same legs to its grammars, by the definition of UVT. But the discussion has disclosed a pervasive broader generalization: every UVT for a given typology exhibits the same *filtration patterns*, working through each leg constraint by constraint from highest ranked to lowest. By the *filtration pattern* of a ranking λ with respect to a collection of candidates, we mean the telescoping sequence of accepted subsets obtained in the course of filtering it by λ. As exemplified here and as shown for the general case in Lemma (123), p. 88, §3.2, any change in filtration patterns, even at intermediate stages, will involve a reassignment of legs to grammars, yielding a different typology.

Even more strikingly, it turns out that to obtain the privileged relations between grammars, which determine the possible typological classes, we need only attend to the border point pairs. To build the MOAT in its entirety, we examine every border point pair in the manner just laid out, gathering the relevant information for each of the EPOs.

The border point pair (69) gives a quantum of information about EPO(m.Ons). Let's complete that EPO, examining first the following three border point pairs.

(73) **Three BPPs in EST**

Leg	Border Point Pair	From Lg.	Type
⟨**1**.a⟩	f.dep ≫ m.NoCoda ≫ ***m.Ons*** **≫ f.max**	1:CV.del	OR, CP
⟨**2**.a⟩	f.dep ≫ m.NoCoda ≫ **f.max ≫ *m.Ons***	2:(C)V.del	OLA, CP
⟨**3**.a⟩	m.NoCoda ≫ f.max ≫ ***m.Ons*** **≫ f.dep**	3:CV.ins	OR, CP
⟨**4**.a⟩	m.NoCoda ≫ f.max ≫ **f.dep ≫ *m.Ons***	4:(C)V.ins	OLA, CP
⟨**5**.a⟩	f.dep ≫ ***m.Ons*** **≫ f.max** ≫ m.NoCoda	5:CV(C).del	OR, CA
⟨**6**.a⟩	f.dep ≫ **f.max ≫ *m.Ons*** ≫ m.NoCoda	6:(C)V(C).del	OLA, CA

The three pairs in (73), along with the pair {⟨**7**.a⟩,⟨**8**.a⟩in (71), establish all four privileged order relations between the languages of EST on the constraint m.Ons.

(74) **Privileged relations of EST on m.Ons**

From {⟨**1**.a⟩,⟨**2**.a⟩}: $\mathbf{1} <_{\text{m.Ons}} \mathbf{2}$

From {⟨**3**.a⟩,⟨**4**.a⟩}: $\mathbf{3} <_{\text{m.Ons}} \mathbf{4}$

From {⟨**5**.a⟩,⟨**6**.a⟩}: $\mathbf{5} <_{\text{m.Ons}} \mathbf{6}$

From {⟨**7**.a⟩,⟨**8**.a⟩}: $\mathbf{7} <_{\text{m.Ons}} \mathbf{8}$

It remains to establish the EPO equivalences of m.Ons. By examining other border point pairs which have prefixes containing m.Ons, we can cull those languages which must share an m.Ons value. The following three pairs will suffice. In each border point pair, the prefix precedes the underlined transposition, and m.Ons is highlighted in blue.

(75) **Border point pairs establishing equivalences** on m.Ons

Leg	Border Point Pair	From Lg.	Type
⟨**1**.b⟩	**m.Ons** ≫ m.NoCoda ≫ f.dep ≫ f.max	**1**:CV.del	OR, CP
⟨**3**.b⟩	**m.Ons** ≫ m.NoCoda ≫ f.max ≫ f.dep	**3**:CV.ins	OR, CP
⟨**5**.a⟩	f.dep ≫ **m.Ons** ≫ f.max ≫ m.NoCoda	**5**:CV(C).del	OR, CA
⟨**1**.c⟩	f.dep ≫ **m.Ons** ≫ m.NoCoda ≫ f.max	**1**:CV.del	OR, CP

⟨**7**.b⟩ **m.Ons** ≫ <u>*f.max* ≫ f.dep</u> ≫ m.NoCoda **7**:CV(C).ins OR, CA
⟨**5**.b⟩ **m.Ons** ≫ <u>f.dep ≫ *f.max*</u> ≫ m.NoCoda **5**:CV(C).del OR, CA

The members of each competing pair of adjacent grammars register as equivalent on all constraints in the prefix. The pairs above license the following relations in EPO(m.Ons):

$\mathbf{1} \sim_{\text{m.Ons}} \mathbf{3}$
$\mathbf{1} \sim_{\text{m.Ons}} \mathbf{5}$
$\mathbf{5} \sim_{\text{m.Ons}} \mathbf{7}$.

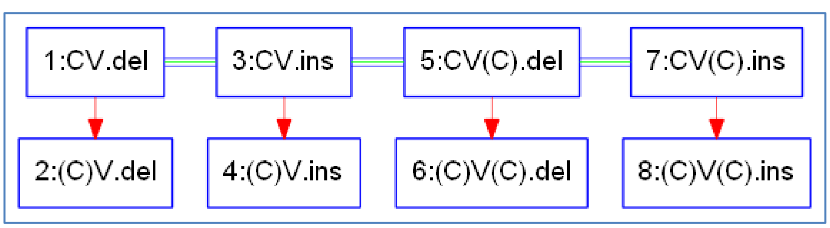
An equivalence relation is symmetric, reflexive, and transitive. Symmetry is guaranteed by prefixal status and by the fact that border point pairs are unordered, so that each of the above may be written the other way around, as e.g. $\mathbf{3} \sim_{\text{m.Ons}} \mathbf{1}$. To achieve full-scale equivalence, the relation $\sim_{\text{m.Ons}}$ must be augmented to reflexivity and transitivity. Once this is done, it follows that the OR languages **1, 3, 5, 7** are all equivalent in EPO(m.Ons).

(76) EPO(m.Ons): equivalent languages
$\mathbf{1} \sim_{\text{m.Ons}} \mathbf{3} \sim_{\text{m.Ons}} \mathbf{5} \sim_{\text{m.Ons}} \mathbf{7}$

Strikingly, there is no border point pair distinguishing **1** and **7** which has m.Ons in its prefix. (See Appendix I for a listing of all legs of the EST with their associated grammars). Nevertheless, we are certain of their relationship.

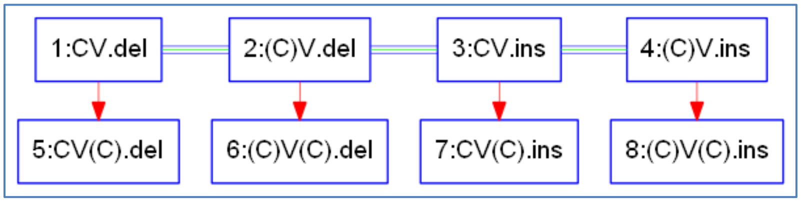
In an EPO diagram, we do not attempt to represent the relation 'equivalent by virtue of some border point prefix'. Equivalence, however derived, is portrayed in any of several convenient layouts, from which all further pairwise equivalences may be easily derived by transitivity. We repeat the EPO diagram given above for m.Ons, which lays out the equivalences and privileged orders just derived.

(77) EPO(m.Ons) in the EST

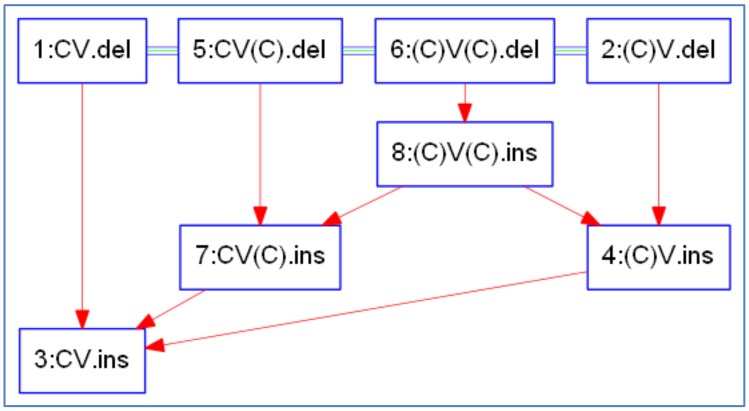


Continuing in this fashion, we can derive every EPO for the EST. Here are the remaining three, completing the MOAT with one EPO for each constraint in the system.

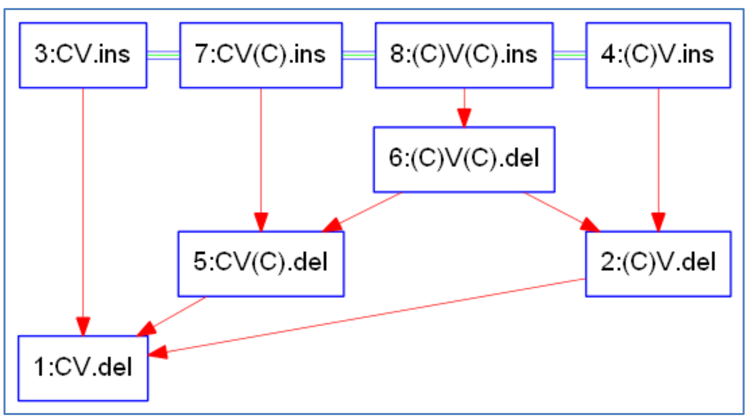
(78) **EPO(m.NoCoda)** in the EST



(79) **EPO(f.dep)** in the EST



(80) **EPO(f.max)** in the EST

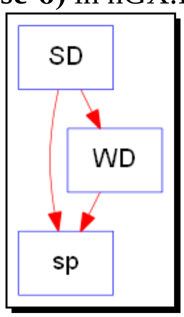


With the full MOAT assembled, it becomes clear that there are striking symmetries in the structure of the typology. Visible in EPOs (77) and (78) is the parallelism between m.Ons and m.NoCoda. EPOs (79) and (80) show a similarly close relation between f.dep and f.max. Concretely, these isomorphisms are prefigured in the formal symmetry of the definitions in CON$_{EST}$ (44), but the cooperation of GEN$_{EST}$ (42) is required to achieve the results. EPO analysis tells us exactly how the definitional parallels play out in the typology and, in the general case, will identify symmetries that may not be otherwise perspicuous.

We conclude with an observation about structural properties of the EPO. Each EPO comes ultimately from the analysis of the border point pairs. The assignment of legs to grammars comes from a UVT, via the general OT definition of optimality. As we have glimpsed in our example, and as we will show for the general case in §§3.1-2, the EPO relations are instantiated as numerical relations in UVTs. They must therefore be consistent with some partial order on the grammars, because they ultimately cash out as the ordered values of a UVT column. The local pairwise relations between grammars determined at their borders must extend to order and equivalence relations that combine together in a way that comports with the way numerical order and equality combine. Graphically, then, a valid EPO cannot have cycles.

The *privileged* relations are all and only those relations of order that come directly from border point pairs. Returning to our first example (27) from the system, nGX.IL we can now see why the EPO of Parse-σ contains the non-Hassean transitively-derivable arrow on the left: it represents a relation that arises at the border of **SD** and **sp**.

(81) **EPO(Parse-σ)** in nGX.IL



Observe that there is no (directed) cycle here, since all arrows point in the same direction: downward.

The link between **SD** and **sp** signals the existence of a border point pair spanning the languages. A little inspection shows it to be the following.

(82) **Border Point Pair, SD and sp in nGX.IL**

SD: ***Parse-σ*** $\gg$ **AFL** $\gg$ Iamb

sp: **AFL** $\gg$ ***Parse-σ*** $\gg$ Iamb

This establishes that the relation **SD**$<_{\text{Parse-}\sigma}$ **sp** is privileged. This fact will have no direct effect on which UVTs are admitted by the EPO, because it follows from other facts that do, namely **SD**$<_{\text{parse-}\sigma}$**WD** and **WD**$<_{\text{parse-}\sigma}$**sp**. We retain it, contra Hasse, in the interests of making apparent in the EPO diagram the structure of privilege that lies within the contents of the EPO object. When we look into the geometry behind the algebra in §6, we will find considerable utility in this decision, because it allows adjacency to be recoverable from EPO diagrams.

In sum: an EPO is a certain kind of bigraph derived from analysis of the border point pairs of a typology. Since any UVT of a typology must give rise to the same border point pairs, the EPO is invariant across all equivalent UVTs. More than that, the EPO relations delimit the entire range of UVT possibilities. The EPO contains the privileged relations that lead to an order on the grammars of the typology; it must therefore be *acyclic*. Acyclicity diagnoses whether a partition is a typology. Acyclicity and privilege determine the possibilities of typological classification. [30]

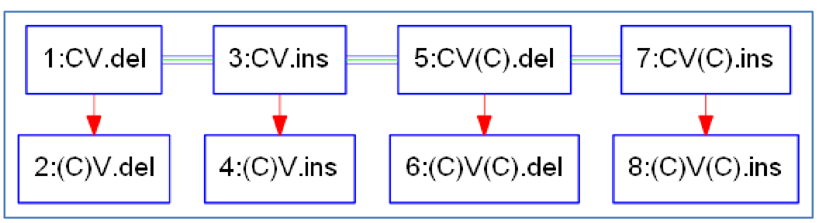
2.3 Why Privilege?

The privileged relations, derived directly from border point pairs, provide the data for determining typological classes from a MOAT. The non-privileged relations, which arise by combining order and equivalence information, are excluded from the MOAT because they do not obstruct the formation of typological classes. In this section, we examine cases that show the force of this crucial distinction.

A numerical UVT will typically induce many pairwise relations of ordering. Some of these follow from the EPO relations and will therefore hold across the set of all typologically-equivalent UVTs. These will appear in the *i*EQO, which records all shared relations. But if we want to use the MOAT for typological analysis, we must omit these universally-present but non-privileged relations and disregard them in typological calculations.

To see why this must be so, it is instructive to re-examine EPO(m.Ons).

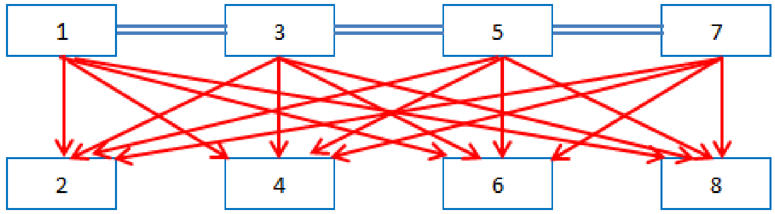
(83) **EPO(m.Ons) in the EST**



The instructive comparison is with *i*EQO(m.Ons), repeated from (67), which directly represents every numerical ordering relation shared by the UVTs of the EST.

[30] These assertions are demonstrated below. EPOs delimit: Theorem (159), §3.2. The privilege relations extend to an order on grammars: Lemma (111),§2.3. Acyclicity described: p. 35, §0.3.2; defined: Definition (176), §3.3.1. When a partition is a typology: Theorem (189), §3.3.3. Acyclicity and typological classification: §5.

(84) *i*EQO(m.Ons) in the EST



Equivalence holds in both EPO and *i*EQO between languages **1**, **3**, **5**, and **7**. Concretely, they are the languages Onset Required (OR) in which every optimal form has only C-initial syllables: the constraint m.Ons is never violated in their optima. The order-contributing arrows have a more interesting relationship. The privileged order relations in the EPO(m.Ons), as we have seen in §2.3, are just these:

(85) **Privileged relations in m.Ons**, various incarnations

EPO	UVT U (55)	*i*EQO
$\mathbf{1} <_{\text{m.Ons}} \mathbf{2}$	m.Ons(**1**) < m.Ons(**2**)	1:CV.del $\lessdot_{\text{m.Ons}}$ 2:(C)V.del
$\mathbf{3} <_{\text{m.Ons}} \mathbf{4}$	m.Ons(**3**) < m.Ons(**4**)	3:CV.ins $\lessdot_{\text{m.Ons}}$ 4:(C)V.ins
$\mathbf{5} <_{\text{m.Ons}} \mathbf{6}$	m.Ons(**5**) < m.Ons(**6**)	5:CV(C).del $\lessdot_{\text{m.Ons}}$ 6:(C)V(C).del
$\mathbf{7} <_{\text{m.Ons}} \mathbf{8}$	m.Ons(**7**) < m.Ons(**8**)	7:CV(C).ins $\lessdot_{\text{m.Ons}}$ 8:(C)V(C).ins

Strikingly, as we have noted, no relations at all hold between the languages allowing onsetless syllables: **2**:(C)V.del, **4**:(C)V.ins, **6**:(C)V(C).del, **8**:(C)V(C).ins.

The *i*EQO contains the same equivalence relations as the EPO and a complete spell-out of all order relations that follow from combining the equivalence and order information (Theorem (158), §3.2).

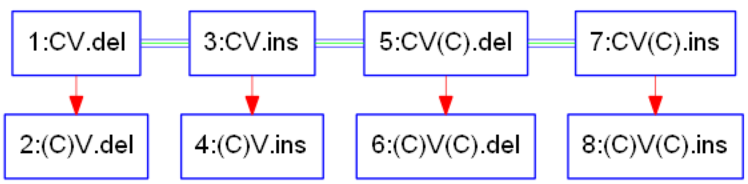
From the EPO equivalences $\mathbf{1} \sim_{\text{m.Ons}} \mathbf{3} \sim_{\text{m.Ons}} \mathbf{5} \sim_{\text{m.Ons}} \mathbf{7}$ taken with the privileged orders in the EPO, we can deduce that in any UVT interpretation of m.Ons every member of {**1**,**3**,**5**,**7**} must be numerically less than every member of {**2**,**4**,**6**,**8**}, exactly as portrayed in the *i*EQO. But the relations portrayed by slanting arrows in (84), such as those between **7** and **2**, **4**, **6,** are nowhere distinguished in the *i*EQO from privileged EPO relations like that between **7** and **8**, the only one involving **7** that comes directly from a border point pair. There is no way of numerically distinguishing these two classes of relations: if a<b and b = c, then we are stuck with a<c. But a special move is required to connect an abstract relation $<_C$ with another abstract relation $\sim_C$ to achieve the parallel conclusion: equivalence is not equality; we decline to close the gap in the MOAT itself.

To see how the privileged order $\mathbf{7} <_{\text{m.Ons}} \mathbf{8}$ restricts typological classification, let's examine the effects of amalgamating **5**:CV(C).del with **8**:(C)V(C).ins in an ill-fated attempt to classify them together to the exclusion of all others.

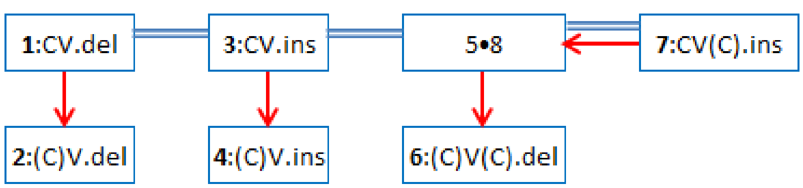
In terms of ranking content, classifying a set of languages together means unioning their legs. Given the EPO bigraph, such a union corresponds to merging the nodes that bear the language names. The node **5•8** thus corresponds to a partition block that contains all and only the legs of grammars **5** and **8**, namely **5**∪**8**. As above, we write N•M for graphical merger of nodes labeled N and M, and N∪M for the union of the legs.[31] Graphical node merger retains all external edges that connect the merged nodes with other nodes, just as union of blocks retains all border points with grammars outside the union. We will see that scrutiny of the border point pairs involving **5**∪**8** confirms the graphical analysis conducted within the EPO.

The effect of merging **5** and **8** in EPO(m.Ons) is seen the bigraph (87). It fails to support a UVT, because it is has a cycle and therefore fails to assert order and equivalence relations that can be realized numerically. No constraint can assign the same number to **7** and **5•8** while at the same time assigning a smaller number to **7** than to **5•8.** For ease of comparison, we repeat the rendering of EPO(m.Ons) from (83).

(86) **EPO(m.Ons) in EST**



(87) **A bigraph tangled by merger**



This merged bigraph is visibly a non-EPO, because of the relations between **5•8** and **7**. The cycle arises from the fact that **5** ═ **7** and **8**←**7**. The node **5•8** inherits both.

(88) **Birth of a bigraph cycle**

Relation in (87)	Source in EPO bigraph (83)	Source in EPO(m.Ons)
5•8 ═ **7**	**5** ═ **7**	**5** $\sim_{\text{m.Ons}}$ **7**
7 → **5•8**	**7** → **8**	**7** $<_{\text{m.Ons}}$ **8**

[31] We tolerate the mild ambiguity of letting the symbols N,M refer to nodes in N•M and legs sets in N∪M.

The class denoted by **5•8** acquires its equivalence with **7** because it incorporates **5**, which is equivalent to **7**. From its **8** component, **5•8** acquires the order relation ‘greater than **7**’. But this cycle defies numerical instantiation.

To see how how border point analysis leads to the same result in the coarsened partition containing the the unioned grammars, we list the legs of **5**∪**8**, which are simply those of **5** and **8** lumped together. Leg designations are retained from §2.2.

(89) All legs of **5**∪**8**, from **5**:CV(C).del and **8**:(C)V(C).ins

Name	Source	Leg
⟨**5**∪**8**.i⟩	⟨**5**.a⟩	f.dep ≫ m.Ons ≫ f.max ≫ m.NoCoda
⟨**5**∪**8**.ii⟩	⟨**5**.b⟩	m.Ons ≫ f.dep ≫ f.max ≫ m.NoCoda
⟨**5**∪**8**.iii⟩	⟨**8**.a⟩	f.max ≫ f.dep ≫ m.Ons ≫ m.NoCoda
⟨**5**∪**8**.iv⟩	⟨**8**.b⟩	f.max ≫ f.dep ≫ m.NoCoda ≫ m.Ons
⟨**5**∪**8**.v⟩	⟨**8**.c⟩	f.dep ≫ m.Ons ≫ f.max ≫ m.NoCoda

Here are the legs of **7**:

(90) Legs of **7**:CV(C).ins

Name	Leg
⟨**7**.a⟩	f.max ≫ m.Ons ≫ f.dep ≫ m.NoCoda
⟨**7**.b⟩	m.Ons ≫ f.max ≫ f.dep ≫ m.NoCoda

To analyze the border point pairs linking **5**∪**8** and **7** requires little more than observing that **5**∪**8** simply inherits the relations with **7** that **5** and **8** have individually. *The legs that determine these relations are still there*. Thus:

● The equivalence **5**∪**8** $\sim_{\text{m.Ons}}$ **7** comes from the relation between ⟨**7**.b⟩ and ⟨**5**∪**8**.ii⟩. We’ve already encountered this in the border point analysis that established the m.Ons EPO, because ⟨**5**.b⟩ and ⟨**7**.b⟩ are a border point pair for the EST, examined in ex. (75) and repeated here.

(91) Equivalence on m.Ons

Legs	Border Point Pair	From Lg.
⟨**5**∪**8**.ii⟩ = ⟨**5**.b⟩	**m.Ons** ≫ <u>f.dep ≫ f.max</u> ≫ m.NoCoda	**5**:CV(C).del
⟨**7**.b⟩	**m.Ons** ≫ <u>f.max ≫ f.dep</u> ≫ m.NoCoda	**7**:CV(C).ins

● The privileged relation **5**∪**8** $<_{\text{m.Ons}}$ **7** comes from the relation between ⟨**8**.a⟩ =⟨**5**∪**8**.iii⟩ and ⟨**7**.a⟩, seen previously in the border point analysis of ⟨**7**.a⟩,⟨**8**.a⟩, ex. (71).

(92) **Relations on m.Ons**

Name	Border Points	From Lg.
⟨**7**.a⟩	f.max ≫ ***m.Ons*** ≫ **f.dep** ≫ m.NoCoda	**7**:CV(C).ins
⟨**5**∪**8**.iii⟩ = ⟨**8**.a⟩	f.max ≫ **f.dep** ≫ ***m.Ons*** ≫ m.NoCoda	**8**:(C)V(C).ins

But this is not consistent with the definitions of $<_X$ and $\sim_X$ for a constraint X. To track its dire consequences, we perform the following deduction among with integers.

(1) m.Ons(⟨**5**∪**8**⟩) = m.Ons(**7**) in every UVT,
(2) m.Ons(⟨**5**∪**8**⟩) < m.Ons(**7**) in every UVT
(3) m.Ons(**7**) < m.Ons(**7**) in every UVT. from (1) & (2)

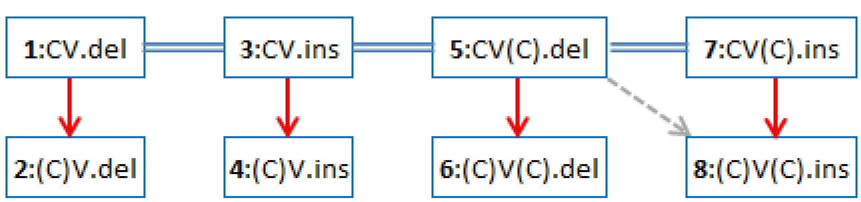
Since no integer is strictly smaller than itself, the conclusion (3) invalidates the premises. We conclude that the relation '$<_{\text{m.Ons}}$' established from the border point pairs of **5**∪**8** and **7** cannot support an order in any numerical instantiation. There is no UVT that yields a typology that is like the EST except for containing **5**∪**8** in place of **5** and **8**.

We have now shown this in two ways. First, by examining the bigraph created by merging **5** and **8** into **5•8.** Second, by analyzing the border points pairs on the coarsened partition that contains **5**∪**8**. Since every typology has a MOAT, we conclude that there is no typology EST′ coarsened from EST by unioning **5** and **8**. Conclusion: **5**∪**8** is not a typological class.

What then of the non-privileged orders that universally accompany the privileged orders? These do not come from border point pairs. They are derivable from an EPO only by combining equivalence and order information. To see how this works out, let's take another look at the **5-7-8** nexus.

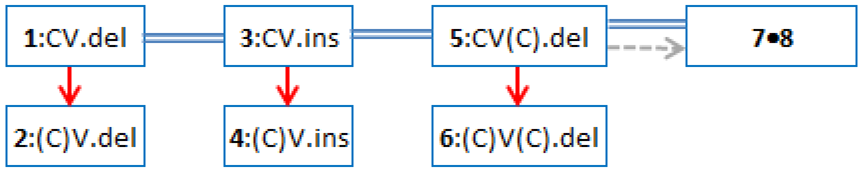
It is assuredly the case that in every UVT m.Ons(**5**) < m.Ons(**8**). We mark this in the following bigraph, which expands the m.Ons EPO diagram in the direction of the *i*EQO. The dashed grey arrow indicates the nonprivileged relation between **5** and **8**.

(93) **EPO(m.Ons) augmented** to include some non-privileged information



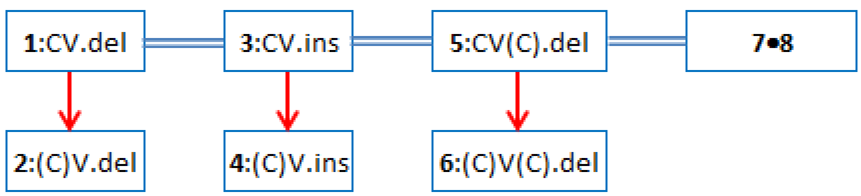
If the gray arrow were taken into account, it would obstruct the merger of **7** and **8** into the class **7•8**. In the EST, this is the class CA.ins: codas allowed, with all unfaithfulness handled by insertion. It generalizes over CV(C).ins and (C)V(C).ins, abstracting away from the different treatment of onsets in the two languages.

(94) **Merger including nonprivileged information**



The cycle that appears between **5** and **7**•8 is illusory. The grammatical class **7**∪**8** is a typologically valid union, and the m.Ons EPO raises no objection. Note that the red arrow between **7** and **8** is a purely internal matter which disappears with the merger. When it disappears, the derived relation indicated by the grey arrow disappears with it.

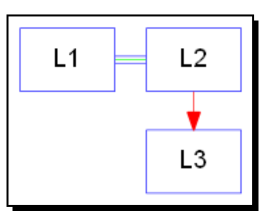
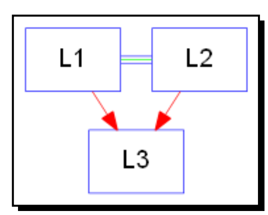
(95) **Merger with only privileged info**



Because there is no border point between **5** and **8**, there is nothing about their relationship that is retained in the merger, and everything runs smoothly. More generally, it is the lack of a border-point based, **5**-to-**8** directed path external to the merger that ensures the typological validity of the **7●8** class. [32]

Let's conclude the discussion by looking at two related but perhaps subtler points. In §1, (33) we noted the constrast between the following EPO bigraphs. These have a different structure but identical numerical instantiations.

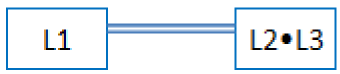
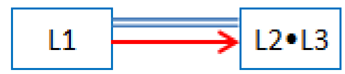
(96) **Distinct EPOs**

EPO(C_1)	EPO(C_2)
	

[32] General bigraphs can contain more elaborate structures like A→B=C→D=E→F. Here, the merger A●F creates a cycle that is not instantiable in numbers. The path between A and F is external to the merger.

In EPO(C_1), the languages L_1 and L_3 are not linked by a border point pair. Nevertheless, it must be that $C_1(L_1) < C_1(L_3)$ in every UVT that instantiates this EPO. By constrast, in EPO(C_2), L_1 and L_3 are so linked. The typological consequence is that EPO(C_1) allows the merger of L_1 and L_3, whereas EPO(C_2) obstructs it, as may be seen when the merger is attempted.

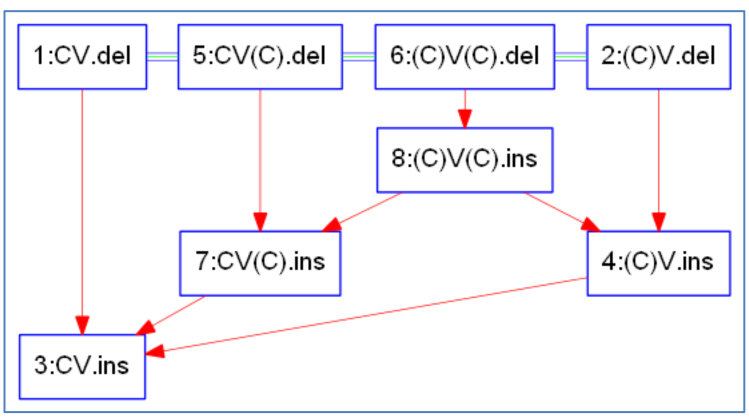
(97) **Mergers in the EPOs**

From EPO(C_1) with merger	From EPO(C_2) with merger
	

Finally, we note that relations derived solely by transitivity from privileged relations, without certification by border point pairs, are absent from the EPO drawing. They obstruct classification because they induce cycles. Schematically, if an EPO contains a path L→M→N, it's clear the the merger L•N is disallowed, whether or not L and N share a border point pair.[33]

Consider the f.dep EPO of the EST.

(98) **EPO(f.dep)** in the EST



Focus on the downward slanting sequence **8**→**7**→**3**. On grounds of privilege alone, we must have numerical f.dep(**8**) < f.dep(**7**) < f.dep(**3**) in every UVT. This follows because of the pairwise privileged relations shown in the EPO (98).

EPO	UVT numerics	EPO bigraph
$\mathbf{8} <_{\text{f.dep}} \mathbf{7}$	f.dep(**8**) < f.dep(**7**)	**8** → **7**
$\mathbf{7} <_{\text{f.dep}} \mathbf{3}$	f.dep(**7**) < f.dep(**3**)	**7** → **3**

[33] As per definition (113), the EPO itself, in contrast to the drawing, contains the relations derived by transitivity from the privileged relations. The EPO relation $<_C$ is a partial order and must be transitive. The drawing is simplified for convenience, without loss of information.

We can immediately deduce that **8∪3** cannot be a typological class. The merger **8•3** creates the cycle **8•3→7→8•**3, which defies numerical instantiation.

The privileged relations are the orders that arise from relations entailed by the transpositions in border point pairs. EPO equivalences arise from the prefixes. The interaction of equivalence and order — ‘hypertransitivity’ — impacts numerical realization in UVTs, but is not privilege. Inclusion of nonprivileged information in a bigraph, as we have seen, disables its relevance to typological classification. The privileged relations, along with the equivalences, suffice to exactly delimit the cases in which merger leads to exit from the realm of typologies.

3 Analysis of the MOAT

SubTOC

Above the forest of the parakeets,
A parakeet of parakeets prevails…

THE UVTS OF A TYPOLOGY DIFFER NUMERICALLY, but the MOAT is invariant: it derives from border point pairs, which are the same no matter what VT or VTs are used to produce the typology. The first goal of this section, carried out in §3.1 and §3.2, is to show that MOAT(T) is not merely unique for a given typology T, but that it *characterizes* T: if a typology has the same MOAT as another, then they are the same typology. Although the MOAT derives from a subset of the legs of T, it determines the full content of every grammar in T. This gives us license to argue, as we have done, from the properties of the MOAT to the properties of the typology it represents.

We take the MOAT concept a step further in §3.3. Not all partitions of Ord(CON_S) are typologies: far from it. How do we know whether a given partition is a typology? To satisfy the definition directly, we might aim to display a UVT that yields it. The method of border point analysis offers another avenue. It applies to any partition and yields a relational structure we call the Generalized MOAT or GMOAT. In §3.3, we show that *acyclicity* of the GMOAT is all that's needed to certify that its sponsoring partition is a typology. The typology qua abstract object is therefore a certain kind of order structure, above and beyond the numerics of the VT.

We conclude by extending ERC logic to interpret the data from the border point pairs of a grammar (§3.4). A border point pair supplies the familiar values for constraints appearing in the prefix (*e*) and the transposition (W,L) but yields no information about the ranking of suffixal constraints. This fourth, undetermined state we reify as a new value u, giving rise to the 4-valued *ERCoid*. Treating u as the identity for fusion, we show how the Fusional Reduction Algorithm (FRed) operates unmodified in various cases to produce ERC grammars from the richer ERCoid representation, suggesting that the procedure is efficacious in general.

The analysis of the MOAT rests firmly but indirectly on the foundation of the integers and their relations of order and equality (§§3.1-2). A typology is certified by the existence of a UVT whose rows yield its grammars under OT filtration. Two distinct objects emerge from the typology thus defined: the *i*OAT and the MOAT. The *i*OAT (nonconstructively) collects relations shared across all the UVTs of a given typology, retaining exactly those aspects of the numerical relations which contribute to the typology's structure. The MOAT, in contrast, has nothing at all to do with integers, and is (constructively) defined by border point pairs that arise from a subset of the legs of adjacent grammars. The fact of a shared typology allows us to transit back and forth between EPO and constraint, between *i*OAT and MOAT, determining the properties of the more abstract relations from, ultimately, the behavior of their more concrete numerical instantiations.

The course of the argument runs as follows. First, we lay out the border-point-based relations between grammars (§3.1), establishing their relationship to the integer relations in the UVTs. On this basis, the abstract relations are shown to be relations of order and equivalence, as we have claimed all along. We then define the key notions: the EPO as a relational structure, and the MOAT as a collection of EPOs.

Next, we establish relevant facts about filtration in OT (§3.2.1), discerning some hitherto unrecognized properties that are basic to typological analysis: No Dead Man Walking (117); Filtration Uniformity (123) and its converse (124). With these in hand, we advance to the analysis of the *i*OAT and its *i*EQOs (§3.2.2). We show that the *i*OAT completely characterizes a typology in the following sense: any UVT for a typology instantiates the order and equivalence relations of its *i*OAT (unsurprising), and (more interesting) any VT at all that instantiates the *i*OAT of a typology is a UVT for that typology.

With the *i*OAT in hand, we show the relations of the MOAT are exactly those of the *i*OAT (§3.2.3). More specifically, we show that the equivalences are the same, and that the order relations become the same when those of the MOAT are extended to take account of equivalence. This extension, which we call 'hypertransitivity', allows us to argue from $a<b$ and $b\sim c$ to $a<c$, a fact which is so familiar among the integers, when equality is the equivalence relation, as to hardly merit notice.

Having established that the *i*OAT and (the hypertransitive closure of) the MOAT are the same, we immediately deduce that the MOAT completely characterizes its typology (§3.2.4). With the main result in place, we shift focus from the UVT to the more typical ecological situation where a typology sponsors many VTs, each arising from a concrete candidate set. EPO equivalence between grammars requires that in each of VTs, the optima selected by those grammars must receive numerically identical values. EPO order between grammars requires, more weakly, that the values be either ordered accordingly or equal: no outright reversal of the order relation is allowed. This shows that the EPO embodies the coordinatewise order on the languages, conceived of as vectors of optima.

3.1 Border Point Pairs, the UVT, and the MOAT

In this section, we develop the notions of order and equivalence that play a role in the EPO, constructing them from the basic border-point information and showing how they are reflected numerically in a UVT. This leads to definition of the EPO and the MOAT.

Let's first recall how filtration works in Concrete OT (COT), expanding on Samek-Lodovici & Prince 1999: 18-19. Suitably transposed into Abstract OT (AOT), this drives the interpretation of border point pairs, which is in turn yields the contents of the EPO.

Given a COT system $S = \langle CON_S, GEN_S \rangle$, every $C \in CON_S$ assigns a nonnegative integer value to each candidate admitted by GEN_S. For any set of candidates K, one of those values must be minimal, because every set of nonnegative integers has a least element. We use the function C to filter K by defining from it a new function that we write C[K], using square brackets to distinguish it, which returns the subset of K whose elements receive the minimal value from C(q), $q \in K$, where C( ) is the familiar constraint function from candidates to $\mathbb{N}$.

(99) **Filtration by a single constraint**

$$C[K] = \{q \in K \mid \nexists z \in K \text{ such that } C(z) < C(q)\}$$

Observe that C( ) and C[ ] are very different functions: where C( ) is a function from CAND, the set of all candidates, to $\mathbb{N}$, the set of nonnegative integers, C[ ] is a function from sets to sets, and therefore from 2^{CAND} to 2^{CAND}. C( ) evaluates and C[ ] chooses.

We then extend the idea to a sequence of constraints CP, where P consists of at least one constraint and may consist of more. Left to right sequential order reflects ranking order. We use P to filter the output of C.

(100) **OT Filtration**

$$CP[K] = P[C[K]]$$

If P consists of a single constraint, we're done, because we already know from (99) how to filter with a constraint. If P is a sequence of more than one constraint, we just reapply the definition, stripping off the first constraint in the sequence repeatedly until we have reduced the process to a composition of single-C filtrations. If, for example, we're looking at C= X and P=YZ, then CP = XYZ, following the linear order $X \gg Y \gg Z$. Filtration by XYZ[K] unfolds likes this:

(101) **Ranking as Filtration**

$$XYZ[K] = YZ[X[K]] = Z[Y[X[K]]]$$

This corresponds to first filtering K by X, then filtering the result of that by Y, and finally filtering that result by Z. Filtration is function composition. (Karttunen 1998, Samek-Lodovici & Prince 1998, Prince 2002:*iv*.) We will sometimes refer to the elements of K that belong to P[K], for any sequence of constraints P, as the 'survivors' of P. We also will say the survivors 'pass through' P. Elements of K that don't survive to P[K] will be said to 'fail on P' or to be 'ejected by P'. The optimal form is the ultimate survivor.

Under a ranking, a candidate set K telescopes down, never expanding and going through a step of potential shrinkage as each constraint has its say: $K \supseteq K_1 \supseteq K_2 \ldots \supseteq K_n$, where K_i, $1 \leq i \leq n$, denotes the result of filtration by the i^{th} constraint in the ranking. If all candidates have distinct violation profiles, as is the case in a UVT as defined in (8), p.16 above, then K_n is a unit set, containing the one candidate that is optimal under the ranking at hand.[34] Because filtration outputs the minimum-valued candidates, choice is forced, and no K_i is empty if we start from a nonempty K, as we always do. If all members of K_i are assigned the same value by the filtering constraint, then no ejection takes place and $K_{i+1} = K_i$. A fortiori, once a singleton set is reached, all further results are identical. No matter what values are assigned after the singleton is reached, further constraints must choose the one survivor again and again.

Privilege and equivalence come from the relation between maximally similar pairs of rankings: the members of such a 'border point pair' belong to different grammars and are identical except for a single, adjacent transposition. To appreciate their significance, we want to see how the *base relations* derived from border point pairs are matched by numerical relations in a UVT. The abstract relations in the EPOs of the MOAT are extended from the base relations.

[34] As noted above §0.1.1 , p. 10, OT evaluation is about violation profiles; candidates provide them, but aspects of candidates unrepresented in violation patterns are unknowable by OT (Samek-Lodovici & Prince 2005).

The MOAT is an object of Abstract OT, whose properties are inherited by Concrete OT. AOT shares with COT the notion of the (ranking) *typology*, a certain kind of partition of the set of all linear orders on the constraint set CON_S, and through this the structures of AOT propagate to COT. In deriving typologies, Abstract OT makes use of the same mode of filtration as Concrete OT, but has a simpler notion of what is filtered. AOT lacks GEN, which provides the entities evaluated by constraints in a COT system: COT evaluates *candidates* through their violation profiles. AOT takes the violation profiles as its starting place, presupposing no particular source for them. A grammar arises when a particular row in a UVT is asserted to be optimal; the entities filtered are therefore the rows of UVT. A constraint in AOT takes a row (or a row label) as it argument and returns the value from a fixed position in that row.[35] A typology is a partition that derives from a UVT, exactly as defined in (12) above, repeated here.

(102) **Definition. Typology**. Given a set of constraints CON_T, a partition of the set of all orders on CON_T is a typology iff there is a UVT U, with columns that correspond 1:1 to the constraints of CON_T and rows that correspond 1:1 to the grammars of T, such that each block in the partition T is the ranking grammar of a row in U.

Let's consider an arbitrary ranking typology $T = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_p\}$ in AOT, arising from some UVT. The Γ_i are ranking grammars, sets of linear orders on a fixed constraint set CON_T. Since there are many such U in $\mathcal{U}(\text{T})$, the set of all UVTs that deliver T, we offer a naming system for their rows that allows for cross-UVT comparison. The columns of every U are named by the constraints in CON_T. The p rows in U are labeled by the elements of $\text{K}_\text{U} = \{\text{L}_1^\text{U}, \ldots, \text{L}_p^\text{U}\}$, where L_i^U names the row in U that requires the grammar Γ_i to render it optimal. For $\text{C} \in \text{CON}_\text{T}$, $\text{C}(\text{L}_i^\text{U})$ is the value of L_i^U in column C. With these developments, the definition of filtration given above in (100) and (101) carries over directly, and we know what it means to say that a given leg $\lambda \in \Gamma_i$ filters the set K_U so that $\lambda[\text{K}_\text{U}] = \text{L}_i^\text{U}$.

We may now ask what consequences a border point pair has for the numerical values in a $\text{U} \in \mathcal{U}(\text{T})$. We examine a generic, arbitrarily indexed border point pair (λ_1, λ_2), which consists of legs $\lambda_1 = \text{P}\underline{\text{XY}}\text{Q}$ and $\lambda_2 = \text{P}\underline{\text{YX}}\text{Q}$, where P, Q are sequences of constraints (possibly empty) and X, Y are individual constraints. Because this is a border point, we have λ_1 and λ_2 lying in distinct ranking grammars of the typology, which we designate Γ_1 and Γ_2 respectively. Since we have fixed the UVT for this discusussion, we will suppress the sub- and superscript U that identifies it. Thus we have, filtrationwise, for $\text{K} = \{\text{L}_1, \ldots \text{L}_p\}$ the following, using the convention that we write a singleton set without braces:

$$\text{P}\underline{\text{XY}}\text{Q}[\text{K}] = \text{L}_1$$

[35] In AOT, a constraint $\text{C}_k \in \{\text{C}_1, \ldots, \text{C}_m\}$, $1 \leq k \leq m$, is therefore a function $\mathbb{N}^m \rightarrow \mathbb{N}$ that projects the k^th component of $\mathbf{q} \in \mathbb{N}^\text{m}$. For further discussion, see Prince 2015b.

$P\underline{YX}Q[K] = L_2$

From this, we will deduce that the following numerical relations hold in U.

$X(L_1) < X(L_2)$

$Y(L_2) < Y(L_1)$

$C(L_1) = C(L_2)$, where C is any constraint in the prefix P.

These are intuitively plausible, because λ_1 and λ_2 differ only in the $\underline{XY}$ / $\underline{YX}$ transposition and any differences in their behavior can only be due to that. Let's pursue the argument.

First, to see that the claimed equalities hold in U, observe that L_1 and L_2 must both survive the shared prefix P, so that $L_1 \in P[K]$ and $L_2 \in P[K]$. If L_1 fails on P, for example, then $P\underline{XY}Q[K] \neq L_1$, contrary to assumption, and similarly for L_2. If $C(L_1) \neq C(L_2)$ for some constraint C in P, then one of L_1, L_2 will fail on C. Because both survive, this can't happen for any C in P. Therefore, L_1 and L_2 must be assigned identical values in U by every constraint in P.

Let's now turn to the claimed inequalities. Because L_1 and L_2 pass through P, the final sequences $\underline{XY}Q$ and $\underline{YX}Q$ face a set of candidates P[K] that contains both L_1 and L_2. Regardless of whatever else may reside in that set, these final sequences must filter $\{L_1, L_2\}$ as follows:

$\underline{XY}Q[\{L_1, L_2\}] = L_1$

$\underline{YX}Q[\{L_1, L_2\}] = L_2$.

If the claimed orders hold, we get the desired result. What happens, then, if the claimed orders do not hold? We consider all possibilities. First, note that neither of the relations can be reversed. If we have $X(L_2) < X(L_1)$, then L_1 is ejected on X and couldn't possibly be optimal on $\lambda_1 = P\underline{XY}Q$. The same form of argument holds for L_2 and Y.

Now suppose that in U, $X(L_1) = X(L_2)$, so that X does not distinguish the two languages. In this case, the rankings $\underline{XY}Q$ and $\underline{YX}Q$ must have the same effect on $\{L_1, L_2\}$, since X contributes nothing to the choice between $\{L_1, L_2\}$, rendering both sequences equivalent to Y**Q** over the set $\{L_1, L_2\}$. But this cannot be, because, by assumption, the rankings λ_1 and λ_2 differ in output on $\{L_1,L_2\}$. The possibility of equality on X is therefore refuted. The same argument holds for Y. To close the argument, we pedantically note that there are only three possible relevant relations between numerical values: less-than, greater-than, equal. Two have been eliminated for X and for Y, so all that's left are the originally claimed relations, which we know to be sufficient to produce the desired results: ergo, $X(L_1) < X(L_2)$ and $Y(L_2) < Y(L_1)$ in U. Since U was chosen arbitrarily, these relations hold in every U that delivers T.

We record this result in the following lemma. In the interest of exactitude, we return to the full L_i^U notation from here on.

(103) **Lemma**. **From Border Points to Numerical Relations**. Let (λ_j, λ_k) be a border point pair $(\mathrm{P\underline{XY}Q}, \mathrm{P\underline{YX}Q})$ in a typology T. Let $\mathrm{K_U}$ be the set of row labels of $\mathrm{U}\in\mathcal{U}(\mathrm{T})$, with $\mathrm{L}_j^{\mathrm{U}}, \mathrm{L}_k^{\mathrm{U}} \in \mathrm{K}$ such that $\lambda_j[\mathrm{K_U}] = \mathrm{L}_j^{\mathrm{U}}$ and $\lambda_k[\mathrm{K_U}] = \mathrm{L}_k^{\mathrm{U}}$. Then the following numerical relations hold for every $\mathrm{U}\in\mathcal{U}(\mathrm{T})$.

$$\mathrm{X}(\mathrm{L}_j^{\mathrm{U}}) < \mathrm{X}(\mathrm{L}_k^{\mathrm{U}})$$
$$\mathrm{Y}(\mathrm{L}_k^{\mathrm{U}}) < \mathrm{Y}(\mathrm{L}_j^{\mathrm{U}})$$
$$\mathrm{C}(\mathrm{L}_j^{\mathrm{U}}) = \mathrm{C}(\mathrm{L}_k^{\mathrm{U}}), \text{ for every C in P}$$

Proof. As in the text above. □

The next step in the direction of the MOAT is to introduce the abstract relations that derive from border point pairs. This requires a certain amount of attention to detail. We must be clear about what objects the relations relate. And because we need to arrive at full partial order and equivalence relations, we have to extend the basic border-point-derived relations so that the end results are transitive, and in the case of the equivalence relation, reflexive as well. Our first step is to define the 'base relations' directly from border point pairs. We begin by repeating the definition of border point pair from 0.

(104) **Definition**. **Border Point Pair**. Let T be a typology on a set of constraints $\mathrm{C{\small ON}_T}$, given as a set of ranking grammars. Let $\Gamma_j, \Gamma_k \in \mathrm{T}$, $\Gamma_j \neq \Gamma_k$. Let $\lambda_1 = \mathrm{P\underline{XY}Q}$ and $\lambda_2 = \mathrm{P\underline{YX}Q}$ be legs over $\mathrm{C{\small ON}_T}$, with P,Q sequences of constraints from $\mathrm{C{\small ON}_T}$ and $\mathrm{X,Y}\in\mathrm{C{\small ON}_T}$. Then (λ_1, λ_2) is a *border point pair* for Γ_j, Γ_k iff $\lambda_1 \in \Gamma_j$ and $\lambda_2 \in \Gamma_k$.

Each border point pair gives rise to relations between the bordering grammars.

(105) **Definition. Base relations from a Border Point Pair.** Given a typology T, we define for each $\mathrm{C}\in \mathrm{C{\small ON}_T}$ the relations $<^b_{\mathrm{C}}$ and $\sim^b_{\mathrm{C}}$.

$\Gamma_j <^b_{\mathrm{C}} \Gamma_k$ iff there is a border point pair for Γ_j, Γ_k, $(\lambda_1, \lambda_2) = (\mathrm{P\underline{XY}Q}, \mathrm{P\underline{YX}Q})$ with C = X.

$\Gamma_j \sim^b_{\mathrm{C}} \Gamma_k$ iff there is a border point pair for Γ_j, Γ_k, $(\lambda_1, \lambda_2 = (\mathrm{P\underline{XY}Q}, \mathrm{P\underline{YX}Q})$ with C in P.

Note that the relations $<^b_{\mathrm{C}}$ are exactly the *privileged relations* discussed in §2. Observe that $<^b_{\mathrm{C}}$ is by no means guaranteed to be a partial order, nor $\sim^b_{\mathrm{C}}$ to be an equivalence relation. Their status depends entirely on whatever border point pairs happen to exist in T. From the definition, we can however establish the following properties.

(106) **Lemma. Properties of the Base Relations.** The relation $<^b_{\mathrm{C}}$ is asymmetric and irreflexive. The relation $\sim^b_{\mathrm{C}}$ is symmetric.

Proof. The relation $\Gamma_j <^b_{\mathrm{C}} \Gamma_k$ is irreflexive by its definition, which requires $\Gamma_j \neq \Gamma_k$. As for asymmetry, recall that $\Gamma_j <^b_{\mathrm{C}} \Gamma_k$ holds only if there is a $\lambda_j \in \Gamma_j$ and $\lambda_k\in\Gamma_k$, $\Gamma_j\neq \Gamma_k$, such that $(\lambda_j, \lambda_k) = (\mathrm{P\underline{CY}Q}, \mathrm{P\underline{YC}Q})$. Now consider any $\mathrm{U}\in\mathcal{U}(\mathrm{T})$. We have, from Lemma (103), $\mathrm{C}(\mathrm{L}_j^{\mathrm{U}}) < \mathrm{C}(\mathrm{L}_k^{\mathrm{U}})$. Suppose that in addition to this, $\Gamma_k <^b_{\mathrm{C}} \Gamma_j$. From this we obtain $\mathrm{C}(\mathrm{L}_k^{\mathrm{U}}) < \mathrm{C}(\mathrm{L}_j^{\mathrm{U}})$ by Definition (105), which gives us a border point pair involving C, and Lemma (103), which cashes out the pair numerically in U. This is a contradiction. The relation $<^b_{\mathrm{C}}$ is therefore asymmetric.

Now consider the case where $\Gamma_j \sim^b{}_C \Gamma_k$. We must have a border point pair $\lambda_j \in \Gamma_j$ and $\lambda_k \in \Gamma_k$ such that (λ_j, λ_k) =(P$\underline{XY}$Q, P$\underline{YX}$Q), in which C∈ P. Then (λ_k, λ_j) is also a border point pair, and from it, along with C∈P, we may conclude $\Gamma_k \sim^b{}_C \Gamma_j$. Therefore, the relation $\Gamma_j \sim^b{}_C \Gamma_k$ is symmetric. □

We may now lift the numerical implications in Lemma (103) to the level of the base relations.

(107) **Lemma**. **From Abstract Base Relations to Numerical Relations**. Given a typology T, we have the following: For every U∈$\mathcal{U}$(T),

$\Gamma_j <^b{}_C \Gamma_k \Rightarrow C(L^U_j) < C(L^U_k)$.

$\Gamma_j \sim^b{}_C \Gamma_k \Rightarrow C(L^U_j) = C(L^U_k)$.

Proof. If for any U∈$\mathcal{U}$(T), we have $L^U_j <^b{}_C L^U_k$, then by definition there is a border point pair (P$\underline{XY}$Q, P$\underline{YX}$Q) from T with C = X. By Lemma (103), $C(L^U_j) < C(L^U_k)$ in U. If $L_j \sim^b{}_C L_k$, then by definition there is a border point (P$\underline{XY}$Q, P$\underline{YX}$Q) from T with C in P. By Lemma (103), $C(L^U_j) = C(L^U_k)$ in U. □

Neither abstract relation is guaranteed to be transitive, and $\sim^b{}_C$ is not guaranteed to be reflexive either. The border point pairs that would ratify these properties may simply not exist. To build the MOAT, we want to be dealing with partial orders and equivalences, so we must enlarge $<^b{}_C$ and $\sim^b{}_C$. We therefore augment both to their transitive closures, and in addition, we reflexively close $\sim^b{}_C$.[36]

This gives rise to the order and equivalence relations we are looking for: $<_C$ and $\sim_C$.

(108) **Definition**. **Partial Orders from the Base Relations.** The relation $<_C$ is the transitive closure of $<^b{}_C$.

(109) **Definition**. **Equivalences from the Base Relations.** The relation $\sim_C$ is the transitive closure of the reflexive closure of $\sim^b{}_C$.

We now show that the relation $<_C$ just defined is, as claimed, a partial order. Our method of proof is to show that the abstract relations behave as desired because of their relation to the integers. We begin with a result that extends this relationship from the base relations $<^b{}_C$, whose behavior has been established in Lemma (107).

(110) **Lemma**. **From Abstract Relations to Numerical Relations**. Given a typology T we have for every Γ_j , Γ_k ∈T and for every C∈CON_T the following: For every U∈$\mathcal{U}$(T),

a. $\Gamma_j <_C \Gamma_k \Rightarrow C(L^U_j) < C(L^U_k)$.

b. $\Gamma_j \sim_C \Gamma_k \Rightarrow C(L^U_j) = C(L^U_k)$.

[36] Transitive closure extends a nontransitive relation by including all further relations derivable through the assumption of transitivity. Reflexive closure includes all pairs (Γ_i, Γ_i) in the relation. The Wikipedia article Transitive Closure provides an account.

Proof. Suppose we have $\Gamma_j <_C \Gamma_k$. Consider any $U \in \mathcal{U}(T)$, and suppose $\Gamma_j <^b{}_C \Gamma_k$. Then $C(L^U_j) < C(L^U_k)$ by Lemma (107). If it's not the case that $\Gamma_j <^b{}_C \Gamma_k$, then because $<_C$ is the transitive closure of $<^b{}_C$, there is a sequence of grammars $\Gamma^{(1)},\ldots,\Gamma^{(n)}$ s.t. $\Gamma^{(i)} \in T$, $1 \leq i \leq n$, with $\Gamma^{(1)} <^b{}_C \ldots <^b{}_C \Gamma^{(n)}$, where $\Gamma^{(1)} = \Gamma_j$ and $\Gamma^{(n)} = \Gamma_k$.

Applying Lemma (107) to each adjacent pair of grammars in this sequence, we have, by the numbers, $C(L^U_j) < \ldots < C(L^U_k)$. Since $<$ is transitive on the integers, we have $C(L^U_j) < C(L^U_k)$, as desired.

A parallel argument establishes that if we have $\Gamma_j \sim_C \Gamma_k$, we must have $C(L^U_k) = C(L^U_k)$. □

Lemma (110) establishes that each U yielding T instantiates the abstract relations of the EPOs, in the sense that the abstract relations of order and equivalence are realized as numerical order and numerical equality among constraint values. From this, it follows immediately that $<_C$ is a partial order.

(111) **Lemma. Order**. $<_C$ is a strict partial order.

Proof. We have it by construction that $<_C$ is transitive, since it is defined to be a transitively closed. We show that it is asymmetric and irreflexive, because $<$ is asymmetric and irreflexive on $\mathbb{N}$. By Lemma (110), in every $U \in \mathcal{U}(T)$, we have $\Gamma_j <_C \Gamma_k \Rightarrow C(L^U_j) < C(L^U_k)$. Suppose that $\Gamma_k <_C \Gamma_j$. Then we have $C(L^U_k) < C(L^U_j)$, impossible for $<$ on the integers. This contradiction establishes asymmetry. For irreflexivity, note that $C(L^U_k) \not< C(L^U_k)$ for all k. By contraposition of Lemma (110), clause (a), this then entails $\Gamma_k \not<_C \Gamma_k$ for all k. Since it is transitive, asymmetrical, and irreflexive, $<_C$ is a strict partial order. □

(112) **Lemma**. **Equivalence**. $\sim_C$ is an equivalence relation.

Proof. The relation $\sim_C$ is transitive because it is constructed by transitive closure; it is reflexive because it is constructed via reflexive closure. The following argument shows that it is symmetric. Consider any $U \in \mathcal{U}(T)$. Assume $\Gamma_j \sim_C \Gamma_k$. Suppose first that $j = k$. All such relations are in $\sim_C$ by construction. They are trivially symmetric. Now suppose that $j \neq k$. Suppose that $\Gamma_j \sim^b{}_C \Gamma_k$. Then, by Lemma (106), we have $\Gamma_k \sim^b{}_C \Gamma_j$, and therefore $\Gamma_k \sim_C \Gamma_j$. If it's not the case for Γ_j and Γ_k that $\Gamma_j \sim^b{}_C \Gamma_k$, then because $\sim_C$ is the transitive closure of $\sim^b{}_C$, there is a sequence of grammars $\Gamma^{(1)},\ldots,\Gamma^{(n)}$ s.t. $\Gamma^{(i)} \in T$, $1 \leq i \leq n$, with $\Gamma^{(1)} \sim^b{}_C \ldots \sim^b{}_C \Gamma^{(n)}$, where $\Gamma^{(1)} = \Gamma_k$ and $\Gamma^{(n)} = \Gamma$. But $\sim^b{}_C$ is symmetric by Lemma (106). We may therefore reverse the sequence, with the guarantee that each adjacent pair in the reversed sequence stands in the relation $\sim^b{}_C$. Each adjacent pair in the sequence therefore stands in the relation $\sim_C$. Since $\sim_C$ is transitive, we conclude that $\Gamma_k \sim_C \Gamma_j$, as desired. □

The EPO for a constraint C is the following structure, a 'setoid' in the standard nomenclature, which collects together the languages and the two relations $<_C$ and $\sim_C$.

(113) **Definition**. **EPO(C) for T**. For a typology T, let $G = \{\Gamma | \Gamma \in T\}$.

$$\text{EPO}(C) = \langle G, <_C, \sim_C \rangle$$

The MOAT collects the EPO(C) for every C in $\text{C}{\small ON}_T$. When it is important to draw attention to the sponsoring typology, we will write $\text{EPO}_T(C)$.

(114) **Definition**. MOAT(T). For a typology T,

$$\text{MOAT}(T) = \{ \text{EPO}(C) | C \in \text{C}{\small ON}_T \}$$

In this section, we have proceeded from the definition of filtration in COT to the base relations derived directly from border point pairs to abstract partial order and equivalence relations on the grammars of the typology, allowing us to define the EPO, and from this, the MOAT.

3.2 Analysis of the MOAT

We want to show that typologies with the same MOAT are the same typology. To reach this goal, we first establish the relevant properties of filtration (§3.2.1), proceed to examine the UVT-based *i*OAT, showing that it completely characterizes the content of the typology it is derived from (§3.2.2), and conclude by establishing that the MOAT, hypertransitively closed, is identical in its relational structure to the *i*OAT (§3.2.3). This gives us our desired result that the MOAT characterizes its typology, because the *i*OAT has been shown to have this property (§3.2.4). The discussion comes to a close with an analysis of the implications of MOAT relations for the evaluation of candidates in a Concrete OT system with many candidate sets.

3.2.1 Filtration Basics

Here we establish properties of OT filtration that play a key role in the argument. We begin by fixing some relevant notions of basic OT.

(115) **Definition**. **Possible optimum**. For a system Σ, possibly abstract, given $\text{C}{\small ON}_\Sigma$ and a candidate set K, possibly abstract, a candidate $q \in K$ is a *possible optimum* for Σ iff there is a ranking λ on $\text{C}{\small ON}_\Sigma$ such that $q \in \lambda[K]$.

This definition casts the net wide, so that it embraces both concrete systems $S_C = \langle \text{C}{\small ON}_{SC}, \text{G}{\small EN}_{SC} \rangle$ and abstract systems $\mathbf{S_A} = \langle \mathbf{C{\small ON}_{SA}, T} \rangle$, for $T = \{\Gamma_1, \ldots, \Gamma_n\}$. In an abstract system, the typology is basic, not derived, and $\text{C}{\small ON}_S$ is a merely a set of labels which are correlated with UVT columns and rows by the apparatus introduced above, in the discussion of ex. (102), p.79. Both concrete and abstract OT systems call on the same notion of optimality and and therefore share its consequences.

We assume throughout that the constraint set is shared between the objects we compare. Why keep CON the same? If $\text{CON}_{S1} \neq \text{CON}_{S2}$ then the typologies are necessarily different, being partitions of different sets, even if they are structurally identical. Calling the constraints by the same names obviates the need for defining a bijection between them.

Most candidates are typically not possible optima, and in systems where their number is infinite, *almost all* of them fail. In the jargon, they are said to be *harmonically bounded* (Prince & Smolensky 1993/2004, Samek-Lodovici & Prince 1999, 2005).

(116) **Definition**. **Harmonically Bounded**. Given CON_Σ and a candidate set K, possibly abstract, a candidate $z \in K$ is *harmonically bounded in* K iff for every ranking λ on CON_Σ, there is a candidate $q \in K$, $q \neq z$, such that that $\lambda[\{z,q\}] = q$.

Here and throughout we indulge in a mild abuse of notation, writing $q \neq z$ to mean that the *violation profiles* of q and z are not equal. This is harmless in the present context, since candidates name violation profiles and have no other internal structure.

If z is harmonically bounded, there can be no leg λ on which $z \in \lambda[K]$. This is because if $z \in \lambda[K]$, then $\lambda[\{z,q\}] = z$ for every $q \in K$, $q \neq z$. But for a bounded z, this is never true: for every ranking, there's at leasts one candidate q that beats it in head-to-head competition: that is selected instead of z from $\{q,z\}$. A candidate is said to be *possible optimum* iff it is not *harmonically bounded*.[37]

We now establish a fundamental property of the filtration of sets of possible optima. Any possible optimum that survives a sequence of constraints is the actual optimum of some continuation of that sequence.

[37] We also like Riggle's 'contender' for 'possible optimum', but prefer to retain the explicit reference to optimality. *A man hears what he wants to hear and disregards the rest*.

(117) **Lemma**. **No Dead Man Walking**. Let S be an OT system, possibly abstract, and let K be a candidate set admitted by S. If $q \in K$ is a possible optimum and $q \in P[K]$, for P a sequence of constraints from C_{ON_S}, then there is a sequence of constraints Q exhausting CON_S such that $q \in PQ[K]$.

Proof. Let P be a sequence of constraints drawn from CON_S, possibly not including all the constraints of CON_S . Say $q \in P[K]$ is a possible optimum for K, the candidate set containing q, where filtration is carried out over a VT involving the candidate set containing q. Assume for purposes of contradiction that there is no Q such that $q \in PQ[K]$, where Q is a sequence of all the constraints not in P. Under this assumption, q would be optimal for some leg not beginning with the sequence P, but not for any leg beginning with P.

Let $B = P[K] \setminus \{q\}$. It must be that B is nonempty, because something other than q wins on each Q; that is, for every such Q, there is some $z \in B$, $z \neq q$, such that $z \in PQ[K]$. We make one additional observation: the members of $B \cup \{q\}$ are all equal on each constraint in P. But these facts entail that q is bounded by B, contradicting the assumption that it is a possible optimum. To spell this out: since the constraints in P do not distinguish the candidates in $B \cup \{q\}$, those constraints can have no effect on any filtration of $B \cup \{q\}$. Whatever the ranking, all decisions are made by the constraints not in P, namely those in Q, which by assumption eject q, whatever their ranking. Thus no ranking selects q, contradicting the assumption that q is a possible optimum. □

A crucial notion in establishing the relation between the MOAT and the UVTs of a given typology is that of the *filtration pattern* of a total order λ: the telescoping sequence of sets of candidates that shrinks to contain only the optima selected by λ.

(118) **Definition**. **Filtration Pattern**. The filtration pattern of a total order $\lambda = C_1 \ldots C_n$, given a candidate set K, is a sequence of sets of candidates, $C_1[K]$, $C_1C_2[K], \ldots$, $C_1 \ldots C_n[K]$, one for each step of filtration as defined in (99) and (100).

Note that in definition (118) the sequences C_1, C_1C_2, …, $C_1 \ldots C_n$, are the prefixes of λ.

We may now define the conditions under which distinct UVTs are said to have *identical* filtration patterns. What this means intuitively is that filtration proceeds identically in both. However, since we're dealing with different UVTs, the candidates involved cannot be literally the same. What we want is for the patterns to be the same, up to renaming the candidates. To speak intelligibly about this, we introduce a function φ that correlates sets of row labels across UVTs.

(119) **Definition. Identical Filtration Patterns**. Given CON_T, two UVTs U, V with columns labeled by CON_T are said to have *identical filtration patterns* if there is a bijection, $\varphi : \mathcal{P}(K_U) \rightarrow \mathcal{P}(K_V)$ such that $\varphi(P[K_U]) = P[K_V]$ for all prefixes P.

Since a total order is a prefix with an empty suffix, we have for every total order λ, $\varphi(\lambda[K_U]) = \lambda[K_V]$. With UVTs, there are no co-optima, so that $\lambda[K_U]$ and $\lambda[K_V]$ are unit sets. Definition (119) thus identifies individual row labels across VTs as well as identifying sets of row labels.

We have set up this definition in the most useful way, mapping from sets to sets. A more incremental approach would map from filtration pattern to filtration pattern: that is, from a sequence of sets to a sequence of sets. The proffered definition (119) entails this one, which in turn entails (119), as may easily be shown. We therefore proceed with (119).

By definition (8) = (102), any typology $T = \{\Gamma_1, \ldots, \Gamma_n\}$ must have a witnessing UVT U, which consists of a finite set of rows $\{ L_i^U, 1 \le i \le n \}$, each of which has a corresponding grammar $\Gamma_i \in T$. It is useful to have an explicit way of going back and forth between the grammar Γ_i and the row L_i^U that sponsors it in U. To that end, we let correspondence be given by bijections $g_U{:}\Gamma_i \mapsto L_i^U$, one for each UVT U yielding T.

First, some useful notation.

(120) **Notation.** T_U. For U, a UVT, $T_U = \{\Gamma_1, \ldots, \Gamma_n\}$ is the typology produced from U.

(121) **Grammar $\sim$ Label Correspondence**. For T_U and Lab(U) the set of row labels in the UVT U, we define the bijection $g_U{:}T_U \rightarrow \mathrm{Lab(U)}$ given by $g_U{:}\Gamma_i \mapsto L_i^U$.

The typology T will certainly have another UVT V that yields it, which like U consists of n rows, $\{ L_i^V, 1 \le i \le n \}$, where the indexing correspondence will be given by the function $g_V{:}\ \Gamma_i \mapsto L_i^V$. We have $T = T_U = T_V$, and we may therefore make use of g_U and g_V to identify row labels across UVTs.

As with U, the indexing of the rows in V is derived from the indexing of the grammars $\Gamma_i \in T$, so that L_i^V and L_i^U each give rise to the same grammar Γ_i. This gives us a natural bijection $f{:}\, L_i^U \mapsto L_i^V$ between the **row labels**, definable as $f = g_U^{-1} \circ g_V$, as shown in the following diagrams.

(122) **Grammar-mediated Label Correspondence**

$$\begin{array}{ccc} & \Gamma_i & \\ {}_{g_U}\swarrow & & \searrow_{g_V} \\ L^U_i & \mapsto & L^V_i \end{array} \qquad\qquad \begin{array}{ccc} & T & \\ {}_{g_U}\swarrow & & \searrow_{g_V} \\ \mathrm{Lab\ (U)} & \longrightarrow & \mathrm{Lab(V)} \end{array}$$

$$g_U^{-1} \circ g_V(L^U_i) = L^V_i \qquad\qquad g_U^{-1} \circ g_V$$

The grammar-mediated correspondence between L_i^U and L_i^V thus allow us to compare filtration patterns across UVTs. In the simple case where they give rise to the same typology, the correspondence is reflected directly in the subscripting. The correspondence functions also allow us to handle cases where we do not assume that the grammars associated with each UVT are identical, as in Lemma (124).

The grammar-label correspondence functions allow us to spell out the bijection $\varphi:\mathcal{P}(K_U)\rightarrow\mathcal{P}(K_V)$ that we need for filtration uniformity as follows:

$\varphi(A) = B$, for $A\subseteq K_U$ and $B\subseteq K_V$, iff $f(A) = g_U^{-1}\circ g_V(A) = B$.

Here we use the common notational shortcut whereby f(A) is the image of A under f, namely $\{f(x)|x\in A\}$.

We now show that all UVTs yielding the same typology have identical filtration patterns.

(123) **Lemma. Filtration Uniformity**. For any typology T, all $U\in\mathcal{U}(T)$ have identical filtration patterns.

Proof. For $U,V\in\mathcal{U}(T)$, $U \neq V$, we have the bijection $f:K_U\rightarrow K_V$ given by $f: L_i^U \mapsto L_i^V$. Recall that L_i^U and L_i^V both give rise to the same grammar Γ_i, a certain set of linear orders on Con_S, which mediates the correspondence between them. From this, we construct a bijection $\varphi:\mathcal{P}(K_U)\rightarrow\mathcal{P}(K_V)$ as follows:

For $S\subseteq K_U$, $\varphi(S) = \{f(L_i^U) = L_i^V \in K_V \mid L_i^U \in S\}$

We assert that $\varphi(P[K_U]) = P[(K_V)]$ for all prefixes P. Suppose, for purposes of contradiction, that this is false. Then there is a prefix P such that $\varphi(P[K_U]) \neq P[(K_V)]$. This means that there is some $L_i^U \in P[K_U]$ such that L_i^V is not in $P[K_V]$, or vice versa. Because of the arbitrariness of U and V, we can assume wlog that there is an $L_i^U \in P[K_U]$ such that $L_i^V \notin P[K_V]$. By *No Dead Man Walking* (117), there's a total order $\lambda = PQ$ such that $L_i^U \in \lambda[K_U]$.

Now, $\lambda = PQ$ is a leg of the grammar Γ_i associated with L_i^U, so that $PQ\in\Gamma_i$. Since $L_i^V \notin P[K_V]$, it must be that $L_i^V \notin PQ[K_V]$. But the grammar Γ_i of L_i^V is the same as the grammar of L_i^U. Contradiction. □

The converse also holds: UVTs with identical filtration patterns yield the same typology.

(124) **Lemma**. **Converse of Filtration Uniformity**. For any two UVTs U,V over the same constraint set Con_S, if their filtration patterns are identical, then $T_U = T_V$.

Proof. We must show that T_U and T_V are the same partition on the set of all rankings of Con_S. A leg is assigned to a grammar Γ_i by a UVT U because it filters the candidate set down to the unit set containing the row label $L_i^U = g_U(\Gamma_i)$ associated with that grammar by g_U. Using the bijection provided by the assumption of identical filtration patterns, it's clear that if aleg λ is assigned to a grammar Γ_i by U, it is assigned to the same grammar Γ_i by V. This is all we need; in the interests of explicitness, we spell out the details here, exercising the apparatus of inter-UVT correspondence.

Because U and V are assumed to have identical filtration patterns, we have by definition a bijection $\varphi:\mathcal{P}(K_U)\rightarrow\mathcal{P}(K_V)$, such that $\varphi(P[K_U]) = P[K_V]$ for every sequence of constraints P. Since every leg is a prefix, albeit one with a null suffix, and since every leg maps the entire set of labels to a set containing a single label, it follows that every unit subset of K_U is in the domain of φ, and is mapped bijectively to a unit set in the range of φ. This gives us a bijection $f:K_U \rightarrow K_V$, whereby $f(L_i^U) = \kappa \in K_V$ such that $\kappa \in \varphi(\{L_i^U\})$.

For any leg λ, there is some L_i^U such that $\lambda[K_U] = \{L_i^U\}$, so that $\lambda \in g_U^{-1}(L_i^U) = \Gamma_i$. Now, $\varphi(\lambda[K_U]) = \lambda[K_V]$ by the definition of identity of filtration patterns (119). Therefore, λ is in the grammar associated with $f(L_i^U)$, i.e. $\lambda \in g_V^{-1} \circ f(L_i^U)$. This establishes that $g_U^{-1}(L_i^U) = \Gamma_i \subseteq g_V^{-1} \circ f(L_i^U)$. By the same reasoning, any leg λ′ in the grammar of $f(L_i^U)$ is in the grammar of $f^{-1} \circ f(L_i^U) = L_i^U$, namely Γ_i, so that $\lambda' \in \Gamma_i$. This establishes that $g_V^{-1} \circ f(L_i^U) \subseteq g_U^{-1}(L_i^U) = \Gamma_i$. From these inclusions, it follows that the grammars associated with L_i^U and $f(L_i^U)$ are the same, namely Γ_i. Since λ is arbitrary, this holds for every leg and every label: the grammars of T_U are exactly those of T_V. □

In this section, we have justified three useful (and, we believe, novel) observations about the way OT filtration works. The No Dead Man Walking Lemma (117) shows the remarkable staying power of a possible optimum: if a possible optimum survives filtration by some nonempty sequence of constraints, then there is a continuation of that sequence that selects it as optimal. Filtration Uniformity (123) establishes that any UVT for a a typology not only assigns the same optima to the same grammars, as the definition of a UVT for T requires, but also does it stepwise in the same way, as filtration proceeds constraint-by-constraint through the legs of the grammars. The Converse of Filtration Uniformity shows (usefully, but perhaps unsurprisingly) that if two UVTs have the same filtration patterns, they yield the same typology.

3.2.2 The *i*EQO and *i*OAT

We show here that the *i*OAT, which collects the relations shared across the entire set of UVTs for a typology, completely characterizes the typology. Qualitatively speaking, the shared relations are exactly those that matter in the course of filtration; the rest is dross. We conclude that any UVT numerically instantiating the abstract relations of the *i*OAT is a UVT delivering the *i*OAT's typology, and thus that sameness of *i*OATs guarantees sameness of their sponsoring typologies. This sets the stage for the next step, correlating MOAT and *i*OAT relationally (§3.2.3), which then leads immediately to the main result, that identical MOATs have identical typologies (§3.2.4).

We begin by fixing some basic concepts.

(125) **Definition**. A bigraph is a triple $\langle S, R_1, R_2 \rangle$ where S is an arbitrary set and $R_1, R_2 \subseteq S \times S$ are relations on S.

(126) **Definition**. An **EQO** is a bigraph in which R_1 is a partial order and R_2 an equivalence relation.

The iEQO(C) for a constraint C is a EQO derived from the intersection of order relations and equivalence relations of all $U \in \mathcal{U}(T)$, as in (65) and (66) above.

We call on two relations $\lessdot_{C:U}$ and $\approx_{C:U,}$ which are determined by the numerics of the specific $U \in \mathcal{U}(T)$ mentioned in the subscript. From these, we obtain the general relations $\lessdot_C$ and $\approx_C$ for $\mathcal{U}(T)$, which constitute the iEQO(C), for each constraint C.

(127) **Definition**. The relations $\lessdot_{C:U}$ and $\approx_{C:U}$ induced by $U \in \mathcal{U}(T)$

(i) Partial Order. $\Gamma_{\boldsymbol{i}} \lessdot_{C:U} \Gamma_{\boldsymbol{j}}$ iff $C(L_i^U) < C(L_j^U)$

(ii) Equivalence. $\Gamma_i \approx_{C:U} \Gamma_j$ iff $C(L_i^U) = C(L_j^U)$

The abstract relations $\lessdot_{C:U}$ and $\approx_{C:U}$ on the grammars of $T = T_U$ derive from the simple numerical relations on U, which are based on values assigned to row labels.

The iEQO(C) consists of two relations, a partial order $\lessdot_C$ obtained by intersecting the partial orders $\lessdot_{C:U}$ over all $U \in \mathcal{U}(T)$, and an equivalence relation $\approx_C$ obtained by intersecting the equivalences $\approx_{C:U}$. This proceeds exactly as above in (65) and (66) , p. 56, which we repeated here for convenience of reference.

The relation $\lessdot_C$ is the intersection of partial orders, and therefore a partial order itself.

(128) **Definition**. $\lessdot_C$ as intersection of order relations

$$\lessdot_C = \bigcap_{U \in \mathcal{U}(T)} \lessdot_{C:U}$$

The relation $\approx_C$ is an intersection of equivalences, therefore an equivalence itself.

(129) Definition. $\approx_C$ as intersection of equivalence relations

$$\approx_C = \bigcap_{U \in \mathcal{U}(T)} \approx_{C:U}$$

The iEQO$_T$(C) is then the triple $\langle T, \lessdot_C, \approx_C \rangle$, where T is, as always, the set of grammars that constitute the typology, over which the relations are defined. (As above, we will write just iEQO(C) when T is clear from context.) Collecting together all iEQOs for the constraints of the system yields the iOAT of $\mathcal{U}(T)$.

(130) **Definition.** iEQO$_T$(C)

$i\text{EQO}_T(C) = \langle T, \lessdot_C, \approx_C \rangle$

(131) **Definition.** iOAT($\mathcal{U}$(T))

$i\text{OAT}(\mathcal{U}(T)) = \{i\text{EQO}_T(C) \mid C \in \text{Con}_S\}$

A quick check confirms that the iEQO$_T$(C) is an EQO, as defined in (126). The iOAT then is defined as the collection of iEQOs, one for each constraint.

The iOAT determines the membership of $\mathcal{U}$(T). To understand why this is so, we need to be clear about the relation between the iOAT and the tableaux U$\in\mathcal{U}$(T). The key bridging notion relates an abstract order or equivalence structure to its natural correlates on the integers. A function assigning numerical values will be said to *respect* order or equivalence if its values map the abstract relations into the appropriate numerical relations.

(132) **Definition**. ***Respect.***
   a. Given an order relation $<_S$ on a set S, a function f:S$\rightarrow\mathbb{N}$ will be said to *respect the relation* $<_S$ iff a $<_S$ b $\Rightarrow$ f(a) $<$ f(b).
   b. Given an equivalence relation $\sim_S$ on a set S, a function f:S$\rightarrow\mathbb{N}$ will be said to *respect the relation* $\sim_S$ iff a $\sim_S$ b $\Rightarrow$ f(a) = f(b).

If we have a function f:S$\rightarrow\mathbb{N}$ that respects $<_S$, we will say of the numerical relation $<$ on the set f(S) that it *instantiates* $<_S$ under f, and similarly for $\sim_S$.

(133) **Definition**. **Instantiate a relation**. Given a function f:S$\rightarrow\mathbb{N}$ that respects $<_S$, we say of the numerical relation $<$ on the set f(S) that it *instantiates* $<_S$ under f, and similarly for $\sim_S$.

Following this usage, a UVT U will be said to *instantiate* the relations of an iOAT on the elements of some T if there is a function f:T$\rightarrow\mathbb{N}$ such that for all C$\in$CON$_S$, the numerical relations $<$ and = on the values that C assigns can be shown to instantiate under f the abstract relations $\lessdot_C$ and $\approx_C$.

(134) **Definition.** ***Instantiate* an *i*EQO**. Let U be a UVT with columns associated with constraints C$\in$CON$_S$. A column C of U is said to *instantiate* an iEQO$_T$(C) iff the numerical relations $<$ and = on the values assigned by C to the elements of K_U *instantiate* the relations $\lessdot_C$ and $\approx_C$.

(135) **Definition**. *Instantiate* an iOAT. A UVT U with columns labeled by C$\in$CON$_S$ *instantiates* iOAT(($\mathcal{U}$(T)) iff every column C of U instantiates iEQO$_T$(C) for every C$\in$CON$_S$.

It is *not* presupposed in the definition that U$\in i$OAT(($\mathcal{U}$(T)). Instantiation can relate any iOAT to any VT U, so long as the columns of U are labeled to allow iEQO(C) to be correlated with a unique column C in U. This freedom allows us to ask *whether* a VT instantiating iOAT(($\mathcal{U}$(T)) necessarily belongs to $\mathcal{U}$(T), a question we will shortly be able to answer in the affirmative.

Let us first establish the truth of the obvious: if U$\in\mathcal{U}$(T), then U instantiates iOAT($\mathcal{U}$(T)). Instantiation requires respect, and the definition of *respect* requires us to proceed in this case from grammars to numbers. Constraints, however, do not assign directly to grammars, but to the labels $L_i^U \in Lab(U)$, which are associated 1:1 with the grammars of T. The respectful function we are looking for is therefore $C \circ g_U : T \rightarrow \mathbb{N}$.

Explicitly, for each $\Gamma_i, \Gamma_j \in T$, to show that $C \circ g_U$ respects the *i*EQO relations, we need

$$\Gamma_i \lessdot_C \Gamma_j \Rightarrow C \circ g_U(\Gamma_i) < C \circ g_U(\Gamma_j)$$
$$\Gamma_i \approx_C \Gamma_j \Rightarrow C \circ g_U(\Gamma_i) = C \circ g_U(\Gamma_j).$$

Equivalently, with rows labeled by $L_i^U = g_U(\Gamma_i)$, we want

$$\Gamma_i \lessdot_C \Gamma_j \Rightarrow C(L_i^U) < C(L_j^U)$$
$$\Gamma_i \approx_C \Gamma_j \Rightarrow C(L_i^U) = C(L_j^U)$$

This follows directly from the definitions of $\lessdot_C$ and $\approx_C$. From (128), we have $\Gamma_i \lessdot_C \Gamma_j$ iff $\Gamma_i \lessdot_{C:U} \Gamma_j$ for *every* $U \in \mathcal{U}(T)$. From (127), we have, in the case of a specific given U, that $\Gamma_i \lessdot_{C:U} \Gamma_j$ iff $C(L_i^U) < C(L_j^U)$. Putting the two together yields the implication $\Gamma_i \lessdot_C \Gamma j \Rightarrow C(L_i^U) < C(L_j^U)$ for every $U \in \mathcal{U}(T)$, the very definition of instantiation for $\lessdot_C$. Similar reasoning applies to the equivalence relation $\approx_{C.}$

(136) **Remark**. If $U \in \mathcal{U}(T)$, then U instantiates *i*OAT($\mathcal{U}$(T))

Proof. Direct from the relevant definitions, as in the text. □

This observation may be paraphrased by saying that the *i*OAT gives *necessary* conditions for a U to be in $\mathcal{U}$(T). We proceed now to show that the *i*OAT gives sufficient conditions as well. This means, intuitively, that intersecting the relations in all UVTs has lost no information that is crucial to the filtration process. With the *i*OAT in hand, then, we can produce any numerical UVT that belongs to $\mathcal{U}$(T) by simply deploying numbers in a way that instantiates the relations $\lessdot_C$ and $\approx_C$.

We prove sufficiency in two steps, the first of which carries the major burden. In it, we show that we can replace any single column C of $U \in \mathcal{U}(T)$ with a variant that instantiates *i*EQO$_T$(C), preserving membership in $\mathcal{U}$(T). We then note that we can start out with any $U \in \mathcal{U}(T)$ and make our way toward any instantiating V by step-wise column-at-time substitutions.

We adopt the notation $U[K_C]$ to denote the VT that is identical to U except that the values in column C are different in a way that is described in context. For $V = U[K_C]$, this substitution has the effect of creating new candidates L_i^V, which are identical to L_i^U except in the values assigned by constraint C. This gives us a natural correspondence $L_i^U \rightarrow L_i^V$ between the rows of U and V, based on the way they are enumerated in U, which is indicated by the subscripts. Note however that we have no a priori assurance that V produces T_U, or indeed that V even satisfies the definition of UVT, which requires that every row give rise to a grammar, and a grammar distinct from all the others. We have no guarantee that the correspondence can be mediated by shared grammars: this is what we need to prove. We must therefore recognize a direct row-to-row bijection ρ from K_U to K_V defined as follows:

(137) **Row-to-Row**. $\rho:K_U \rightarrow K_V$ s.t. $\rho(U[n]) = V[n]$, where $U[n]$ is the n^{th} row of U.

Following standard practice, we also write $\rho(S)$ to denote the image of $S \subseteq K_U$ under ρ.

(138) **Lemma**. **Columnar Interchange**. For $U \in \mathcal{U}(T)$ and C a column of U, if $V = U[K_C]$ instantiates $i\mathrm{EQO}_T(C)$, then $V \in \mathcal{U}(T)$.

Proof. By assumption, V is identical to U except in column C, where we are given that column C instantiates $i\mathrm{EQO}_T(C)$. Instantiation requires the existence of a function $f:T \rightarrow \mathbb{N}$ which respects the relations of the *i*OAT. In this case, we can use $C \circ \rho \circ g_U$, which maps from $\Gamma_i \in T$ to L_i^U to L_i^V to $\mathbb{N}$.

Assuming instantiation under this function gives us

$$\Gamma_i \lessdot_C \Gamma_j \Rightarrow C \circ \rho \circ g_U(\Gamma_i) < C \circ \rho \circ g_U(\Gamma_j)$$
$$\Gamma_i \approx_C \Gamma_j \Rightarrow C \circ \rho \circ g_U(\Gamma_i) = C \circ \rho \circ g_U(\Gamma_j).$$

In terms of the row labels of V, $L_i^V = \rho(L_i^U) = \rho(g_U(\Gamma_i))$, we have

$$(*)\ \Gamma_i \lessdot_C \Gamma j \Rightarrow C(L_i^V) < C(L_i^V)$$
$$(**)\ \Gamma_i \approx_C \Gamma j \Rightarrow C(L_i^V) = C(L_i^V).$$

Observe that we have made no assumptions about the grammars associated with the L_i^V We proceed to show that they are exactly the $\Gamma_i \in T_U$.

Consider any leg λ= PCQ, with P or Q possibly empty. We show $\rho(PC[K_U])) = PC[(K_V)]$. Let $S_U = P[K_U]$ and $S_V = P[K_V]$. Since U and V do not differ on any constraints in P, $\rho(S_U) = S_V$. Let $L_z^U \in S_U$. We have two cases to consider with respect to filtration by C: ejection and acceptance.

Suppose first that C ejects L_z^U from S_U, so that $L_z^U \notin C[S_U]$. Then there is some $L_q^U \in S_U$ such that $C(L_q^U) < C(L_z^U)$.This relation holds in *all* $U \in \mathcal{U}(T)$ because all have the same filtration patterns by Filtration Uniformity (123). Therefore $\Gamma_q \lessdot_C \Gamma_z$ in *i*EQO(C). In particular, we may now follow this over to V via (*), which gives us

$$\Gamma_q \lessdot_C \Gamma_z \Rightarrow C(L_q^V) < C(L_z^V).$$

Therefore C ejects $L_z^V = \rho(L_z^U)$ from S_V.

Now, suppose $L_q^U \in C[S_U]$. Then $C(L_q^U) < C(L_z^U)$ for every $L_z^U \in S_U$ which is such that $C(L_q^U) \neq C(L_z^U)$. This relation holds in *all* $U \in \mathcal{U}(T)$ because all have the same filtration patterns by Filtration Uniformity (123). Therefore $\Gamma_q \lessdot_C \Gamma_z$ in *i*EQO(C) for all $\Gamma_z \neq \Gamma_q$. Since V instantiates *i*EQO(C) by $C \circ \rho \circ g_U$, we have $C(\rho(L_q^U)) < C(\rho\ (L_z^U))$, that is, $C(L_q^V) < C(L_z^V)$ for $L_q^V \neq L_z^V$. Thus $L_q^V = \rho(L_q^V) \in PC[K_V]$.

With both filtration possibilities covered, we conclude $\rho(PC[K_U])) = PC[(K_V)]$.

Since this argument applies to any prefix, including the empty prefix, it follows that V has the same filtration pattern as U, and therefore by converse of Filtration Uniformity (124), $V \in \mathcal{U}(T)$. □

(139) **Theorem**. **R.E.S.P.E.C.T.** If V is a UVT that instantiates *i*OAT(T), then $V \in \mathcal{U}(T)$.

Proof. We simply interchange the columns of V with those of any UVT in $\mathcal{U}(T)$. Specifically, define the sequence $U_0,\ldots,U_n$ where U_0 is any UVT in $\mathcal{U}(T)$ and $U_n = V$, and for $1 \le i \le n$, U_i is the VT that is identical to U_{i-1} except that the values in the i^{th} column of U_i are those of the i^{th} column of V. By Lemma (138), each $U_i \in \mathcal{U}(T)$. □

The *i*OAT($\mathcal{U}$(T)) thus completely determines the typology T. Every UVT that belongs to $\mathcal{U}$(T) instantiates the *i*OAT($\mathcal{U}$(T)), as noted in Remark (136). And every UVT that instantiates the *i*OAT($\mathcal{U}$(T)) belongs to $\mathcal{U}$(T), as shown in Theorem (139). A UVT gives rise to exactly one typology. If two typologies have the same *i*OAT, they are produced by exactly the same set of UVTs, and are therefore identical. We record this observation as a corollary to Theorem (139).

(140) **Corollary**. **Shared *i*OAT**. Let T_1, T_2 be typologies over a set of constraints $\text{C}\small{\text{ON}}_S$. If *i*OAT($\mathcal{U}(T_1)$) = *i*OAT($\mathcal{U}(T_2)$), then $T_1 = T_2$.

Proof. Any UVT U that instantiates *i*OAT($\mathcal{U}(T_1)$) also instantiates *i*OAT($\mathcal{U}(T_2)$). By Theorem (139), $U \in \mathcal{U}(T_1)$ and $U \in \mathcal{U}(T_2)$. Since U gives rise to a unique typology, $T_1 = T_2$. □

Having shown that *i*OAT($\mathcal{U}$(T)) gives us T and only T, we turn to the relation between the UVT-based *i*OAT and the border-point based MOAT.

3.2.3 The *i*OAT and the hypertransitive closure of the MOAT

The definition of the *i*EQO(C) is grossly nonconstructive, since it involves every one of an infinite number of UVTs that generate the same typology T. It delimits *i*EQO(C) but does not deliver it into our hands. But we will see that it can be obtained algorithmically directly from the border points of T, using the EPO(C).

We will show that the relations of MOAT(T) and those of *i*OAT($\mathcal{U}$(T)) become identical when MOAT equivalence and order relations are appropriately combined with each other via what we will call *hypertransitive closure*, defined in (150) below. First, we demonstrate that MOAT(T) and *i*OAT($\mathcal{U}$(T)) have the same equivalences. Then we show that the partial order relations of the *i*OAT are identical to the partial order relations of the MOAT when extended by hypertransitive closure to take account of relations between equivalents. From this it will quickly follow that the MOAT gives all the information in the iOAT (§3.2.4).

Since we are now concerned with the EPO, we repeat the definitions and statements of fact that relate it on the one hand to Border Point Pairs and on the other to values in UVTs. Let's begin with the base relations $\sim^b_C$ and $<^b_C$ from (105).

(141) **Definition. Base relations from a Border Point Pair.** Given a typology T, we define for each C∈$\mathrm{CON_T}$ the relations $<^b_{\mathrm{C}}$ and $\sim^b_{\mathrm{C}}$ where $\Gamma_j <^b_{\mathrm{C}} \Gamma_k$ iff there is a border point pair for Γ_j , Γ_k, (λ_j, λ_k) = (P$\underline{\mathrm{XY}}$Q, P$\underline{\mathrm{YX}}$Q) with C = X, and $\Gamma_j \sim^b_{\mathrm{C}} \Gamma_k$ iff there is a border point pair for Γ_j , Γ_k, (λ_j, λ_k) = (P$\underline{\mathrm{XY}}$Q, P$\underline{\mathrm{YX}}$Q) with C in P.

From this, EPO equivalence is defined as follows, repeated from (109):

(142) **Definition**. **Equivalences from the Base Relations.** The relation $\sim_{\mathrm{C}}$ is the transitive closure of the reflexive closure of $\sim^b_{\mathrm{C}}$.

We now have already established in Lemma (110), repeated here as (143), that the abstract relations in MOAT(T) enforce themselves on the individual UVTs of $\mathcal{U}$(T)

(143) **Lemma**. **From Abstract Relations to Numerical Relations**. Given a typology T we have for every Γ_j , Γ_k ∈T and for every C∈ $\mathrm{C{\small ON}_T}$ the following for every U∈$\mathcal{U}$(T),

a. $\Gamma_j <_{\mathrm{C}} \Gamma_k \Rightarrow \mathrm{C}(\mathrm{L}^{\mathrm{U}}_j) < \mathrm{C}(\mathrm{L}^{\mathrm{U}}_k)$.

b. $\Gamma_j \sim_{\mathrm{C}} \Gamma_k \Rightarrow \mathrm{C}(\mathrm{L}^{\mathrm{U}}_j) = \mathrm{C}(\mathrm{L}^{\mathrm{U}}_k)$.

We begin with an analysis of the various equivalence relations.

Lemma (143) means, in terms of the relationship between EPO(C) and *i*EQO(C), that $\Gamma_j \sim_{\mathrm{C}} \Gamma_k \Rightarrow \Gamma_j \approx_{\mathrm{C}} \Gamma_k$. We now establish the converse**:** $\Gamma_j \approx_{\mathrm{C}} \Gamma_k \Rightarrow \Gamma_j \sim_{\mathrm{C}} \Gamma_k$, which allows us to conclude the equivalence relations in EPO(C) and *i*EQO(C) are identical.

To proceed, we need to recognize a basic structure imposed by EPO(C: the equivalence classes imposed on the grammars by the equivalence relation $\sim_{\mathrm{C}}$. For a given grammar Γ_k∈T and constraint C∈$\mathrm{C{\small ON}_T}$, we denote its equivalence class, which we will refer to informally as its 'Γ-cloud', by $\Gamma_k^{*\mathrm{C}}$. We omit the superscript C when it is clear from context which constraint we're referring to.

(144) **Notation**. For C∈$\mathrm{CON_T}$, Γ_k∈T, we write $\Gamma_k^{*\mathrm{C}}$ for the set $\{\Gamma | \Gamma \sim_{\mathrm{C}} \Gamma_k\}$.

Since $\sim_{\mathrm{C}}$ is the transitive (and reflexive) closure of the base relation $\sim^b_{\mathrm{C},}$ we have it that $\Gamma_k^{*\mathrm{C}}$ is the set of Γ∈T related to Γ_k∈T through a sequence of grammars related pairwise by $\sim^b_{\mathrm{C}}$. Thus:

$\Gamma_k^{*\mathrm{C}} = \{\Gamma | \exists\, \Gamma^{(1)},\ldots,\Gamma^{(n)}$ s.t. $\Gamma^{(i)}$∈T, $1 \le i \le n$, with $\Gamma^{(1)} \sim^b_{\mathrm{C}} \ldots \sim^b_{\mathrm{C}} \Gamma^{(n)}$, $\Gamma^{(1)} = \Gamma_k$ and $\Gamma^{(n)} = \Gamma\}$.

This observation allows us to articulate the relative positions of the Γ-clouds ("equivalence classes") with respect to a given UVT, which is fundamental to establishing that *i*EQO equivalence entails EPO equivalence.

(145) **Lemma. From *i*EQO to EPO: Equivalence.** Let T be a typology, C∈CON_T, and let $\Gamma_i, \Gamma_j \in$T. If $\Gamma_i \approx_\text{C} \Gamma_j$, so that $\text{C}(\text{L}_i^\text{U}) = \text{C}(\text{L}_j^\text{U})$ for every U∈$\mathcal{U}$(T), then $\Gamma_i \sim_\text{C} \Gamma_j$.

Proof. Let T be a typology, C∈ CON_T, and suppose that Γ_i, $\Gamma_j \in$T with $\text{C}(\text{L}_i^\text{U}) = \text{C}(\text{L}_j^\text{U})$ for every U∈$\mathcal{U}$(T). We want to show that $\Gamma_i \sim_\text{C} \Gamma_j$. Assume for purposes of contradiction that $\Gamma_i \nsim_\text{C} \Gamma_j$.

Consider the two Γ-clouds $\Gamma_i^{*\text{C}}$ and $\Gamma_j^{*\text{C}}$. By our reductio assumption $\Gamma_i \nsim_\text{C} \Gamma_j$, these two clouds are distinct, and therefore their intersection is empty: $\Gamma_i^{*\text{C}} \cap \Gamma_j^{*\text{C}} = \varnothing$.

Take some U∈ $\mathcal{U}$(T). By assumption $\text{C}(\text{L}_i^\text{U})$ and $\text{C}(\text{L}_j^\text{U})$ have the same nonegative value, call it *n*. We will see that $\Gamma_i^{*\text{C}} \cap \Gamma_j^{*\text{C}} = \varnothing$ leads to a contradiction.

To reach this goal, we define a new UVT V by manipulating the values of C and leaving the values of other constraints the same. Let U and V have the same column labels. This means that we can compare T_U and T_V because they are both partitions of the linear orders of the same constraint set. Let V have the same number of rows as U, and for every constraint D ≠ C, let $\text{D}(\text{L}_k^\text{U}) = \text{D}(\text{L}_k^\text{V})$ for all D∈CON_T, $1 \leq k \leq m$, *m* the number of rows in V.

Now we turn to C. Recall that $\text{C}(\text{L}_i^\text{U})$ and $\text{C}(\text{L}_j^\text{U})$ have the same nonegative value, *n*. For our purposes, the values assigned by C in U fall into 3 classes: those less than *n*, those greater than *n*, and *n* itself. On C, for all languages L_k^V in V with $\text{C}(\text{L}_k^\text{U}) < n$, we set $\text{C}(\text{L}_k^\text{V}) = \text{C}(\text{L}_k^\text{U})$, leaving their values the same as their cognates in U. For all languages L_k^V with $\text{C}(\text{L}_k^\text{U}) > n$, we set $\text{C}(\text{L}_k^\text{V}) = \text{C}(\text{L}_k^\text{U}) + 1$.

Now we set the values in V for the languages L_x^V such that $\text{C}(\text{L}_x^\text{U}) = n$. For L_x^V such that $\Gamma_x^\text{U} \in \Gamma_i^{*\text{C}}$, let $\text{C}(\text{L}_x^\text{V}) = n$. For L_x^V such that $\Gamma_x^\text{U} \notin \Gamma_i^{*\text{C}}$, let $\text{C}(\text{L}_x^\text{V}) = n+1$. We now have a complete VT V which differs from U only in column C.

This manipulation does not differentiate the grammars Γ_m^V with $\text{C}(\text{L}_m^\text{U}) \neq n$ from their correlates in U. For these $\Gamma_m^\text{V} = \Gamma_m^\text{U}$. Relations of order and equality between all constraint values have remained the same. But we have cleared space in V for adjusting the C values of the languages that all receive the value *n* in U. This allows us to probe, and find a contradiction from, the assumption that there are languages L_i^U and L_j^U such that $\text{C}(\text{L}_i^\text{U}) = \text{C}(\text{L}_j^\text{U}) = n$, belonging to distinct equivalence classes mod $\sim_\text{C}$. i.e. with $\Gamma_i \nsim_\text{C} \Gamma_j$.

Note that $\text{T}_\text{V} \neq \text{T}_\text{U}$ by Lemma (138) and Corollary (139) R.E.S.P.E.C.T, because T_V does not instantiate the equivalence relation of *i*EQO(C) in numerical equality. Specifically we have $\text{C}(\text{L}_j^\text{V}) > \text{C}(\text{L}_i^\text{V})$ where in *i*EQO(C), $\Gamma_i \approx_\text{C} \Gamma_j$ and therefore $\text{C}(\text{L}_j^\text{U}) = \text{C}(\text{L}_i^\text{U})$.

Now, because $\text{T}_\text{U} \neq \text{T}_\text{V}$, there must be a total order λ = PCQ and corresponding languages L_w^U, L_w^V such that L_w^U wins on λ while L_w^V loses on λ, i.e. ∃λ such that

(i) $\text{L}_w^\text{U} \in \lambda[\text{K}^\text{U}] = \text{PCQ}[\text{K}^\text{U}]$
$\text{L}_w^\text{V} \notin \lambda[\text{K}^\text{V}] = \text{PCQ}[\text{K}^\text{V}]$.

We can further refine this statement by noting that U and V only differ in values on C, which is the sole point of distinction between the two UVTs. This ensures that suffix Q in λ plays no role in ejecting L_w^V, so that a stronger statement is true:

(ii) $\text{L}_w^\text{U} \in \text{PC}[\text{K}^\text{U}]$
$\text{L}_w^\text{V} \notin \text{PC}[\text{K}^\text{V}]$.

At this point, we have not yet established the C values of L_w^U and L_w^V. We will argue from the assumption that $\lambda[K^U]=\{L_w^U\}$ and $\lambda[K^V] \neq \{L_w^V\}$ to conclude that $C(L_w^U) = n$ and $C(L_w^V) = n+1$. Because L_w^U comes from U and L_w^V comes from V, the argument requires a certain amount of back-and-forth between the two UVTs.

First, observe that no language in V differs on any constraint in P from its U correlate; in particular L_w^V does not differ from L_w^U on constraints in P, since C∉P. It follows that L_w^V also passes through the prefix P, so we have

(iii) $L_w^U \in P[K^U]$, and therefore
$L_w^V \in P[K^V]$.

Now, L_w^V loses on PC to some V-language since $L_w^V \notin PC[K^V]$. Call this language L_a^V.

Since $L_w^V \in P[K^V]$, it must be that $L_a^V \in P[K^V]$, otherwise L_w^V will be ejected on P in favor of L_a^V. Necessarily, then, $C(L_a^V) < C(L_w^V)$ since L_a^V faces off against L_w^V on C in PC.

(iv) $L_a^V \in PC[K^V]$
$L_a^V \in P[K^V]$
$C(L_a^V) < C(L_w^V)$

This establishes that C in PCQ is the constraint where L_w^V is ejected in favor of L_a^V. To establish the C value of L_a^V, we look back at U, from which V is derived.

Like all languages in V, L_a^V has a corresponding language L_a^U in K^U. Because L_a^V and L_a^U agree on all values of the constraints in P, and because L_a^V survives the prefix P by (iv), we conclude that L_a^U survives P as well. More concisely, $L_a^V \in P[K^V]$ entails $L_a^U \in P[K^U]$.

(v) $L_a^V \in P[K^V]$, and therefore
$L_a^U \in P[K^U]$

From above (i), we have it that L_w^U wins on the total order λ = PCQ, ensuring that L_w^U passes through each prefix of λ including both P and PC, so that $L_w^U \in P[K^U]$ and $L_w^U \in PC[K^U]$, as in (ii) and (iii). But this means that L_a^U and L_w^U compete on C in PC. We have thus pulled the competition between L_a^V and L_w^V on C in V back into U. We know that L_a^V wins overall on λ and that L_a^U loses on λ. Since L_w^U is the winner on λ, L_w^U must either beat or tie L_a^U on C, so that we have in U either

[a] $C(L_a^U) > C(L_w^U)$ or
[b] $C(L_a^U) = C(L_w^U)$.

In the first case, L_a^U is ejected from λ = PCQ on C. (In the second case, it would have to be ejected somewhere in Q.)

We aim to establish [b] $C(L_a^U) = C(L_w^U)$, and from this that they are both equal to *n*. We do this by eliminating possibility [a]. Shifting our attention to UVT V, we have previously shown that L_a^V beats L_w^V on C, so $C(L_a^V) < C(L_w^V)$ as in (iv). Now by the construction of V, languages in V can differ from their corresponding languages in U on C by at most one, therefore by [a], the following is true.

[a′] $C(L_a^V) \geq C(L_w^V)$

But L_a^V must beat L_w^V on C, by (iv), and therefore [a′] cannot be true, and so [b] is the only remaining possibility.

Now, by (iv), $C(L^V_a) < C(L^V_w)$. But the only values of C in U for which corresponding languages in V can differ on C is *n*. Therefore, $C(L^U_a)=n$.

Crucially, this implies that $L^U_a \in \Gamma^*_i$ because $C(L^V_w) = C(L^U_w)+1$, and $C(L^V_a)= n$ because $C(L^V_a)<C(L^V_w)$. This also entails that $L^U_w \notin \Gamma^*_i$.

But what we've shown is that we have a prefix PC on which L^U_a and L^U_w compete, and both pass through. This connects us with the EPO world, because this establishes that $L^U_a \sim'_C L^U_w$ and puts L^U_w in Γ_i*. This is a contradiction. This comes directly from the assumption that $\Gamma_i \not\approx_C \Gamma_j$, which ensures that $\Gamma^{*C}_i \cap \Gamma^{*C}_j = \varnothing$, allowing us to construct the (hypothetical and nonexistent) different typology V.

□

With this lemma established, we supply a couple of definitions that will allow us to deal concisely with the equivalence relations in *i*OAT and MOAT.

(146) **Definition.** For a typology, T, $i\text{OAT}_{\text{equiv}}(\mathcal{U}(\text{T})) =_{\text{df}} \{\approx_C \mid C \in \text{C}\small{\text{ON}}_T\}$.

(147) **Definition.** For a typology, T, $\text{MOAT}_{\text{equiv}}(\text{T}) =_{\text{df}} \{\sim_C \mid C \in \text{C}\small{\text{ON}}_T\}$.

(148) **Theorem. Equivalence Relations Shared**. For a typology, T, and $C \in \text{C}\small{\text{ON}}_T$, $i\text{OAT}_{\text{equiv}}(\mathcal{U}(\text{T})) = \text{MOAT}_{\text{equiv}}(\text{T})$.

Proof. LR. Let T be a typology, $C \in \text{C}\small{\text{ON}}_T$, and $\Gamma_i, \Gamma_j \in$T with $\Gamma_i \approx_C \Gamma_j$. By definition this means that $C(L^U_i) = C(L^U_j)$ in every $U \in \mathcal{U}(\text{T})$. Invoking Lemma (145), we have that $\Gamma_i \sim_C \Gamma_j$. This establishes inclusion from left to right, so that $i\text{OAT}_{\text{equiv}}(\mathcal{U}(\text{T})) \subseteq \text{MOAT}_{\text{equiv}}(\text{T})$.

RL. Let T be a typology, $C \in \text{CON}_T$, and $\Gamma_i, \Gamma_j \in$T with $\Gamma_i \sim_C \Gamma_j$. By Lemma (110)b, we have $\Gamma_i \sim_C \Gamma_j \Rightarrow C(L^U_i) = C(L^U_j)$ in every $U \in \mathcal{U}(\text{T})$. Therefore by definition (127), $\Gamma_i \approx_C \Gamma_j$ and we have established that $\text{MOAT}_{\text{equiv}} \subseteq i\text{OAT}_{\text{equiv}}$. □

We now turn to the partial order relations in the MOAT and *i*OAT. For a given constraint C, it is not necessarily the case that the EPO relation $<_C$ is the same as the *i*EQO relation $\lessdot_C$. This is because the *i*EQO relation responds to the arithmetical fact that a<b and b=c implies that a<c, while the the EPO relation $<_C$ takes no account of EPO equivalence $\sim_C$. We therefore need to extend the EPO relation in a way that parallels the behavior of numerical order and equality, so that a<b and b∼c implies a<c. We call this extension 'hypertransitive closure'.

We start with a preliminary definition of an equivalence-extended relation, or EER.

(149) **Definition. Equivalence-extended relation (EER)**. Given a structure $\Sigma = \langle S,<,\sim\rangle$, we define EER($\Sigma$) to be $\langle S, <^{\sim} \rangle$, where $<^{\sim}$ is a relation on S, defined as follows. For all a, b, c $\in$ S,

$$\begin{array}{ll} a < b & \Rightarrow\ a <^{\sim} b \\ a \sim b\ \&\ b < c & \Rightarrow\ a <^{\sim} c \\ a \sim b\ \&\ c < b & \Rightarrow\ c <^{\sim} a. \end{array}$$

Note that $<^{\sim}$ is not necessarily a partial order, even though it is derived from one, because it may lack transitive closure. It *is* asymmetric and irreflexive, because it inherits these properties from $<$. We can construct a partial order $\leqslant$ from $<^{\sim}$ by simply taking its transitive closure.

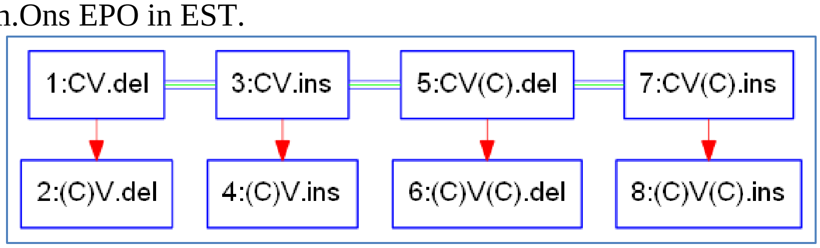
(150) **Definition. Hypertransitive closure of EPO**. For an EPO(C) = $\langle K, <_C, \sim_C\rangle$. The hypertransitive closure of EPO(C), which we denote htcEPO(C), is defined to be the bigraph $\langle K, \leqslant, \sim\rangle$, where $\leqslant$ is the transitive closure of the relation $<^{\sim}$ of EER(EPO(C)) and $\sim_C$ is the relation $\sim_C$ in EPO(C).

We now show that for a given EPO(C), the order relations in htcEPO(C) are exactly the same as those in the corresponding *i*EQO(C). We first introduce the term '*n*-band' for the set of grammars whose corresponding row labels all receive the same value from C.

(151) **Definition**. $\mathbf{C^{[U:\mathit{n}]}}$, **the *n*-band for C in U**. For a typology T, U$\in\mathcal{U}$(T), and C$\in$CON$_T$, define the *n*-band $C^{[U:n]}$ for C to be $g_U^{-1}(C^{-1}(n)) = \{\Gamma_i \mid C(L_i^U) = n\}$.

Recall that the Γ-cloud of Γ_k is $\{\Gamma \mid \Gamma \sim_C \Gamma_k\}$. From (143)b, we know that all members of the Γ-cloud of Γ_k will be assigned the same value by C: that is, will belong to the same *n*-band in every UVT U. The converse is not guaranteed, It's entirely possible that nonequivalent Γ_i and Γ_j, if noncomparable in EPO(C), may end up in the same *n*-band in some UVT. For example, consider EPO(m.Ons) from EST, repeated from (68):

(152) m.Ons EPO in EST.



Here, grammar **1, 3, 5, 7** are in a single Γ-cloud and must be assigned the same m.Ons value in every UVT. But grammars **2**, **4**, **6**, **8** are noncomparable and live inside their own Γ-clouds. Each may each be assigned any numerical value greater that that assigned to **1, 3**, **5, 7**. In particular, they may all be assigned the *same* value in some U, in which case the *n*-band in U for that value would consist of four distinct Γ-clouds.

Nevertheless, for any typology T, it is possible to find a V for T such that each nonempty n-band $C^{[V:n]}$ consists of a single Γ-cloud.

(153) **Lemma**. **Purifying the band**. For a typology T, there is a $V \in \mathcal{U}(T)$ in which for every $C \in \text{C}_{\text{ON}_T}$, and for every $n \geq 0$, $C^{[V:n]} = \varnothing$ or $C^{[V:n]} = \Gamma^*_i$, for some $\Gamma_i \in T$.

Proof. Let $C \in \text{C}_{\text{ON}_T}$. We prove the lemma by induction. Let $U \in \mathcal{U}(T)$. Consider $C^{[U:0]}$. If $C^{[U:0]} = \varnothing$, the lemma holds. Assume $C^{[U:0]} \neq \varnothing$. Claim: $C^{[U:0]} = \Gamma_i^*$ for some Γ_i. Proof of claim: every L_i^U whose corresponding grammar Γ_i is in $C^{[U:0]}$ makes it through C in every total order that begins with C since it has the minimal value on C. Therefore all such corresponding grammars, Γ_i, are in the same Γ-cloud. Assume that the lemma holds for all $k < n$. Assume $C^{[U:n]} \neq \varnothing$. If $C^{[U:n]} = \Gamma^*_i$, we are done. If not, it is the finite union of a set of disjoint Γ-clouds. Wlog $C^{[U:n]} = \Gamma^*_i \cup \Gamma^*_j$, where $\Gamma^*_i \cap \Gamma^*_j = \emptyset$. We lose no generality because the following process constructing the relevant UVT can be repeated any finite number of times. Consider the VT, V, having the same column labels and the same number of rows as U, having identical values in all positions except that in C all bands above n are shifted up one, so that the (n+1)-band is empty in V. Now, consider V′ in which Γ^*_j is moved to the (n+1)-band. Claim: $V, V' \in \mathcal{U}(T)$. The fact that $V \in \mathcal{U}(T)$ is clear since all relations between languages on C are identical to those of U. Now because $\Gamma^{*U}_i \cap \Gamma^{*U}_j = \emptyset$, there is no compulsion from the iOAT(C) that any corresponding language labels have the same value in $\mathcal{U}(T)$. By R.E.S.P.E.C.T. (139), if U is any VT that respects the orders and equivalences of iOAT(T), then $U \in \mathcal{U}(T)$, and so $V' \in \mathcal{U}(T)$. What we have just done is purify the n-band, and so the lemma holds for all n. □

We now recall a familiar term from the theory of partial orders.

(154) **Definition** (standard)**. Cover**. In the arbitrary partial order relation $<$, if $a<b$ and there is no c such that $a<c<b$, then we say that b *covers*, or is a *cover* for, a.

Two preliminary lemmas lead us to our desired theorem.

(155) **Lemma**. For a typology $T = \{\Gamma_1, \ldots, \Gamma_n\}$, if $\Gamma_i \lessdot_C \Gamma_j$ and Γ_j covers Γ_i in the iEQO(C), then $\Gamma_i <_C \Gamma_j$ in htc(EPO(C)).

Proof. Start out with any U in which L_i^U and L_j^U reside in adjacent n-bands, which is possible because Γ_j covers Γ_i . Now, consider the two Γ-clouds, Γ_i^* and Γ_j^*.

First, suppose there are $L_a^U \in \Gamma_i^*$, $L_b \in \Gamma_j^*$, such that $\Gamma_a <_C \Gamma_b$ in the EPO(C). This puts $\Gamma_i <_C \Gamma_j$ in htc(EPO(C)).

Next, for purposes of contradiction, suppose that in the two clouds there are no languages whose grammars are comparable across the clouds, i.e. $\nexists\, \Gamma_a \in \Gamma_i^*, \Gamma_b \in \Gamma_j^*$ such that $\Gamma_a <_C \Gamma_b$. By use of the construction method in the proof of Purifying the Band (153), it is possible to find a UVT $U' \in \mathcal{U}(T)$ in which $L_i^{U'}$ and $L_j^{U'}$, corresponding to Γ_i and Γ_j, are in adjacent, pure bands.

Observe that since Γ_i^*, Γ_j^* have noncomparable languages, this means there does not exist grammars $\Gamma_a \in \Gamma_i^*$, $\Gamma_b \in \Gamma_j^*$ such that their corresponding languages $L_a^{U'}$ and $L_b^{U'}$ compete on some λ = PCQ where $L_a \in PC[K_{U'}]$, $L_b \notin PC[K_{U'}]$ and $L_b \in P[K_{U'}]$.

But this means that the respective values of $L_a^{U'}$ and $L_b^{U'}$ on C are irrelevant to any filtration pattern because whenever $L_a^{U'}$ and $L_b^{U'}$ face off on C, they both lose to some language whose value on C is less than $L_a^{U'}$'s value on C.

Therefore, one can swap $L_a^{U'}$ and $L_b^{U'}$'s C values without changing the filtration pattern. But this produces a new UVT, U″ ∈ $\mathcal{U}$(T) with $C(L_a^{U''}) > C(L_b^{U''})$. This implies that it is not the case that $\Gamma_a \lessdot_C \Gamma_b$ since there is at least one UVT, U″ ∈ $\mathcal{U}$(T) in which $C(L_a^{U''}) > C(L_b^{U''})$. This is a contradiction since $\Gamma_a \in \Gamma_i^*$ and $\Gamma_b \in \Gamma_j^*$, but $\Gamma_i \lessdot_C \Gamma_j$, and therefore $\Gamma_a \lessdot_C \Gamma_b$. □

(156) **Lemma**. If $\Gamma_i \lessdot_C \Gamma_j$ in the *i*EQO(C), then $\Gamma_i <_C \Gamma_j$ in htcEPO(C).

Proof. There's a sequence of $\Gamma_i \lessdot_C \ldots \lessdot_C \Gamma_j$ which has a covering relation at each step ($\lessdot_C$ is a partial order over a finite set of grammars in any T). Repeated applications of Lemma (155) put each step in htcEPO(C), and therefore $\Gamma_i <_C \Gamma_j$ follows in htcEPO(C) as well because $<_C$ is a partial order and therefore transitive. □

(157) **Theorem**. **Order Relations Shared**. For a typology T and constraint $C \in \text{Con}_T$, the order relation $\leqslant_C$ of htcEPO(C) is identical to the order relation $\lessdot_C$ of *i*EQO(C).

Proof. Since $\lessdot_C$ and $\leqslant_C$ are partial orders on the same set of grammars T, we want to show that the ordered pairs of grammars $\pi_{ij} = (\Gamma_i, \Gamma_j)$ that are in each of the partial orders are identical. We already have the subset relation in one direction from Lemma (156), namely $\pi_{ij} \in \lessdot_C \Rightarrow \pi_{ij} \in \leqslant_C$, or more concisely, $\lessdot_C \subseteq \leqslant_C$.

Now we show the reverse, namely $\leqslant_C \subseteq \lessdot_C$. We have $<_C \subseteq \lessdot_C$ from Lemma (110) and $\sim_C = \approx_C$ from Theorem (148). If $\Gamma_a \leqslant_C \Gamma_b$, then there's a sequence of grammars $\Gamma^{(1)}, \ldots, \Gamma^{(n)}$, where $\Gamma^{(1)} = \Gamma_a$ and $\Gamma^{(n)} = \Gamma_b$, with $\Gamma^{(i)} <_C \Gamma^{(i+1)}$ or $\Gamma^{(i)} \sim_C \Gamma^{(i+1)}$. Now, every EPO relation corresponds to a relation in the *i*EQO, and therefore in every UVT in the guise of < and = on the integers, so that the hypertransitive closure of the EPO is in the *i*EQO because the integers are hypertransitively closed. □

Theorem (148) 'Equivalence Relations Shared' and Theorem (157) 'Order Relations Shared' together establish that htcMOAT(T) and *i*OAT($\mathcal{U}$(T)) are identical in every respect. We record this explicitly here:

(158) **Theorem**. htcMOAT = *i*OAT. For any typology T, htcMOAT(T) = *i*OAT($\mathcal{U}$(T)).

Proof. This asserts $\langle T, \leqslant, \sim\rangle = \langle T, \lessdot, \approx\rangle$. Theorem (148) establishes the equality of $\leqslant$ and $\lessdot$. Theorem (157) establishes the equality of $\sim$ and $\approx$. □

The information in the *i*OAT derives (by nonconstructive, global definition) from every UVT that yields the target typology. We have shown that the very same information can be obtained effectively and locally from the typology itself, through analysis of its Border Point Pairs.

3.2.4 The MOAT determines

We are now but a short step away from showing that MOAT(T) completely characterizes the typology T. First, we observe that any UVT instantiating MOAT(T) delivers T.

(159) **Theorem**. **Instantiation**. If V is a UVT that instantiates MOAT(T), then $V\in\mathcal{U}(T)$.

Proof. By Theorem (158), htcMOAT(T) = *i*OAT($\mathcal{U}$(T)). By Theorem (139), if V is a UVT that instantiates *i*OAT($\mathcal{U}$ (T)), then $V\in\mathcal{U}(T)$. Therefore, if V instantiates the MOAT(T), then V necessarily instantiates the htcMOAT(T). And if V instantiates the htcMOAT(T), then $V\in\mathcal{U}(T)$. But if V instantiates htcMOAT(T), it also instantiates MOAT(T), because because the hypertransitive closure of the MOAT adds to it relations that are present in the integers, which come hypertransitively closed. □

This leads us immediately to our main result.

(160) **Theorem. Our MOATish Mother**. Let T_1, T_2 be typologies over the same set of constraints CONS. Then $MOAT(T_1) = MOAT(T_2) \Rightarrow T_1 = T_2$.

Proof. Let U be a UVT that instantiates $MOAT(T_1)$. Then it also instantiates $MOAT(T_2)$. From Theorem (159), we have $U\in\mathcal{U}(T_1)$ and $U\in\mathcal{U}(T_2)$. Since a UVT yields a unique typology, $T_1=T_2$. □

We might want to think of this in a somewhat more general way, considering the situation where T_1 and T_2 are not written over the *same* CON_S. In that case, we would want to speak of a bijection between $\text{CON}(T_1)$ and $\text{CON}(T_2)$ that respects the relations of the MOAT. But this amounts to no more than identifying the names of the constraints across the two sets, which can be more crudely accomplished by simply giving them the *same* names, as we have done throughout.

We briefly note that there is an alternate geometric proof: as we establish in section 6, grammars delimit connected regions on the permutohedron, and the border points of a grammar are exactly the borders of the region defining the grammar. Therefore, grammars in two typologies that have identical border points consisting of exactly the same total orders. This shows that $T_U = T_V$.

MOAT(T) determines the structure of any UVT that yields T. It is ecologically typical, however, for an OT system to sponsor many csets, often an unbounded number.With the UVT results in hand, we now show that the MOAT also determines the behavior of the constraints within each of these, though in the case of the order relation, not quite as strictly as in the case of the single UVT.

● *Equivalence* in EPO(C) entails that within every cset the constraint C will assign equal values to the candidates belonging to the languages associated with C-equivalent grammars.

● *Order* between two grammars in EPO(C) entails that within every cset the values assigned by constraint C will respect the nonstrict version of the order between the candidates belonging to the languages associated with the C-ordered grammars. In short, '$<_C$' leads to numerical '$\leq$' within individual csets, with the caveat that in any universal support — any collection of csets from which the typology may be derived — there must be at least one cset in which the order is strict.

To see why this is so, consider a set of VTs $\mathbb{V} = \{V_1,\ldots,V_n\}$, each with columns indexed to the same set of constraints $\textsc{Con}_S$. Each VT $V_i \in \mathbb{V}$ has a set of candidates A_i, which we take to be simply the rows of the VT. A language of $\mathcal{V}$ is an element $\mathbf{a} \in A_1 \times \ldots \times A_n$ which is such that there is a grammar Γ over $\textsc{Con}_S$ under which each candidate $\mathbf{a}_i \in A_i$ optimal in A_i. For convenience, and without terrific loss of generality, we assume that each optimum is unique.

We can derive a UVT U from $\mathbb{V}$ by taking its Minkowski sum $U = \oplus V_i$ and plucking out the harmonically bounded candidates. Let's consider two candidates $\mathbf{a}^{\oplus}$, $\mathbf{x}^{\oplus} \in U$, which we notate as follows, subscripting the candidates to indicate their cset.

$$\mathbf{a}^{\oplus} = \mathbf{a}_1+\ldots+\mathbf{a}_n$$
$$\mathbf{x}^{\oplus} = \mathbf{x}_1+\ldots+\mathbf{x}_n$$

A constraint C assigns a value to each candidate: $C(\mathbf{a}_i)$, $C(\mathbf{x}_i)$ are the values assigned to candidates $\mathbf{a}_i$, $\mathbf{x}_i \in A_i$, which are laid out as rows of $V_i \in \mathbb{V}$

We first establish a key observation: any strict order inside a single V_i on a given constraint C can be magnified so as to determine the UVT order relation on C between entire languages, without affecting the grammatical structure of the typology.

(161) **Lemma**. **Inflation**. Let T be a typology over $\textsc{Con}_S$, and let $\mathbb{V} = \{V_1,\ldots,V_n\}$ be a set of VTs over $\textsc{Con}_S$ which provide a universal support for T. Let U be $\oplus V_i$ with all harmonically bounded rows removed, therefore a UVT for T. Let $\mathbf{a}^{\oplus}$, $\mathbf{x}^{\oplus}$ be rows of U, corresponding to Γ_a, $\Gamma_x \in T$. Suppose for some $C \in \textsc{Con}_S$, $C(\mathbf{a}_1) > C(\mathbf{x}_1)$. Then there is a UVT V for T such that $C(g_V(\Gamma_a)) > C(g_V(\Gamma_x))$.

Proof. Suppose that $C(a_1) > C(x_1)$ but $C(a_1)+\ldots+C(a_n) \leq C(x_1)+\ldots+C(x_n)$, so that in U, we have $C(\mathbf{a}^{\oplus}) \leq C(\mathbf{x}^{\oplus})$. We manipulate this expression to isolate the terms from V_1.

$$(C(a_1)+\ldots+C(a_n)) - ((C(x_1)+\ldots+C(x_n)) \leq 0$$
$$(C(a_1) - C(x_1)) + \quad \ldots \quad + (C(a_n) - C(x_n)) \leq 0$$
$$(C(a_1) - C(x_1)) \leq (C(x_2) - C(a_2)) +\ldots+ (C(x_n) - C(a_n))$$

Observe that the expression on the left hand side of the last inequality is positive, because by assumption $C(a_1)>C(x_1)$. By the Archimedean property of the integers, $\exists$ $n > 0$ s.t

$$n\cdot(C(a_1) - C(x_1)) > (C(x_2) - C(a_2)) +\ldots+ (C(x_n) - C(a_n))$$

Running through the manipulations in reverse, we arrive at the following:

$$(*)\ n\cdot C(a_1)+ \ldots+C(a_n) > n\cdot C(x_1)+\ldots+C(x_n)$$

Now consider the typology T* derived from the collection of VTs $\mathbb{V}$* obtained by replacing V_1 in $\mathbb{V}$ by $n\cdot V_1$, the matrix in which all values of $\mathbb{V}$ are multiplied by n. (Equivalently, one may think of $\mathbb{V}$* as expanding $\mathbb{V}$ by adding n–1 copies of V_1 to it.) In either case, the set of grammars is clearly unchanged, so that T = T*.

Now construct a UVT U* from the Minkowski sum $\oplus\mathbb{V}$* by removing all harmonically bounded rows. Because T = T*, we have Γ_a, $\Gamma_x \in$ T*, and the candidates from $\mathbb{V}$* that belong to the associated languages are exactly the same except in the case of the replacement for V_1, where their values have been multiplied by n, and the relevant sums that appear in the UVT U* are as in (*) above. From (*), it is clear that $C(g_{U*}(\Gamma_a)) > C(g_{U*}(\Gamma_x))$, as desired. □

From this lemma, we may immediately derive the effects of EPO relations on the individual VTs that are derived from an OT system.

First, we are guaranteed that EPO equivalence $\sim_C$ forces numerical equality on C in every VT of the system. This follows because the only way we can achieve equality in all of the UVTs of the typology is to have equality in all the VTs.

(162) **Theorem**. **Equivalence Respected**. Let T = $\{\Gamma_1,\ldots, \Gamma_n\}$ be a typology over CON_S. Suppose that in EPO(C) for some C$\in\text{CON}_S$, we have $\Gamma_i \sim_C \Gamma_j$ for some Γ_i, $\Gamma_j \in$T. Let V be any VT allowed by S, where **a** is a candidate of V that is optimal under Γ_i and **x** is a candidate of V that is optimal under Γ_j. Then C(**a**) = C(**x**).

Proof. Because by assumption $\Gamma_i \sim_C \Gamma_j$ in EPO(C), we have by Theorem (158) that $\Gamma_i \approx_C \Gamma_j$ in iEQO(C). This means that $C(L_i^U) = C(L_j^U)$ for every U$\in\mathcal{U}$ (T). Now consider a VT V in which the candidate **a** is optimal under Γ_i and the candidate **x** is optimal under Γ_j. Suppose for purposes of contradiction that C(**a**)$\neq$C(**x**). We may also suppose wlog that C(**a**)>C(**x**). By the Inflation Lemma, we are assured of a UVT V$\in\mathcal{U}$(T) in which $C(L_i^V) > C(L_j^V)$, a contradiction. □

Second, we are guaranteed that EPO order $<_C$ requires numerical $\leq$ on C in every VT of the system. The only way to achieve a strict order in all UVTs of the typology is for it never to be reversed in any VT of the system.

(163) **Theorem**. **Order Weakly Respected**. Let T = $\{\Gamma_1,\ldots, \Gamma_n\}$ be a typology over CON_S. Suppose that in EPO(C) for some C$\in\text{CON}_S$, we have $\Gamma_i <_C \Gamma_j$ for some Γ_i, Γ_j $\in$T. Let V be any VT allowed by S, where **a** is a candidate of V that is optimal under Γ_i and **x** is a candidate of V that is optimal under Γ_j. Then C(**a**) $\leq$ C(**x**).

Proof. Because by assumption $\Gamma_i <_C \Gamma_j$ in EPO(C), we have by Theorem (158) that $\Gamma_i \lessdot_C \Gamma_j$ in *i*EQO(C). This means that $C(L_i^U) < C(L_j^U)$ for every $U \in \mathcal{U}(T)$. Now consider a VT V_k allowed by the system in which the candidate **a** is optimal under Γ_i and the candidate **x** is optimal under Γ_j. Suppose for purposes of contradiction that $C(\mathbf{a}) > C(\mathbf{x})$. By the Inflation Lemma (161), we have a UVT $V \in \mathcal{U}(T)$ in which $C(L_i^V) > C(L_j^V)$, a contradiction. □

Observe that if we have a collection of csets that provides a universal support for T, it must be the case that in at least one of them the EPO relation $\Gamma_i <_C \Gamma_j$ is instantiated as numerical $<$. Since the universal support can be Minkowski-summed into a UVT (with removal of irrelevant, harmonically bounded rows), and since $\Gamma_i <_C \Gamma_j$ is instantiated in every UVT, it cannot be that relation $\leq$ on C-values is everywhere realized as equality.

In addition to equivalence and strict order, we know that the EPO also allows noncomparability between grammars. In the world of UVTs, we know that this means that numerical instantions include both $<$ and $>$ (as well as $=$) in different $U \in \mathcal{U}(T)$. From the theorem, we now also have it that the same variety can appear for noncomparables in various component VTs allowed by the system.

Regarding languages $\mathbf{a}_i \in A_1 \times \ldots \times A_i \times \ldots$, where the A_i are candidate sets, we recognize that the EPO orders and equivalences between their grammars are induced by the familiar *coordinatewise* order and equality relations derived from these candidate vectors. Coordinatewise relations on vectors are defined as follows from relations on the components, writing $C(\mathbf{x})$ for $(C(\mathbf{x}_1), \ldots)$:

- Equality. $C(\mathbf{a}) = C(\mathbf{b})$ iff for all i, $C(\mathbf{a}_i) = C(\mathbf{b}_i)$.
- Order. $C(\mathbf{a}) < C(\mathbf{b})$ iff for all i, $C(\mathbf{a}_i) \leq C(\mathbf{b}_i)$ and for some j, $C(\mathbf{a}_j) < C(\mathbf{b}_j)$.

These are exactly the relations inherited by languages from the EPO relations, which derive from the intrinsic structure of typologies, which itself derives without additional premises from the definition of optimality in OT.

We conclude by observing that, for a given typology T, if any collection of csets (1) gives rise to *k* grammars and (2) obeys the order and equivalence restrictions of Theorems (162) and (163), then it will also produce T. This is because the Minkowski sum of these csets instantiates MOAT(T), and therefore by Theorem (159) yields T.

3.2.5 Summary

In this section we have shown how the UVT-based *i*OAT completely characterizes the order and equivalence relations that must obtain in any UVT that instantiates a given typology (§3.2.2). We then have shown how the boarder-point pair constructed MOAT, when hypertransitiviely closed, has an identical relational structure to the *i*OAT. This leads to the conclusion, established in (160), that the MOAT characterizes a typology in that any two typologies having identical MOATs are themselves identical.

3.3 The Well-formed MOAT

I was lucky in the order.
– William Munny

A typology is a certain kind of partition: one that is imposed by a UVT. We repeat the definition that we have used throughout.

(164) **Definition. Typology**. Given a set of constraints $\mathrm{C{\scriptstyle ON}}_S$, a partition of Ord(S), the set of all orders on $\mathrm{C{\scriptstyle ON}}_S$, is a typology iff there is a UVT U, with columns that correspond 1:1 to the constraints of $\mathrm{C{\scriptstyle ON}}_S$ and rows that correspond 1:1 to the grammars of T, such that each block in the partition T is the ranking grammar of a row in U.

Each typology has a unique MOAT, which derives directly from the contents of the partition, without reference to a UVT or a set of UVTs. This suggests an alternate route to the notion *typology*, based on its intrinsic order and equivalence structure rather than on its plentiful numerics. Our target, then, is to show that a partition is a typology iff it has a MOAT.

Some care is required, because the MOAT concept has been developed in terms of a prior assumption that its sponsoring partition is already known to be a typology under definition (164). We must therefore characterize the relevant order and equivalence structure in a way that doesn't depend on reference to a witnessing UVT.

Observe first that border point analysis may be applied to any partition of a set of linear orders on a reference set S, which we have denoted as Ord(S). Any such partition consists of disjoint sets that support a concept of adjacency based on pairwise transposition. Between adjacent blocks of such a partition, two types of relation may be discerned: let's call them R_o and R_e, to indicate their connection with the relations of order and equivalence that have been shown to arise when the partition is a typology according to definition (164). From these, we can construct a generalized version of the MOAT, call it the GMOAT, that exactly parallels the MOAT, consisting of generalized EPOs (GEPOs) for each element of S. We will then be in a position to show that a partition possessing a certain kind of GMOAT is a typology.

There are two routes to the GMOAT of a given partition π. We may directly examine π through border point analysis and construct the GMOAT from the results. Or we may regard π as the result of unioning blocks from another partition π' that is already known to be a typology. In this case, we can construct the GEPOs of GMOAT(π) as bigraphs by merging nodes in the EPO bigraphs of the background typology π', with graphical node merger corresponding to union of the blocks denoted by the nodes.

The second tactic sounds like it might be of limited applicability because it depends on a prior typology. But it is in fact guaranteed to work for arbitrary π. The maximally refined partition, in which every block contains a single element of the underlying set, is certainly a typology: it is the *Discrete Typology* (DT), in which every grammar consists of a single linear order.[38] Any other partition is a coarsening of this one: any partition at all can be derived by unions of the elements of the DT. It follows that the GEPOs of any GMOAT can be derived by mergers of nodes in the EPOs of the DT's MOAT. We are thus guaranteed that an arbitrary partition can be arrived at by coarsening some typology.

The two paths to the GMOAT can be represented in the following diagram, where *U* is a function mapping a typology T to a partition π by block unions, *merge* is a function on graphical structures that merges the nodes corresponding to the blocks unioned by *U*, M designates *MOAT*, GM *GMOAT*, and *bpa* is the function that delivers the GMOAT from a partition, which is guaranteed to be MOAT when the partition is a typology.

(165) **Paths to the GMOAT**

$$\begin{tikzcd} T \arrow[r, "U"] \arrow[d, "bpa"'] & \pi \arrow[d, "bpa"] \\ \mathrm{M}(T) \arrow[r, "merge"] & \mathrm{GM}(\pi) \end{tikzcd}$$

The key fact is that $bpa \circ U = merge \circ bpa$. This holds because merger of nodes preserves graphical information in a way that exactly parallels the preservation of border point information under union.

To see this, consider the following. The border of a union of blocks is exactly the union of the their borders, less those original border point pairs that link unioning blocks. On the graphical side, merger of nodes produces a node whose outgoing and incoming edges are exactly those of the original nodes, less those edges that run between pairs of merging nodes. Since the surviving edges of the graph are derived from the surviving border point pairs, their exact correspondence is maintained.

We want the conditions under which π = *U*(T) is a typology. Arbitrary unions, even of adjacent blocks, can fail to produce a typology. Failure is associated with the production of cycles in the bigraph structure: a cyclic GEPO cannot be instantiated as a column in a VT. Acyclicity is clearly a necessary condition for MOAT-hood. We assert that it is also *sufficient*, in the following sense: for a partition π, if GMOAT(π) is acyclic, then π is a typology and GMOAT(π) is its MOAT. When this thesis is established below, we will

[38] The Discrete Typology is easily shown to meet the requirements of definition (164). For $\mathrm{CON}_{DT} = \{X_1,\ldots,X_n\}$, let the rows of a VT consist of all $n!$ permutations of the integers $\{0,1,\ldots, n\}$. Each row selects the linear order on CON_{DT} for which, according to that order, the values occur in a strictly increasing sequence. This VT is therefore a UVT for the DT based on CON_{DT}.

have characterized the notion 'typology' as a kind of order structure, above and beyond the relationship with UVTs.

(166) **Thesis**. For a partition π, if GMOAT(π) is acyclic, then π is a typology and GMOAT(π) is its MOAT.

Our strategy will be first to show that this holds when the partition π is derived from a known typology by node-merger/block-union, and then to observe, as we have just done, that every partition can be so derived.

3.3.1 Acyclicity

We first clarify how border point analysis works on a general partition and use the results to define the key notion of acyclicity.

Reconstructing border point analysis in this context involves little more than deploying notation. In the original development, we used decorated versions of the symbols '$<$' and '$\sim$', which were suitable because the various relations involved were navigating in the direction of order and equivalence. In the general case, the assurance of order fails, but the formal construction is basically identical and will serve to focus attention on the key issues.

For each constraint X∈$\text{C}_{\text{ON}_\text{S}}$, we are interested in its generalized EPO, or GEPO(X), which will contain the relations $R_{o/X}$ and $R_{e/X}$ — the analogs of typological $<_X$ and $\sim_X$ in the world of general partitions on Ord(CON_S). As with the MOAT construction of §3.1 definition (105) , we start out with *base relations* $R^b_{o/X}$ and $R^b_{e/X}$ defined directly from linear orders. These record particular facts about the relations between blocks that follow directly from their contents.

(167) **Definition. Base relations from a Border Point Pair.** Given a partition π of Ord(CON_π), with π ={B_1,…B_n}, we define for each X∈CON_π the relations $R^b_{o/X}$ and $R^b_{e/X}$ on π.

a. $R^b_{o/X}(B_j, B_k)$ iff there is a border point pair P$\underline{XY}$Q∈B_j, P$\underline{YX}$Q∈B_k.

b. $R^b_{e/X}(B_j, B_k)$ iff there is a border point pair PQ∈B_j, PQ′∈B_k with X∈P.

Observe that $R^b_{e/X}$ is symmetrical, because the relevant border point pair shares the determining prefix. By contrast, $R^b_{o/X}$ —the analog of $<_X$ — is not guaranteed to be asymmetrical. Nothing forbids a block in a general partition from containing border points P$\underline{XY}$Q, and P′$\underline{YX}$Q′.

To obtain $R_{o/X}$ from $R^b_{o/X}$ — as with $<_{X,}$ in definition (108) — we transitively close $R^b_{o/X}$. This will ensure that $R_{o/X}$ is the partial order $<_X$ when π is a typology.

(168) **Definition**. $R_{o/X}$. $R_{o/X}$ is the transitive closure of $R^{b}_{o/X}$.

Paralleling the construction of $\sim_X$, we obtain the relation $R_{e/X}$ by transitively closing the reflexive closure of $R^{b}_{e/X.,}$ as in definition (109).

(169) **Definition**. $R_{e/X}$. $R_{e/X}$ is the transitive closure of the reflexive closure of $R^{b}_{o/X}$.

Observe that $R_{e/X}$ is symmetric because it is derived from $R^{b}_{e/X}$ in a way that does not affect its native symmetry. Since $R_{e/X}$ is reflexive and transitive by construction, it follows that it is an equivalence relation. We therefore replace $R_{e/X}$ with $\sim_X$.

We may now define the notions GEPO and GMOAT, paralleling EPO and MOAT.

(170) **Definition**. GEPO(X) for π. For a partition π of Ord($\textsc{Con}_\pi$)

$$\text{GEPO(X)} = \langle \pi, R_{o/X}, \sim_X \rangle$$

The GMOAT collects all GEPO(X) for every X in $\textsc{Con}_\pi$.

(171) **Definition**. GMOAT(π). For a partition π,

$$\text{GMOAT}(\pi) = \{\text{GEPO(X)} \mid X \in \textsc{Con}_\pi\}$$

The GEPO(X), like EPO(X), is interpreted as a mixed graph with both directed and undirected edges. The undirected edges arises from the relation ∼X. The directed edges arise from $R_{o/X}$, paralleling the graphical interpretation of the EPO(X).

(172) Interpeting $R_{o/X}(B_j, B_k)$. Suppose that $R_{o/X}(B_j, B_k)$ in π. Then there is a directed edge (B_j, B_k) from B_j to B_k in GEPO(X).

(173) Interpreting $B_j \sim_X B_k$. Suppose that $B_j \sim_X B_k$ in π. Then there is an undirected edge (B_j, B_k), equivalently (B_k, B_j), in GEPO(X).

(174) Finis. There are no other edges in GEPO(X).

We are now in a position to define acyclicity. Two nodes a_j, a_k are *adjacent* if (a_j, a_k) is an edge. A *path* is a sequence of nodes in which each sequential pair is adjacent. A *directed path* $(a_1,\ldots,a_n)$ is a path in which each subsequence a_k, a_{k+1} is either an undirected edge or a directed edge (a_k, a_{k+1}). In such a path all directed edges go in the same direction. We require that a directed path contain at least one directed edge. A *directed cycle* is a directed path from a node back to itself: a directed path $(a_1,\ldots,a_n)$ in which $a_1 = a_n$. We pull out this last notion.

(175) Directed cycle in a GEPO. A directed path $(a_1,\ldots, a_n)$ is a *directed cycle* iff $a_1 = a_n$.

We may now specify what it means for a GEPO bigraph to be acyclic.

(176) **Definition**. **Acyclic**. A GEPO bigraph is *acyclic* iff it contains no directed cycles.

A GMOAT is said to be acyclic iff all of its GEPOs are acyclic. With this, we have now specified the content of thesis (166), repeated here.

(177) **Thesis**. For a partition π, if GMOAT(π) is acyclic, then π is a typology and GMOAT(π) is its MOAT.

To see what acyclicity amounts to in the realm of relations, we must define the hypertransitive closure of $R_{o/X}$ with respect to $\sim_X$, paralleling the notion of hypertransitive closure of $<_{X.}$. As with the other definitions here, we closely follow the approach of §3.2.

(178) **Definition. Equivalence-extended relation (EER)**. Given a structure $\Sigma = \langle S, R_1, \sim\rangle$, with R_1 a binary relations on the set S, and '$\sim$' an equivalence relation on s, define EER(Σ) to be $\langle S, R_1^{\sim}\rangle$, where $R_1^{\sim}$ is a relation on S, defined as follows. For all a, b, c $\in$ S,

$$R_1(a,b) \Rightarrow R_1^{\sim}(a,b)$$
$$a{\sim}b \ \&\ R_1(b,c) \Rightarrow R_1^{\sim}(a,c)$$
$$a{\sim}b \ \&\ R_1(c,b) \Rightarrow R_1^{\sim}(c,a)$$

Note that $R_1^{\sim}$ is not necessarily a partial order, even if R_1 is a partial order, because it may lack transitive closure. If R_1 happens to be a (strict) partial order, $R_1^{\sim}$ will be asymmetric and irreflexive, because it will inherit these properties from R_1. To give $R_1^{\sim}$ a chance to be a partial order when R_1 is compatible with that possibility, we construct an extended order $\hat{R}_1^{\sim}$ from $R_1^{\sim}$ by simply taking its transitive closure. This we will call the 'hypertransitive closure' of R_1 with respect to $\sim$. We also apply the term to the GEPO structure itself.

(179) **Definition. Hypertransitive closure of GEPO**. For GEPO(X) = $\langle \pi, R_{o/X}, \sim_X \rangle$, the hypertransitive closure of GEPO(X), which we denote htcGEPO(X) is defined to be the bigraph $\langle \pi, \hat{R}^{\sim}_{o/X}, \sim_X \rangle$, where $\hat{R}^{\sim}_{o/X}$ is the transitive closure of the relation $R^{\sim}_{o/X}$ and $\sim_X$ is the relation $\sim_X$ in GEPO(X).

The assertion that a GEPO bigraph is acyclic now translates to the claim that in htcGEPO(X), the relation $\hat{R}^{\sim}_{o/X}$ is a partial order.

3.3.2 Rootedness

The *root* of a directed graph, in one sense of the term, is "a distinguished node r, such that there is a directed path from r to any node other than r." (See "Rooted Graph," Wikipedia.) We generalize the notion to embrace a set of equivalent nodes in a bigraph. It may directly applied to a bigraph 'modded out' by equivalence, in which equivalent nodes are identified (see fn 18 p. 35 above). Here we give a direct account of the notion.

(180) **Definition**. Rootedness. A bigraph $\beta = \langle S, R, \sim\rangle$, for a partial order R and its hypertransitive closure, also a partial order, is *rooted* if there is at least one element $x\in S$ such that $\forall y\in S$, $x\leqslant y$ or $x\sim y$, where $\leqslant$ is the order relation in htc(β). The *root* of β is the set of all such x.

The elements of the root are strictly less than all other elements under the relation $\leqslant$, and they are equivalent to each other under the relation $\sim$.

We note two basic properties of selection that derive from the character of initial sequences of constraints in a leg, which we term 'prefixes'. First, if a grammar contains a leg with a certain prefix P, then among the rows selected by P will be the row corresponding to that grammar.

(181) **Lemma.** Prefixal Selection**.** If $\lambda = PQ\in G_k$, then $r_k\in P[U]$ for all $U\in\mathcal{U}(T)$.

Proof. By def $\lambda[U] = r_k$. Ergo, $r_k\in P[U]$ since $\lambda=PQ$. □

Conversely, if a grammar contains no legs beginning with a certain prefix P, then the prefix P selects only rows that do not correspond to the grammar.

(182) **Lemma**. Prefixal Exclusion. Suppose G_k contains no legs $\lambda = P\ldots$ for some given sequence of constraints P. Then $r_k\notin P[U]$ for any $U\in\mathcal{U}(T)$.

Proof. Let G_k have no legs $\lambda = P\ldots$. Suppose *per contradictio* that $r_k\in P[U]$. Then by No Dead Men Walking, Lemma (117), there is a continuation Q of P such that $r_k\in PQ[U]$. But this means that $\lambda = PQ\in G_k$. Contradiction. □

Filtration applies to the rows of a UVT U. The rows correspond 1:1 to grammars in T_U. Through this correspondence, there is a parallel filtration process induced by a linear order on the *grammars* of T_U, playing out among the EPOs of MOAT(T_U). This creates objects that we will call fEPOs, for 'filtered EPOs'. These are EPOs with the ejected grammars removed. The bigraph structure among the retained grammars is of considerable interest.

(183) **Definition**. fEPO(X)|P. The 'filtered EPO of X with respect to P', written fEPO(X)|P, is defined recursively as follows.

(1) fEPO(X)|C is the sub-bigraph of EPO(X) equal to $\langle K, <_k, \sim_k\rangle$, where K⊆S consists of the grammars in the root of EPO(C).

(2) fEPO(X)|PC, for P a sequence of constraints, is the sub-bigraph of EPO(X) equal to $\langle K, <_z, \sim_z\rangle$, where K⊆S consists of the grammars in the root of fEPO(X)|P.

The definition presupposes that both EPO and fEPO are rooted. The claim is that a (f)EPO contains the crucial structure that allows for OT filtration: the presence of a set of minimal, equivalent elements — a root. We now show this to be true.

(184) **Lemma**. Rootedness. Let P be a sequence of constraints and Z a single constraint not in P. Then fEPO(Z)|P is rooted.

Proof. The proof requires two parts. Recall that PZ[U]⊆P[U].

1. The grammars corresponding to rows in PZ[U] are equivalent in EPO(Z).
2. The grammars corresponding to rows in P[U]\PZ[U] stand in the relation $>_Z$ to those in PZ[U].

Proof of 1. Let $J^{PZ} = \{\Gamma_1,\ldots, \Gamma_k\}$ be the grammars of T containing a leg λ_j = PZ… . By the Prefixal Selection Lemma (181), they're all in PZ[U], and by the following *i*OAT-based considerations, they must be equivalent on every constraint in PZ.

U is an arbitrary instantiation of MOAT(T). Since the rows corresponding to the grammars of J^{PZ} all pass through PZ together in U, they must pass through PZ together in every V∈ $\mathcal{U}$(T), by Filtration Uniformity (123). The rows corresponding to the grammars are equal numerically in every such V. By the definition of *i*OAT, they are therefore Z-equivalent in the iOAT(T) and therefore equivalent in EPO(Z), by Theorem (158).

Proof of 2. Let J^P = grammars corresponding to rows in P[U] for arbitrary U∈$\mathcal{U}$(T). Consider $J^P\backslash J^{PZ}$. For S⊆T, let ρ(S) be be the rows in U corresponding to these grammars. Then $\rho(J^P\backslash J^{PZ})$ contains the rows that pass through P and then are ejected by Z. Numerically, for any $r\in\rho(J^P\backslash J^{PZ})$ and any $s\in\rho(J^{PZ})$, Z(s)<Z(r). Because U is arbitrary, this relation holds in every V∈ $\mathcal{U}$(T). Therefore, for each $\Gamma\in J^P\backslash J^{PZ}$ and each $\Gamma'\in J^{PZ}$, $\Gamma' <_Z \Gamma$ in EPO(Z).

1 & 2 combine to show that fEPO(Z)|P is rooted. □

An EPO is a fEPO, so it follows immediately that EPOs are rooted as well.

(185) **Corollary**. Let $X\in \textsc{Con}_S$ in typology T_S. Then EPO(X) is rooted.

Proof. Let P=∅ in the Rootedness Lemma (184). □

3.3.3 The MOAT characterizes the Typology Object

We will now establish the order properties of the GMOAT are sufficient to determine its status as MOAT.

(186) **Thesis**. For a partition π, if GMOAT(π) is acyclic, then π is a typology and GMOAT(π) is its MOAT.

In this section, we assume that we have a typology $T = \{\Gamma_1, \ldots, \Gamma_n\}$ over $\text{C}{\small\text{ON}}_S$ = $\{X_1, \ldots X_m\}$, with M(T) as its MOAT. In addition, we have a partition of Ord($\text{C}{\small\text{ON}}_S$) $\pi = \{B_1, \ldots, B_p\}$, $p \leq n$, where each B_i is the union of some set of grammars of T. Finally, we assume that its GMOAT GM(π) is acyclic.

The relations of GM(π) are inherited from those of M(T), in this sense: when two grammars of T reside in different blocks of π, then a relation in M(T) between the two grammars manifests as a relation between those two blocks in GM(π).

(187) **Lemma.** Inheritance. For $B_j, B_k \in \pi$, $C \in \text{C}{\small\text{ON}}_S$

a. If there are $\Gamma_j \subseteq B_j$ and $\Gamma_k \subseteq B_k$ such that $\Gamma_j \sim_C \Gamma_k$, then $B_j \sim_C B_k$.

b. If there are $\Gamma_j \subseteq B_j$ and $\Gamma_k \subseteq B_k$ such that $\Gamma_j <_C \Gamma_k$, then $B_j <_C B_k$

Proof.

a. Equivalence. If $\Gamma_j \sim_C \Gamma_k$, then there is a chain of pairwise equivalences between them, graphically an undirected path. Under merger, this yields a chain of equivalences between merged blocks, since the edges are preserved between merged blocks. So $B_j \sim_C B_k$.

b. Order. If $\Gamma_j <_C \Gamma_k$, then Γ_j and Γ_k are the endpoints of a directed path containing no equivalences. In π, this directed path is merged down to at least two blocks. Edges are preserved between merged blocks, so the result is still a directed path. So $B_j <_C B_k$. □

Inheritance from an EPO endows a GEPO with key EPO structure.

(188) **Lemma**. GEPO Rootedness. Each GEPO in an acyclic GMOAT is rooted.

Proof. Let $\Gamma \in T$ be in the root of EPO(X). This means that $\Gamma <_X \Gamma'$ or $\Gamma \sim_X \Gamma'$ for every other grammar $\Gamma' \in T$. There's some block $B \in \pi$ such that $\Gamma \subseteq B$. Claim: $B \in$ root(GEPO(X)). Note that the Γ' must lie inside blocks of π as well. Let $B_i \in \pi$. Then $\exists\, \Gamma_i \in T$, such that $\Gamma_i \subseteq B_i$ and $\Gamma <_X \Gamma_i$ or $\Gamma \sim_X \Gamma_i$. By inheritance, $B <_X B_i$ or $B \sim_X B_i$. This means, by the definition of root, that GEPO(X) has a root and B is in it. □

An acyclic GMOAT can be instantiated as a VT V. To establish our thesis, we will show that the typology of V is identical to the partition π from which the GMOAT was derived.

Since the GMOAT was constructed in exactly the same way that a MOAT is constructed, if the partitions are the same, then the GMOAT must be the same as the MOAT.

(189) **Theorem.** Let π be a partition produced by union of blocks from a typology T over $\textsc{Con}_S$. If each GEPO(X), $X \in \textsc{Con}_S$, is acyclic, then π is a typology and GMOAT(π) is MOAT(π).

Proof. Let $T = \{\Gamma_1, \ldots, \Gamma_n\}$ be a typology over $\textsc{Con}_S = \{X_1, \ldots X_m\}$ and let M(T) be its MOAT. Union collections of grammars to produce a new partition π of Ord($\textsc{Con}_S$) such that its GMOAT GM(π) is acyclic.

We have $\pi = \{B_1, \ldots, B_p\}$, $p \leq n$, where each B_i is the union of some set of grammars of T. Because by assumption GEPO(X) is acyclic for any X, it may be instantiated numerically by assigning non-negative integers to each B_j in a way that respects the relations of GEPO(X). We may therefore instantiate GMOAT(π) in a VT V whose k^{th} column instantiates GEPO(X_k) and whose j^{th} row r_j consists of the instantiations of B_j.
We recognize a set of instantiating functions, one for each $X_k \in \textsc{Con}_S$. This provides the abstract analog of the constraint functions of Concrete OT. We therefore name the instantiating functions by the constraints in $\textsc{Con}_S$.

$X_k: \pi \rightarrow \mathbb{N}$

This allows us to spell out V entry-wise as

$V_{jk} = X_k(Bj)$

V then consists of rows r_j, $1 \leq j \leq p$, each of which is associated with $B_j \in \pi$ via this expression.

The VT V gives rise to a typology $T_V = \{G_1, \ldots, G_q\}$ on $\textsc{Con}_S$. We are not, however, guaranteed that the grammars of T_V are equal to the blocks of π. We are not even guaranteed that V is a licit UVT, because we haven't shown that it has no harmonically bounded rows; thus, we can't even assume safely that $p = q$, where p is the number of blocks in π and q the number of grammars in T_V.

Consider V to be a set of rows $\{r_1, \ldots, r_p\}$. The grammars of T_V are related to V by filtration: for every $\lambda \in G_j \in T_V$, $\lambda[V] = r_j$. The rows of V are related to the blocks $B_j \in \pi$ by instantiation: the entries of r_j instantiate the order relations of B_j.

$$\begin{tikzcd} G_j \in T_V \arrow[dr, "select"'] & & B_j \in \pi \arrow[dl, "instantiate"] \\ & r_j \in V & \end{tikzcd}$$

Our goal is to show that $G_j = B_j$. If this is true, then $\pi = T_V$, so that the partition π is a typology and GM(π) = M(T_V), establishing the theorem.

We need to show that, for every block $B_j \in \pi$, every linear order $\lambda \in B_j$ *selects* the row that instantiates B_j in V, so that $B_j \subseteq G_j$; and no others do, so that $G_j \setminus B_j = \varnothing$.

Consider any $\lambda \in \mathrm{Ord}(S)$. The linear order λ belongs to some block in π and this block has an instantiating row $r \in V$. We will call this block B^r. To show that the blocks of π are the grammars of T_V, it suffices to establish the following two statements about λ and B^r.

(*) if $\lambda \in B^r$, then $\lambda[V] = r$	'if λ is in B^r, then λ selects r'
(**) if $\lambda[V] = r$, then $\lambda \in B^r$	'if λ selects r, then λ is in B^r'

Like all the blocks of π, B^r is the union of grammars of T. Among these grammars will be one to which λ belongs: call this Γ^λ.

$$B^r = \bigcup_{j=1}^{m} \Gamma_{a_j} = \ldots \cup \Gamma^\lambda \cup \ldots = \{\ldots, \lambda, \ldots\}$$

To establish (*) and (**), we show that for arbitrary prefix $P_i\boldsymbol{C_i}$, where $P_i = C_1 \ldots C_{i-1}$, that $r \in P_iC_i[V]$ if and only if B^r contains a linear order with the prefix $P_i\boldsymbol{C_i}$. Here we use the term 'prefix' to mean any initial sequence of a linear order. Since λ is itself a prefix, the desired results will follow. We will prove this claim by induction on *i*.

Induction hypothesis. $r \in P_iC_i[V]$ if and only if B^r contains a linear order with the prefix $P_i\boldsymbol{C_i}$.

Base case. $i = 1$.To establish the base case, we show that $r \in C_1[V]$ iff B^r contains a linear order C_1Q.

RL. Assume the linear order $\lambda = C_1Q$ is in B^r. Observe that if B^r is minimal in $\mathrm{GEPO}(C_1)$, then $C_1(r)$ is numerically minimal in the C_1-column of V, and therefore $r \in C_1[V]$. And so to prove the base case, it suffices to show that B^r is minimal in $\mathrm{GEPO}(C_1)$.

The linear order λ belongs to some grammar of T, which we will denote as Γ^λ. We have $B^r = \ldots \cup \Gamma^\lambda \cup \ldots$. Now, Γ^λ is minimal in $\mathrm{EPO}_T(C_1)$ by Lemma (181), because Γ^λ contains a leg, namely λ, beginning with C_1. From the Inheritance Lemma (187), we have that $B^r \in \mathrm{GEPO}_\pi(C_1)$ inherits from $\Gamma^\lambda \in \mathrm{EPO}_T(C_1)$ all relations except those between B^r's merged grammars. Therefore, B^r is minimal in $\mathrm{GEPO}_\pi(C_1)$ because Γ^λ is minimal in $\mathrm{EPO}_T(C_1)$.

We conclude that $r \in C_1[V]$ because B^r is minimal in $\mathrm{GEPO}(C_1)$ and V instantiates $\mathrm{GEPO}(C_1)$, establishing the base case right-to-left.

LR. We show the contrapositive: if B^r contains no linear order starting with C_1, then $r \notin C_1[V]$.

Assume that B^r has no leg in it beginning with C_1. $B^r = \Gamma_1 \cup \ldots \cup \Gamma_m$ is composed entirely of grammars lacking C_1-initial legs. Back in the $EPO_T(C_1)$, none of these grammars are in the root, by the Prefixal Exclusion Lemma (182). Let Γ be in the root of $EPO_T(C_1)$. Since Γ is in the root, $\Gamma <_{C1} \Gamma_k$, for $1 \leq k \leq m$. Thus, Γ stands in the relation $<_{C1}$ with respect to all of the constituent grammars of B^r. And for some block $B^{r'} \in \pi$, $\Gamma \subseteq B^{r'}$. By the Inheritance Lemma (187), $B^{r'} <_{C1} B^r$. This shows that $r \notin C_1[V]$, since $C_1(r') < C_1(r)$. This establishes the base case left-to-right, concluding this step of the proof .

Induction step. Now assume for every $k<i$ that $r \in P_k C_k[V]$ if and only if B^r contains a linear order with the prefix $P_k C_k$. We show that the inductive hypothesis holds for i.

RL. Suppose that B^r contains a linear order $\lambda = P_i XQ$, where $X = C_i$. We will show that $r \in P_i X[V]$. Of concern: the rows $s \in P_i[V]$, which are the surviving competitors of r after filtration by P_i. We need to show that $X(r) \leq X(s)$ for all $s \in P_i[V]$ in order to get $r \in P_i X[V]$.

Consider the blocks B^r and B^s for some arbitrary $s \in P_i[V]$. By the induction hypothesis, each B^s contains a linear order starting with P_i.

Returning our attention to T, we note that B^s includes a grammar $\Gamma^s \in T$ which contains a linear order starting with P_i. It must be that $\Gamma^s \in fEPO(X)|P_i$. Now $fEPO(X)|P_i$ is rooted by Lemma (184), and since $\lambda = P_i XQ \in B^r$, the linear order λ has a corresponding grammar Γ^λ that is in the root in $fEPO(X)|P_i$. But this means that $\Gamma^\lambda <_X \Gamma^s$ or $\Gamma^\lambda \sim_X \Gamma^s$ in $EPO_T(X)$. By inheritance, $B^r <_X B^s$ or $B^r \sim_X B^s$ in $GEPO_\pi(X)$. Now, V instantiates GEPO(X) so $X(r) \leq X(s)$ for $s \in P[V]$. This establishes $r \in PX[V]$, the right-to-left assertion of the induction step.

LR. We want to show that if $r \in P_i X[V]$ then B^r contains a linear order with the prefix $P_i X$.

Suppose for purposes of contradiction that B^r does not contain a linear order with prefix $P_i X$. Of concern: the rows $s \in P_i[V]$. We know by the induction hypothesis that B^r, along with each B^s, contains a linear order with the prefix P_i. This entails that back in T there is a grammar $\Gamma^r \subseteq B^r$ containing a total order with prefix P_i. Similarly, each B^s includes a grammar Γ^s with a total order prefixed by P_i. Each of these grammars are in $fEPO(X)|P_i$. By the Prefixal Exclusion Lemma (182), because Γ^r does not have a total order starting with $P_i X$ it is not in the root of $fEPO(X)|P_i$. This means there is some $\Gamma^{s'}$ in $fEPO(X)|P_i$ with $\Gamma^{s'} <_X \Gamma^r$. This relation is inherited by B^r and $B^{s'}$. Therefore, $X(s') < X(r)$. But this means that $r \notin P_i X[V]$, a contradiction. This establishes that if $r \in P_i X[V]$, then B^r contains a linear order with the prefix $P_i X$, establishing induction step left-to-right. □

(190) **Corollary.** A partition π of Ord(S) is a typology iff GMOAT(π) is acyclic.

Proof.

LR. If π is a typology, its GMOAT is a MOAT, and every MOAT is acyclic, because the hypertransitive closure of $<_X$ with respect to $\sim_X$ is a partial order, as noted in §3.2, above ex. (150), p. 99.

RL. Any partition π of Ord(S) arises from mergers on the Discrete Typology, which has a MOAT. If the GEPOs of GMOAT(π) are acyclic, it follows from Theorem (189) that they form a MOAT. □

To conclude the discussion, let's put partitions behind us and consider the prospects of deriving a typology from a set of general bigraphs. Suppose we have a MOAT-like structure MLS on a set of objects S, which consists of a set of EPO-like bigraphs ELB_i, $\langle S, <_i, \sim_i\rangle$, $i\in\mathbb{N}$, where the only restriction on the bigraphs is that they are acyclic and prefix-rooted, in the following sense.

Definition. Prefix Rootedness. A set of ELB's is prefix-rooted iff, given any order P on a finite subset of indices $A=\{a_1,\ldots a_n\}$, $fELB_k|P$ is rooted as per definition (180).

Definition. fELB. We recursively define the filtered ELB, denoted fELB.

(1) $fELB_k|q$ is the sub-bigraph of ELB_k equal to $\langle K, <_k, \sim_k\rangle$, where $x\in K\subseteq S$ is in the root of ELB_q.

(2) $fELB_k|Pz$, for P a sequence of indices, is the sub-bigraph of ELB_k equal to $\langle K, <_z, \sim_z\rangle$, where every $x\in K\subseteq S$ is in the root of $fELB_k|P$.

If we have an MLS on a finite set S, then it has an OT instantiation in the sense that it has set of VTs associated with it, each of which produces the same typology. The relations in the MOAT of this typology are a subset of the relations in the MLS, given the natural correspondence between ELBs and EPOs.

Suppose we interpret a set of bigraphs as a set of preferences relating the objects of its reference set S. Then if the bigraphs form an MLS, we are guaranteed that the preference system can be instantiated as an OT typology. If the preference system does not form an MLS, then it cannot be so instantiated. This gives us a general criterion for OT analyzability.

3.4 Grammar at the edge: the ERC and the ERCoid

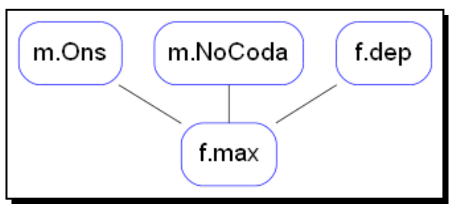
The border point pairs of typology determine its MOAT, and therefore all of its grammars through any instantiating UVT. Each individual grammar has its own border, and we explore here how the legs in its border determine the entire contents of the grammar. We will find that border points participate in an extended version of ERC representation, whose basic components we will call 'ERCoids', which in addition to the familiar {W,L,*e*} deploy a fourth value 'u' indicating lack of information. We will show how the notion of fusion can be extended to ERCoids by taking u to be the identity. Under this assumption, the Fusional Reduction Algorithm (FRed: Brasovenau & Prince 2005/11) can operate on border point information, yielding ERC grammars from sets of ERCoids. Proof of general efficacy is not yet complete, but the approach is sufficiently promising to warrant discussion here, given its current level of success.

3.4.1 Grammars from Border Points

Returning to our touchstone example, we consider two grammars from the EST, which indicate how ERCoids emerge and combine.

First we consider the grammar of 1:CV.del.

(191) 1:CV.del

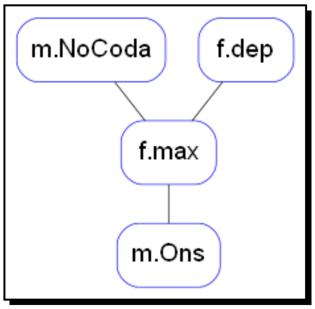
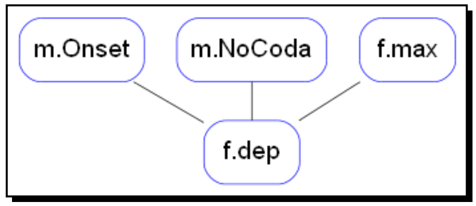
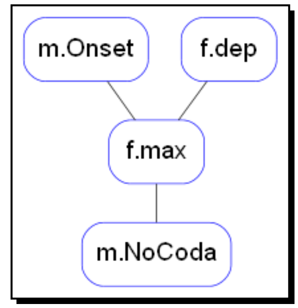


(192) ERC grammar of 1:CV.del in MIB form

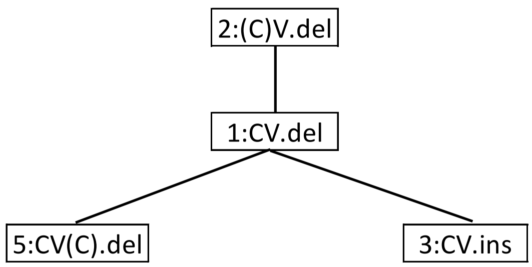
1:m.Ons	2:m.NoCoda	3:f.dep	4:f.max
W			**L**
	W		**L**
		W	**L**

The grammar 1:CV.del is adjacent to three other grammars: 2:(C)V.del, 3:CV.ins, 5:CV(C).del. For convenience, we give them here as Hasse diagrams.

(193) Hasse Diagrams of grammars adjacent to 1:CV.del

2:(C)V.del	3:CV.ins	5:CV(C).del
		

Anticipating the full geometrical development of §4, we can portray the adjacencies of 1:CV.del in a simple diagram:



Each adjacent grammar shares two border point pairs with 1:CV.del. Since the members of a pair differ only in ordering of the two constraints in the prefix, we show one for each pair.

(194) Border point pairs of 1:CV.del

Lgs	**Border point pairs**	**Constraints transposed**
1:CV.del	m.NoCoda ≫ f.dep ≫ ***m.Ons*** ≫ **f.max**	m.Ons, f.max
2:(C)V.del	m.NoCoda ≫ f.dep ≫ **f.max** ≫ ***m.Ons***	
1:CV.del	m.Ons ≫ m.NoCoda ≫ ***f.dep*** ≫ **f.max**	f.dep, f.max
3:CV.ins	m.Ons ≫ m.NoCoda ≫ **f.max** ≫ ***f.dep***	
1:CV.del	f.dep ≫ m.Ons ≫ ***m.NoCoda*** ≫ **f.max**	m.NoCoda, f.max
5:CV(C).del	f.dep ≫ m.Ons ≫ **f.max** ≫ ***m.NoCoda***	

From the transposition in the pair (**1**, **2**), we learn that $\mathbf{1}<_{\text{m.Ons}}\mathbf{2}$ and $\mathbf{2}<_{\text{f.max}}\mathbf{1}$**.** From the prefix we lean that $\mathbf{1}\sim_{\text{m.NoCoda}}\mathbf{2}$ and $\mathbf{1}\sim_{\text{f.dep}}\mathbf{2}$. These facts translate directly into an ERC formula which expresses exactly these relations:

(195) BP-derived ERC: **1**∼**2**

	m.Ons	m.NoCoda	f.dep	f.max
1:CV.del∼**2**:(C)V.del	W	*e*	e	L

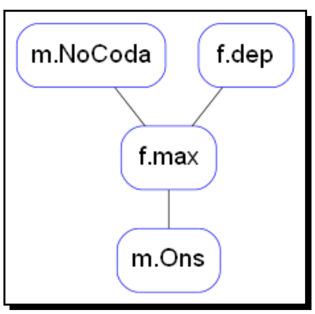
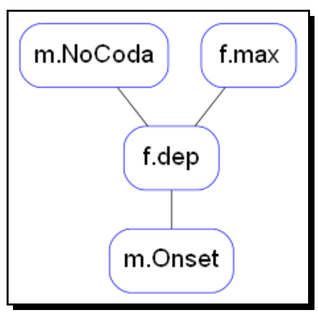
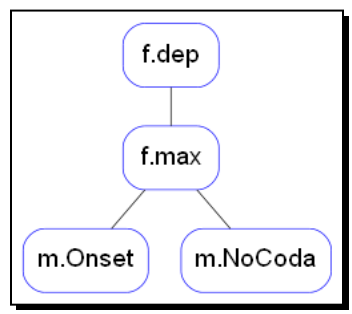
Continuing in this vein, we derive two more ERCs from the two other border pairs in (194). Putting everything together, we get the full grammar of 1:CV.del, which is exactly as in ex. (192)

(196) ERC table for 1:CV.del

	m.Ons	m.NoCoda	f.dep	f.max
1 ∼ 2	W	*e*	*e*	L
1 ∼ 3	*e*	*e*	W	L
1 ∼ 5	*e*	W	*e*	L

In the case, border point analysis yields the grammar directly. A subtlety arises in cases like that of 2:(C)V.del, in which the results of border point analysis require further interpretation.

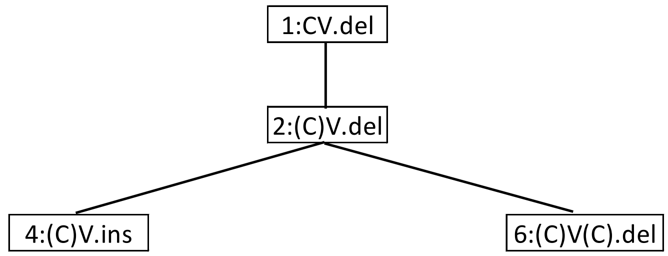
The grammar 2:(C)V.del is also adjacent to 3 other grammars: 1:CV.del, 4:(C)V.ins, 6:(C)V(C).del. For convenience, we give them here as Hasse diagrams.

1:CV.del	**4**:(C)V.ins	**6**:(C)V(C).del
		

(197) ERC Grammar of 2:(C)V.del in MIB form

	m.Ons	m.NoCoda	f.dep	f.max
ERC#1	L	W		L
ERC#2	L		W	L
ERC#3	L			W

We portray the adjacencies of 2:(C)V.del diagrammatically:



Grammars **2** and **1** meet at two border pairs, as noted above in the discussion of **1**; we show one of them. Grammars **4** and **6** share one border pair each with grammar **2**.

(198) Border points pairs of grammar 2:(C)V.del

Lgs	Border point pairs	Constraints transposed
2:(C)V.del	m.NoCoda ≫ f.dep ≫ ***f.max*** ≫ **m.Ons**	f.max, m.Ons
1:CV.del	m.NoCoda ≫ f.dep ≫ **m.Ons** ≫ ***f.max***	
2:(C)V.del	m.NoCoda ≫ ***f.dep*** ≫ **f.max** ≫ m.Ons	f.dep, f.max
4:(C)V.ins	m.NoCoda ≫ **f.max** ≫ ***f.dep*** ≫ m.Ons	
2:(C)V.del	f.dep ≫ ***m.NoCoda*** ≫ **f.max** ≫ m.Ons	m.NoCoda, f.max
6:(C)V(C).del	f.dep ≫ **f.max** ≫ ***m.NoCoda*** ≫ m.Ons	

From the pair involving 2 and 1, we derive the ERC [2~1], the negation of [1~2].

(199) BP Derived ERC

	m.Ons	m.NoCoda	f.dep	f.max
2:(C)V.del∼**1**:CV.del	L	*e*	e	W

But the pair (**2**,**4**) yields only partial information: in particular, it yields no information about the relative values on m.Ons.

● The prefix consists solely of m.NoCoda, establishing $\mathbf{2}\sim_{\mathrm{m.NoCoda}}\mathbf{4}$ and placing an *e* in the m.NoCoda column of [**2**~**4**].

● The transposition yields $\mathbf{2}<_{\mathrm{f.dep}}\mathbf{4}$ and $\mathbf{4}<_{\mathrm{f.max}}\mathbf{2}$, so that a W is placed in f.dep and and L in f.max.

● m.Ons resides in the suffix, from which nothing can be deduced about the relation between **2** and **4** on m.Ons.

In the case of 1:CV.del , all relations between it and its bordering grammars could be determined from border points. This permitted us to construct ERCs between 1:CV.del and each of its bordering grammars. Here, the relation between **2** and **4** on m.Ons is not determined at the border between **2** and **4.** To represent this states of affairs we expand the vocabulary of comparison to include a fourth value 'u' to denote the case where the value is undetermined. A vector over the expanded vocabulary will be called an 'ERCoid'. Using this construction, we write the results of border point analysis of (**2**,**4**) as follows:

(200) ERCoid derived from border points of **2** and **4**.

	m.Ons	m.NoCoda	f.dep	f.max
2:(C)V.del∼**4**:(C)V.ins	u	*e*	W	L

The same mode of analysis applied to **2** and **6** produces the following ERCoid:

(201) ERCoid derived from border points of **2** and **6**.

	m.Ons	m.NoCoda	f.dep	f.max
2 ∼ 6	u	W	*e*	L

The entirety of the information that can be deduced from border point relations of 2 with its neighbors is included in this table.

(202) ERCoid table for 2:(C)V.del

	m.Ons	m.NoCoda	f.dep	f.max
2 ∼ 1	L	*e*	*e*	W
2 ∼ 4	u	*e*	W	L
2 ∼ 6	u	W	*e*	L

The value u associated with m.Ons is locally indeterminate, but context determines what values m.Ons may assume. Here, because of the L in m.Ons in [**2**∼**1**], the two u's may be realized as *any* value from {W,L,*e*}. Whatever the specification is, logical combination with [**2**∼**1**] will always yield L.

It is not the case, however, that u can simply be replaced *salve veritate* by anything anywhere. It is specifically the L in the m.Ons column of (202) that liberates its possibilities. To see this, consider a 2-grammar typology on the 3 constraints {X, Y, Z}, where the first grammar Γ_1 is defined by the single ERC WWL, and the second Γ_2 by the ERC LLW. Γ_1 meets Γ_2 in two border point pairs, (a) and (b).

(203) Border Points Pairs

Lgs	**Border point pairs**	**Constraints transposed**

Γ_1(a)	***X***≫**Z**≫ Y	X,Z
Γ_2(a)	**Z**≫ ***X***≫ Y	

Γ_1(b)	***Y***≫**Z**≫ X	Y,Z
Γ_2(b)	**Z**≫ ***Y***≫ X	

These give rise to the following two ERCoids. Note that Y is in the suffix of border points (a) and X is in the suffix of border points (b).

(204) BP Derived ERC

	X	Y	Z
Γ_1(a)~ Γ_2(a)	W	u	L
Γ_1(b)~ Γ_2(b)	u	W	L

These must combine to give the ERC WWL, which characterizes Γ_1. Thus u cannot freely take on any value in this example. If either or both of the u's were L, the resulting grammar would contain an extra L in either X or Y (or both), and consequently fail to be a grammar of Γ_1.

If we want to derive grammars from border point information, we will have to combine ERCoids in way that parallels ERC combination in the FRed algorithm (Brasoveanu & Prince 2005/2011). To do so, we need to take account of the behavior of u with respect to fusion. Recall that the ERC-logic operation works like this (Prince 2002):

(205) Fusion of ERC-logic values, V∈{W,L,*e*}

a. L∘V = L — 'L is dominant'

b. V∘V = V — 'fusion is idempotent (like *and* and *or*)'

c. *e* ∘V = V — '*e* is the identity'

Fusion is symmetric, so that the order of fusands doesn't matter.

To bring u into the fold, we observe that u indicates an absolute lack of information. Consequently, u will function as an identity with respect to all the standard ERC values, which carry information. This displaces *e* as the identity, since *e*∘u = *e*.

(206) Fusion of ERCoid values

a. L∘V = L	'L is dominant'
b. V∘V = V	'fusion is idempotent'
c. e∘V = V for V∈{e, W, L}	'e is an identity for ERC values'
d. u∘V = V	'u is the identity for all values'

If we think of fusion as delivering the maximal element of the fusing pair, as determined by a scale of values, then standard ERC fusion operates on the scale *e*<W<L. In the ERCoid extension of this logic, we merely append u to the bottom, so that the augmented fusion scale is u<*e*<W<L. This allows for the concise definition x∘y = y iff x≤y.

We hypothesize that with these extensions, FRed runs in its familiar way. For this to work, it must be the case the fusions produced by FRed never contain u, with the u's are absorbed into some ERC value {W,L,*e*}, a phenomenon we will refer to as 'ERCoid confinement'. In the end, nothing compares to 'u'. This, of course, will require proof, yet to be obtained.

If this is right, the definition of FRed persists unchanged. In particular, an 'Info Loss Configuration' in FRed is defined to be a column fusing to W which contains both e and W. This definition is retained without modification. This means that a column containing u and W without *e* is *not* an Info Loss Configuration, just as is shown in ex. (204). Qualitatively speaking, fusion of u and W does not lose information, since there is no information in u to lose.

To see how this works, let's apply FRed to the complete set of BP-derived ERCoids associated with the boundary of 2:(C)V.del.

FRed

Step 1: Fuse all.

Result: LWWL. Status of fusion: entailed by ILRs {abc}, {abd}.

ERC		m.Ons	m.NoCoda	f.dep	f.max
a	[**2** ~ **1**](a)	L	*e*	*e*	W
b	[**2** ~ **1**](b)	L	*e*	*e*	W
c	**2** ~ **4**	u	*e*	W	L
d	**2** ~ **6**	u	W	*e*	L
a∘b∘c∘d		**L**	**W**	**W**	**L**

We have two Info Loss Residues: ILR_1={a,b,c} and ILR_2={a,b,d}. Since these collectively include every ERCoid {a,b,c,d}, their fusion ILR_1∘ILR_2 entails the fusional result of the first step, by virtue of being identical to it. We therefore dismiss it and proceed to examine each ILR separately.

Step 2a. ILR {a,b,c}:

ERC		m.Ons	m.NoCoda	f.dep	f.max
a	[**2** ~ **1**](a)	L	*e*	*e*	W
b	[**2** ~ **1**](b)	L	*e*	*e*	W
c	**2** ~ **4**	u	*e*	W	L
a◦b◦c		**L**	***e***	**W**	**L**

The ILR here is {a,b}, which consists of identical ERCs. Thus a◦b = *a* = *b* = L*ee*W, which doesn't entail a◦b◦c = L*e*WL. We therefore enter a◦b◦c into the MIB.

Step 2b. ILR {a,b,d}

ERC#		m.Ons	m.NoCoda	f.dep	f.max
a	[2 ~ 1](a)	L	*e*	*e*	W
b	[2 ~ 1](b)	L	*e*	*e*	W
d	2 ~ 6	u	W	*e*	L
a◦b◦d		**L**	**W**	***e***	**L**

The ILR here is also {a,b}, which does not entail the fusion LW*e*L. Therefore we also enter a◦b◦d = LW*e*L into the MIB.

Step 3. ILR {a,b}. This is the ILR from both preceding steps, and as noted, is just LeeW, which has no ILR. We therefore enter this into the MIB and we're done. The MIB discovered is exactly that of (197), repeated below, which can be calculated from any UVT.

(207) ERC Grammar of 2:(C)V.del in MIB form

	m.Ons	m.NoCoda	f.dep	f.max
a◦b◦d	L	W		L
a◦b◦c	L		W	L
a◦b	L			W

This analysis has shown that to apply FRed to ERCoids, we must assume u◦L=L. To show that u◦W=W, let's examine the grammar Γ_1 = WWL on {X, Y, Z} from ex. (204) above. It has two border point pairs: a = (*XY*Z, *YX*Z), and b = (*YZ*X, *ZY*X). Their associated ERCoids and its fusion are given below. This gives us the first step of FRed.

(208) BP Derived ERCoids

	X	Y	Z
Γ_1(a)~ Γ_2(a)	W	u	L
Γ_1(b)~ Γ_3(b)	u	W	L
fu(all)	**W**	**W**	**L**

There are no Info Loss Configurations. We are done and the MIB is WWL, as promised.

A final simple example shows that it is necessary to set u∘*e* = *e*, as claimed. Consider the grammar H = WL*e* on {X,Y,Z}. Its two border point pairs are a = (*XY*Z, *YX*Z) and b = (Z*XY*, Z*YX*). Their associated ERCoids and its fusion are given below.

(209) BP Derived ERCoids

	X	Y	Z
H(a) ~ H′(a)	W	L	u
H(b) ~ H″(b)	W	L	*e*
fu(all)	W	L	*e*

Here too there are no Info Loss Configurations. To achieve the grammar WL*e*, we must fuse u and *e* to *e*.

3.4.2 ERCoid confinement

It is natural to hope that u can be incorporated into a 4-valued logical system, some kind of extension of ERC logic, but it is by no means clear that this is the case. ERCoids form a kind of pre-logic: they combine to give the elements of ERC logic but do not themselves support the basic operations and relations expected in a logic.

Suppose we begin with the observation that ‘u’ represents lack of information. Then combining two ERCoids in which one has a strict superset of the information in the other ought to be completely harmless. For example, WL*e*u should combine with WLeL to yield WL*e*L. Similarly, WL*e*u should combine with WL*e*W to yield WL*e*W. In these cases, the ERCs WL*e*L and WL*e*W both have strictly more information than WL*e*u.

Such considerations suggest a notion of ‘weak composition’ that applies to pairs of ERCoids which are identical except in components where one has u.

(210) **Def**. Weak composition. The *weak composition* ERCoid κ = α*β of two ERCoids α, β is defined componentwise as follows:

a. κ[*i*] = α[*i*] when α[*i*] = β[*i*]
b. κ[*i*] = α[*i*] when β[*i*] = u
c. κ[*i*] = β[*i*] when α[*i*] = u
d. When for some *i*, α[*i*] ≠ β[*i*] and neither is equal to u, weak composition of α and β is not defined.

Since u means “no information,” it might seem that weak composition is info preserving. However, this is not true in context, in the following sense:

For {α, β, γ}, {α*β, γ} is not necessarily the logical equivalent of {α, β*γ} or indeed {α, β, γ} itself.

In short, how the ERCoids are *packaged* (weakly composed) can affect the interpretation of an ERCoid set.

Consider the following ERCoid set:

(211) ERCoid set with non-equivalent weak combinations

a	W	L	u	L
b	W	L	*e*	u
c	W	L	W	u
d	W	L	*e*	W

All are of the form WLxy, with u showing up in position x or y . The issue is the relation between the 1st and 4th components. If we weakly compose a and b, we have a*b=WL*e*L, which requires 1≫4. But weak composition of a and c as a*c = WLWL reduces the relation between 1 and 4 to a mere disjunct.

Thus, the order of weak composition matters, defying naïve intuition. This is apparent from considering various weakly composed versions of {a,b,c,d}.

{a*b, c, d} = {WL*e*L, WLWu WL*e*W } = {WL*e*L, WLWu} = {WL*e*L}

{ a*c, b*d} = { WLWL, WL*e*W}

In the end**,** this is perhaps not too shocking. The glory of the u-containing ERCoid is that it can compose (weakly) to give either conjunction or disjunction, which does not suggest that weak combination is harmless. In particular, this property undermines the hopes of defining entailment on ERCoids.

A set of ERCoids, then, does not have anything like the status of a set of ERCs. Any consistent ERC set defines a grammar. But not so any set of ERCoids. The viability of an ERCoid set is tied to its origin as a representation of a set of border point pairs; therefore, it is closely tied to particularities of the structure of the border points of a grammar. Thus, we do not know, right off, whether the kind of ambiguous situation constructed above actually arises ecologically.

It may be relevant that the ERCoid set (211) cannot arise from the border point pairs of two grammars. This is because all the (possibly many) ERCoids that arise along a shared boundary weakly must weakly compose to a single unique ERCoid, which we will call a Unitary Border Ercoid (UBE). When one of the border ERCoids has W,L,*e* in a coordinate, any other is either identical, or has u in that coordinate. There is no distinction between the specified information they carry; it is only the lack of information in one allows another to provide further information.

(212) **Lemma**. Unitary Border Ercoid. Let Γ_1 and Γ_2 share a set of *n* border points pairs $\Gamma_1|\Gamma_2 = \{p_1,\ldots,p_n\}$. Let $\mathcal{E} = \{\alpha_1, \ldots,\alpha_n\}$ be the set of ERCoids derived from the border point pair in $\Gamma_1|\Gamma_2$. Then, if $\alpha_i[j] = x \in\{W, e, L\}$, then for every k, $1\leq k\leq n$, $\alpha_k[j] = x$ or $\alpha_k[j] = u$.

Proof. Suppose $\alpha_i[j] = x \in\{ W, e, L\}$. Consider the EPO of constraint *j*. The border point pair p_i giving rise to ERCoid α_i imposes a relation between Γ_1 and Γ_2 that appears in EPO(*j*) and is determined by the value *x*. If $x = W$, then $\Gamma_1 <_j \Gamma_2$. If $\underline{x} = L$, then $\Gamma_2 <_j \Gamma_1$. If $x = e$, then $\Gamma_1 \sim_j \Gamma_2$. No other ERCoid in $\mathcal{E}$ can contradict this relation.

□

If FRed is to succeed on ERCoids without modification, as is it used above, it must handle the u value properly. It cannot stumble on weak combination effects; it must work correctly on the full assembly of border point ERCoids, without regard to their provenance. The lemma shows that we can without prejudice represent each border between two grammars by a single ERCoid. As yet, it is not clear whether it is necessary to pre-package the border point ERCoids in this way.

Even more fundamentally, the fusions produced by FRed must all free of u. Were this to be false, the algorithm simply does not run, because of the Entailment Check Step, in which the fusion is tested for entailment against the collection of Info Loss Residues. General ERCoids do not support a notion of entailment. And, of course, the result of FRed must be an ERC set: a grammar.

We want the output of integrating ERCoids over the boundary of the target grammar Γ via FRed to be a pure ERC set, identical to the unique MIB of Γ. But if we have a column that consists entirely of u's, we can only get u out of it. To resolve this matter, we'd have to show that no column containing only u shows up at any step of FRed. We will not complete this task here, but we will take the first step, showing that in the entire ensemble of border points of Γ, it cannot be the case that a constraint evaluates everywhere to u.

Observe that if a constraint C has u in every ERCoid associated with Γ's boundary, then u must be in the suffix of every border point pair. This cannot happen.

(213) **Theorem**. No constraint can be entirely suffixal in the border points of a grammar.

Proof. Let $\mathcal{E} = \{\alpha_1, \ldots,\alpha_n\}$ be the collection of ERCoids derived from the border points of a grammar Γ. Assume for a constraint C that for every k, $1\leq k\leq n$, $\alpha_k[C] = u$. Now consider EPO(C). The grammar Γ has no relations of equivalence or order with any other grammars. Thus EPO(C) is unrooted, an impossibility by Corollary (185) to the Rootedness Lemma (184) . □

This result can also be approached through consideration of the leg content of a grammar, yielding an alternate proof of Theorem (213) that allows us to generalize from suffixes to prefixes.

(214) **Theorem**. A. No constraint can be entirely suffixal in the border points of a grammar. B. No constraint can be entirely prefixal in the border points of a grammar.

Proof of A. Let **X** be a constraint that is entirely suffixal in the border points of Γ. First, we establish that X is freely positioned in the legs of Γ. For any point p = …BX…∈Γ, consider p′ = …XB…, where the material is '…' is sequentially identical in p and p′. It must be that p′∈ Γ, else p is a border point and (p, p′) a border point pair in which X is involved in the transposition, contrary to the assumption that it is suffixal in every border point of Γ. The same holds going rightward for p = …XD…∈Γ: it must be that p′ = …DX…∈Γ, else (p, p′) is a border point pair in which X is non-suffixal. We may iterate this argument so that for any leg λ∈G, X can be displaced via a sequence of adjacent transpositions to any position, respecting the order of the other constraints. The resulting leg must still belong to Γ.

Since X appears freely, it must be that some legs of Γ must begin with X. We show that every leg beginning with X must belong to Γ. Suppose to the contrary that not every leg beginning with X is included in G. Consider any leg p_1∈Γ that begins with X and some leg p_n∉Γ that begins with X. From the algebra of permutations, we know that there is a trail of legs $p_1,\ldots,p_n$ where p_j and p_{j+1} differ only by a single adjacent transposition, and where every p_j, $1\leq j\leq n$ begins with X. Since p_1∈G and p_n∉G, it follows that at least one sequential pair of points is a border point pair. But for this pair, X is in the prefix, contradicting the assumption that X is never prefixal. Therefore every point beginning with X belongs to G.

This gives us as members of G all legs of the form X followed by every order on the remaining constraints. But we know from the first argument that X is freely positioned in the legs of Γ Therefore, for each order of the remaining constraints, X may be positioned anywhere among them. This gives us every ranking of the contraints. Therefore Γ is the trivial grammar that includes every possible ranking and has no border points at all. This gives us an alternate proof of claim A.

Proof of B.The same form of argument shows that we cannot have a constraint that is always *prefixal* in every border point of a grammar. Suppose, for purposes of contradiction, that X is always prefixal in every border point of a grammar Γ. Consider any p = …BX… . ∈Γ. As above, it must be that p′ = …XB…∈Γ, else (p,p′) is a border point in which X is non-prefixal. The same holds for p =…XD…, entailing that p′=…DX…∈Γ. As above, we may iterate the argument to show that X occurs freely among the legs of Γ.

Now consider any leg $p_1 = \ldots X \in \Gamma$. We show that all such legs ending with X must be in Γ. If not, there is some leg p_n with X as its bottom-most constraint, with $p_n \notin \Gamma$. Exactly as above, there is a trail of adjacent legs running between p_1 and p_n, with X at the bottom of each leg in the trail. But one of these must be a border point for Γ, in which X is suffixal, not prefixal. Therefore no such p*n* can exist and all legs ending in X belong to Γ. But any leg at all may be derived from one of these by transposing X upward. So all legs must belong to Γ. Γ is the trivial grammar that includes every possible ranking and has no border points at all. □

It follows then, that the distribution of border points is such that, for each grammar a constraint X either participates in a border transposition, or if not, appears both prefixally and suffixally, generating both *e* and u values in the border point ERCoids. This is a crucial part of the workings of OT that hides, as it were, the fourth possible value of ordinal comparison ('noncomparable'), even though it plays an inner role in the evaluation of ranking information directly from border point pairs.

3.4.3 ERCoids from the MOAT

The extension of ERCs to ERCoids allows us to represent a grammar's position in each EPO with an ERCoid. Given any two grammars Γ_1, Γ_2 in a typology, the ERCoid $[\Gamma_1\sim \Gamma_2]$ has coordinate values W, L,*e*, u which express the following relations in the C^{th} coordinate:

$[\Gamma_1\sim \Gamma_2][C] = W$	$\Gamma_1 <_C \Gamma_2$	
$[\Gamma_1\sim \Gamma_2][C] = L$	$\Gamma_2 <_C \Gamma_1$	
$[\Gamma_1\sim \Gamma_2][C] = e$	$\Gamma_1 \sim_C \Gamma_2$	
$[\Gamma_1\sim \Gamma_2][C] = u$	$\Gamma_1 \parallel_C \Gamma_2$	'Γ_1 is noncomparable to Γ_2'

The u value arises when Γ_1 and Γ_2 do not stand in either of the two relations $<_C$ or $\sim_C$, as is notated by $\parallel_C$. If we compare each grammar with every other in this way, we arrive at an unambiguous algebraic representation of EPO(C) which consists entirely of ERCoids.

What is the relationship of these MOAT ERCoids to the Unitary Border ERCoids (UBEs) of the grammars? Recall that the UBE of Γ with respect to Γ′ is the weak composition of all the border point ERCoids arising from the shared border points of Γ and Γ′. We can deduce the answer by examining the way the information in the MOAT differs from that in the border points.

Only adjacent grammars have Border Point ERCoids . The MOAT, by contrast, provides ERCoids relating every pair of grammars. We might wish to disregard MOAT ERCoids between nonadjacent grammars by fiat. Nevertheless, those that remain may still differ from Border Point ERCoids in the relations they represent.

The base relations $<^b_C$ and $\sim^b_C$ are transitively closed to produce the EPO relation $<_C$ and $\sim_C$. An EPO relation may therefore be a proper superset of the base relation from which it is derived. In ERCoid terms, this means that the some u's in a UBE $[\Gamma_1 \sim \Gamma_2]_{BP}$ can be matched to W's, L's, or e's in the MOAT ERCoid $[\Gamma_1 \sim \Gamma_2]_M$. In terms of weak composition, we may write $[\Gamma_1 \sim \Gamma_2]_{BP} * [\Gamma_1 \sim \Gamma_2]_M = [\Gamma_1 \sim \Gamma_2]_M$.

EPO diagrams show $<^b_C$ directly without its transitive closure, in the interests of representational clarity. This allows us, if we wish, to entertain an alternate definition of the MOAT ERCoid in which W and L are restricted to base relations, and therefore exactly represent the unitary Border Point ERCoids in these values.

But the relation $\sim_C$ is represented in EPO diagrams without regard to adjacency or transitive closure, and therefore in this case the base relations cannot be extracted from an EPO diagram. Thus, a MOAT ERCoid based on what is represented in an EPO diagram may still have a strict superset of *e*'s in it.

3.5 Retrospect

A typology has many UVTs but just one MOAT. A MOAT contains the key information that delimits every VT representation of its typology and excludes all others. The primary thrust of this section has been to prove these assertions, which go well beyond anything that can be established through examples.

Our line of attack has been to connect the MOAT with another natural characterizing structure, the *i*OAT, which is defined in terms of the set of all UVTs yielding T. The MOAT by contrast has no definitional relation with a UVT. It derives directly from the grammars of the typology through border point analysis, where *grammars* are taken to be sets of linear orders on $\text{C}\small{\text{ON}}_T$. Despite their distinct origins, these two objects are shown to contain identical order and equivalence information. Since the *i*OAT characterizes, the MOAT must as well.

The MOAT consists of order and equivalence structures — EPOs — which submit to graphical representation. As with the familiar Hasse diagram, which lacks the equivalence component, the EPO bigraph is *acyclic*, admitting no directed path that revisits a node. Border point analysis, when applied to the blocks of any partition of a set of linear orders, produces a Generalized MOAT (GMOAT), which imposes a relational structure on those blocks. When the GMOAT is acyclic in the relevant sense, the partition is guaranteed to be a typology (§3.3). This result indicates that the MOAT not only distinguishes among typologies but also distinguishes typologies from mere partitions.

Border point analysis lends itself to representation in terms of the ERCoid, which contains a fourth value — *unknown* — and which may, we conjecture, submit to the same algorithms that yield canonical grammar representations (MIBs, SKBs) from ERCs, thereby further integrating the leg structure of grammars into the theory of OT.

4 Working out the MOAT: Jump to the CSys

SubTOC

SINCE TYPOLOGIES ARE COMPOSITE OBJECTS, it is often illuminating to study restricted subparts in isolation from the whole. For example, we can instructively distinguish in the EST between the treatment of underlying C and that of underlying V, since these function independently, as may be seen from the Universal Support (45), where different csets from different inputs determine their behaviors.

To put it more generally: in examining the typology T of a system S, it will often be valuable to study a coarser typology T′ that amalgamates some of the grammars of T into typological classes. Coarsening T to T′ means abolishing some distinctions that are respected in T. Two roads lead to MOAT(T′). We may be able to restrict GEN$_{S}$ concretely in such a way that the T′ emerges from a Universal Support of candidate sets. Or, more abstractly, we can merge nodes of the MOAT of the finer T in such way that the MOAT of coarser T′ is produced, with no concern about whether it is even possible to achieve the same result within the conception of linguistic structure embodied in the concrete system S underlying T. When both approaches are possible, the paths look like this:

(215) **To the MOAT of coarsened T**, concretely and abstractly

$$\begin{tikzcd} \mathrm{S{:}T} \arrow[r, "bpa"] \arrow[d, "mod\ \mathrm{GEN}_S"'] & \mathrm{MOAT(T)} \arrow[d, "merge\ nodes"] \\ \mathrm{S'{:}T'} \arrow[r, "bpa"'] & \mathrm{MOAT(T')} \end{tikzcd}$$

Here we will examine a coarsened version of the EST which recognizes only the distinctions in the treatment of consonants. We will call it the *C-System* of the EST, abbreviated to *CSys*, or more fully to *EST.CSys*. To obtain it, we effectively disable the constraint m.Ons by modifying GEN$_{S}$ so that m.Ons makes no distinctions, while retaining it in CON$_{CSys.}$ Because CON$_{CSys}$ = CON$_{EST,}$ we can still directly compare the two typologies because their grammars are built from the same constraints.

We modify GEN$_{EST}$ so as to disallow all inputs containing ‘problematic’ V, those that admit unfaithful optima. We retain CON$_{EST}$ in its entirety. Distinctions in the handling of onsets are thereby lost, because m.Ons is never violated in the admitted candidates. This tactic creates a system in which our view of the handling of consonants is unimpeded by irrelevancies related to vowels.

There are a couple of more abstract ways to construct the CSys, which we are worthy of mention because of their potential for further development. We could hold $\text{G}\textsc{en}_{\text{EST}}$ constant but on the constraint side, redefine m.Ons so that it assigns the same value to each candidate, say 0. The constraint $\text{m.Ons}_{\text{CSys}}$ is a different function on $\text{G}\textsc{en}_{\text{EST}}$ than $\text{m.Ons}_{\text{Csys}}$ and therefore formal steps would have to be taken to identify them, along the lines of how the language labels are identified across typologies in our analysis of the MOAT above (§§2-3). A second approach develops this idea with more algebra. In place of the concrete linguistic forms defined by $\text{G}\textsc{en}_{\text{EST}}$ as candidates, we could treat candidates as equivalence classes, where two candidates are equivalent if they are evaluated identically by all constraints *except* m.Ons. With this definition we replace a candidate set K with another K /m.Ons, where the distinction in m.Ons has been 'modded out'. This must be accompanied by a redefinition of $\text{C}\textsc{on}_{\text{EST}}$ as well, because the constraints now evaluate equivalence classes rather than simple candidates. Here's how it can be done. (a) On constraints in $\text{C}\textsc{on}_{\text{CSys}}$ other than $\text{m.Ons}_{\text{CSys}}$, all members of an equivalence class agree, so the value assigned by a non-m.Ons constraint to any member of the class can naturally be chosen as the value it assigns to the entire class. (b) Let the value assigned by m.OnsCSys be the same for every candidate. Observe that under this conception, the three inputs from /V/ that are possible optima in EST will each end up in its own equivalence class in CSys, because they all differ on faithfulness constraints. But all except the equivalence class containing the faithful candidate are harmonically bounded.

Setting aside whatever virtues these approaches may have, we will proceed by limiting modification to $\text{G}\textsc{en}_{\text{EST}}$. The CSys will therefore be defined in the in the usual way: articulating $\text{G}\textsc{en}_{\text{CSys}}$ and $\text{C}\textsc{on}_{\text{CSys}}$, calculating the typology from a Universal Support, and then determining the MOAT via border point analysis.

But modifying $\text{G}\textsc{en}_{\text{S}}$ or $\text{C}\textsc{on}_{\text{S}}$ or both is cumbersome. The most concrete approach in particular lacks generality and depends on the unusually simple way that the candidate sets of EST are structured. We therefore pursue a second route to the CSys MOAT, one which frees us entirely from the need for such manipulations. We start from the MOAT of EST and merge nodes within in it, focusing on those nodes whose corresponding languages vary only in the way they deal with onset issues. The mergers produce a new MOAT: one that is structurally identical to the MOAT of the concrete EST.CSys. Identical MOAT structure ensures exact grammatical correspondence, allowing us to investigate our system either in the specialized setting of Concrete OT or in the general setting of Abstract OT.

4.1 A Universal Support for EST.CSys

We're interested in the effect of unioning the pairs of grammars that differ only in the treatment of onsets. The languages pair up as follows. Recall the abbreviations OR 'Onset Required', OLA 'Onsetlessness Allowed', CP 'Coda Prohibited', CA 'Coda Allowed.' Here we depart from the rigors of §3, where it was important to use L_i^U and Γ_i to distinguish language from grammar, and commit the abuse of using the same name for both.

(216) Languages of EST

Name	**Outputs**	**IO disparities**	**Output Type**
1:CV.del	{ [CV] }*	deletion	**OR**, CP
2:(C)V.del	{ [(C)V] }*	deletion	**OLA**, CP
3:CV.ins	$\{\ [CV]\ \}^+$	insertion	**OR**, CP
4:(C)V.ins	$\{\ [(C)V]\ \}^+$	insertion	**OLA**, CP
5:CV(C).del	{ [CV(C)] }*	deletion	**OR**, CA
6:(C)V(C).del	{ [(C)V(C)] }*	deletion	**OLA**, CA
7:CV(C).ins	$\{\ [CV(C)]\ \}^+$	insertion	**OR**, CA
8:(C)V(C).ins	$\{\ [(C)V(C)]\ \}^+$	insertion	**OLA**, CA

The grammars we're looking for are then **1**∪**2**, **3**∪**4**, **5**∪**6**, **7**∪**8**. The extensional effect is to eliminate the distinction between syllables with and without onsets, while retaining all the other distinctions. The retained distinctions involve *problematic* C, those which do not sit before V in the input. At issue is whether to retain a problematic C when possible, as a coda; or, unfaithfully, to delete a problematic C or insert a vowel to support it.

In this particular case, we are fortunate in that a concrete version of the system can be obtained by omitting one candidate set from our universal support for EST, given in (45), which involves from just three inputs: /V/, /C/, /CVC/. The input /V/ is the only one in which m.Ons is violated in optima. Eliminating it ensures that m.Ons always gives the value 0 for every admitted optimum in the support. To achieve this omission, we set up $\text{G}\textsc{en}_{\text{Csys}}$ so as to not produce /V/ or anything like it in the relevant respect, while retaining the input /C/ and the input /CVC/. These latter two pose no problems for m.Ons, as their optima have onsets consisting of input C. This, then, gives us a universal support for the C-System of EST, shown below:

(217) EST.CSys Universal Support

input	output	m.Ons	m.NoCoda	f.dep	f.max	Type
C_1	ε	0	0	0	1	del
	[C_1**V**]	0	0	1	0	ins
C_1VC_2	[C_1VC_2]	0	1	0	0	F
	[C_1V_2]	0	0	0	1	del
	[C_1V] [C_2**V**]	0	0	1	0	ins

The abbreviations in the rightmost "Type" column specify the character of the IO map from the input to each output: as usual, we write *F* for faithful, *del* for deletion, *ins* for insertion. As before, we distinguish epenthetic V typographically.

To arrive at the CSys, we specify GEN_{CSys} and CON_{CSys}.

- $\text{CON}_{\text{CSys}} = \text{CON}_{\text{EST}}$.
- GEN_{CSys} must undergo a modification to exclude problematic V from inputs.

To meet the GEN requirement, we eliminate from the set of inputs produced by GEN_{CSys} all V not preceded by C. Other than that, the two specifications remain the same, as does every other aspect of admitted structure. The grammars of this new typology can be generated from just the two candidate sets of (217).

(218) GEN of CSys and of EST

GEN_{CSys}		**GEN_{EST}**	
IN	$= \{ C, CV\}^+$	IN	$= \{ C,V \}^+$
OUT	$= \{ [\,(C)V(C)] \}^*$	OUT	$= \{ [\,(C)V(C)] \}^*$

Our GEN_{CSys} disallows *inputs* beginning with V as well as those containing sequences of V. In order to maintain as close a relation to GEN_{EST} as possible, we set GEN_{CSys} to allow exactly the same outputs as GEN_{EST}. This means that deletional maps are allowed to produce onsetless syllables in the output.

To see why this doesn't undermine the enterprise of avoiding m.Ons violations in optima, consider these maps, where the only change between input and output is assumed to be the deletion indicated by *ε*.

a) $C_1V_2\ldots \quad \rightarrow \varepsilon[V_2\ldots$
b) $\ldots V_iC_jV_k\ldots \rightarrow \ldots Vi]\varepsilon[Vk\ldots$

We may assume without loss of generality that the only faithfulness breaches anywhere in the maps are deletional: in EST_{CSys}, because f.dep≫f.max or vice versa in any grammar, any candidates with both insertion and deletion are going to be harmonically bounded by candidates with one or the other. In both (a) and (b), there is deletion leading to violation of m.Ons. But both are simply bounded in EST.Csys (and indeed in EST) by the following more faithful maps. Here we assume that the sections denoted '…' in (a′) and (b′) are entirely identical in C,V-content, IO-correspondence structure, and syllabic affiliation to their sequentially corresponding sections in (a) and (b) respectively.

a′) $C_1V_2\ldots \rightarrow [C_1\ V_2\ldots$

b′) $\ldots V_iC_jV_k\ldots \rightarrow \ldots V_i][C_jV_k\ldots$

Violationwise, candidates (a) and (a′), (b) and (b′) are identical except for the cited deletions in (a) and (b). Thus the more faithful (a′) and (b′) fare exactly the same as their counterparts (a) and (b) with respect to m.NoCoda and f.dep, but do not incur the m.Ons violation or the f.max violation. This observation establishes that all maps deleting a single intervocalic or initial C in such a way as to incur an m.Ons violation are harmonically bounded, because for each of them there is a better map which doesn't delete that C. The fate under deletion of whole clusters of C in the same positions follows the same pattern. Such clusters can't be deleted down to nothing, since that would entail harmonic bounding by candidates in which deletion leaves behind a single C. See Prince & Smolensky 1993/2004, ch. 6, for this kind of reasoning, as well as Prince 2006 and Alber, DelBusso, and Prince 2016 for further use of it.

Our newly defined $\text{G}{\scriptsize\text{EN}}_{\text{CSys}}$ along with the assumption that $\text{C}{\scriptsize\text{ON}}_{\text{CSys}} = \text{C}{\scriptsize\text{ON}}_{\text{EST}}$ yields a factorial typology of four languages. The languages differ along two binary dimensions, each addressed in a separate cset.

(a) CA/CP: whether to allow codas (CA) or to prohibit them (CP).
- determined by /CVC/.

(b) del/ins: whether to *delete* to handle all problematic input C , or to *insert* a vowel to support their output correspondents, necessarily as an onset in optima.
- determined by /C/.

We use the following descriptors to identity the languages as CA.del, CA.ins, etc.

(219) **Extensional languages of EST.CSys**

Language	Optimal Outputs	Faithfulness breach
1∪**2**:CP.del	$\{\ [\text{CV}]\ \}^*$	del
3∪**4**:CP.ins	$\{\ [\text{CV}]\ \}^+$	ins
5∪**6**:CA.del	$\{\ [\text{CV(C)}]\ \}^*$	del
7∪**8**:CA.ins	$\{\ [\text{CV(C)}]\ \}^+$	ins

Because $\text{C}{\scriptsize\text{ON}}_{\text{EST}} = \text{C}{\scriptsize\text{ON}}_{\text{CSys}}$, direct comparison is possible between ranking grammars of EST and CSys. Each of the four ranking grammars of the CSys is literally the union of two ranking grammars from EST. This alignment is recorded in the naming of the four languages of CSys. The legs of CP.del, for example, combine the legs of **1**:CV.del, named OR.CP.del by our conventions, with those of **2**:(C)V.del, namable as OLA.CP.ins. See Appendix I for a classified list of all legs of the EST.

(220) **1**∪**2**:CP.del = **1**:OR.CP.del ∪ **2**:OLA.CP.del

CSys Lg.	EST Lg.	Leg#	Legs						
1∪**2**:CP.del	**1**	1	m.Ons	≫	m.NoCoda	≫	f.dep	≫	f.max
	1	2	m.Ons	≫	f.dep	≫	m.NoCoda	≫	f.max
	1	3	m.NoCoda	≫	m.Ons	≫	f.dep	≫	f.max
	1	4	m.NoCoda	≫	f.dep	≫	m.Ons	≫	f.max
	1	5	f.dep	≫	m.Ons	≫	m.NoCoda	≫	f.max
	1	6	f.dep	≫	m.NoCoda	≫	m.Ons	≫	f.max
	2	7	m.NoCoda	≫	f.dep	≫	f.max	≫	m.Ons
	2	8	f.dep	≫	m.NoCoda	≫	f.max	≫	m.Ons

Legs 1-6 are those of **1**:OR.CP.del, as can be seen from the fact that f.max sits in bottom position in the rankings. Legs 7 and 8 are those of **2**:OLA.CP.del from EST, with m.Ons in bottom position and both m.NoCoda and f.dep dominating f.max.

4.2 Unitary VT of CSys

Equipped with the Univeral Support for EST.CSys in tableau (217) we perform Minkowski summation over its two csets to get a unitary VT from which the MOAT may be calculated (Prince 2015). In this VT, CP.del is the constraint-wise sum of the violation profile of OR.CP.del with that of OLA.CP.del, and so on.

(221) UVT for EST.CSys

/C/⊕/CVC/

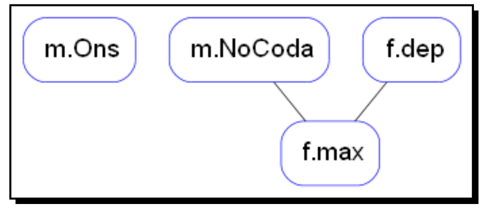
Cand. Lg.	**m.Ons**	**m.NoCoda**	**f.dep**	**f.max**
1∪**2**:CP.del	0	0	0	2
3∪**4** CP.ins	0	0	2	0
5∪**6**:CA.del	0	1	0	1
7∪**8**:CA.ins	0	1	1	0

Now we have left Concrete OT with its inputs, outputs, and associated paraphernalia. Once again in the realm of Abstract OT, we turn to construction of the MOAT.

Here is a list of all the grammars of the CSys, as computed from UVT (221):

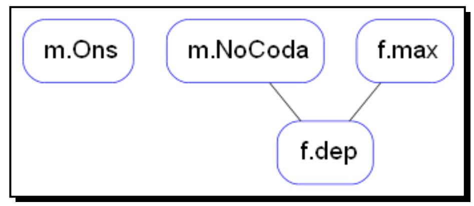
(222) **1**∪**2**:CP.del

m.Ons	m.NoCoda	f.dep	f.max
	W		**L**
		W	**L**



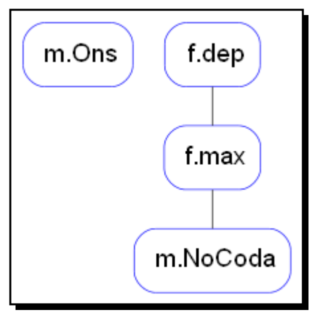
(223) **3**∪**4** CP.ins

m.Ons	m.NoCoda	f.dep	f.max
	W	**L**	
		L	**W**



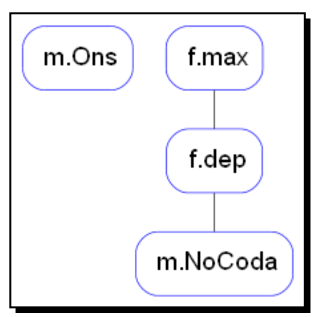
(224) **5**∪**6**:CA.del

m.Ons	m.NoCoda	f.dep	f.max
		W	**L**
	L		**W**



(225) **7**∪**8**:CA.ins

m.Ons	m.NoCoda	f.dep	f.max
		L	**W**
	L	**W**	



4.3 The CSys MOAT

The MOAT consists of a collection of EPOs, one for each constraint of the system, constructed from the border point pairs.

(226) All 12 Border Point Pairs of CSys

Grammars	Border Point Pairs	Constraints transposed
1∪**2**:CP.**del**	m.Ons ≫ m.NoCoda ≫ ***f.dep*** ≫ **f.max**	f.dep, f.max
3∪**4**:CP.**ins**	m.Ons ≫ m.NoCoda ≫ **f.max** ≫ ***f.dep***	
1∪**2**:CP.**del**	m.NoCoda ≫ m.Ons ≫ ***f.dep*** ≫ **f.max**	f.dep, f.max
3∪**4**:CP.**ins**	m.NoCoda ≫ m.Ons ≫ **f.max** ≫ ***f.dep***	
1∪**2**:CP.**del**	m.NoCoda ≫ ***f.dep*** ≫ **f.max** ≫ m.Ons	f.dep, f.max
3∪**4**:CP.**ins**	m.NoCoda ≫ **f.max** ≫ ***f.dep*** ≫ m.Ons	
3∪**4**:**CP**.ins	m.Ons ≫ f.max ≫ ***m.NoCoda*** ≫ **f.dep**	m.NoCoda, f.dep
7∪**8**:**CA**.ins	m.Ons ≫ f.max ≫ **f.dep** ≫ ***m.NoCoda***	
3∪**4**:**CP**.ins	f.max ≫ m.Ons ≫ ***m.NoCoda*** ≫ **f.dep**	m.NoCoda, f.dep
7∪**8**:**CA**.ins	f.max ≫ m.Ons ≫ **f.dep** ≫ ***m.NoCoda***	
3∪**4**:**CP**.ins	f.max ≫ ***m.NoCoda*** ≫ **f.dep** ≫ m.Ons	m.NoCoda, f.dep
7∪**8**:**CA**.ins	f.max ≫ **f.dep** ≫ ***m.NoCoda*** ≫ m.Ons	
7∪**8**:CA.**ins**	m.Ons ≫ ***f.max*** ≫ **f.dep** ≫ m.NoCoda	f.dep, f.max
5∪**6**:CA.**del**	m.Ons ≫ **f.dep** ≫ ***f.max*** ≫ m.NoCoda	
7∪**8**:CA.**ins**	***f.max*** ≫ **f.dep** ≫ m.Ons ≫ m.NoCoda	f.dep, f.max
5∪**6**:CA.**del**	**f.dep** ≫ ***f.max*** ≫ m.Ons ≫ m.NoCoda	
7∪**8**:CA.**ins**	***f.max*** ≫ **f.dep** ≫ m.NoCoda ≫ m.Ons	f.max, f.dep
5∪**6**:CA.**del**	**f.dep** ≫ ***f.max*** ≫ m.NoCoda ≫ m.Ons	
5∪**6**:**CA**.del	m.Ons ≫ f.dep ≫ ***f.max*** ≫ **m.NoCoda**	f.max, m.NoCoda
1∪**2**:**CP**.del	m.Ons ≫ f.dep ≫ **m.NoCoda** ≫ ***f.max***	
5∪**6**:**CA**.del	f.dep ≫ m.Ons ≫ ***f.max*** ≫ **m.NoCoda**	f.max, m.NoCoda
1∪**2**:**CP**.del	f.dep ≫ m.Ons ≫ **m.NoCoda** ≫ ***f.max***	
5∪**6**:**CA**.del	f.dep ≫ ***f.max*** ≫ **m.NoCoda** ≫ m.Ons	f.max, m.NoCoda
1∪**2**:**CP**.del	f.dep ≫ **m.NoCoda** ≫ ***f.max*** ≫ m.Ons	

We'll go through the system constraint by constraint, constructing each EPO, explicitly gathering relations between grammars induced by the border point pairs. Above we identify each border point by the grammar it comes from. Note that the cited leg is just one among several from that grammar.

Let us proceed to the EPOs.

1. m.Ons

No distinctions of order are made, as can be seen in the unitary VT (217), in which all m.Ons values are the same, by design. This means that m.Ons can go anywhere in a leg without changing the outcome. Consequently, for every adjacent pair of languages there is border point pair that has m.Ons somewhere in its prefix. Indeed, there will always be such a pair with m.Ons undominated, making for easy collection. Hence, every grammar will be equivalent on m.Ons, yielding the following EPO:

(227) **EPO$_{\text{CSys}}$(m.Ons)**

1∪2:CP.del ═ 3∪4:CP.ins ═ 5∪6:CA.del ═ 7∪8:CA.ins

2. m.NoCoda

The crucial information about order and equality need not come from a unique source. For m.NoCoda, several border points provide the same information, sufficient to produce its EPO. We present a minimal set of border point pairs, which yields all the privileged relations for m.NoCoda., namely:

CP.del $<_{\text{m.NoCoda}}$ CA.del

CP.ins $<_{\text{m.NoCoda}}$ CA.ins.

No other privileged order relations obtain on m.NoCoda. Each border point gives information about another constraint as well, which we include for completeness.

(228) **Privileged orders** in Border Point Pairs involving m.NoCoda

Grammars	Border Point Pairs	Privileged Relations
1∪**2**:**CP**.del	m.Ons ≫ f.dep ≫ ***m.NoCoda*** **≫ f.max**	**CP**.del $<_{\text{m.NoCoda}}$ **CA**.del
5∪**6**:**CA**.del	m.Ons ≫ f.dep ≫ **f.max ≫** ***m.NoCoda***	**CA**.del $<_{\text{f.max}}$ **CP**.del

3∪**4**:**CP**.ins	f.max ≫ m.Ons ≫ ***m.NoCoda*** **≫ f.dep**	**CP**.ins $<_{\text{m.NoCoda}}$ **CA**.ins
7∪**8**:**CA**.ins	f.max ≫ m.Ons ≫ **f.dep ≫** ***m.NoCoda***	**CA**.ins $<_{\text{f.dep}}$ **CP**.ins

Equivalence relations follow from border point pairs in which m.NoCoda is *not* one of the two constraints transposed and lies in the prefix. The following pair meets this description:

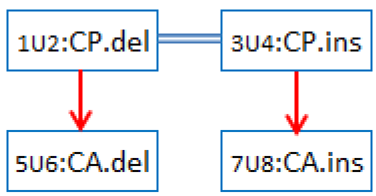
(229) **Equivalence on m.NoCoda**

Grammars	Border Point Pair	EPO Equivalence
1∪**2**:CP.**del**	**m.NoCoda** $\gg$ *f.dep* $\gg$ f.max $\gg$ m.Ons	CP.**del** $\sim_{\text{m.NoCoda}}$ CP.**ins**
3∪**4**:CP.**ins**	**m.NoCoda** $\gg$ f.max $\gg$ *f.dep* $\gg$ m.Ons	

Here languages CP.del and CP.ins both pass through m.NoCoda in the first position of these two rankings. (Concretely, in these languages all syllables are open because no optimum violates m.NoCoda.) Hence CP.del $\sim_{\text{m.NoCoda}}$ CP.ins. Note that equivalence on a constraint C is necessitated by any *single* total order in which multiple languages pass through C. The cited border point pair gives us two such instances.

Coupling the order relations and equivalence relations produces EPO(m.NoCoda).

(230) **EPO$_{\text{CSys}}$(m.NoCoda)**



3. f.dep

Here transitive order relations obtain, ensuring that f.dep must make a three-way distinction among the languages in any instantiation of CSys. There are three privileged order relations:

CP.del $<_{\text{f.dep}}$ CP.ins
CA.del $<_{\text{f.dep}}$ CA.ins
CA.ins $<_{\text{f.dep}}$ CP.ins.

These emerge from various border point pairs, three of which are shown in (231).

(231) **Privileged orders** on f.dep

Grammars	Border Point Pairs	Privileged Relations
1∪**2**:CP.**del**	m.NoCoda $\gg$ m.Ons $\gg$ ***f.dep*** $\gg$ **f.max**	CP.**del** $<_{\text{f.dep}}$ CP.**ins**
3∪**4**:CP.**ins**	m.NoCoda $\gg$ m.Ons $\gg$ **f.max** $\gg$ ***f.dep***	CP.ins $<_{\text{f.max}}$ CP.del

5∪**6**:CA.**del**	m.Ons $\gg$ ***f.dep*** $\gg$ **f.max** $\gg$ m.NoCoda	CA.**del** $<_{\text{f.dep}}$ CA.**ins**
7∪**8**:CA.**ins**	m.Ons $\gg$ **f.max** $\gg$ ***f.dep*** $\gg$ m.NoCoda	CA.ins $<_{\text{f·max}}$ CA.del

7∪8:**CA**.ins	f.max ≫ ***f.dep* ≫ m.NoCoda** ≫ m.Ons	**CA**.ins $<_{\mathrm{f.dep}}$ **CP**.ins
3∪4:**CP**.ins	f.max ≫ **m.NoCoda ≫ *f.dep*** ≫ m.Ons	CP.ins $<_{\mathrm{m.NoCoda}}$ CA.ins

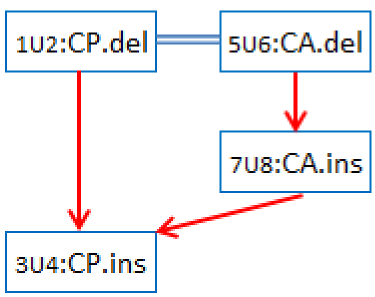
There are only two languages that are equivalent in EPO(f.dep): CA.del and CP.del. Either of the two total orders from the border point pair below demonstrates this, as both languages pass together through highest-ranked f.dep in each total order.

(232) **Equivalence on f.dep**

Grammars	Border Point Pair	EPO equivalence
5∪6:**CA**.del	**f.dep** ≫ *f.max* ≫ m.NoCoda ≫ m.Ons	**CA**.del $\sim_{\mathrm{f.dep}}$ **CP**.del
1∪2:**CP**.del	**f.dep** ≫ m.NoCoda ≫ *f.max* ≫ m.Ons	

The border-point derived relations are represented in the following bigraph.

(233) **EPO$_{\mathrm{CSys}}$(f.max)**



4. f.max

The structure of the f.max EPO mirrors that of the f.dep EPO: one can be obtained from the other by swapping the *ins* and *del* versions of output-identical languages. As we would expect from this symmetry, there are three privileged order relations in the f.max EPO:

CP.ins $<_{\mathrm{f.max}}$ CP.del
CA.ins $<_{\mathrm{f.max}}$ CA.del
CA.del $<_{\mathrm{f.max}}$ CP.del.

The following three border point pairs establish these relations.

(234) **Privileged relations on f.max**

Grammars	Border Point Pairs	Privileged Relations
3∪4:CP.**ins**	m.NoCoda ≫ m.Ons ≫ ***f.max* ≫ f.dep**	CP.**ins** $<_{\mathrm{f.max}}$ CP.**del**
1∪2:CP.**del**	m.NoCoda ≫ m.Ons ≫ **f.dep ≫ *f.max***	CP.del $<_{\mathrm{f.dep}}$ CP.ins

7∪**8**:CA.**ins**	m.Ons ≫ ***f.max* ≫ f.dep** ≫ m.NoCoda	CA.**ins** $<_{\text{f.max}}$ CA.**del**
5∪**6**:CA.**del**	m.Ons ≫ **f.dep ≫ *f.max*** ≫ m.NoCoda	CA.del $<_{\text{f.dep}}$ CA.ins

5∪**6**:**CA**.del	f.dep ≫ ***f.max* ≫ m.NoCoda** ≫ m.Ons	**CA**.del $<_{\text{f.max}}$ **CP**.del
1∪**2**:**CP**.del	f.dep ≫ **m.NoCoda ≫ *f.max*** ≫ m.Ons	CP.del $<_{\text{m.NoCoda}}$ CA.del

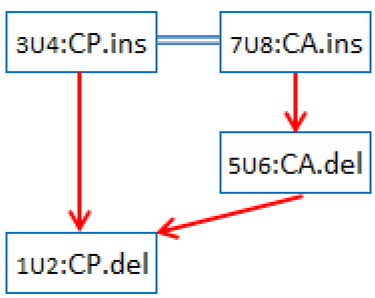
One equivalence relation holds: CP.ins $\sim_{\text{f.max}}$ CA.ins. The prefix of either border point in the following pair gives this result:

(235) **Equivalence on f.max**

Grammars	**Border Point Pair**	**EPO Equivalence**
3∪**4**:**CP**.ins	**f.max** ≫ *m.NoCoda* ≫ f.dep ≫ m.Ons	**CP**.ins $\sim_{\text{f.max}}$ **CA**.ins
7∪**8**:**CA**.ins	**f.max** ≫ f.dep ≫ *m.NoCoda* ≫ m.Ons	

Putting these relations together yields an EPO bigraph.

(236) **EPO$_{\text{CSys}}$(f.max)**



The collection of these four EPOs constitutes the MOAT of CSys, which we have derived by analyzing border point pairs from the concretely-derived Universal Support (217).

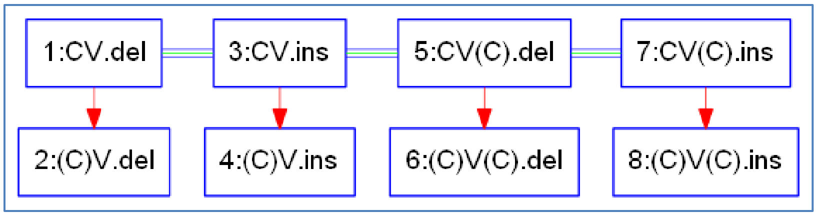
Let's now approach the CSys from the fully abstract side, starting from the EST MOAT and merging its nodes in such a way that that their associated grammars are typological classes which line up with the grammars of the CSys.

How do we connect the abstract calculation with the concrete results from the CSys? Because the EST and the CSys share CON, we may directly identify the grammars of one with the grammars of the other through their ranking content. Our strategy is to merge nodes in the EST MOAT, producing derived EPOs in which the constituent languages have ranking grammars that match those of the CSys leg for leg.

As above, we advance EPO by EPO. We begin with m.Ons in the EST.

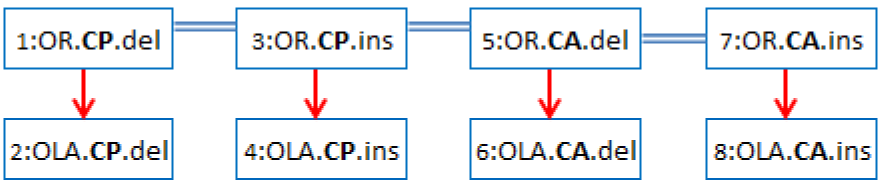
1. From EPO_{EST}(m.Ons)

(237) **EPO_{EST}(m.Ons)** with iconic labels



Using the descriptive labels, the m.Ons EPO looks like this:

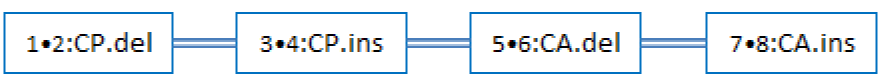
(238) **EPO_{EST}(m.Ons)** with descriptive labels



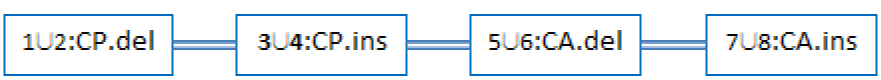
Merging nodes **1** and **2** corresponds to unioning their legs. Since **1** is OR.CP.del and **2** is OLA.CP.del, merging them into a node **1•2** gives us the graphical correlate of CP.del, as desired. Similarly, merging **3**:OR.CP.ins with **4**:OLA.CP.ins gives **3•4**:CP.ins, and so on.

This operation collapses distinctions between the languages on the onset dimension, leaving the four merged nodes equivalent, and consequently induces no cycles. This new EPO is structurally identical to the EPO for m.Ons for CSys (227), reproduced as (240).

(239) **EPO_{CSys}(m.Ons)** derived from node mergers on EPO_{EST}(m.Ons)

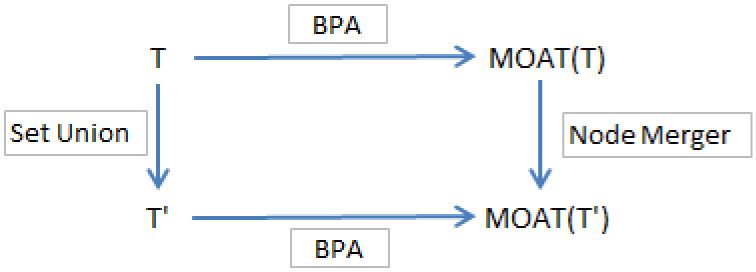


(240) **EPO_{CSys}(m.Ons)** derived by BPA on the CSys.



Node merger in the graph corresponds to union in the ranking grammars, and the CSys is built directly from unions, as shown above in (220). These EPOs therefore denote identical leg content amongst the corresponding grammar. The significance is that we are able to construct the EPO of the coarsened typology without analyzing the legs of its grammars: once we have the MOAT, all such calculations can be performed through merger. Two courses of action lead to the same result. We can represent this fact in a diagram.

(241) Two routes to the EPOs of a coarsened typology



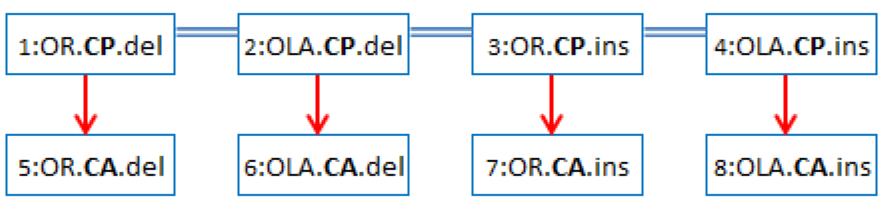
This diagram resembles that of (215), which compares node merger with modifying $\mathrm{GEN}_{\mathrm{EST}}$ as modes of coarsening the EST. But here the action is more abstract, taking place at the level of grammars and typologies, making no direct contact with the extensional languages. The diagram indicates that the composition of set union (in the typology) with BPA yields the same result as the composition of BPA with node merger (in the MOAT). For this to work, set union must be typological coarsening and node merger must not introduce cycles. When node merger preserves acyclicity, then as we've shown (§3.3) set union is typological coarsening; and vice versa.

With the overall picture in place, let's continue with the details.

2. From $\mathbf{EPO_{EST}}$(m.NoCoda)

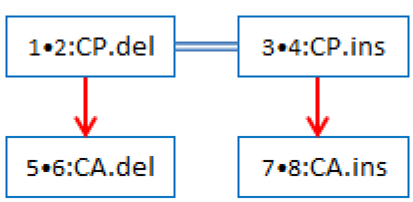
$\mathrm{EPO}_{\mathrm{EST}}$(m.NoCoda) mirrors $\mathrm{EPO}_{\mathrm{EST}}$(m.Ons), mutatis mutandis, overall. In it, the four CP grammars are equivalent ($\mathbf{1} \sim_{\mathrm{m.NoCoda}} \mathbf{2} \sim_{\mathrm{m.NoCoda}} \mathbf{3} \sim_{\mathrm{m.NoCoda}} \mathbf{4}$), with each of the four ordered above one grammar from the CA series.

(242) **$\mathbf{EPO_{EST}}$(m.NoCoda)**



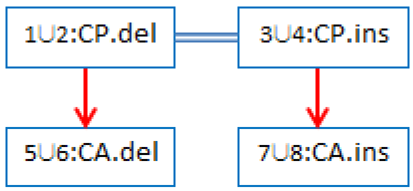
We merge the four pairs of horizontal neighbors (**1**, **2**), (**3**, **4**), (**5**, **6**), (**7**, **8**), leading to a well-formed EPO with no cycles.

(243) **$\mathbf{EPO_{CSys}}$(m.NoCoda)** from node merger



Our direct calculation by border point analysis on the CSys yielded the following result, derived in (228) and (229):

(244) **EPO$_{CSsys}$(m.NoCoda)** from BPA

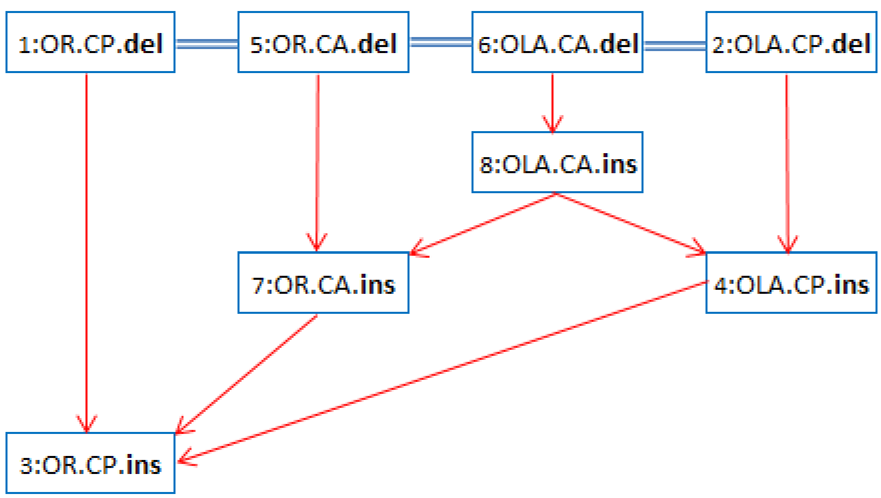


As before, these EPOS are equivalent, denoting the same languages with the same order and equivalence relations holding among them.

3. From EPO$_{EST}$(f.dep)

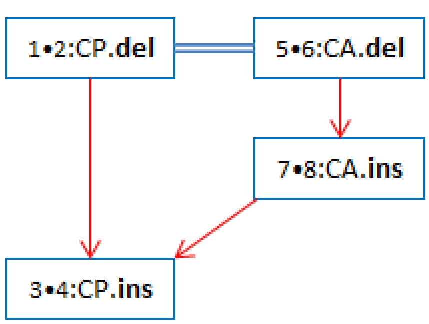
EPO$_{EST}$(f.dep) is given below, repeated from (79).

(245) **EPO$_{EST}$(f.dep)**



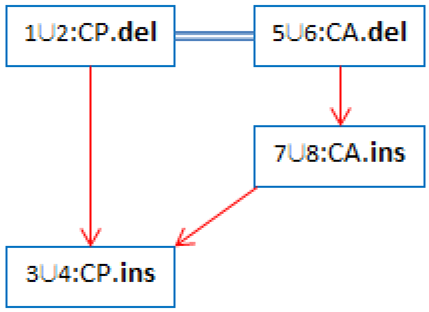
Forming, as always, the mergers **1•2**, **3•4**, **5•6**, **7•8** will derive the following:

(246) **EPO$_{CSys}$(f.dep)** via node merger



Calculating from border points in the CSys yields the following result, as in

(247) **EPO$_{CSys}$(f.dep)** via BPA

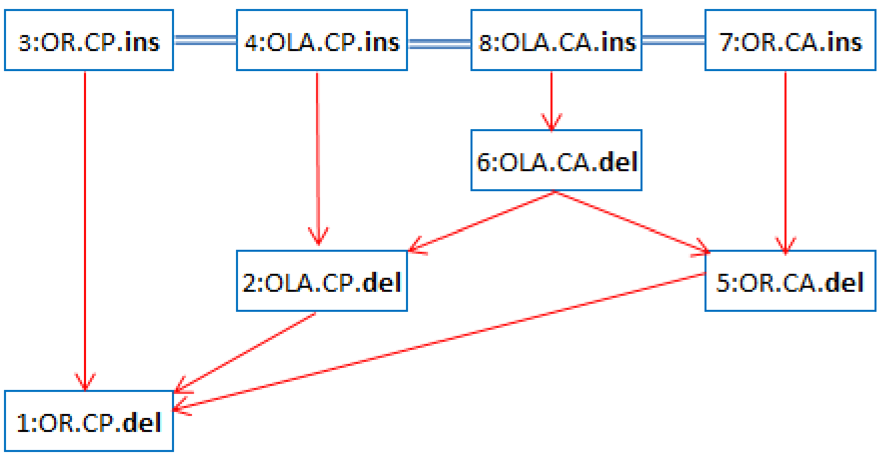


Once again, inevitably, isomorphic results are obtained.

4. f.max EPO for EST

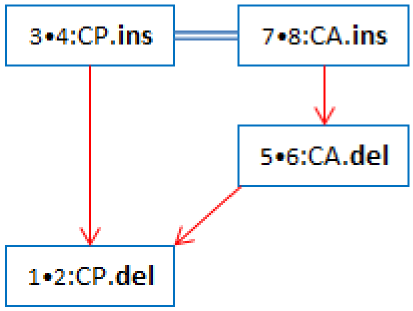
We conclude by examining the last remaining EPO in the EST MOAT: f.max, which symmetrically reflects that of f.dep. The EST version of the f.max is repeated from (80).

(248) **EPO$_{EST}$(f.max)**



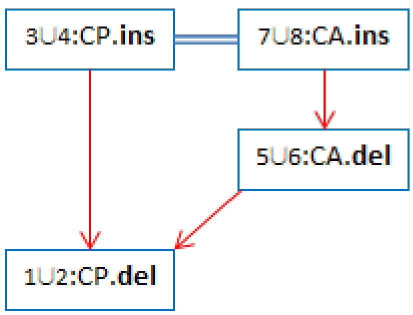
Merging the nodes that differ only in onset specification yields this EPO:

(249) **EPO$_{CSys}$(f.max)** via node merger



Direct calculation from the concrete CSys yields this EPO, reproduced from (236):

(250) **EPO$_{CSys}$(f.max)** via BPA



The two paths arrive at the same goal. This concludes the demonstration. □

The merger of nodes in every EPO corresponds, in this case, to grouping pairs of grammars, and thence pairs of extensional languages, together into empirically meaningful classes. By merging EST grammar 1:OR.del with 2:OLA.del, and 3:OR.ins with 4:OLA.ins, and so on, we obtain a coarser typology whose grammars are classes of grammars and whose languages are classes of EST languages. These classes abstract away from the distinctions in treatment of onsets, retaining the EST distinctions in consonant behavior and modes of unfaithfulness. This gives an analysis of one aspect of the EST. Similarly, by merging nodes of grammars that differ only in the treatment of consonants, we can achieve a parallel analysis of onset behavior — the VSys.

In this simple case, these abstract maneuvers are reflected exactly in the concrete OT system under analysis. The CSys may be obtained by tailoring GEN$_{CSys}$ so that an onsetless configuration never arises. From the abstract point of view, however, there is no need to search for a concrete instantiation, which may not exist, within the structural strictures of the original typology. In the stress typologies studied by Alber & Prince, for example, we can abstractly merge information from forms of different lengths, even though no concrete form can have both an even and an odd number of syllables. The formal system of MOAT manipulation based on node merger provides the entire inventory of possible typological classes, a starting point for principled classification of language types in terms of ranking relations that define structural patterns. This strategy of abstract analysis pushes the classification program forward in both practice and theory.

5 Cohabitation & the Join

SubTOC

Classifying grammars together is mirrored by merger of their nodes in the EPOs of a MOAT. We have seen in §4 how EST grammars merge to form classes like CP.del and CP.ins in the coarser CSys typology. But merger has its perils: when the result contains cycles, typological status is lost.

There are two sources for this outcome. First, the merger may not be a grammar at all: in this case, it is not ERC-characterizable, not a 'grammatical class' and it has no hope of being hosted in a typology. Second, and more subtly, a set of grammars may merge to a perfectly fine abstract grammar but may nonetheless fail to be EPO-orderable with its neighbors, failing to be a typological class of its host typology.

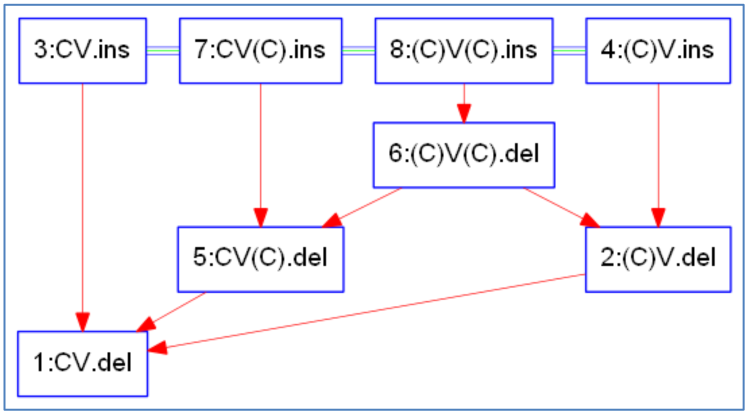
In studying the first case (§5.1), we show how the question of grammatical class status can be resolved in a typology-independent way by the ERC-logic based operation of the *join* (Merchant 2008, 2011). The second case is fleshed out with two examples which show different aspects of MOAT structure at play (§5.2), leading inter alia to the conclusion that a set of grammars may when merged form a typological class even though when unmerged it cannot sit in any typology due to internal cycles.

5.1 To Form a More Perfect Union

All typological classes can be obtained by node merger in the MOAT, but when merger yields a cyclic bigraph, it takes us outside the class of well-formed MOATs altogether. The block of legs denoted by the merged nodes in a cycle does not form a typological class because merger does not derive a typology, and it may not even constitute a *grammatical* class — an abstract grammar, characterizable by an ERC set. In this section, we examine a method based in ERC logic, the 'join' (Merchant 2008, 2011) that allows us determine when grammars can union into a grammatical class, regardless of their position in any surrounding typology.

We begin by examining in detail the character of a cycle-inducing merger in the full EST. Let us return to the extensionally-motivated grouping 'Onset Required' (OR), consisting of languages **{1**,**3**,**5**,**7}** which contrasts with the grouping 'Onset Lack Allowed' (OLA). Let's examine the merger in the EPO_{EST}(f.max). We repeat it here in its unmerged form:

(251) **EPO$_{EST}$(f.max)**



←OR | OLA→

Let's approach the problem incrementally, by first merging languages that only disagree along the (vertical) ins/del axis in diagram (251), such as **1**:CV.*ins* and **3**:CV.*del* at the far left. These mergers are **1•3**, **2•4**, **5•7**, **6•8**. In terms of the descriptive labels, we have:

(252) **EST unions** of ins/del pairs

Unions: X.del ∪ X.ins			Result	Mergers
1:OR.CP.del	⋃	3:OR.CP.ins	OR.CP	**1•3**
5:OR.CA.del	⋃	7:OR.CA.ins	OR.CA	**5•7**
2:OLA.CP.del	⋃	4:OLA.CP.ins	OLA.CP	**2•4**
6:OLA.CA.del	⋃	8:OLA.CA.ins	OLA.CA	**6•8**

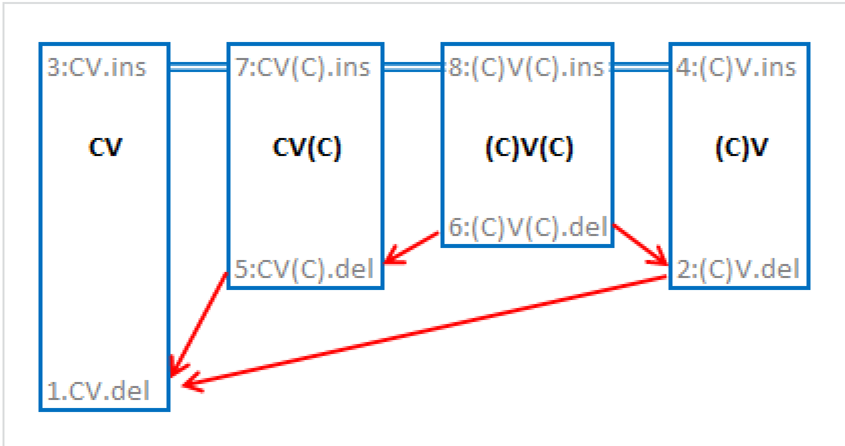
This serves as an instructive step on the way to merging all nodes **1**,**3**,**5**,**7** on the OR side and all nodes **2**,**4**,**6**,**8** on the OLA side. These four mergers generalize away from faithfulness distinctions and will divide EST into the four familiar Jakobsonian output classes (P&S: §6):

(253) **Output classes** of the EST

CV	OR.CP
CV(C)	OR.CA
(C)V	OLA.CP
(C)V(C)	OLA.CP

The results of merger are shown graphically below.

(254) **Bigraph of EPO(f.max)** with merger of ins/del classes



←OR | OLA→

The Jakobsonian bigraph is already cyclic in a way that spells doom for the project of getting the OR/OLA distinction from typological classes. The diagram has been constructed to indicate how prior relations between the components of distinct merged nodes persist into the merger.

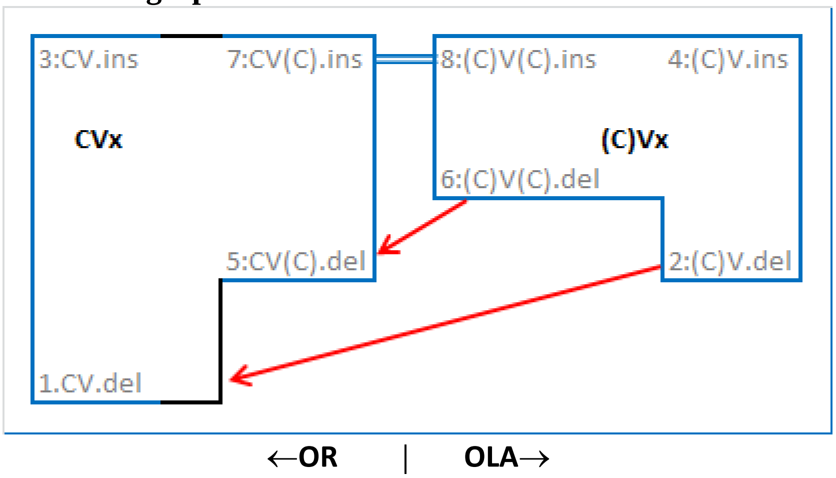
Every pair of nodes in bigraph (254) is involved in a cycle.[39] By virtue of their 'ins' components, the members of any pair are connected by '═', signaling equivalence of the grammars they denoted. By virtue of their 'del' components, all pairs are forced into an order relationship, in some cases directly, in some cases by transitivity of the '$<_{f.max}$' relation denoted by '→'. By 'hypertransitivity', which collates the consequences of order and equivalence taken together, we have a plethora of cycles.

Some of the cycles are erased in the further mergers that create the OR/OLA distinction. For example, we see in (254) that the inherited, retained relations **3═7** and **5→1** produce a cycle between the left-hand pair of nodes, which comprise the OR half of the typology. This particular cycle will be erased when we merge them to produce OR = **1•3•**5•7, because the obstructing relation **5→1** is no longer external to the merged node. But the relations between components of OR (left-side pair) and components of OLA (right-side pair) have the same character and will undermine the typological status of the OR and OLA groupings.

To see this, we merge again to produce the OR/OLA bigraph.

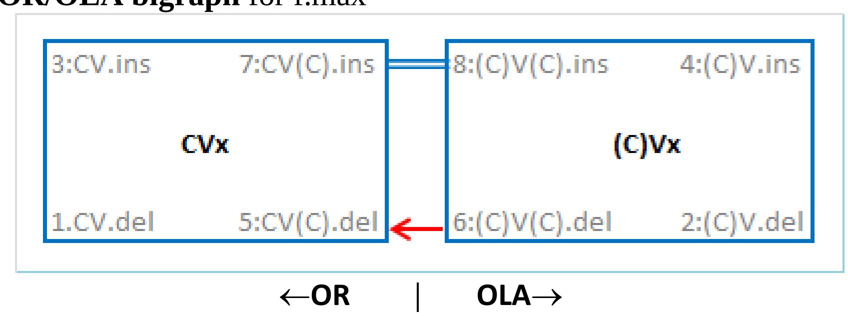
[39] This includes pairs that are not shown as adjacent in diagram (254). Of the pairs, only 5•7, 2•4 requires touring outside the pair. Recall that equivalences are represented sparsely, so that although all the nodes are equivalent, only three '═' are shown — diagrammatically convenient, and enough to entail the others.

(255) **OR/OLA bigraph** for f.max with all external relations shown



Consolidating the order information yields the following concise form.

(256) **OR/OLA bigraph** for f.max



The class OR **1•3•5•7** must be EPO-equivalent and UVT-equal to OLA **2•4•6•8** by virtue of relations involving **3**, **7** (OR) and **4**,**8** (OLA). At the same time **1•3•5•7** must be greater than than **2•4•6•8** by virtue of relations between **1**,**5** (OR) and **2**, **6** (OLA). No consistent assignment of numbers is in the offing.

The algebraic details run like this:

- OLA $<_{\text{f.max}}$ OR, inherited from e.g. **2**$\subseteq$OLA $<_{\text{f.max}}$ **1**$\subseteq$OR
- OR $\sim_{\text{f.max}}$ OLA, inherited from e.g **7**$\subseteq$OR $\sim_{\text{f.max}}$. **8**$\subseteq$OLA
- OLA $\lessdot_{\text{f.max}}$ OLA, by hypertransitivity, a contradiction.[40]

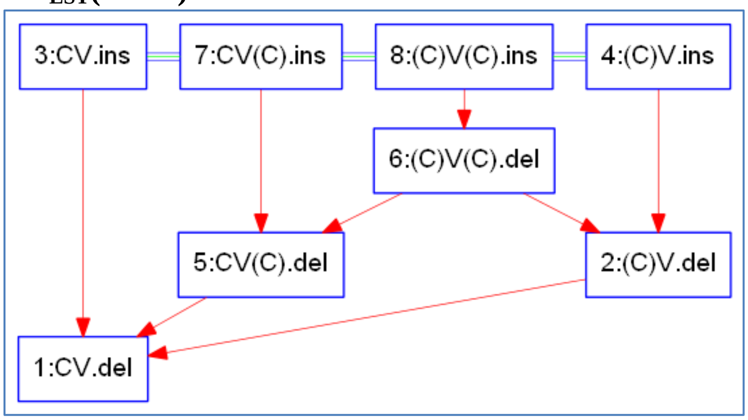
Observe that the project was doomed at the lowest level: the class CV, descriptively OR.CP, tries to amalgamate the grammar of 1:CV.del, in which f.max is dominated by every other constraint, with that of 3:CV.ins, in which f.dep dominated by every other constraint. But in this collocation, there is no single constraint that is necessarily dominated throughout, and therefore no ERC that expels some legs from the presumed

[40] Recall that '$\lessdot_X$' is the strict partial order relation obtained by combining information from '$<_X$' and '$\sim_X$'. As such, it must be implemented as numerical '$<$' in any UVT.

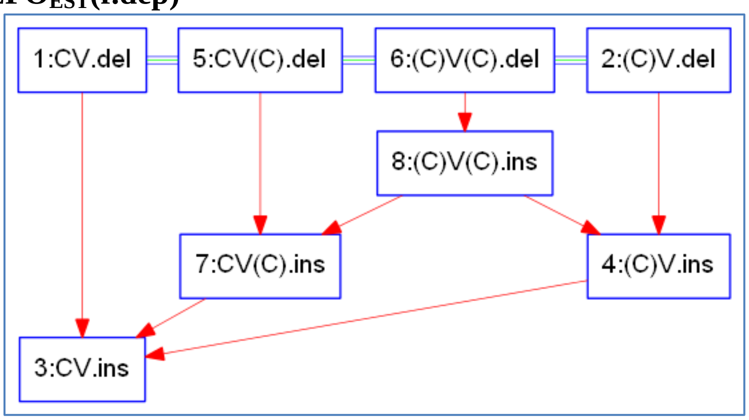
grammar. Thus CV, descriptively OR.CP, is not even a *grammatical* class; and similarly for the others.

We have arrived at the border of what a grammatical or typological classification system allows. Alber & Prince (2015, in prep.) observe that the issue arises because classes of constraints may be systematically related in ways that go beyond the logic of single ERCs. In the system at hand, f.dep and f.max behave symmetrically with respect to the grammars of the system, as is readily visible in their EPOs, repeated here for ease of comparison.

(257) **EPO$_{\mathbf{EST}}$(f.max)**



(258) **EPO$_{\mathbf{EST}}$(f.dep)**



In such cases, the parallelism indicates that constraints form classes to which analysis must refer. The relevant class here is 'Faithfulness', abbreviated as F, with F = {f.max, f.dep} for EST. The classification theory of Alber & Prince 2015 allows reference to the member of that class which is *subordinate* in a leg λ — lowest ranked among its class — as 'F.sub(λ)'. Applying this to the present case, we see that OR meets the condition 'm.Ons(λ)≫F.sub(λ)' in every λ in OR, because in every leg of the OR grammars, the occurrence of m.Ons dominates that member of the faithfulness pair which is subordinate to the other. OLA meets the rankingwise opposite condition 'F.sub(λ)≫m.Ons(λ)'. In this case, since in any leg the F constraints are ordered with respect to each other, when F.sub(λ) dominates m.Ons, it follows from transitivity of domination that *both* faithfulness constraints dominate m.Ons. When the theory is further augmented by the ability to refer to *dominant* member of a constraint class in a leg as e.g. F.dom(λ), a theory of reference to constraint classes emerges that generalizes the ERC. See Alber & Prince 2015, in prep. for development. For present purposes, the significant finding is that the theory of typological classification, whose structure we are investigating here, sets the stage for analyzing such further articulations, of which it forms a proper part.

The MOAT allows us to determine, via node merger, whether the union of grammars results in a grammar that belongs to a coarsened version of a reference typology. We may also focus on the grammars and pose the question independent of a typological surround: given two grammars, is their leg-union also a grammar? Taken together, do they form a *grammatical class*? Recall that a grammatical class is a set of grammars whose union is itself a grammar, whereas a typological class within a typology T is a set of grammars within T whose union forms a coarser typology T′.

The MOAT allows us to answer the narrower question — when does a collection of grammars form a *typological* class ?— narrower because a typological class is *a fortiori* a grammatical class. The more general question may be answered directly within the realm of ERC grammars without reference to a containing typology. There, the use of the *join*, an operation on grammars introduced in Merchant 2008, 2011, making use of ERC logic (Prince 2002), determines the smallest grammar that contains the legs of the union.

To see how this works in practice, we begin by observing that the class OR has the following leg collection, which includes all the legs of the four grammars for languages in which all well-formed syllables have onsets.

(259) **OR legs**

1:CV.del	m.NoCod	≫	f.dep	≫	**m.Ons**	≫	**f.max**
	f.dep	≫	m.NoCod	≫	**m.Ons**	≫	**f.max**
	m.NoCod	≫	**m.Ons**	≫	f.dep	≫	**f.max**
	f.dep	≫	**m.Ons**	≫	m.NoCod	≫	**f.max**
	m.Ons	≫	m.NoCod	≫	f.dep	≫	**f.max**
	m.Ons	≫	f.dep	≫	m.NoCod	≫	**f.max**
5:CV(C).del	f.dep	≫	**m.Ons**	≫	**f.max**	≫	m.NoCod
	m.Ons	≫	f.dep	≫	**f.max**	≫	m.NoCod
3:CV.ins	m.NoCod	≫	f.max	≫	**m.Ons**	≫	**f.dep**
	f.max	≫	m.NoCod	≫	**m.Ons**	≫	**f.dep**
	m.NoCod	≫	**m.Ons**	≫	f.max	≫	**f.dep**
	f.max	≫	**m.Ons**	≫	m.NoCod	≫	**f.dep**
	m.Ons	≫	f.max	≫	m.NoCod	≫	**f.dep**
	m.Ons	≫	m.NoCod	≫	f.max	≫	**f.dep**
7:CV(C).ins	f.max	≫	**m.Ons**	≫	**f.dep**	≫	m.NoCod
	m.Ons	≫	f.max	≫	**f.dep**	≫	m.NoCod

As we've just shown by a MOAT-based argument, the legs of all OR languages, taken together, do not form a grammar in any typology coarsened from the EST, indeed in any typology. By inspecting table (259), we can deduce, independent of typological considerations, that there is no ERC set that delimits OR.

The simplest argument would simply be to audit the leg table, and to note, paralleling the the discussion of ex. (256), that each of the four constraints appears in top position in some leg. Since no constraint is crucially dominated, there is no ERC grammar that excludes any leg. A nontrivial ERC grammar, one that excludes some rankings, requires at least one constraint to be subordinated in every leg.

We may also expand on an observation of greater generality, first made in §0.3.3, p. 40. Admission to the OR leg set requires *either* m.Ons≫f.dep *or* m.Ons≫ f.max. An ERC says that *some* constraint assessing W must dominate *every* constraint assessing L, so that disjunction among the dominated is not ERC representable. Nor can there be a *set* of ERCs that embodies this disjunction. A set imposes the conjunction of the requirements of each ERC in the set. Every ERC must be satisfied, and there's no general way to say, disjunctively, that some ERC *or* another must hold. This argument establishes that OR is not a grammar and applies to any case where a leg set is only describable with disjunction among the dominated. The Alber & Prince classification theory, which refers to classes like OR by the operators *dom* and *sub*, takes us (via *sub*) outside the descriptive capacity of the ERC and the ERC set.

What this means in the realm of ERC grammars is that the *join* of the four OR grammars, which we represent as **1**+**3**+**5**+**7**, has more legs in it than **1**∪**3**∪**5**∪**7**, which is shown in table (259). The join of the four includes every leg in EST. This superfluity means that the simple union of the four is not a grammatical class.

The logic of the join is that of coordinatewise ERC disjunction. Just as in propositional logic the formula p∨q is the 'smallest' formula that is entailed by both p and q individually, so with ERC grammars P, Q, the join P+Q is the 'smallest' grammar containing both P and Q.[41] 'Smallest' means here that P+Q contains the legs of P and the legs of Q, along with possibly more, with the proviso that any ranking grammar R containing the legs of P and the legs of Q also contains those of P+Q.

The join of two ERC grammars is constructed from a suitably full ERC representation of both: each ERC in the first is disjoined in the coordinatewise ERC logic manner with each ERC in the second. The result is an ERC set — a grammar — shown by Merchant to be the smallest grammar legwise that contains both joining grammars. We sketch the operation here.

ERC logic disjunction is like Boolean disjunction in that it has the properties of idempotence (X ∨ X = X) and commutativity (X ∨ Y = Y ∨ X). The W value behaves like T and L behaves like F; the third value *e* is interpolated between them. See Prince 2002:51 for the details. The basic recipe runs like this:

(260) **ERC logic disjunction** (applies to each coordinate). For $X \in \{W, e, L\}$,

$$W \vee X = W$$
$$L \vee X = X$$
$$e \vee e = e$$

Compare Boolean T∨X = T, F∨X = X. We may also think of this as working from a scale W> *e* > L. Disjunction of values returns the greatest value among the disjuncts. Boolean logic, from this perspective, operates on the binary scale T>F.

The join of two grammars G_1+G_2 disjoins every ERC in an ERC grammar of G_1 with every ERC in an ERC grammar of G_2. (Caveat: the operation requires these grammars to be suitably rich in ERC content.[42]) An ERC grammar unique up to logical equivalence results. Because the join, like disjunction itself, is commutative and associative, we may easily extend the definition to arbitrary finite sets of languages.

[41] Details, details: of course q∨p and Q+P fill the bill as 'smallest' as well. To go full Lindenbaum, one must deal with equivalence classes of formulae.

[42] In particular, the grammars must be such that for every α with $G \vDash \alpha$, there must be an ERC $\gamma \in G$ such that $\gamma \vDash \alpha$. Merchant uses the fusional closure of G to ensure that this condition is met.

What, then, is the result of *joining* the grammars of OR, namely **1**+**3**+**5**+7? Equivalently: what is the smallest grammar that contains the legs of all of these? We know already that it is the trivial grammar that includes all legs. Why does the join give us this result?

As Merchant observes, to produce a nontrivial ERC in the join, one that is not true of every ranking, a disjoining pair of ERCs must share an L in some constraint. According to the disjunction table (260), the only way to get L in the output is to disjoin two Ls in the input: L∨L = L. Since **1**,**3**,**5**, and **7** do not share even one dominated constraint, the join is going to be trivial in the sense that it admits all rankings.

By contrast, compare the nontrivial join of **1**:CV.del (OR.CP.del) and **2**:(C)V.del. (OLA.CP.del). These two grammars disagree on the obligatoriness of onsets, but both ban codas in optima and achieve conformity with the CP requirement by deletion. In the joined grammar **1**+**2**:CP.del the distinction between the onset requirements in the joinards is wiped out. The constraint f.max is crucially dominated in both.

(261) **ERC grammars** of 1:CV.del and 2:(C)V.del

(a) **1:CV.del**

m.Ons	m.NoCoda	f.dep	f.max
W			**L**
	W		**L**
		W	**L**

(b) **2:(C)V.del**

m.Ons	m.NoCoda	f.dep	f.max
L		**W**	
	W		**L**
		W	**L**

The join comes out like this:

(262) **Join: 1:CV.del + 2:(C)V.del**

m.Ons	m.NoCoda	f.dep	f.max
	W		**L**
		W	**L**

The join may be formed by disjoining every ERC of (261)a with every ERC of (261)b, or more economically, by disjoining those pairs of ERCs, one from each grammar, which share L in some coordinate. Those which share no L disjoin to an ERC without L, which is true of every ranking, and may be discarded, or ignored uncomputed, as uninformative.

Because both **1** and **2** contain the ERC *e*W*e*L as row 2 in each of the tableaux in (261), the join will contain the idempotent disjunction *e*W*e*L$\vee$ *e*W*e*L = *e*W*e*L. The same holds for row 3 in both tableaux of (261), which contains the ERC *ee*WL. All other ERCs in the join of grammars 1 and 2 are entailed by these two and omitted as redundant from our representation of the join in (262).

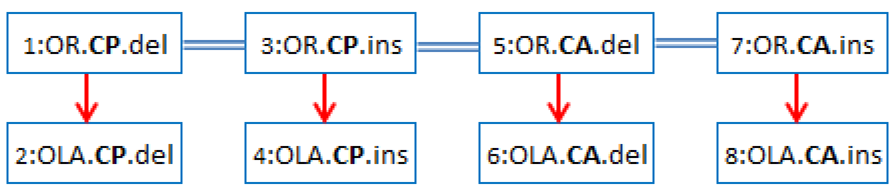
The join **1**+**2** is nontrivial and *conservative*, in the sense that the leg set of the join is exactly the union of the leg sets of the participating grammars. The smallest language containing both **1**:CV.del and **2**:(C)V.del thus contains only legs from one or the other. The join **1**+**3**+**5**+**7** of the OR languages, by contrast, is *nonconservative* since it includes every total order on CON$_{\text{EST}}$.

(263) **Definition. Conservativity of Join**. Let G_1, G_2 be ranking grammars. Let G_1+G_2 denote the ranking grammar of join of G_1 and G_2. The join of G_1 and G_2 is *conservative* iff $G_1+G_2 = G_1 \cup G_2$. Otherwise, the G_1+G_2 is *nonconservative*.

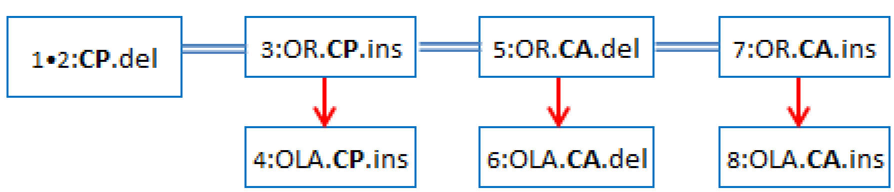
The join **1**+**2** is also a typological class of EST. To validate this claim, we show the m.Ons and f.max EPOs with the merger 1•2. The other two EPOs mirror these structurally, mutatis mutandis, so we need not examine them. To make it easier to see the relation to the EST proper, we also include the unmerged EPOs.

(264) **EPO$_{\text{EST}}$(m.Ons)**, unmerged and merged

a. **m.Ons**: EST unmerged

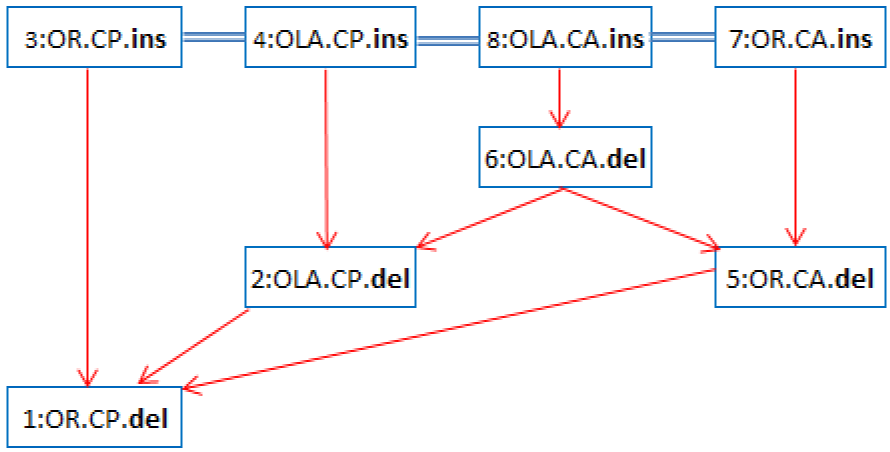


b. **m.Ons** with 1•2 merger

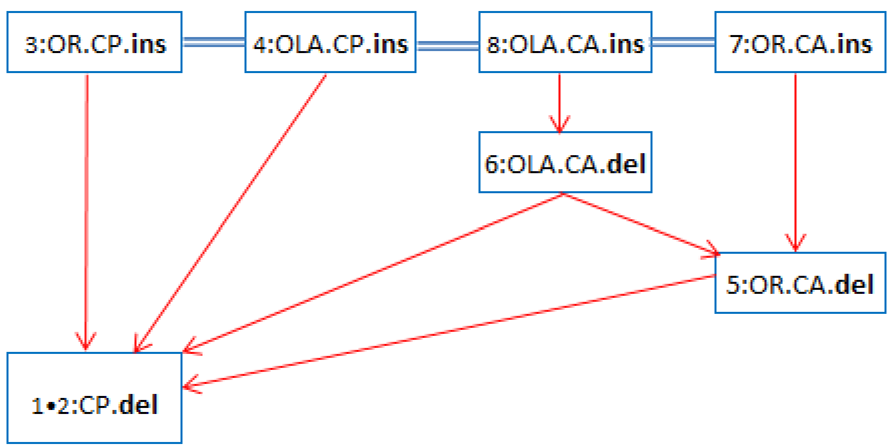


(265) f.max EPOs, unmerged and merged.

a. **f.max**:EST, unmerged



b. **f.max** with **1•2** merged



Numerical remark. The contrast between the behavior of **1•2** in the various EPOs shows how the join resists a simple numerical definition within a UVT. In EPO b, **1•2** assumes the ordinal position of grammar **1**. But **1** holds the *minimal* value for m.Ons and the *maximal* value for f.max. Therefore no simple rule of numerical combination — that the join assumes the maximum, or that it assumes the minimum of the joinards — will properly characterize its functioning. Constructing a UVT for the join requires a detour through the MOAT.

5.2 Problem 2: Typological Compatibility

In the cases examined so far, the notions 'grammatical class' and 'typological class' track each other perfectly. It is natural to conjecture that whenever grammars are conservatively joinable, they may be joined in any typology that hosts them, coarsening their host typology. This conjecture is false: MOAT structure imposes nontrivial conditions on the coexistence of grammars within a typology. We present two examples where conservatively joinable grammars cannot be merged in a typology without introducing cycles. The first case (§5.2.1) involves hypertransitivity: an ill-fated combination of equivalence and order. The second (§5.2.2) involves only order and also provides the extreme case of a set of pairwise disjoint grammars that are conservatively joinable but cannot be co-resident in any typology. The notions of grammar class and typological class are therefore distinct. The join retains its singular value in being able to compute the first from the internal content of grammars, regardless of the typology they sit in, while the MOAT stands as the arbiter of typological status.

5.2.1 The Split Bots

To find divergences between the notions of typological and grammatical classes, we consider abstract typologies on 4 constraints, named for convenience x, y, z, w. Our first example derives from a simple typology with four grammars. Each grammar has one constraint ranked at the bottom, beneath the others, with no further order restrictions. For mnemonic purposes, we call this starter typology the 4 Bots, and each 'Bot' grammar is named after its bottom-most constraint: x-Bot is {y,z,w≫x}, and so on.[43]

(266) **The 4 Bots**

Name	Grammar
x-Bot	y, z, w ≫ **x**
y-Bot	x, z, w ≫ **y**
z-Bot	x, y, w ≫ **z**
w-Bot	x, y, z ≫ **w**

The following UVT produces The 4 Bots typology:

(267) **UVT for The 4 Bots**

Name	**x**	**y**	**z**	**w**
x-Bot	**1**	0	0	0
y-Bot	0	**1**	0	0
z-Bot	0	0	**1**	0
w-Bot	0	0	0	**1**

The typology of interest is slightly more articulated. It splits both z-Bot and w-Bot into two grammars each. In one half of the split, x≫y; in the other, y≫x. Since neither z-Bot nor w-Bot imposes any ranking relation between x and y, we are free to refine the

[43] For the sake of conciseness, we write 'y≫x & z≫x & w≫x' as 'y,z,w≫x', and similarly for the others.

typology by doing so. We also apply the splitting conditions to x-Bot and y-Bot, but their grammars are untouched. They either satisfy or contradict its components; for example, x-Bot entails y≫x and contradicts x≫y. In first case x-Bot 'splits' to itself; in the second, no grammar results. The grammars of the Split Bot typology are these:

(268) The Split Bots

Name	**Grammar**
x-Bot	y, z, w ≫ **x**
y-Bot	x, z, w ≫ **y**
z-Bot-a	x, y, w ≫ z & x≫y
z-Bot-b	x, y, w ≫ z & y≫x
w-Bot-a	x, y, z ≫ w & x≫y
w-Bot-b	x, y, z ≫ w & y≫x

We can create the Split Bots by adjoining an abstract cset to the UVT for the 4 Bots:

(269) The Split Bots

		x	**y**	**z**	**w**
xy-split	x>>y	0	**1**	0	0
	y>>x	**1**	0	0	0
4 Bots	x-Bot	**1**	0	0	0
	y-Bot	0	**1**	0	0
	z-Bot	0	0	**1**	0
	w-Bot	0	0	0	**1**

From this, we derive a UVT for the Split Bots typology by Minkowski summation. The crucial entries differentiating x≫y from y≫x are boxed. In z-Bot-a, for example, the original z-Bot profile (0,0,1,0) splits to (1,0,1,0) and (0,1,1,0), introducing the distinction between x≫y and y≫x while maintaining the status of z as the bottom-ranked constraint.

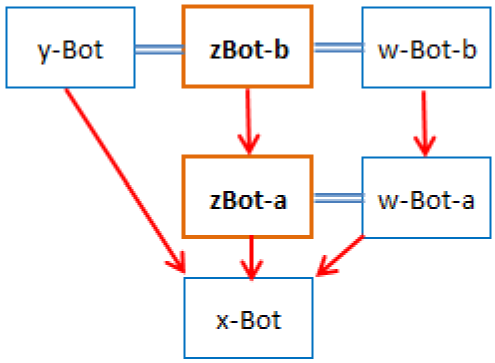
(270) **UVT for Split Bots**

Names	**x**	**y**	**z**	**w**
x-Bot	**2**	0	0	0
y-Bot	0	**2**	0	0
z-Bot-a	**1**	0	**1**	0
z-Bot-b	0	**1**	**1**	0
w-Bot-a	1	0	0	**1**
w-Bot-b	0	1	0	**1**

The key observation is that z-Bot-a and z-Bot-b are conservatively joinable into z-Bot. Their join simply neutralizes the x≫y / y≫x distinction and returns us to z-Bot. We may therefore ask whether we can create a coarsened version of the Split Bots typology, in which *only* z-Bot-a and z-Bot-b are joined. Our target would be a typology with just one pair of split Bot languages, all the others being straight Bots.

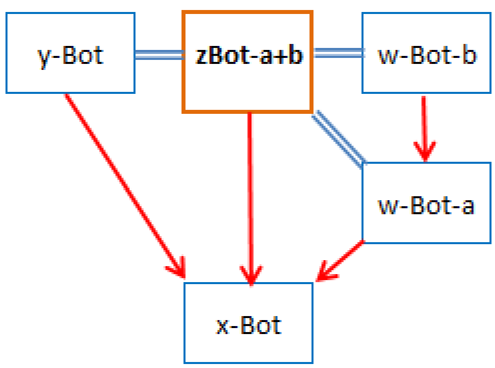
The acid test occurs in the node mergers of the Split Bot MOAT. Here is the EPO for x.

(271) **EPO(x) in the Split Bots**



Merging z-Bot-a and z-Bot-b, the two halves of z-Bot, results in a bigraph that is not an EPO, but rather a generalized EPO or GEPO (§3.3), because it contains a cycle consisting of two equivalences and a strict order. The cycle, shown on the right side of the following diagram, involves the merged node z-Bot-a • z-Bot-b and the two halves of w-Bot. Because z-Bot-a and z-Bot-b are conservatively joinable, we label this merged node z-Bot-a+b. It is of course equivalent to z-Bot.

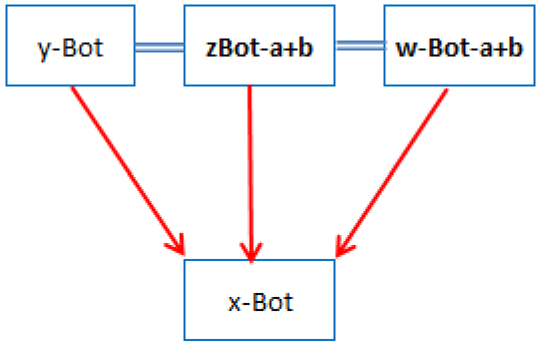
(272) **GEPO(x) from merger** of z-Bot-a and z-Bot-b



The joined language z-Bot-a+b is, by virtue of its z-Bot-b component, equivalent to w-Bot-b (top row). By virtue of its z-Bot-a component, the node z-Bot-a+b is equivalent to w-Bot-a (second row). But w-Bot-b and w-Bot-a are in a strict order relation with each other. Therefore, z-Bot-a+b cannot be equivalent to both of them.

Observe that if we merge w-Bot-a and w-Bot-b as well, we recover w-Bot. This leads us back to the original, entirely healthy EPO(x) of the 4 Bots.

(273) $EPO_{4Bots}(x)$



This confirms the observation that z-Bot-a and z-Bot-b are conservatively joinable. But their join in the 4 Bots is typologically valid only when w-Bot-a and w-Bot-b are simultaneously joined to w-Bot so that we recover the 4 Bots. This example shows that membership in a typology imposes restrictions on fellow languages, restrictions which follow from the relations that hold between them in the MOAT.

5.2.2 The Contradictory Snake

The Split Bots example shows a debilitating interaction between equivalence and order. Here we show that contradiction can be achieved with order alone. In addition, we find a conservatively joinable set of pairwise disjoint grammars that cannot sit together in any typology.

Our reference typology with constraints x, y, z, w contains five 5 languages; let's call it the Snake, for reasons that will shortly become apparent. The following UVT derives it.

(274) The Snake

Names	**x**	**y**	**y**	**w**
x-Top	0	1	1	1
S_1	1	1	0	1
S_2	2	0	0	1
S_3	3	0	1	0
S_4	4	1	0	0

We first observe that the languages labeled S_1 and S_4 are conservatively joinable, as determined by the Join Explorer of OTWorkplace. We reproduce here ERC grammars of both, along with their join, all in MIB form.

(275) MIB of S_1

S_1	x	y	z	w
a	**L**	**L**	**W**	**L**
b	**W**	**L**		**L**

(276) MIB of S_4

S_4	x	y	z	w
c	**L**	**L**	**W**	
d	**L**	**L**		**W**

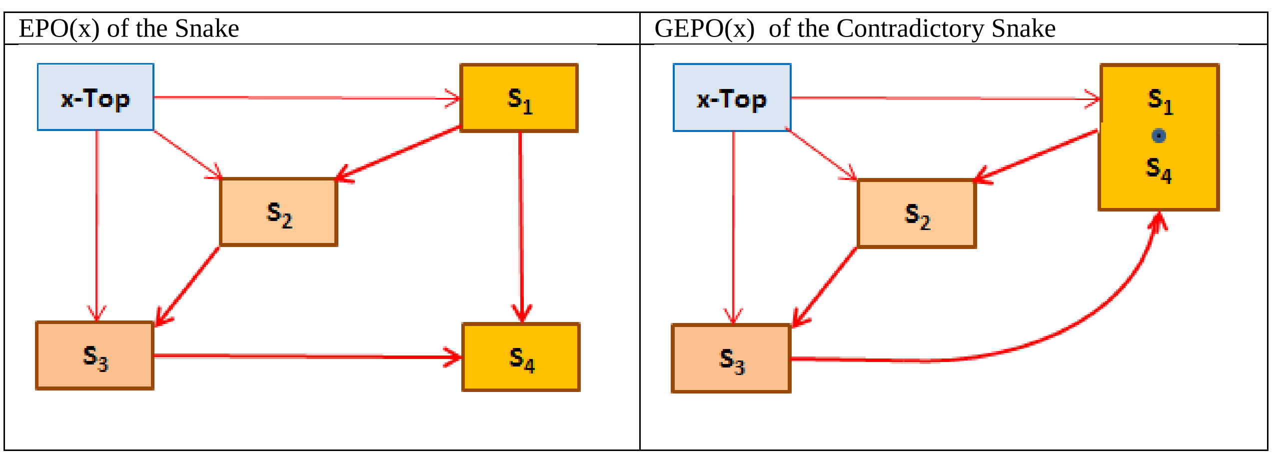
(277) Join(S_1, S_4)

$J(S_1,S_4)$	x	z	y	w
$a \vee c$	**L**	**L**	**W**	
$b \vee d$	**W**	**L**		**W**

To show that this is conservative, it suffices to enumerate the legs of each of these, and establish that the third is equal to the union of the first two. The Join Explorer of OTWorkplace imposes the ERCs of the join (277) as fixed ranking conditions on the Snake's UVT (274), and determines that S_1 and S_4 together provide the only legs in this restriction of the typology.

Turning to the MOAT, we derive the Contradictory Snake partition by merging S_1 and S_4, the head and tail of the serpentine substructure of the EPO. The crucial action takes place in the x-bigraph, GEPO(x).

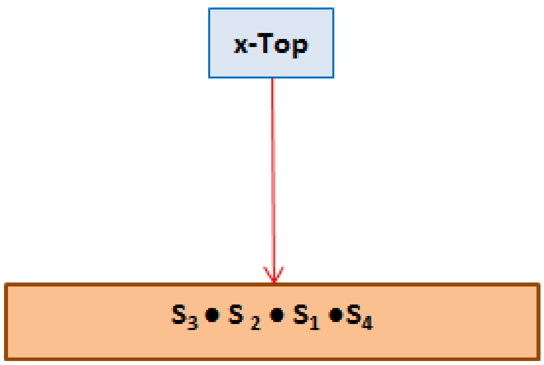
(278) **Deriving the Contradictory Snake**



Node merger induces a cycle that is defined purely in terms of order relations. From this it follows that union of grammars S_1 and S_4, though exactly equal to their join, cannot be imposed on the Snake typology to coarsen it. The result of merger partitions the entire leg set on four constraints into valid grammars, but it is a non-typology nonetheless, because its order relations do not respect the logic of OT.

The three grammars in the ouroboros-like cycle, namely $\{S_1 \bullet S_4, S_2, S_3\}$ collectively form a join that is both conservative and typological.

(279) **Joining to gain acyclicity**



Their join is the complement of x-Top, LWWW. This shows that it is possible for a set of grammars to be typologically joinable, and therefore conservatively joinable, even though they contain an internal cycle that bars them from appearing together unjoined in any typology.

6 Geometry

SubTOC

The notion of 'border point pair' arises from a natural geometry on the set of all total orders, one that has been studied since the early 20th century (Schoute 1911). Each total order is identified as a vertex of a graph, with geometric adjacency holding between points distinguished by a single adjacent transposition. The resulting object is known as the *permutohedron*. It provides a perspective on typological structure that engages a new range of concepts and analytical tools.

We introduce the permutohedron through a series of examples (§6.1). Since grammars are connected regions on the permutohedron, we can define a graphical representation of a typology, the *typohedron*, in which grammars are shrunk to single vertices that are connected to other grammars exactly when they share border point pairs (§6.2). Because the typohedron represents the adjacency structure of grammars in a typology, it supports a representation of the order and equivalence relations of any EPO, giving alternate views of the MOAT.

We conclude (§6.3) with an analysis of typologies on the permutohedron, using a notion of distance — the Riggle metric (Riggle 2012) — that allows us to show that OT grammars have a surprising geometric coherence, in that they are not only connected regions but are also convex. We introduce a constructive means of producing a shortest path between points, Recursive Constraint Promotion (RCP), which allows us to establish these unexpected properties relating geometric and algebraic structures.[44]

[44] Making us thereby, perhaps, as Descartes puts it, masters and possessors of nature.

6.1 The Permutohedron

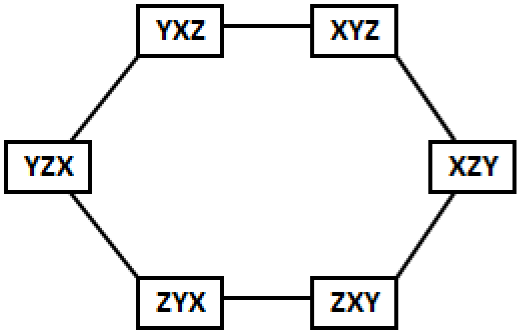
Behind the geometry of grammars lies a method of generating all permutations of a finite set. Consider all length-3 sequences, XYZ, ZYX, and so on, — permutations of letters X, Y, Z. Start out from any one such sequence and flip an adjacent letter pair, continuing on in the same fashion with the resulting permuted sequences. Take, for example, the sequence XYZ as the point of departure, using underlining to draw attention to the transposition. Flipping the first two letters yields XYZ→YXZ. Flipping the last two yields XYZ→XZY. Deal with each of the derived sequences in the same way; so that we get YXZ→YZX and XZY→ ZXY. And yet once more, accumulating all 6 permutations. The procedure of repeatedly flipping adjacent pairs always works to assemble the entirety of any finite set of permutations. This fact is familiar from group theory. Although a given permutation is defined over the entire sequence, it can be related to any other through a sequence of local actions — permutations of just two adjacent elements.

A *Cayley graph* represents this process by connecting each permutation with all the others that differ from it in exactly one adjacent flip.[45] The resulting structure can be displayed as a geometrical figure known as a *permutohedron* (originally, *permutoèdre*:[46] (coined by Guilbaud & Rosenstiehl 1963, first studied by Schoute 1911), a geometrical figure in which each permutation labels a vertex. The permutohedron on three objects X, Y, Z has 3! = 6 vertices. Each vertex is adjacent to two others. It looks like this:

[45] A Cayley Graph is a way of representing the structure of a group, here S_n, the group of permutations of *n* elements. A Cayley Graph is of a group G is based on a set of generators S for G. Each g∈G is a vertex; for each generator s∈S, g is connected to gs by a directed edge of color C_s. In the present instance, the generators are ‘swap adjacent elements at position *k*’ and the *n*–1 different colors are neutralized. The edges are not directed because $s^{-1} = s$, so that (gs)s = g, ensuring a returning edge for each outgoing edge.

[46] “Le mot *permutoèdre* est barbare, mais il est facile à retenir; soumettons le aux critiques des lecteurs.” References from Wikipedia article “Permutohedron.”

(280) **Permutohedron on sequences** of X,Y,Z



For convenient viewing, we compile the clockwise-moving flips here:

$$X\underline{YZ} \rightarrow X\underline{ZY}$$
$$\underline{XZ}Y \rightarrow \underline{ZX}Y$$
$$Z\underline{XY} \rightarrow Z\underline{YX}$$
$$\underline{ZY}X \rightarrow \underline{YZ}X$$
$$Y\underline{ZX} \rightarrow Y\underline{XZ}$$
$$\underline{YX}Z \rightarrow \underline{XY}Z$$

The *n*! total orders of an OT system of *n* constraints can be represented on a permutohedron with *n*! vertices. Each vertex is labeled with a single linear order, and its neighbors are the *n*–1 legs that differ from it by a single adjacent flip.

The 3 constraint permutohedron is a hexagon, as in ex. (280). The 4 constraint permutohedron is a 3-dimensional object, the truncated octahedron (aka omnitruncated tetrahedron). The general permutohedron that accommodates the permutations of *n* objects lives in *n*–1 dimensions (the omnitruncated regular simplex). After 4 constraints, it becomes somewhat more challenging to visualize the permutohedron, but its high degree of regularity makes easier to deal with than one might imagine at first glance.

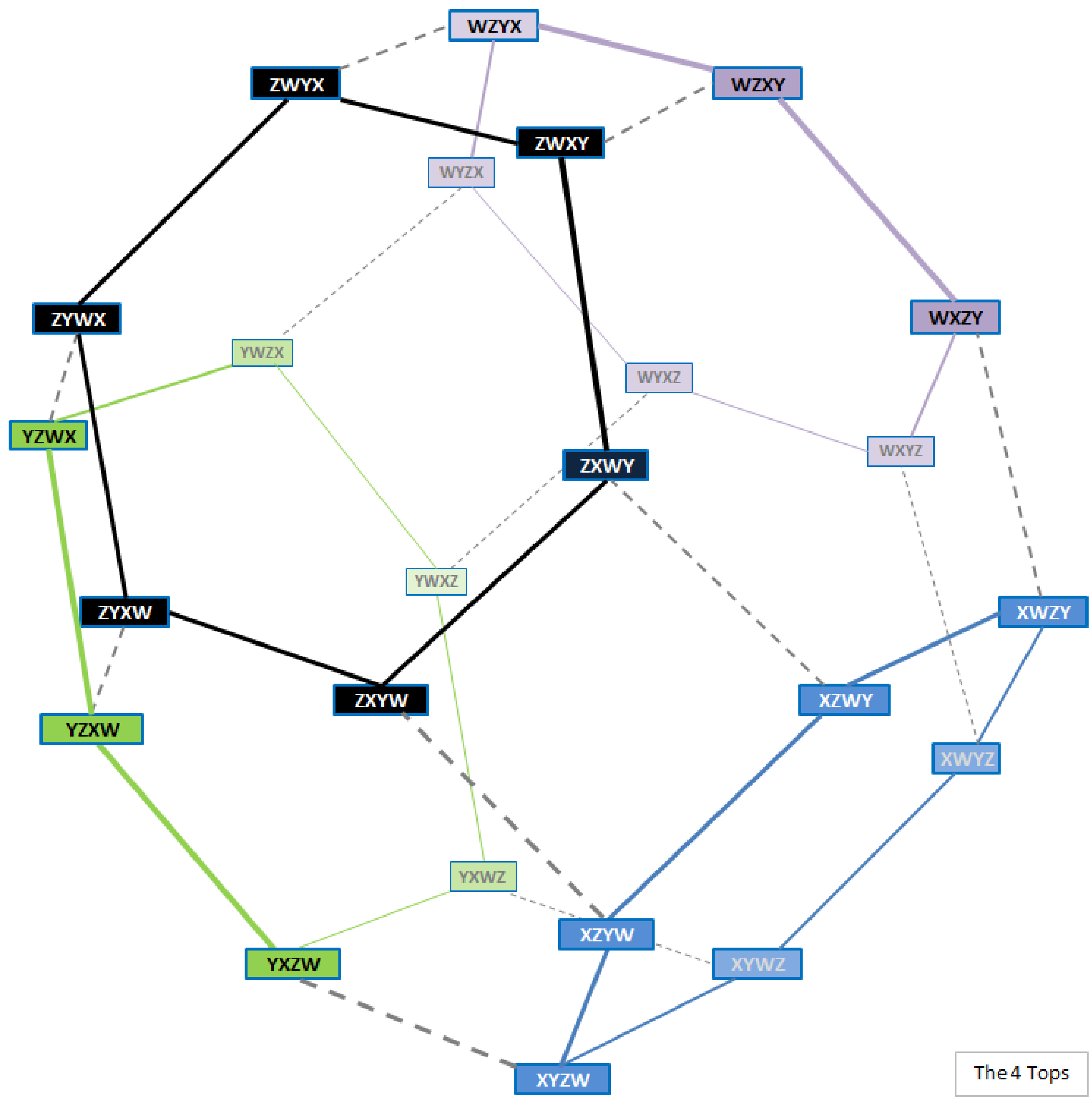
A grammar, geometrically, is a collection of vertices. It is a remarkable fact, first discovered in the 1980’s in work on antimatroids (Dietrich 1987, Riggle 2010, Merchant & Riggle 2016), that the legs of a grammar are *connected*, in the sense that for any two legs of the grammar, there is a trail of legs connected by adjacent flips that leads from one to the other, staying entirely within the grammar. A grammar, then, is a *region* of the permutohedron, where by *region* we mean a connected set of vertices. As we will see in §6.3 below, we can take this one important step farther: Riggle 2012 finds that when *distance* is defined between vertices in the right way, a grammar includes all *shortest* paths between its legs. A *typology* of ranking grammars on *n* constraints is therefore a certain kind of collection of disjoint regions, each of which is connected, that entirely covers the permutohedron.

The notion of a *border point pair* in OT, as defined above in ex. (70), rests on the same concept of adjacency as in permutation theory. A leg of a grammar is a *border point* if there is an adjacent flip of constraints that produces a leg belonging to another grammar.

The members of a *border point pair* are adjacent, and each lies in a different region. All points of the region which are not border points are *interior* points, and will be said to reside in the *interior* of the region.

Here is a view of the order-4 permutohedron. Each vertex has three neighbors.

(281) **The 4 element permutohedron** with the 4 Tops displayed



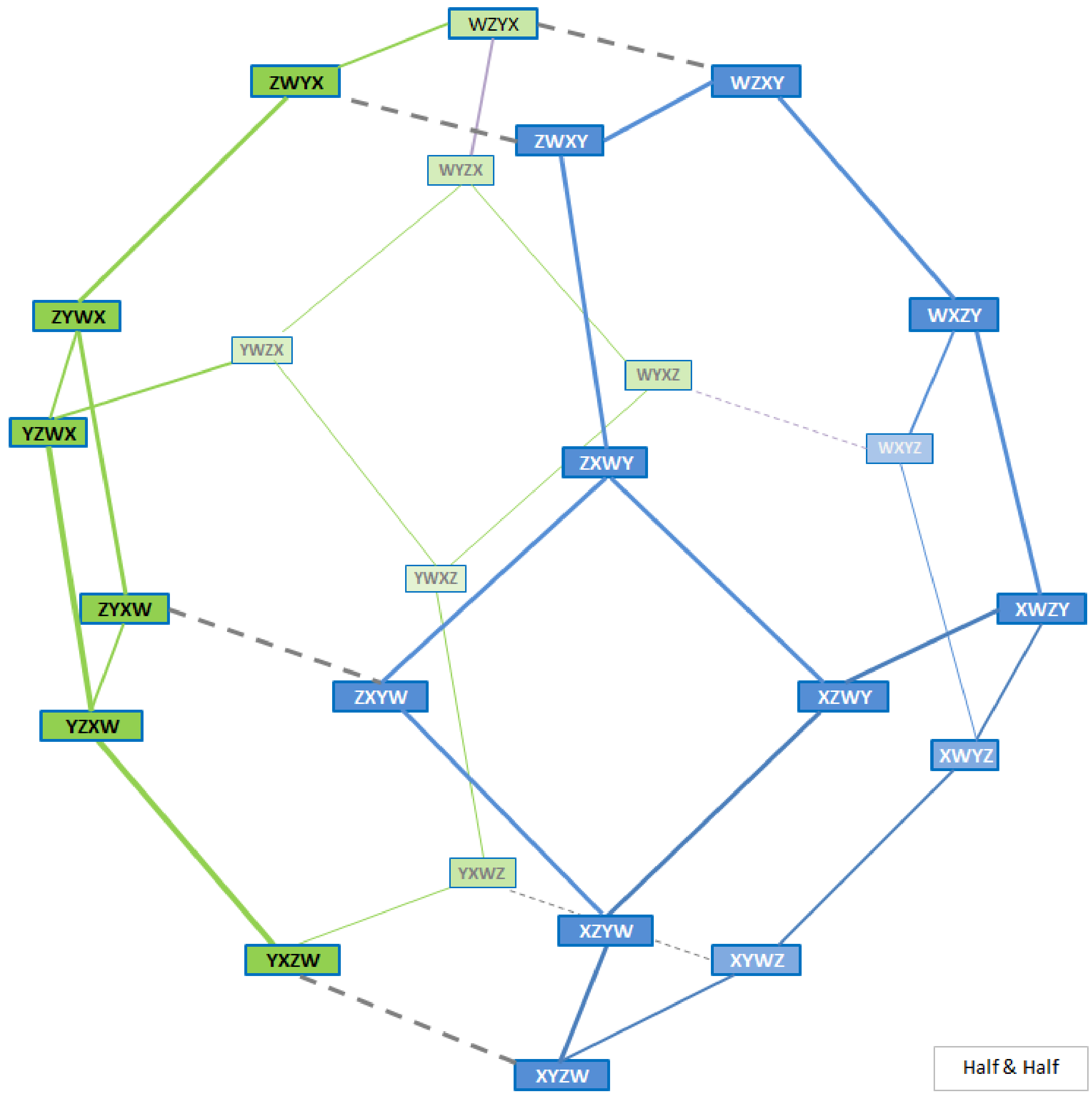
Node and edge colors distinguish the regions that correspond to the grammars of the '4 Tops', a typology comprising four of the 8 hexagonal faces of the permutohedron; each face represents a grammar in which one constraint dominates all the others. The legs of

$X \gg \{Y,Z,W\}$, which would be called X-Top in the parlance of the preceding section, are shown in blue; those of Y-Top in green, and so on.[47]

The dashed lines in the diagram connect border point pairs. For example, XYZW (bottom-most) is adjacent to YXZW (to its northwest) and belong to different languages, X-top and Y-top respectively. Every point is a border point; the regions have no interiors. Consider, by contrast, the simple 2-grammar typology that splits the permutohedron into two symmetrical halves: Γ_1 = {legs such that $X \gg Y\rangle$ and Γ_2 = {legs such that $Y \gg X$}.

[47] X-Top has 6 six legs because its defining condition allows Y,Z,W to occur in any order, so long as they are all dominated by X. Similarly, mutatis mutandis, for X-Bot. The square faces have the form $X\&Y \gg Z\&W$, etc. Each 'top' is thus an image of order-3 permutohedron. The order-4 permutohedron is obtained by arranging 4 copies of the order-3 permutohedron, stepping up one dimension to get enough room. The *n*-element permutohedron is built from *n* copies of the order $n-1$ permutohedron, just as $n!$ is defined by the equation $n! = n \times (n-1)!$ This extreme simplicity of construction provides a way to grasp what's going on even as complexity increases.

(282) **Half & Half** (X,Y)



Twelve of the 24 vertices lie in the interior of one or the other half: these are the vertices with only solid lines connecting them to their three neighbors.

Each grammar visibly occupies a *region*. In the 4 Tops, a grammar circumscribes a hexagonal face of the permutohedron. In Half & Half, each grammar embraces 2 hexagonal and 2 square faces.

The metaphor of the 'border point' is now concrete: border points reside where one region of the typology abuts another. They come in pairs, with one in one grammar, the other in its neighbor. Regions may have very large interiors, consisting of legs adjacent only to other legs of the same grammar, or no interiors at all.

Remarkably, the significant relations between grammars are completely determined at the borders. This follows because the MOAT is constructed entirely from border point pairs, and the MOAT determines all order and equivalence relations between the grammars in the typology (§3.1-2).

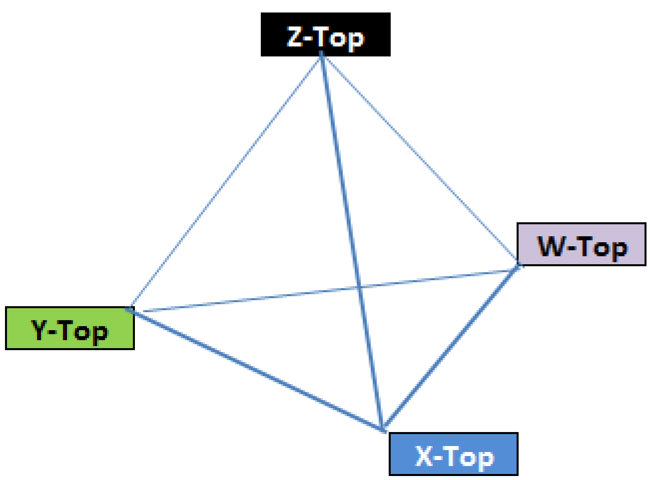
6.2 The Typohedron

As we've seen, two legs are adjacent in the order-based sense when they differ only by a single flip of sequentially adjacent constraints. On the permutohedron, they are geometrically adjacent. Going up a level, two regions_of the permutohedron may be said to be *adjacent* when they are connected by adjacent vertices: a border point pair.

Grammars partition the permutohedron, and from the partitioned permutohedron we may construct a simpler object by shrinking each region to a point while retaining its external connections. In this condensed representation, each vertex now represents an entire grammar. Simplifying yet further, we connect vertices with a single edge when their associated regions are adjacent, perhaps at many points in the permutohedron.[48] The resulting object we call a *typohedron*. The vertices in the typohedron inherit their adjacencies from the regions they represent.

Applying this construction to the 4 Tops, we see that its adjacency structure is that of the tetrahedron.

(283) **4-Tops typohedron**



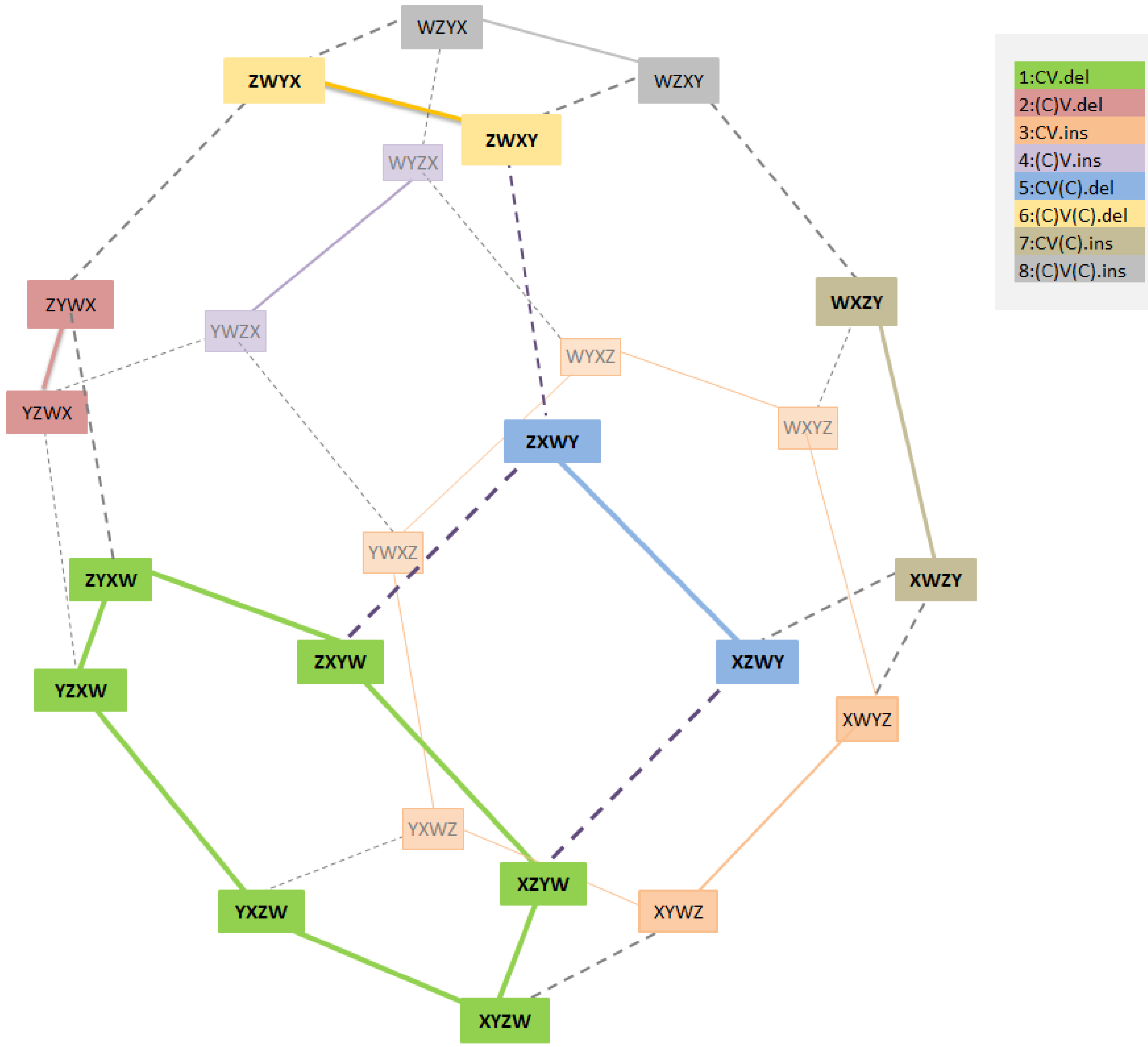
The Half & Half typohedron poses no challenges to visualization:

[48] This is the usual way that (geographical) maps are transformed into graphs so that certain of their properties can be studied, like how many colors it takes to distinguish their regions.

(284) **Half & Half typohedron**

Returning to the main theme of our analysis, we first portray the EST typology as a partition of the permutohedron, using again the convention that solid lines connect the vertices within a region and dashed lines connect regions across border points.

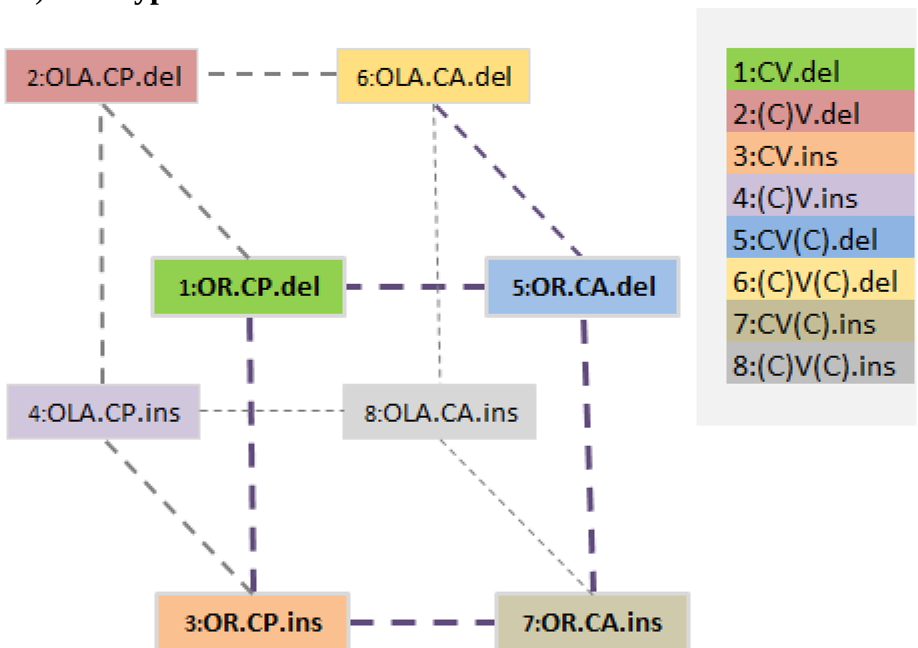
(285) **EST partition of the permutohedron**



Key: X = m.Ons
Y = m.NoCoda
Z = f.dep
W = f.max

Reduced to the typohedron, the EST takes the form of a cube.

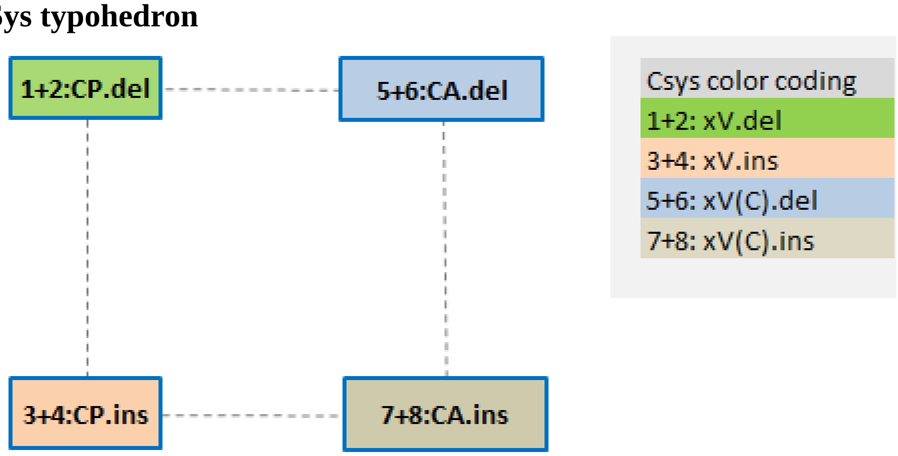
(286) **EST typohedron**



The front face nodes correspond to the OR languages with syllables CVx; the rear face, the OLA languages with syllables (C)Vx , where x is restricted at the language level. The top face contains the deleters; the bottom face, the inserters. The left face has all those that disallow codas, with syllables xV, descriptively CP; the right face, all those that permit codas, xV(C), descriptively CA.

The typohedron for the CSys shows the effect of merging each vertex on the front face (OR) of diagram (285) with its neighboring vertex on the rear face (OLA), collapsing the cube to a square. Here we arbitrarily retain front-face colorations.

(287) **CSys typohedron**



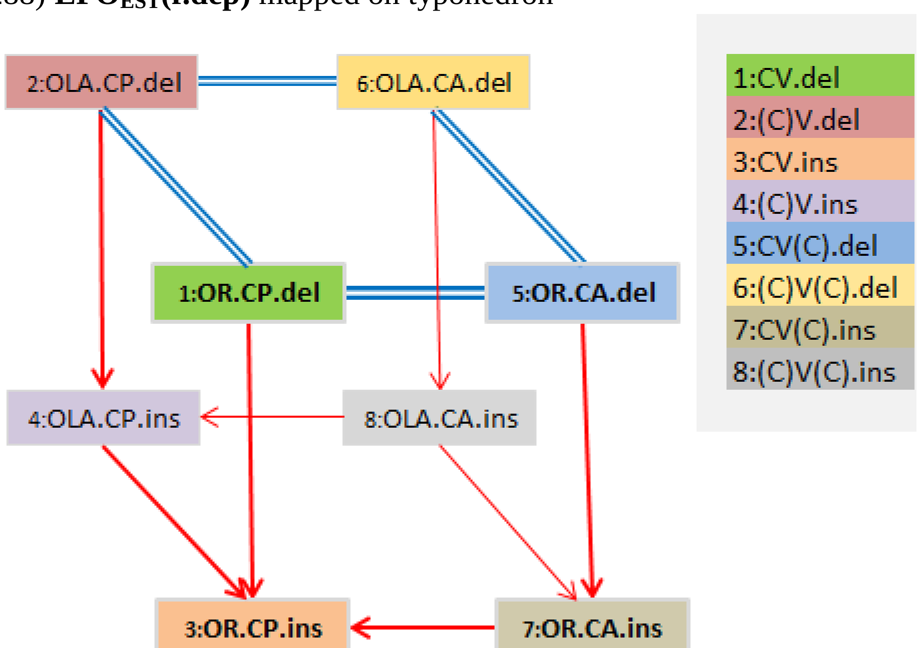
The connectivity of the typohedron is reflected in the base orders that inhabit its EPOs (§3.1). Border point analysis examines every border point pair and returns the concomitant privileged order relations. If any two grammars have a privileged order in any EPO, they are adjacent in the typohedron. In terms of our graphical conventions, the typohedron may be assembled from the MOAT by marking as adjacent any two nodes connected by a red arrow in some EPO. This algorithm is used in OTWorkplace to produce the typohedron.

The typohedron represents adjacency of grammars, which is relevant to typological joinability. We're never going to be able to join 1:CV.del and 6:(C)V(C).del conservatively, regardless of what other relations hold between them, because they are separated by other nodes. Typohedron adjacency provides a necessary but not sufficient condition for joinability, both conservative and typological.

In the EST, for example, as shown in ex. (256), §5.1, it is not typologically valid to merge along the vertical dimension (faithfulness) of typohedron (286), thereby generalizing away from the distinction between *del* and *ins*. This is true despite the fact that the would-be joinards are neighbors. To see the cause of the failure in terms of typohedral structure, we deploy EPO-style annotations, where edges are rendered as blue double line if the connected vertices are equivalent in the EPO and as red arrows if an order relation exists between them.

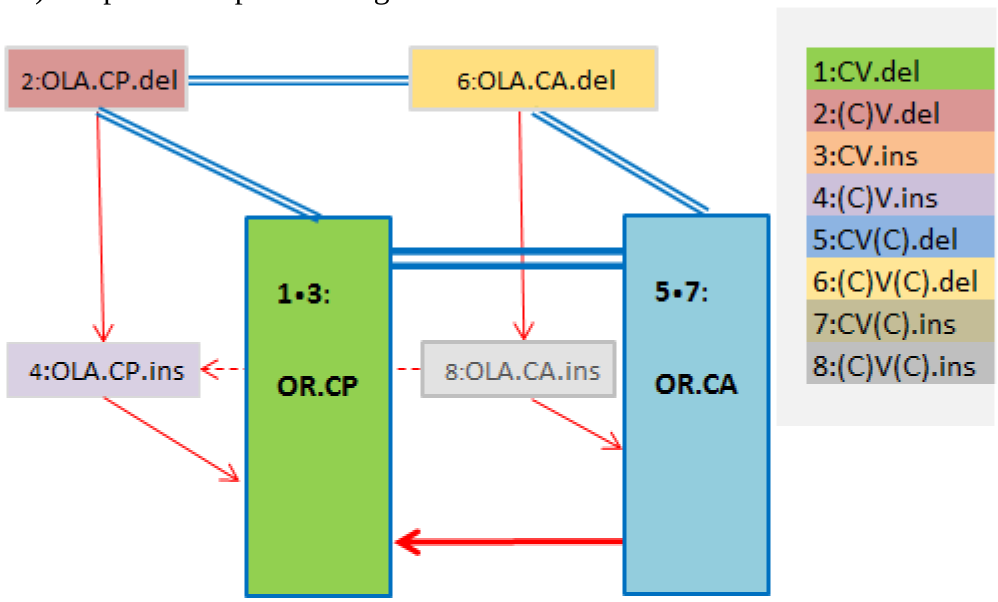
Consider the f.dep EPO of the EST, mapped out on the EST typohedron.

(288) **EPO$_{\text{EST}}$(f.dep)** mapped on typohedron



The front face contains the four OR grammars. Vertical merger of front-face elements on the left gives the CP (coda-prohibited) subset of OR languages: all the CV languages. On the right, it gives the CA (coda-allowed) subset: all the CV(C) languages. When we construct these mergers on the EPO-annotated typohedron, the obstruction to typological join emerges as clearly as in the EPO itself.

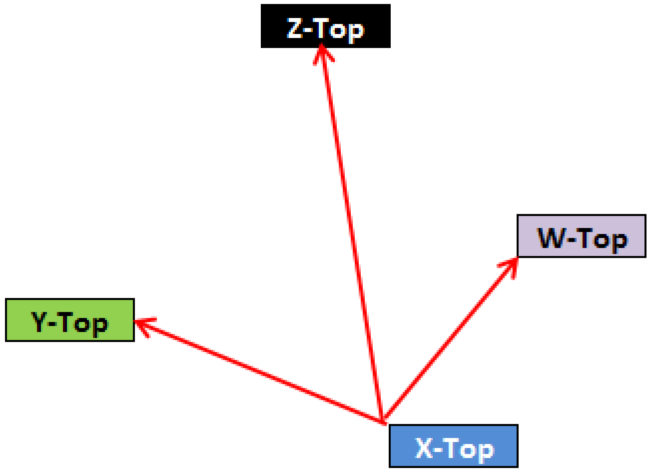
(289) f.dep EPO map with mergers across del/ins divide



Using familiar EPO-type reasoning, we observe that the OR.CP node cannot be both equivalent to OR.CA and ordered below it on f.dep. More elaborate cycles may also be observed on the left and right faces of the merged typohedron.

Applying the same technique to the 4-Tops typohedron (283), we obtain the following map of the X-EPO.

(290) **X-EPOhedron of the 4 Tops**



X-Top names the grammar $X \gg \{Y, Z, W\}$. In X-Top, the choice between languages is decided at X, the first constraint in each of its 6 legs, which allows only one language through. We name the languages L_X, L_Y, L_Z, L_W after the constraint that allows them to pass, losing all companions. Any leg of X-Top produces the following filtration, in which decision is immediate.

(291) **Filtration by X in X-Top**

$$\{L_X, L_Y, L_Z, L_W\}] \rightarrow_X \{L_X\}$$

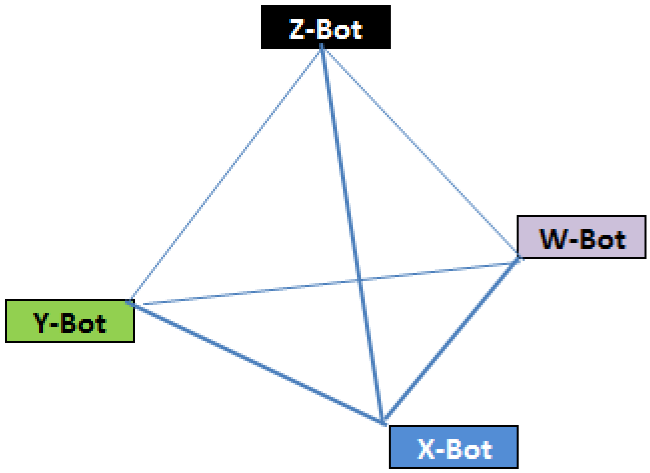
Therefore X may assign any values whatever to L_Y, L_Z, L_W so long as they are greater than the value it assigns to L_X.

The EPOs of the 4 Tops MOAT are all structurally identical and differ only in the grammar that is ordered above all the others. From the X-EPO (290), it is therefore possible to see that any merger whatever will be acyclic and therefore typologically valid. There is simply no possibility of creating a cycle via node merger.

The situation with the 4 Bots is strikingly different. The 4 Bots typohedron is isomorphic to that of the 4 Tops, as may be intuited from the fact that the four hexagonal Bot faces lie on the permutohedron in the same relation to each other as the four hexagonal Top faces. Bot and Top may be swapped by exchanging each face of the permutohedron with the one that is opposite to it. We can achieve this effect with the 'antipodal map', globally reversing the internal order of the permutations, as e.g. $XYZW \mapsto WZYX$. Border point pairs transform into border point pairs, so all connectivity between regions is preserved.

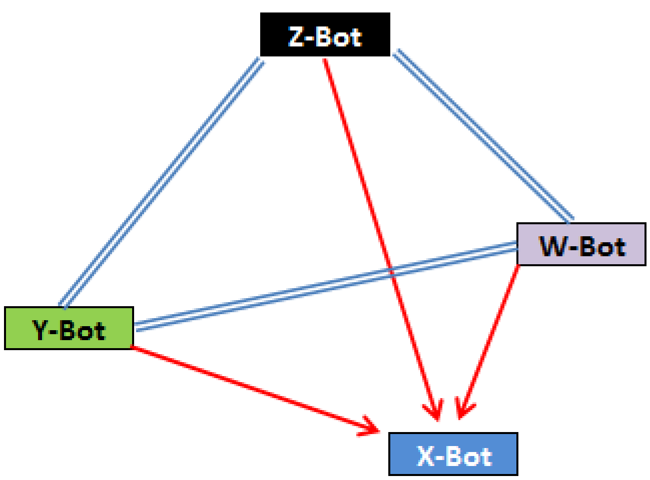
The 4 Bots typohedron therefore looks like this:

(292) **The 4 Bots**



EPO Relations do not simply echo those of the 4 Tops. In addition to the expected reversal, whereby there is one node in each EPO that receives a set of incoming arrows rather than sponsoring a set of outgoers, there is a set of equivalences to contend with. Here is the X-EPO of the 4 Bots.

(293) **X-EPOhedron of the 4 Bots**



We find equivalence rather than noncomparability because the antipodal map exchanges suffix and prefix in the border point pairs. All border crossings in the 4 Tops involve transpositions of the first 2 constraints: for example stepping off Y-Top to arrive at W-Top via the pair {<u>YW</u>ZX, <u>WY</u>ZX}. By contrast, transiting from one Bot to another always involves the last two constraints in the order. Thus, the pair {XZ<u>WY</u>, XZ<u>YW</u>} takes us from Y-Bot to W-Bot. But here, antipodally, both X and Z fall in the prefix, and this ensures that Y-Bot and W-Bot are equivalent in EPO(X) as shown.

The filtration patterns in the two typologies are therefore quite different. In X-Top, the constraint X immediately accepts L_X and forsakes all others, ending the nontrivial part of the filtration. In the 4 Bots, filtration runs through 3 nontrivial stages before settling. Let L^*_X be the language *rejected* by constraint X, and name the others similarly.[49] Crucially, language L^*_X is the one deemed optimal by the grammar X-Bot, L^*_Y by Y-Bot, and so on.

To see how filtration unfolds in the 4 Bots, observe that the constraint X not only *rejects* L^*_X but accepts all the others. The same is true, mutatis mutandis, for constraints Y, Z,W with respect to L^*_Y, L^*_Z, and L^*_W, respectively. This can easily be seen in a UVT.

(294) **4 Bots UVT**

4 Bots	X	Y	Z	W
L^*_X	1	0	0	0
L^*_Y	0	1	0	0
L^*_Z	0	0	1	0
L^*_W	0	0	0	1

Here, for example, is the filtration pattern of the leg YZWX.

(295) **Filtration pattern** of the leg YZWX in X-Bot

$$\{ L^*_X, L^*_Y, L^*_Z, L^*_W \} \to_Y \{ L^*_X, L^*_Z, L^*_W \} \to_Z \{ L^*_X, L^*_W \} \to_W \{ L^*_X \} \to_X \{ L^*_X \}$$

To see the equivalences imposed by X from the filtration point of view, we look at legs in which languages pass through X together. Any leg with X in first position will show the pattern: we follow XYZW.

(296) **Filtration pattern** for the leg XYZW in W-Bot

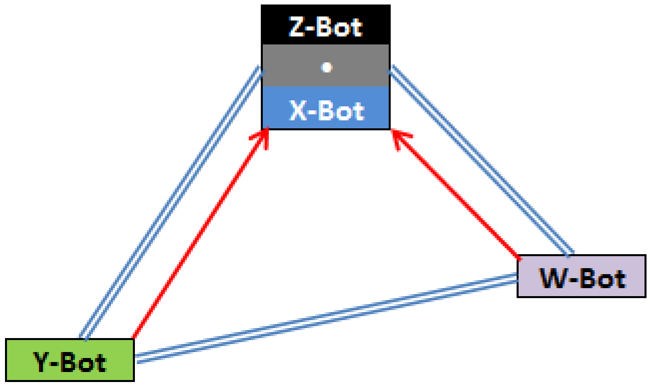
$$\{ L^*_X, L^*_Y, L^*_Z, L^*_W \} \to_X \{ L^*_Y, L^*_Z, L^*_W \} \to_Y \{ L^*_Z, L^*_W \} \to_Z \{ L^*_W \} \to_W \{ L^*_W \}$$

Observe that L^*_Y, L^*_Z, L^*_W all pass through X together in the first step, ensuring their equivalence in EPO_{4Bots}(X), as shown in the X-EPOhedron (293).

The system of equivalences and orders entails that *no* two grammars of the 4 Bots may be typologically joined without creating a cycle in some EPO. X-Bot, for example, cannot be joined with any other grammar because of the behavior EPO_{4Bots}(X). Merging X-Bot and Z-Bot, for example, produces the following cyclic monstrosity.

[49] L^*_X can be given the violation profile (1,0,0,0), L^*_Y given (0,1,0,0), and similarly for L^*_Z and L^*_W.

(297) **Merger of X-Bot and Z-Bot** in the X EPO.



6.3 Putting the Metric in Geometric

SubTOC

The permutohedron is a geometric object in the informal sense that is used of any collection of points and edges — any graph. But geometry in the mathematical sense requires a notion of distance. Riggle has asserted a way of assigning distance between vertices of the permutohedron that characterizes the notion of grammar in metric terms: namely, that the shortest path between any two points in grammar always lies entirely within that grammar, a property known as 'geodesic convexity'. In the following sections, we first introduce the Riggle metric, showing how it works through examples (§6.3.1). We then go on to discuss its relationship to the UVT and the MOAT (§6.3.2), showing that the Riggle metric may be derived from the UVT for the Discrete Typology and vice versa, but that the MOAT is not derivable from the Riggle metric in the general setting and therefore remains irreplaceable as a mode of characterizing typologies. In the course of proving basic claims about the Riggle metric (§6.3.3), we provide an algorithm, Recursive Constraint Promotion (RCP), which produces a shortest path between arbitrary points. This allows us to establish the geodesic convexity of OT grammars.

6.3.1 The Riggle metric, Geodesic convexity, and the Join

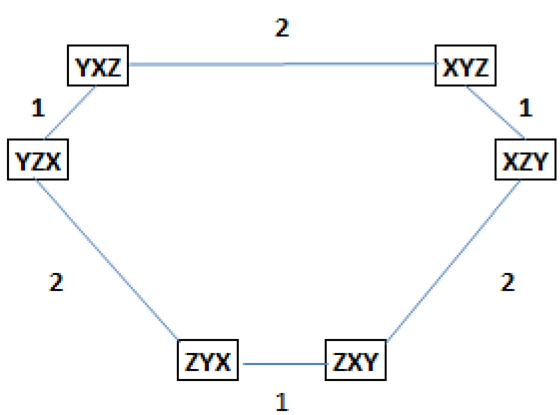
Jason Riggle (2012) has announced a result of great relevance to the present enterprise. In addition to the established techniques for construing orders in terms of adjacency, adopted here, he introduces a notion of *distance* between vertices on the permutohedron. The distance between two adjacent permutations P<u>XY</u>Q and P<u>YX</u>Q is given by |Q|+1, where |Q| is the number of elements in the sequence Q. The effect, qualitatively, is that permutations of higher-ranked constraints result in greater distances than permutations of lower-ranked constraints.

In a 3 element system on {X, Y, Z}, the Riggle distance d_R between <u>XY</u>Z and <u>YX</u>Z is 2 because |Z|+1 = 2. By contrast, d_R(X<u>YZ</u>, X<u>ZY</u>) = 1, because |Q| = 0, since Q is empty for this pair.

The notion generalizes in a natural way to define the distance between two arbitrary points, not necessarily adjacent, because we are guaranteed the existence of a path between them that is made up of adjacent points. Consider any such path, sum up the distances between the adjacent points along it to get the length of the path, and regard the shortest such path as giving the distance.

The geometric effect of imposing the Riggle distance on the permutohedron can be seen directly in the 3 constraint case, shown below. The scale is slightly exaggerated for visual clarity.

(298) **3C Permutohedron with Riggle distances**



With three elements, there are always just 2 paths between any two points, excluding paths that visit a point more than once. For example, from YZX at the far left to XZY on the far right, there's the up-and-over path π_1 = ⟨**YZX**, YXZ, XYZ, **XZY**⟩, and there's the down-and-under path π_2 = ⟨**YZX**, ZYX, ZXY, **XZY**⟩. Writing $|\pi|$ for the length of a path $|\pi|$, we may easily calculate the length of these paths from the diagram:

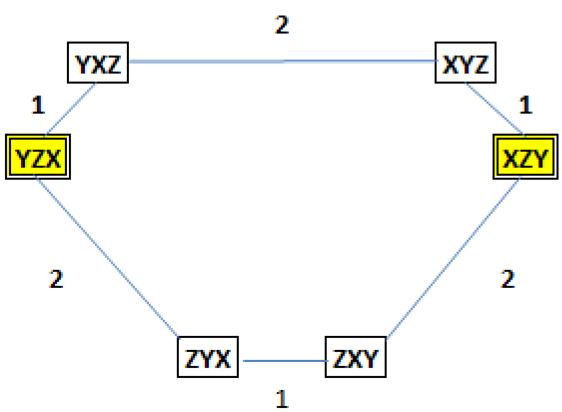
$|\pi_1| = 1+2+1 = 4$

$|\pi_2| = 2+1+2 = 5$
The Riggle distance between YZX and XZY is therefore 4.

Riggle's main finding is that every shortest path between two points in a grammar always lies *within* that grammar. A *geodesic* is the shortest path between two points in a general space that has distance defined on it: a grammar (and equivalently, an antimatroid) is then *geodesically convex* under the Riggle metric.[50] This suggests that it might be possible to establish a new way of characterizing what a grammar is: a convex region of the permutohedron.

The consequences of geodesic convexity can be seen in our example. Consider the leftmost and rightmost points in the 3C permutohedron as we have rendered it.

(299) **3C permutohedron with distances and selected points**



Suppose we wish to find the smallest grammar that contains both YZX (leftmost) and XZY (rightmost). We have computed that the shortest path π_1 between them has length 4 and consists of {YZX, YXZ, XYZ, XZY}. Since this also contains the shortest paths between all of the legs along the path, it is a grammar. It is the ERC grammar {WWL}: "X or Y dominates Z."

What then of the other path, π_2, of length 5? By geodesic convexity, any grammar containing the legs of π_2 must also contain those of π_1 because π_2 contains the nodes {YZX, XZY}and therefore the shortest path between them: π_1. But π_1 and π_2 together exhaust the nodes of the {X,Y,Z} permutohedron. The smallest grammar containing π_2 is therefore the trivial grammar that includes every leg. This grammar has no restrictive ranking conditions on it at all, and can be rendered by any ERC that contains no L's.

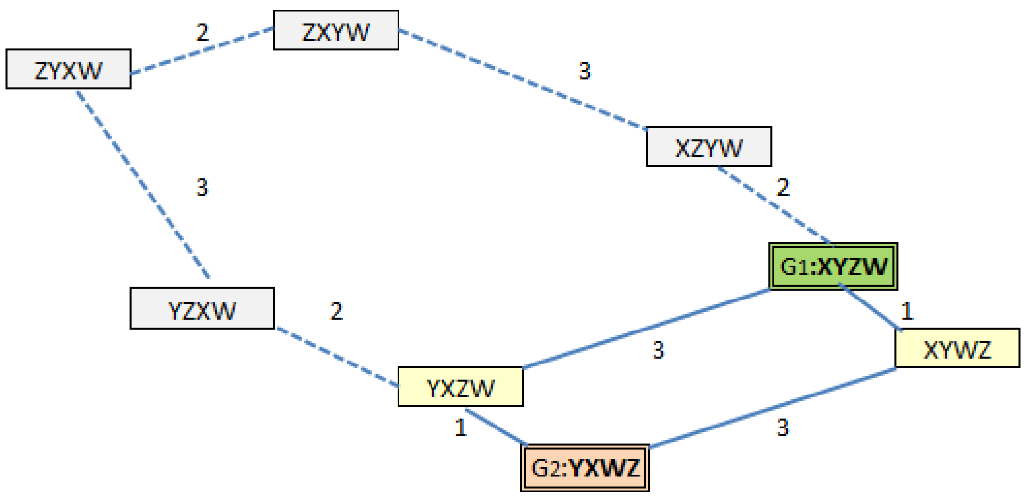
The join of two ERC grammars (§5.1) is the smallest grammar that contains both (Merchant 2008, 2011). The geodesic convexity property gives us another perspective on

[50] See for example "Geodesic Convexity" in Wikipedia.

what's happening when two languages are joined. Since the result is a grammar, it must contain all shortest paths between pairs of its legs. Among these legs are those coming from *either* of the two joinards. It must be that the join computes an ERC description of the set of all legs that are needed to render the union of the joinards into a valid grammar: all shortest paths between any legs in either, plus all shortest paths between the legs in those paths, and so on.

Our example shows a nonconservative join absorbing the entire set of legs in the typology, but the broader generalization is that the the join expands on the union only to the point where a valid grammar is reached, where all shortest internal paths are included. To see this subtler, less radical effect, it is sufficient to examine a simple 4 constraint case. Consider any typology on {X,Y,Z,W} which has among it grammars G_1 = {XYZW} and G_2 = {YXWZ}. They sit across from each other on a quadilateral face of the permutohedron. A relevant patch of their local environment looks like this, drawn to reflect Riggle distances.

(300) **G_1= {XYZW} and G_2 = {YXWZ} on the measured Permutohedron** (local view)



Suppose we wish to join G_1 (green) and G_2 (tan). We must include all vertices on any minimal-length path between them. Heading left from G_1 in the region shown here, we encounter two paths from G_1 to G_2 which do not revisit any vertices.

π_a = **XYZW** – YXZW – **YXWZ**	Length = 4 = 3+1
π_b = **XYZW** – XZYW – ZXYW – ZYXW –YZXW – **YXZW**	Length = 13 = 2+3+2+3+2+1

The minimal path π_a has length 4, while π_b has length 13. It follows that on the left side of G_1 only the leg YXZW will be included in the join G_1+G_2. But since the right descending path from G_1 to G_2 also has length 4, the leg XYWZ must also be part of G_1+G_2. From metric considerations alone we can deduce, without further calculation, that G_1+G_2 = {XYZW, YXZW, XYWZ, YXWZ}. This join is nonconservative but not radically so.

The example also shows that under the Riggle metric, there may be more than one shortest path between two points. Compare the fact that there are many shortest routes from pole to pole on a sphere. Geodesic convexity means that all the legs along every shortest route must be included in the join. It is striking that this calculation can be carried out in ERC space via the join operation with no mention of points or distances. Conversely, joining languages, though originally conceived as a purely algebraic operation, can be determined with this metric.

The join of two grammars is independent of any typology in which the joinards may be embedded. Whether joining grammars within a given typology leads to another valid, coarser typology is, as we've seen in the Split Bots and the Contradictory Snake examples of section §5.2, contextually determined. At the typological level, the order and equivalence requirements of the MOAT are inescapable, and they may obstruct even a conservative join. We know from the Contradictory Snake (§5.3) there are conservatively joinable sets of disjoint grammars which cannot be embedded in any typology unjoined.

The specific metric proposed by Riggle and discussed here is one of many assignments of value to edges of the permutohedron which will carry the geodesic convexity property. Compare the fact that the standard Euclidean metric of daily life is unchanged if we double its value, or change from inches to centimeters. Metrics of the Riggle class are even more flexible, in that only the relative order on paths imposed by their length is important, not for example the ratio of distances from one object to another, which is constant in the Euclidean metric no matter what units we express it in.

The essence of the Riggle metric lies in defining distance based on the position of the adjacent flip that distinguishes neighboring points, increasing geometric distance as the flip recedes from the end of the sequence. This gives us convexity, but it does not give us clustering, in that the legs inside a grammar are by no means guaranteed to be closer to each other than to legs in another grammar. It is an interesting project to redefine the Riggle metric as a pseudometric relativized to a typology so that all points within a region are assigned 0 distance from each other. (In a metric per se, points at zero distance must be the same point; in pseudometric, this condition — the coincidence axiom — is dropped.[51]) The idea is that only certain flips would contribute distance, those within border point pairs, which determine the ranking requirements of grammars. On this conception, in the typology we have named Half & Half above, the only flip contributing distance would be the one exchanging adjacent $X \gg Y$ and $Y \gg X$. In the 4 Tops, the only flips contributing distance would be those between first and second position in the orders. We leave this idea for future contemplation.

[51] See for example the article "Metric" in Wikipedia.

6.3.2 The Riggle metric, the UVT, and the MOAT

We now have two numerical representations of a typology: the UVT, with its violation values licensed by EPO structure, and the partitioned permutohedron, with its Riggle distances between vertices. Here we investigate their relationship. First, we show how the Riggle metric may be derived from the violation value system. This construction also allows us to go the other way, deriving the UVT for the Discrete Typology, in which every grammar has a single leg, from the Riggle Metric. Then, moving from the Discrete Typology to the general typological landscape, we find that a parallel method of deriving UVTs from the Riggle metric is not in the offing, because of the way that relations between grammars develop through order-based interactions. This confirms the irreducible centrality of the MOAT in typological analysis.

The Discrete Typology provides an exact image of the permutohedron: each vertex represents one grammar. As always, the Discrete Typology will have many UVTs, but one of them is conspicuously minimal, in the sense that it uses the smallest possible integers for its values. More precisely put, the largest integer that it uses is as small as possible. Furthermore, this UVT is unique up to renaming of rows and columns. For the Discrete Typology $\mathrm{DT}^{(n)}$ on n constraints, we will call its minimal UVT $\mathrm{U}_0^{(n)}$. Let's begin by settling the layout of $\mathrm{U}_0^{(n)}$.

We may view a UVT from two different angles, horizontally or vertically, as it were. Consider first the n-length row vectors over the integers $\{0,1,\ldots, n\text{–}1\}$, where each vector contains one instance of each of these integers. Assign each component of the vector to a different constraint in some arbitrary but fixed fashion. Each vector then represents a different permutation of the first n non-negative integers, so that there are $n!$ vectors in total. There are also $n!$ grammars in the Discrete Typology and therefore $n!$ rows in any of its UVTs. Assign each vector to one of languages as its violation profile. We claim that this gives us the UVT we're looking for.

To see this, consider the selection process in the VT we have defined. The language L_k will be selected by the leg λ_k that orders the constraints in such a way that they assign strictly increasing values to L_k.

The highest ranked constraint in λ_k, call it C_1, assigns 0 to L_k, so L_k survives filtration by it. This means that every survivor of C_1 is assigned 0. The next highest ranked constraint in λ_k, call it C_2, assigns 1 to L_k. But no survivor of C_1 can have a 0 in C_2: this would put two 0's in a violation profile, an impossibility by construction of the profiles. Therefore, 1 is the smallest value that C_2 assigns to any survivor of C_1, and L_k also survives C_2. The same argument replicates, mutatis mutandis, down the hierarchy of λ_k, leaving L_k as the only survivor. The argument is generic: it follows that each language will be chosen by some leg, so that every language of the DT is an optimum of this VT. Furthermore, each leg chooses a distinct language. This establishes that we have a UVT for the DT on n constraints.

Now shift perspective to the vertical. Consider any constraint X and some n-constraint leg that it initiates, call it $\lambda_1 = \mathrm{XYZ}\ldots\mathrm{W}$. Consider the set of grammars, i.e. single legs, obtained by shifting X downward by a series of adjacent flips:

(301) **Leg Sequence**

$\lambda_1 = \underline{\mathbf{X}\mathrm{Y}}\mathrm{Z}\ldots\mathrm{W}$

$\lambda_2 = \underline{\mathrm{Y}\mathbf{X}}\mathrm{Z}\ldots\mathrm{W}$

$\ldots$

$\lambda_{n-1} = \mathrm{YZ}\ldots\underline{\mathbf{X}\mathrm{W}},$

$\lambda_n = \mathrm{YZ}\ldots\underline{\mathrm{W}\mathbf{X}}.$

Each pair $\{\lambda_i, \lambda_{i+1}\}$, $1 \le i \le n-1$, is a border point pair. Therefore in EPO(X), these grammars form a strictly ordered sequence $\lambda_1 <_X \lambda_2 <_X \ldots <_X \lambda_{n-1} <_X \lambda_n$. Instantiating these relations numerically in a UVT requires n integers. The set $\{0,1,\ldots, n-1\}$ offers the smallest possible non-negative integers that will do the job. This shows that violation profiles used above to create a UVT for $\mathrm{DT}^{(n)}$ cannot be improved upon by using a smaller maximum integer. We conclude that the UVT is unique, up to renaming the constraints and candidates, and is therefore the promised $\mathrm{U}_0^{(n)}$.

With 3 constraints at play, the UVT looks like this, labeling each language with mention of the leg that selects it.[52]

(302) **UVT for DT$^{(3)}$**, the 3C Discrete Typology

$\mathrm{U}_0^{(3)}$	X	Y	Z
$\mathrm{L}_{\mathbf{X}\mathrm{YZ}}$	0	1	2
$\mathrm{L}_{\mathrm{Y}\mathbf{X}\mathrm{Z}}$	1	0	2
$\mathrm{L}_{\mathrm{YZ}\mathbf{X}}$	2	0	1
$\mathrm{L}_{\mathbf{X}\mathrm{ZY}}$	0	2	1
$\mathrm{L}_{\mathrm{Z}\mathbf{X}\mathrm{Y}}$	1	2	0
$\mathrm{L}_{\mathrm{ZY}\mathbf{X}}$	2	1	0

Returning now to the n-length sequence in (301), consider how values must be assigned to the languages in $\mathrm{U}_0^{(n)}$. From ex. (301) and the ordering that it implies, we have $\mathrm{X}(\mathrm{L}_1) = 0$, $\mathrm{X}(\mathrm{L}_2) = 1$, … , $\mathrm{X}(\mathrm{L}_n) = n-1$. For any L_k, we have $\mathrm{X}(\mathrm{L}_k) = k-1$, which is exactly the serial position of X in the constraint order, minus 1. This pattern may be easily seen for $n = 3$ in ex. (302).

By comparison, the Riggle distance between any pair $\{\lambda_k, \lambda_{k+1}\}$ is 1 plus the length of the sequence following X in λ_{k+1}. The length of that suffixal sequence is given by $n - (k+1)$.

[52] It may be observed that the violation profile of a language is the inversion of the permutation that selects it, enumerating *positions* in the sequence starting from 0.

$$\begin{aligned} d_R(\lambda_k, \lambda_{k+1}) &= 1 + n - (k+1) \\ &= n - k \\ &= (n-1) - X(L_k)\ . \\ &= (n-1) - Y(L_{k+1}) \end{aligned}$$

To state it more genererally, consider any border point pair {PXYQ, PYXQ} and let the constraints assign minimal values in the way described. Then we have:

(303) **Riggle distance between adjacent nodes** in $DT^{(n)}$. For |PXYQ| = |PYXQ| = n.

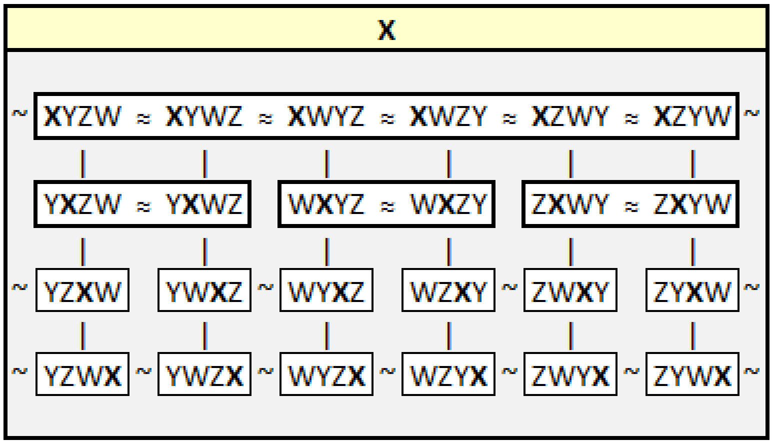
$$\begin{aligned} d_R(\text{PXYQ}, \text{PYXQ}) &= (n-1) - X(L_{PXYQ}) \\ &= (n-1) - Y(L_{PYXQ}). \end{aligned}$$

This gives us complete interconversion between the Riggle distances between adjacent nodes and their correlated evaluations in $U_0^{(n)}$. The Riggle metric giving distances between arbitrary nodes is projected from these local distances: the Riggle metric has therefore been successfully interpreted in violation values for $DT^{(n)}$, as promised. Conversely, by juggling the equations of (303), the violation values in $U_0^{(n)}$ can be stated in terms of distances.

At this point, it is natural to imagine that the Riggle distance between points can be generalized to a distance between regions that allows us to derive a UVT in way that parallels or extends the results of (303). But regions interact orderwise in a way that will dash this hope. The first hint of a red flag can be seen in the gross shape of the Snake typology, which imposes in EPO(X) a linear order on its 5 grammars. This will require 5 distinct integers to instantiate it, as indeed is visible in UVT (274). But there are only 4 Riggle distances in a 4-constraint permutohedron.

To see the source of this effect, it is useful to lay out a flattened version of the permutohedron, running from X-initial orders (top) to X-final orders (bottom), with each vertical thread following the pattern of the leg sequence in (301), where X transposes stepwise through the sequence.

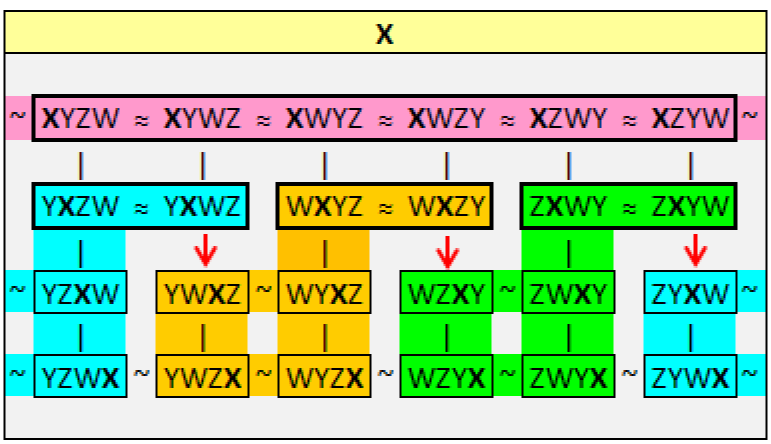
(304) **X-view of Permutohedron**



Nodes are connected by ≈ when adjacent and X-equivalent, by | when adjacent and ordered, and by ∼ when merely adjacent, with the order $<_X$ running from top to bottom. Note that leftmost and rightmost nodes are connected in all except the 2nd row from the top. To aid with visual parsing, X-equivalent nodes are boxed in heavier lines.

It is instructive to portray the Contradictory Snake in this format, replacing the crucial vertical links with arrows. Let's call the grammars after the colors that mark them out.

(305) **The Contradictory Snake**



The red arrows indicate the crucial ordering interaction. From left to right, we have

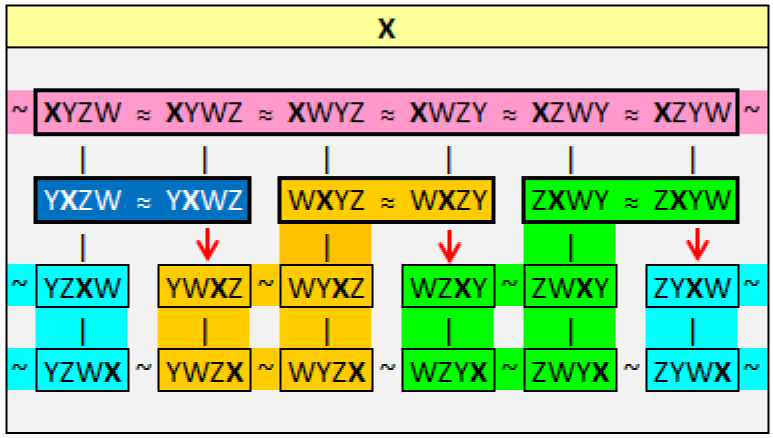
Blue $<_X$ Orange

Orange $<_X$ Green

Green $<_X$ Blue.

This results in the cycle that renders the Contradictory Snake a non-typology. The problem is easily fixed by severing blue into two parts, interrupting the cycle and yielding the Snake of UVT (274).

(306) **The Snake**



We can now assign coherent X-values to the (languages corresponding to the) grammars, based on the ordering Pink$<_X$ DarkBlue $<_X$ Orange $<_X$ Green $<_X$ Blue.

(307) **X-values for the Snake**

Language	X
Pink	0
DarkBlue	1
Orange	2
Green	3
Blue	4

The problem is now immediately apparent: the Riggle distance is the same between the border points of pairs of languages linked by the red arrows in diagram.

d_R(DarkBlue, Orange) = d_R(Y<u>XW</u>Z, Y<u>WX</u>Z) = 2
d_R(Orange, Green) = d_R(W<u>XZ</u>Y,W<u>ZX</u>Y) = 2
d_R(Green, Blue) = d_R(Z<u>XY</u>W, Z<u>YX</u>W) = 2.

Thus there is no hope of translating the local distance between a region and its neighbors into a useful violation value. Any attempt to construct the violation values must be non-local, taking account of the transitivity of order. To see what this amounts to, one must turn to the EPO. But metric considerations are then uninformative and add nothing to what the EPO already says.

6.3.3 The Geodesic Convexity of Grammars

In this section we prove the major results asserted in the previous sections. Starting with the notion of adjacency used throughout, we define distance between points on the permutohedron. From this, we obtain the Riggle Metric, which we show to impose a metric in the formal sense. We then define a way of creating a path between any two points, Recursive Constraint Promotion (RCP). We prove that this path stays inside any grammar containing the two points it connects, and further the path has minimal length under the Riggle Metric. This establishes that grammars are geodesically convex, exactly as announced by Riggle.

(308) **Definition. Adjacent.** Vertices p_1 and p_2 of a permutohedron are *adjacent* iff p_1 = $P\underline{XY}Q$ and $p_2 = P\underline{YX}Q$, for a prefix P, some X and Y, and suffix Q.

Note that a vertex is not adjacent to itself.

(309) **Definition. Distance between adjacent vertices**. Given adjacent nodes p_1 and p_2, with $p_1 = P\underline{XY}Q$ and $p_2 = P\underline{YX}Q$, the distance between p_1 and p_2 is $\text{adjDist}(p_1,p_2) = |Q|+1$.

(310) **Definition. Path between two vertices**. Given vertices p_1, p_2, a path $\pi(p_1, p_2)$ between p_1 and p_2 is a sequence of vertices $\pi(p_1, p_2) = (r_1, r_2, \ldots, r_n)$ with $r_1 = p_1$, $r_n = p_2$, and each pair r_i, r_{i+1}, $1 \leq i \leq n-1$ is adjacent.

(311) **Definition. Length of a path**. Given a path $\pi(p_1, p_2) = (r_1, r_2, \ldots, r_n)$, the length of $\pi(p_1, p_2)$, denoted $\text{Len}(\pi)$, is the sum of the distances between sequential points on the path.

$$\text{Len}(\pi) = \sum_{k=1}^{n-1} \text{adjDist}(r_k, r_{k+1})$$

(312) **Definition. Distance between vertices.** Given arbitrary vertices p_1, p_2, the distance between p_1 and p_2, $d_R(p_1,p_2)$, is the shortest path between the two if they are distinct, and zero if $p_1 = p_2$.

$$\begin{aligned} d_R(p_1,p_2) &= \min\{\text{Len}(\pi) \mid \pi \text{ is a path between } p_1 \text{ and } p_2\}, \text{ if } p_1 \neq p_2 \\ &= 0, \text{ if } p_1 = p_2. \end{aligned}$$

The following lemma justifies our calling $d_R(p_1, p_2)$ a 'distance'.

(313) **Lemma. d_R is a metric**.

Proof. There are four parts to showing that d_R is a metric on the vertices of the permutohedron.

(1) **Non-negativity**. We need to show that $d_R(p,q) \geq 0$. This is clear from the definition of $d_R(p,q)$ since $d_R(p,q)$ is the smallest value of a set that consists of sums of positive numbers, when it is not zero.

(2) **Coincidence**. Next we show that $d_R(p,q)=0$ iff $p = q$. If $p = q$, then by definition (312), $d_R(p,q) = 0$. Suppose $d_R(p,q) = 0$. If $p \neq q$, then every path between them contains at least two vertices, p and q. The smallest distance between adjacent

vertices is 1, occuring when the shared suffix in the adjacent pair is zero. But this means that $d_R(p,q) \neq 0$. Therefore, p = q.

(3) **Symmetry**. $d_R(p,q) = d_R(q,p)$. Given a path from p to q, $\pi(p,q) = (r_1, r_2, \ldots, r_n)$, we can immediately construct a path from q to p, $\pi(q,p)=(r_n, r_{n-1}, \ldots, r_1)$, by reversing the order of the vertices. Furthermore, these paths have the same length since they contain the same pairs of adjacent vertices. This means that a minimal length path from p to q is also a minimal length path from q to p.

(4) **Triangle inequality**. We must show $d_R(p,q) \leq d_R(p,s) + d_R(s,q)$. The distance $d_R(p,q)$ gives a minimal path $\pi_{min}(p,q)$ between them. Similarly we have minimal paths $\pi_{min}(p,s)$ and $\pi_{min}(s,q)$. Concatenating the paths $\pi_{min}(p,s)$ and $\pi_{min}(s,q)$ yields a path between p and q; denote this path $\pi(p,q)$. Now

$$\begin{aligned} d_R(p,q) &= \mathrm{Len}(\pi_{min}(p,q)) \\ &\leq \mathrm{Len}(\pi(p,q)) \\ &= \mathrm{Len}(\pi_{min}(p,s)) + \mathrm{Len}(\pi_{min}(s,q)) \\ &= d_R(p,s) + d_R(s,q). \end{aligned}$$ □

We now establish some basic facts about the geometry of points in the permutohedron.

(314) **Lemma**. For adjacent p_1 and p_2 where $p_1 = P\underline{XY}Q$ and $p_2 = P\underline{YX}Q$, $d_R(p_1,p_2) = \mathrm{adjDist}(p_1,p_2)$.

Proof. First note that because (p_1,p_2) is a path between p_1 and p_2, the distance between the two vertices must be less than or equal to $\mathrm{adjDist}(p_1, p_2)$. More concisely, $d_R(p_1,p_2) \leq \mathrm{adjDist}(p_1,p_2)$. We make the claim that there is no shorter path than (p_1, p_2). To see this, consider any arbitrary path $\pi = (p_1,\ldots,p_2)$ between p_1 and p_2. Note that in π the constraint Y must be transposed from the $|P|+2^{nd}$ position to the $|P|+1^{st}$ at some step in the path. This incurs a cost of $|Q|+1$ which is exactly the value of $\mathrm{adjDist}(p_1, p_2)$. Any other transpositions will only increase the length of π. Therefore (p_1, p_2) has a length less than or equal to all other paths between p_1 and p_2. In fact, it is the unique least path because all other paths must include other points, including positive increments to the path length. □

Given arbitrary point *s* and *t* in the same language, we construct a path between them that is both minimal and entirely contained within the language.

Given two points $s,t \in \mathrm{Ord}(S)$, where S is a set of *n* constraints, there is a first position $k < n$ at which they differ. Therefore, they share a prefix P, possibly empty, with $k = |P|+1$. Thus, s = PQ and t = PXR, with X occurring non-initially in Q. We construct a path (s, t), where *s* is the start point and *t* the terminus, by a procedure that we call *Recursive Constraint Promotion* (RCP). We outline RCP verbosely here and spell it out immediately below in (317).

RCP: First we find X in *s*, somewhere in Q; then we move it up to right below P by pairwise flips, leaving everything else the same, creating a path from *s* to a new point f(*s*) that now shares PX with *t*. This steps reduces our problem by extending the shared prefix by one constraint. If f(s) is not t, we reapply the same procedure to f(s) and t, creating a path between f(s) and f(f(s) = $f^{(2)}(s)$, a point that is identical to t in one more prefixal position. If $f^{(2)}(s)$ is not t, we continue onward, pushing the shared prefix rightward, until the result of the procedure $f^{(k)}(s)$ is identical to t. Pasting together all of the intermediate paths gives us a path from s to t. Consequences: we show first that this path stays within any grammar that includes s and t, and then that this is the shortest path between s and t. Riggle's finding is thereby established.

(315) **Definition. Start point**. The start point of a path $\pi = (s, \ldots, t)$ is the total order s.

(316) **Definition. Terminus**. The terminus of a path $\pi = (s, \ldots, t)$ is the total order t.

We now turn to the definition of RCP. We use the notation p[k] to refer to the k^{th} constraint in the total order p, counting left to right and starting with p[1]. We use the notation π^{-} to represent the path that results from removing the last point from the path π. We write $\pi_1 + \pi_2$ to denote the path resulting from the concatentation of paths π_1 and π_2 in that order, an operation defined only when the terminus of $\pi 1$ is adjacent to the start point of π_2.

(317) **Recursive Constraint Promotion. RCP(*s*,*t*)**

Step 0. Identify the first constraint $t[k]$ in which s and t differ.

Step 1. Find the constraint $t[k]$ in s. This must be in some position $m>k$ in s. Move it up to position k by a series of $m–k$ adjacent flips to create a path $\pi_k = (s, q_1, \ldots, q_{m–k})$. Constraint $s[m]$ is now in position k in $q_{m–k}$. Consequently, $q_{m-k}[i] = t[i]$ for all i, $1 \leq i \leq k$.

Step 2. If $q_{m-k} = t$, then RCP(s,t) = π_k.
Else RCP(s,t) = π_k^{-} + RCP(q_p, t).

Remark. The algorithm terminates because there's only a finite number of differences between t and s, i.e. k increases strictly with every recursive call to RCP, and $k \leq |s| = |t|$.

Remark. RCP proves constructively that Ord(S) is path-connected.

We now consider the relation between RCP(s,t) and a grammar containing both s and t.

(318) **Lemma**. The path RCP(s,t) lies within any grammar than contains s and t.

Proof. We show by contradiction that the path RCP(s,t) can never stray outside a grammar containing s and t. Let t = PXQ where X is the first constraint on which s and t differ. We are given that $s,t \in \Gamma_1$ for some grammar Γ_1. Suppose *per contradictio* that there is some flip involving X in the path RCP(s,t) that exits Γ_1 and arrives in Γ_2. There is then a border point pair {PR$\underline{XY}$T, PR$\underline{YX}$T} with PR$\underline{YX}$T$\in\Gamma_1$

and PR$\underline{XY}$T$\in\Gamma_2$. For this to happen, X must select (the language label corresponding to) Γ_2 and eject (the language label corresponding to) Γ_1.

From previous discussion (§3.1, 3.2), we have that in any UVT U instantiating a typology containing Γ_1 and Γ_2, their corresponding language labels $L^U_{\Gamma_1}$ and $L^U_{\Gamma_2}$ are numerically equal on all the constraints in PR and therefore on all the constraints in P. Thus both $L^U_{\Gamma_1}$ and $L^U_{\Gamma_2}$ survive P. Furthermore, $X(L^U_{\Gamma_2}) < X(L^U_{\Gamma_1})$ because PR$\underline{XY}$T $\in \Gamma_2$ and PRYXT$\notin \Gamma_2$. Thus, PX ejects $L^U_{\Gamma_1}$ in favor of $L^U_{\Gamma_2}$ and possibly others. This is a contradiction, because, by assumption, t = PXQ $\in \Gamma_1$.

Therefore every link in the RCP path stays within any grammar to which *s* and *t* belong. □

Consider any path π = (s,…,t) in P_n. Any given constraint X participates in a certain number of adjacent flips in the path. Each flip defines an edge in π of a certain length. Let the *travel* of X in π, denoted tr(X,π), be the sum of the length of the edges that its movements give rise to. From this, we may define the *action* *S*(π) to be the sum over all constraints of their travel in π.

(319) **Action of a path**

$$S(\pi) = \sum_{X \in \mathrm{CON}_N} \mathrm{tr}(X, \pi)$$

The action of a path has a simple relationship to its length. Each flip involves two constraints; therefore, the length of the edge associated with a flip is added to the travel of two different constraints. Thus, the action of a path is twice its length.

(320) **Lemma.** S(π) = 2 ||π||

Proof. Follows from text above. □

This means that properties of the action translate immediately into properties of path length. In particular, if we find a path of least action, we have found a path of minimal length. We use this fact to show that RCP(s,t) provides a minimal length path from s to t.

(321) **Lemma**. For any *s*,*t* $\in P_n$, RCP(s,t) is a shortest path between *s* and *t*.

Proof. Let *s*, *t* be points on the permutohedron. Consider any constraint X occurring as s[*m*] and t[*k*]. We begin by establishing the minimal number of flips that involve X in any path between *s* and *t*, as it moves from its initial position *m* to its terminal postion *k*. We make this calculation without regard to whether there is a path that contains only these flips.

Observe that in *s*, there is a some set of constraints D above X that appear below X in *t*. In any path from *s* to *t*, these constraints must move *down* past X to reach their position in t.

Similarly, in *s*, there is also a set of constraints U below X that appear above X in *t*. In any path from *s* to *t*, these constraints must move *up* past X to reach their position in *t*.

In addition to these, there is a set *A* of constraints lying above X in both *s* and *t*, and a set *B* of constraints lying below X in both *s* and *t*.

Writing [S,T] for a sequence of elements constructed from the elements of sets S,T, we can schematize the general situation as follows:

$$s = [A, \mathrm{D}]\ \mathrm{X}\ [B, \mathrm{U}]$$

$$t = [A, \mathrm{U}]\ \mathrm{X}\ [B, \mathrm{D}]$$

This notation is not meant to imply that the mentioned sets of constraints *A*, *B*, D, U, lie in separate contiguous blocks, merely that they are in a sequence with the elements of whatever sets they are bracketed with.

There are |D| constraints that must flip downward past X in any path from *s* to *t*. Each of these flips moves X up by one position. Therefore X must minimally flip upward |D| times. Similarly, there are |U| constraints that must flip up past X in any path from *s* to *t*. Each of these flips moves X one position downward. Therefore X must minimally flip down |U| times. Since these flips must occur in any path that goes from *s* to *t*, every such path contains at least |D| + |U| flips that involve X.

Of course, there may be paths between *s* and *t* that include other flips, but all paths must certainly include these. Claim: for any X, the path RCP(s,t) includes *only* these flips. RCP proceeds by finding the first point where *s* and *t* differ. Each element $u \in \mathrm{U}$ will be detected in its position k_u in *t*, and its avatar in position m_u in *s* will be flipped up to position k_u. In this process, it passes X exactly once, shifting X down once, so that X is shifted down precisely |U| times in the processing of U. Note that in RCP a constraint is shifted down only when a constraint that must be above it in *t* is moved up past it. This means that constraint X participates in no other downflips besides those involving the elements of U.

Now consider what happens when RCP reaches X in its left-right sweep of *t* looking for disparities between *s* and *t*. The path under construction currently ends on a point s_X that looks like this:

$$s_\mathrm{X} = [A, \mathrm{D}, \mathrm{U}]\ \mathrm{X}\ [B]$$

At this point, RCP has put every constraint [*A*, U] — everything that precedes X in *t* — into its correct position. This includes all of *A* and U and excludes all of D, because all elements of D follow X in *t*, so that nothing in D has yet been positioned by RCP. The positions occupied by the elements of *A* and U are the first |*A*|+|U| positions in s_X, because they occupy those positions in *t*. Therefore at this stage *all* of the D lie in a sequence that immediately follows [A,U] and immediately precedes X. Thus s_X may be more tightly schematized as follows:

$$(*) \qquad s_\mathrm{X} = [\mathrm{A}, \mathrm{U}]\ [\mathrm{D}]\ \mathrm{X}\ [\mathrm{B}]$$

Since all of the constraints in D must follow X in *t*, RCP will flip X up past each of them, with one flip per constraint in D. After this, there are no other up flips of X, because everything that must be above it (U) and every that must be below it (D) has been correctly positioned with respect to X.

These arguments establish that RCP flips X exactly |U|+|D| times, which is the minimal number of flips involving X in any path from *s* to *t*. Since X is arbitrary, RCP flips every constraint the minimal number of times it must be flipped in any path from *s* to *t*. The path RCP(*s*,*t*) has therefore the minimal number of points in it.

We now show that the action *S*(RCP(s,t)) is minimal. We begin by showing that the travel of each constraint in RCP(s,t) is minimal..

First, consider the *best possible set of positions* in which the downflips of X could occur, without reference to any particular path. We must start with X in position *m* in *s*. We know that X must flip down |U| times. Claim: the downflips of X contribute minimally to the travel of X if they take place when X is in positions *m*, *m*+1, …, *m*+|U|. X will occupy each of these positions when the elements of U are flipped past it without any intervening flips involving X. If this sequence is interrupted by flipping X up, then the length of a necessary downflip can only be increased, so this option may be dismissed. The only way to decrease the length of any of these downflips would be to move X rightward prior to the flip by moving some constraint $C \notin U$ up past X before the flip takes place. However, this is equally futile because the travel of X still includes a flip (with C) at the position it was moved down from. Therefore, to achieve minimal travel on the constraints of |U|, they must be moved up past X without interruption, starting with X is its initial position *m*, as claimed. This is precisely what RCP does.

Now consider the best possible positions in which the |D| upflips of X could occur. Since X must end up at position *k*, this sequence of upflips should start at position *k*+|D| and proceed up to position *k* to minimize the upflip cost. In RCP, X is reached for processing, after all its downflips, and X is in precisely this position, as shown in schema (*).When X is reached, X has participated in |U| downflips and no other flips. It has |D| flips to go. But this means that it is in position *k*+|D|. Therefore RCP incurs the minimal cost for the |D| upflips.

Now observe that there can be no compensation in downflip cost for choosing a path that increases the cost of the upflips, because the downflip cost cannot be reduced below its minimum. Similarly for attempting a path with supraminimal downflip cost. Since there is no way to lower either downflip or upflip costs associated with RCP, the sum of the two is also minimal. Thus, for each constraint X, RCP(*s*,*t*) incurs minimal travel. By the same reasoning just employed, there can be no other less-travel path that has overall less action. Therefore, RCP(s,t) is the least action path between *s* and *t*. Because path length is simply half the action, RCP(s,t) is the shortest path between *s* and *t*. □

(322) **Definition**. **Geodesic Convexity**. A set of points S is *geodesically convex* with respect to a metric μ iff for every $p_1, p_2 \in S$, the *geodesic* or shortest path $\gamma(p_1, p_2)$ under μ is such that $\gamma(p_1, p_2) \in S$.

(323) **Theorem**. **Geodesic Convexity of Grammars**. OT grammars are geodesically convex with respect to the Riggle metric.

Proof. For any p_1, $p_2 \in \Gamma$, the RCP path between p_1 and p_2 lies within Γ, by Lemma (318). By Lemma (321), this is the shortest path between p_1 and p_2. $\square$

O mama, can this really be the end?

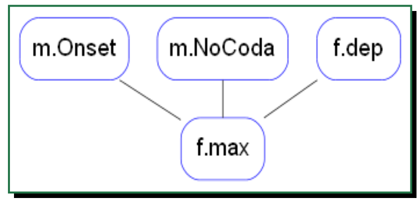
Appendix I. Leg Set Partition of EST

1:CV.del	m.Ons	≫	m.NoCoda	≫	f.dep	≫	f.max
	m.Ons	≫	f.dep	≫	m.NoCoda	≫	f.max
	m.NoCoda	≫	m.Ons	≫	f.dep	≫	f.max
	m.NoCoda	≫	f.dep	≫	m.Ons	≫	f.max
	f.dep	≫	m.Ons	≫	m.NoCoda	≫	f.max
	f.dep	≫	m.NoCoda	≫	m.Ons	≫	f.max
2:(C)V.del	m.NoCoda	≫	f.dep	≫	f.max	≫	m.Ons
	f.dep	≫	m.NoCoda	≫	f.max	≫	m.Ons
3:CV.ins	m.Ons	≫	m.NoCoda	≫	f.max	≫	f.dep
	m.Ons	≫	f.max	≫	m.NoCoda	≫	f.dep
	m.NoCoda	≫	m.Ons	≫	f.max	≫	f.dep
	f.max	≫	m.Ons	≫	m.NoCoda	≫	f.dep
	m.NoCoda	≫	f.max	≫	m.Ons	≫	f.dep
	f.max	≫	m.NoCoda	≫	m.Ons	≫	f.dep
4:(C)V.ins	m.NoCoda	≫	f.max	≫	f.dep	≫	m.Ons
	f.max	≫	m.NoCoda	≫	f.dep	≫	m.Ons
5:CV(C).del	m.Ons	≫	f.dep	≫	f.max	≫	m.NoCoda
	f.dep	≫	m.Ons	≫	f.max	≫	m.NoCoda
6:(C)V(C).del	f.dep	≫	f.max	≫	m.Ons	≫	m.NoCoda
	f.dep	≫	f.max	≫	m.NoCoda	≫	m.Ons
7:CV(C).ins	m.Ons	≫	f.max	≫	f.dep	≫	m.NoCoda
	f.max	≫	m.Ons	≫	f.dep	≫	m.NoCoda
8:(C)V(C).ins	f.max	≫	f.dep	≫	m.Ons	≫	m.NoCoda
	f.max	≫	f.dep	≫	m.NoCoda	≫	m.Ons

Appendix II. EST: SKBs and Hasse Diagrams

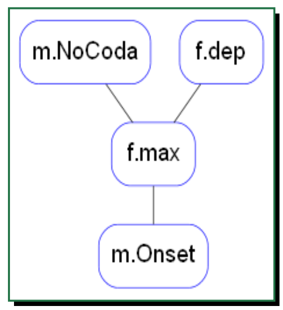
(324) **1:CV.del**

m.Onset	m.NoCoda	f.dep	f.max
W			L
	W		L
		W	L



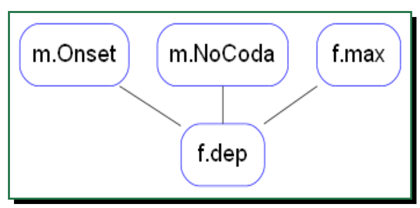
(325) **2:(C)V.del**

m.Onset	m.NoCoda	f.dep	f.max
	W		L
		W	L
L			W



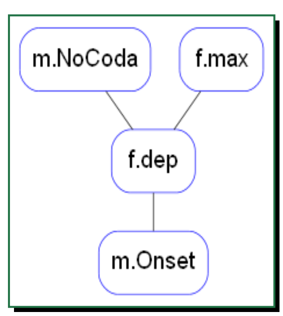
(326) **3:CV.ins**

m.Onset	m.NoCoda	f.dep	f.max
W		L	
	W	L	
		L	W



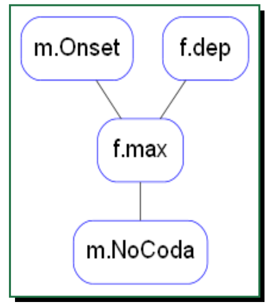
(327) **4:(C)V.ins**

m.Onset	m.NoCoda	f.dep	f.max
	W	L	
		L	W
L		W	



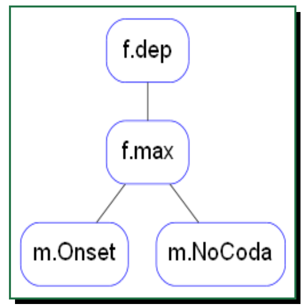
(328) **5:CV(C).del**

m.Onset	m.NoCoda	f.dep	f.max
W			L
		W	L
	L		W



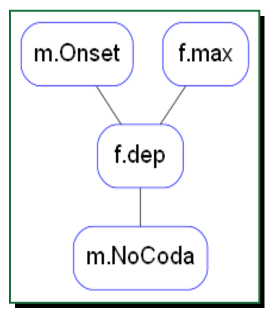
(329) **6:(C)V(C).del**

m.Onset	m.NoCoda	f.dep	f.max
		W	L
L	L		W



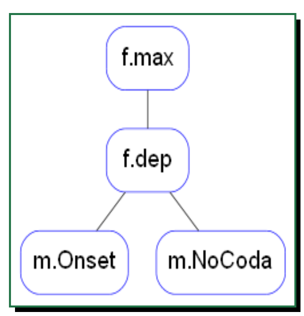
(330) **7:CV(C).ins**

m.Onset	m.NoCoda	f.dep	f.max
W		L	
		L	W
	L	W	



(331) **8:(C)V(C).ins**

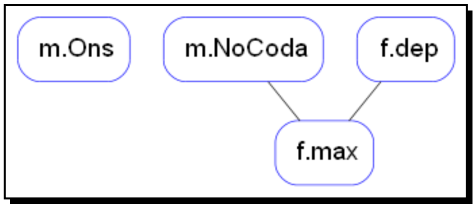
m.Onset	m.NoCoda	f.dep	f.max
		L	W
L	L	W	



Appendix III. EST.CSys: SKBs and Hasse Diagrams

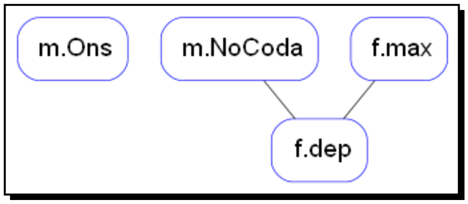
(332) **1**∪**2**:CP.del

m.Ons	m.NoCoda	f.dep	f.max
	W		**L**
		W	**L**



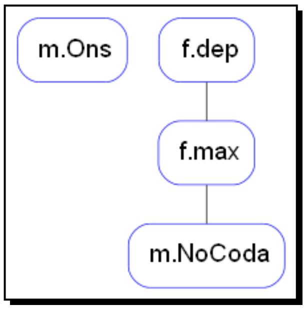
(333) **3**∪**4** CP.ins

m.Ons	m.NoCoda	f.dep	f.max
	W	**L**	
		L	**W**



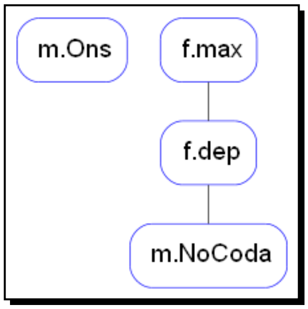
(334) **5**∪**6**:CA.del

m.Ons	m.NoCoda	f.dep	f.max
		W	**L**
	L		**W**



(335) **7**∪**8**:CA.ins

m.Ons	m.NoCoda	f.dep	f.max
		L	**W**
	L	**W**	



Appendix IV. Notation

Symbol	Meaning		Page Reference
GEN$_S$	GEN for a system, S		7
CON$_S$	constraints for a system, S		7
Ord(CON$_S$)	set of all linear orders on CON$_S$		8
$C_i \gg C_j$	constraint C_i is ranked above C_j		8
cset	candidate set		8
leg	linear extension of a grammar		11
XT(S)	extensional typology for system S		13
IT(S)	intensional typology for system S		13
UVT	Unitary Violation Tableau		15
G_R(L)	ranking grammar for language L		9, 10
T_U	typology associated with UVT U		16
*x:P(x)	constraint operator yielding matches of P(x)		24
$\in$	element of		24
card(S)	the cardinality of set S		24, 43
Parse-σ	*o	(card{σ∈out(κ)| σ∉F})	24
Iamb	*[$_F$ σ′	card{ [$_F$ σ′ ∈out(κ)}	24
AFL	*(σ,F): σ…F	card{(σ,F)∈out(κ)| σ precedes F}	24
sp	sparse: words taking the form Fo^n		24
WD	weakly dense: words taking form F^n(o)		24
SD	strongly dense: words taking form F^n		24
$\Gamma_1 \bullet \Gamma_2$	node merger of grammars in MOAT		38
$x \sqsubseteq s$	x is a contiguous substring of the string s		43
m.Ons	number of onsetless syllables in a candidate's output, card{ "[V" ⊑ *out*}		43
m.NoCoda	number of syllables that have a coda in a candidate's output card{ "C]" ⊑ *out*}		43
f.max	number of input segments that lack output correspondents card{x ⊑ *in* | x∈{C, V} and ¬∃y⊑ *out*, y = c(x)}		43
f.dep	number of output segments that lack input correspondents card{y ⊑ *out* | y∈{C, V} and ¬∃x⊑ *in*, y = c(x)}		43
OR	onset required		45
OLA	onset lack allowed		45
CP	coda prohibited		45
CA	coda allowed		45
$\oplus$	Minkowski sum		49
$\approx_{C:U}$	equivalence relation induced from UVT, U, for constraint C		52
$\lessdot_{C:U}$	order relation induced from UVT, U, for constraint C		52?
L_j^U	row label for row *j* of UVT U corresponding to grammar Γ_j		52
aRb	for relation R, (a, b)∈R		54

$\mathcal{U}$(T)	set of UVTs that yield typology T	55
$\lessdot_{C}$	for constraint C, intersection of $\lessdot_{C:U}$ over the U$\in\mathcal{U}$(T)	55, 87
$\approx_{C}$	for constraint C, intersection of $\approx_{C:U}$ over the U$\in\mathcal{U}$(T)	56, 87
*i*EQO(C)	intersection of all equivalence and order relations for column C, the ordered pair $\langle \lessdot_{C}, \approx_{C} \rangle$ Note: 87 and 56 differ in subscripting of typology T. Resolve this here or in the text.	56, 87
*i*OAT($\mathcal{U}$(T))	intersection of all tableaux for typology T, the collection of all *i*EQO(C)	56, 87
BPP	border point pair	58
$<_{C}$	EPO order relation for constraint C	59 (need EPO ref)
$\sim_{C}$	EPO equivalence relation for constraint C	59
C( )	Constraint function for C from CAND to $\mathbb{N}$	74
C[ ]	Constraint function for C from 2^{CAND} to 2^{CAND}	74
P[ ]	OT filtration function for a sequence of constraints P	74
K_{U}	The row labels for UVT U, $\{ L^{U}_{1}, \ldots, L^{U}_{p} \}$	76
$<^{b}_{C}$	base order relation for constraint C	78
$\sim^{b}_{C}$	base equivalence relation for constraint C	78
Lab(U)	the set of row labels for UVT U	84
g_{U}	bijection from grammars of typology to row labels of U	84
$\langle S, R_1, R_2 \rangle$	a bigraph: R_1 and R_2 are relations on set S	86
EQO	a bigraph in which R_1 is a partial order and R_2 an equivalence relation	86
Γ_{k}^{*C}	the Γ-cloud of Γ_k with respect to constraint C, $\{\Gamma \mid \Gamma \sim_{C} \Gamma_k\}$	92
*i*OAT$_{\text{equiv}}$($\mathcal{U}$(T))	the set of equivalence relations from the *i*OAT, $\{\approx_{C} \mid C \in \text{CON}_{T}\}$	95
MOAT$_{\text{equiv}}$(T)	the set of equivalence relations from the MOAT, $\{\sim_{C} \mid C \in \text{CON}_{T}\}$	95
EER(Σ)	equivalence-extended relation on $\Sigma=\langle S,<,\sim\rangle$, $<^{\sim}$	97, 109
$<^{\sim}$	the EER(Σ) relation	97
htcEPO(C)	hypertransitive closure of EPO(C), the bigraph $\langle K, \leqslant, \sim \rangle$	98, 109
$\leqslant$	the partial order relation of the htcEPO(C)	98
$C^{[U:n]}$	the *n*-band for C in U, $\{\Gamma_i \mid C(L^{U}_{i}) = n\}$	98
$\mathbf{a}^{\oplus}$	sum of rows $a_1, \ldots, a_n$ from VTs, $V_1, \ldots, V_n$	102
Ord(S)	the set of all linear orders on a reference set S	104
GMOAT	generalized MOAT constructed from any partition of Ord(S)	105, 108
GEPO(C)	generalized EPO constructed for C$\in$S from any partition of Ord(S)	105, 108
DT	the discrete typology	105
M(T)	MOAT for typology T	106
GM(π)	GMOAT for partition π	106
bpa	border point analysis function from typologies to M(T)	106

merge	node merger function from M(T) to GM(π)	106
U( )	grammar union function from typologies to partitions	106
$R^{b}_{o/X}$	order base relation for constraint X	107
$R^{b}_{e/X}$	equivalence base relation for constraint X	107
$R_{o/X}$	order relation obtained by the transitive closure of $R^{b}_{o/X}$	107
$R_{e/X}$	equivalence relation obtained by the transitive closure of the reflexive closure of $R^{b}_{e/X}$	107
$\sim_X$	alternate notation for $R_{e/X}$	107
fEPO(X)|P	the filtered EPO of X with respect to prefix P	110
T_U	typology associated with the UVT U (this is diff notation from T(U) – see above, p16 – shd we standardize?)	113
B^r	for partition π with instantiating UVT, V, the block in π corresponding to row $r \in V$	113
ELB_k	an indexed EPO-like bigraph	115
$fELB_k\|P$	filtered EPO-like bigraph for a sequence of indices P	115
ERCoid	a finite-dimensional vector with entries from {W, L, *e*, u}	120
$\alpha * \beta$	the weak composition of ERCoids α and β	124
UBE[$\Gamma \sim \Gamma'$]	unitary border ERCoid between grammars Γ and Γ'	126, 129
$[\Gamma \sim \Gamma']_{BP}$	alternate notation for UBE[$\Gamma \sim \Gamma'$]	129
$[\Gamma \sim \Gamma']_{M}$	MOAT-derived ERCoid between grammar Γ and Γ'	129
$\Gamma_i \parallel_C \Gamma_j$	grammars Γ_i and Γ_j are noncomparable on constraint C	128
$\Gamma + \Gamma'$	the join of grammars Γ and Γ'	153
joinard	member of joining set: e.g. Γ or Γ' in $\Gamma + \Gamma'$	157
d_R	the Riggle metric	178, 185
$\delta_x^{\uparrow}(L_i, L_j)$	the upward quasi-pseudodistance between L_i and L_j in the X-view of the typohedron	183
adjDist(p,q)	distance between adjacent vertices p and q in the permutohedron	185
π(p, q)	path between vertices p and q in the permutohedron	185
Len(π)	length of a path π	185
$DT^{(n)}$	the discrete typology on *n* constraints	185
$U_0^{(n)}$	the minimal UVT of $DT^{(n)}$	185
RCP	Recursive Constraint Promotion	187

References

Alber, Birgit. 2005. Clash, Lapse, and Directionality. *Natural Language* & *Linguistic Theory* 23.3, 485-542.

Alber, Birgit & Alan Prince. In Prep. *Typologies.* Ms. University of Verona and Rutgers University.

Alber, Birgit, and Alan Prince. 2015. Outline of Property Theory [Entwurf einer verallgemeinerten Eigenschaftstheorie]. Ms. University of Verona and Rutgers University.

Alber, Birgit, Natalie DelBusso, and Alan Prince. To appear 2016. From Intensional Properties to Universal Support. *Language: Phonological Analysis*. ROA-1235.

Atkin, Emily. 2013. First Climate Article On Nate Silver's Data Website Uses 'Deeply Misleading' Data, Top Climatologists Say. *Think Progress*: March 19, 2014.

Bane, Max and Jason Riggle. 2012. Consequences of Candidate Omission. *Linguistic Inquiry* 43(4):695-706. ROA-1108.

Blair, Eric A. (as George Orwell). 1949. *1984*.

Borges, Jose Luis. 1942. La muerte y la brújula. *Sur*: 92, 27-39. Buenos Aires.

Brasoveanu, Adrian & Alan Prince. 2005/2011. Ranking and Necessity: the Fusional Reduction Algorithm. ROA-794.

Crowhurst, Megan J. and Mark Hewitt. 1995. 'Directional footing, degeneracy, and alignment', in J. Beckman (ed.), Proceedings of NELS 25, Vol. 1, GLSA, Amherst. pp. 47–61, ROA-65.

Descartes, René. 1637. Discourse on Method: Part VI.

Dietrich, Brenda J. 1987. A Circuit Set Characterization of Antimatroids. *Journal of Combinatorial Theory*, Series B 43, 314-321 (1987).

Eastwood, Clint and David Webb Peoples. 1992. *Unforgiven*. Warner Bros.

Guilbaud, G. Th.; Rosenstiehl, P. 1963. Analyse algébrique d'un scrutiny. Mathématiques et sciences humaines 4: 9–33. Via Wikipedia article "Permutohedron."

Howard, Irwin. 1972. *A Directional Theory of Rule Application in Phonology*. PhD dissertation: MIT, Cambridge, MA.

Hyde, Brett. 2012. Alignment constraints. *Natural Language* & *Linguistic Theory* 30: 789-836.

Ito, Junko. 1986. *Syllable Theory in Prosodic Phonology*. PhD dissertation, University of Massachusetts at Amherst.

Ito, Junko. 1989. A Prosodic Theory of Epenthesis. *Natural Language & Linguistic Theory* 7:2, 217-250.

Johnson, C. Douglas. 1972. *Formal aspects of phonological description.* (*Monographs on linguistic analysis*). Mouton: Den Haag. From 1970 UCSD dissertation of same title.

Karttunen, Lauri. 1998. The Proper Treatment of Optimality in Computational Phonology. ROA- 258.

Keats, John. 1816. On first looking into Chapman's Homer. *Examiner*: London.

Korte, Bernhard; Lovász, László; Schrader, Rainer (1991), Greedoids, Springer-Verlag, pp. 19–43.

Krugman, Paul. 2014. Tarnished Silver. New York Times: March 23, 10:48am.

Lombardi, Linda. 1999. Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language* & *Linguistic Theory* 17:267-302.

Merchant, Nazarré. 2008. *Discovering underlying forms: contrast pairs and ranking*. PhD dissertation, Rutgers University. ROA-964.

Merchant, Nazarré. 2011. Learning ranking information from unspecified overt forms using the join. Proceedings of the CLS 47. ROA-1146.

Merchant, Nazarré and Jason Riggle. 2016. OT grammars, beyond partial orders: ERC sets and antimatroids. *Natural Language* & *Linguistic Theory* 34.1: 241-269. As "Erc Sets and Antimatroids," ROA-1158.

Miryam, Yeshua bar. ca. 30. Quoted in κατὰ Ματθαῖον εὐαγγέλιον, 7:26 (PC to author).

Prince, Alan. 2002. *Entailed Ranking Arguments*. ROA-500.

Prince, Alan. 2009. RCD – the Movie. ROA1057. 69pp.

Prince, Alan. 2013. Metrical Theory as a Portal on Theory. Youtube video.

Prince, Alan 2014. Act Accordingly: a User's View of Theory. Slides from talk at FLYSM I, St. Petersburg, FL.

Prince, Alan. 2014. Testing the Boundaries: Aspects of Typological Structure in OT. Slides from talk at Stanford University.

Prince, Alan. 2015a. What is OT? ROA-1271.

Prince, Alan. 2015b. One Tableau Suffices. *Linguistic Inquiry*: rejected. ROA-1250.

Prince, Alan, with Birgit Alber and Naz Merchant 2014. The Structure of OT Typologies: First Steps. Slides from talk at OCP (Old world Conference in Phonology).

Prince, Alan, with Birgit Alber and Nazarré Merchant. 2015. Intrinsic Structure in OT Typologies. Slides from talk at Workshop on the Formal Structure of OT Typologies. Rutgers University.

Prince, A. and P. Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell. Also: ROA-537.

Riggle, Jason. 2004. *Generation, Recognition, and Learning in Finite State Optimality Theory*. Ph.D. Dissertation, UCLA: Los Angeles.

Riggle, Jason. 2010. Sampling Rankings. ROA-1075.

Riggle 2012. Geodesic Convexity. Talk at the Rutgers Optimality Research Group. Department of Linguistics, Rutgers, The State University of New Jersey, at New Brunswick.

Samek-Lodovici and Prince.1999. *Optima*. ROA-363.

Schoute, Pieter Hendrik. 1911. Analytic treatment of the polytopes regularly derived from the regular polytopes. *Verhandelingen der Koninklijke Akademie van Wetenschappen te Amsterdam* 11 (3): 87 pp. Googlebook, 370–381. Also online on the KNAW Digital Library.

Silver, Nate. 2016. How I acted like a Pundit and Screwed up on Donald Trump. May 18: FiveThirtyEight.

Steriade, Donca. 1982. *Greek prosodies and the nature of syllabification*. Ph.D. Dissertation, MIT. Cambridge, MA.

Stevens, Wallace. 1923. The bird with the coppery, keen claws. *Harmonium*. Alfred E. Knopf.

Wolchover, Natalie. 2013. A new tool to help mathematicians pack. *Quanta Magazine*. Dec. 20.

Wikipedia .
Directed graph
Geodesic convexity
Graph
Graph drawing
Hasse Diagram
Metric
Order
Permutohedron
Rooted Graph
Setoid
Transitive closure

La primera letra del Nombre ha sido articulada.