The Structure of OT Typologies

Chapter 1: Introduction to Property Theory (Mar, 2021)

Birgit Alber
Free University of Bozen-Bolzano
Alan Prince
Typohedron.org

Acknowledgments

The fundamental ideas behind this work emerged in 2010-2011when one author was visiting the other at the University of Verona to pursue work begun in 2008-2009, supported in both instances by Cooperint grants to the traveler. Since then the proposals have been filtered, augmented, refined, applied and clarified, often through interaction with others interested in the issues and the theory.

Special thanks go to Natalie DelBusso and Nazarré Merchant for many valuable discussions at all levels of detail, as well as to Hope McManus, Paula Houghton, Will Bennett, Martin Krämer, Sabine Arndt-Lappe for contributions that have significantly influenced the direction of the enterprise. We thank Roberto Zamparelli and Jane Grimshaw for discussion of many related issues.

We would also like to thank the participants in the first Workshop on the Formal Structure of OT Typologies (SOTA 1), held at Rutgers University in 2015 and crucially supported by the School of Arts and Sciences there, which gave a significant boost to the coherence of the theory. We thank John Alderete, Arto Anttila, Eric Bakovic, Nick Danis, Natalie DelBusso, Paula Houghton, Brett Hyde, Luca Iacoponi, Junko Ito, René Kager, Paul Kiparsky, Giorgio Magri, Hope McManus, Nazarré Merchant, Armin Mester, Jason Riggle, and Paul Smolensky for their presentations. Additional progress was made at the 2nd and 3rd Workshops on the Formal Structure of OT Typologies (SOTA 2, 3), held at the University of Verona in 2016 and 2019. The 4th Workshop (SOTA 4), was held at Eckerd College in St. Petersburg, Florida, in 2020, with support by NSF grant #1823827 to Nazarré Merchant and additional support from the Dean's Office at Eckerd. SOTA 4 was based on a week-long course with participants from the University of Verona, The Arctic University of Norway (Tromsø), The University of California at San Diego, The University of California at Santa Cruz, Eckerd College, and Rhodes University. We would like to thank Eirini Apostolopoulou, Eric Bakovic, Jennifer Bellik, Will Bennett, Richard Bibbs, Erin Griesenauer, Nick van Handel, Brett Hyde, Junko Ito, Nick Kalivoda, Annalise Kendrick, Joachim Kokkelmans, Anna Mai, Armin Mester, Thor Håvard Solhaug, Max Tarlov, Kate Zhakun for their participation. We are grateful to Martin Krämer of the Arctic University of Norway for organizing the PhD course "Analyzing linguistic typologies with OTWorkplace & Property Theory," which took place in Tromsø in 2018. Alber, Merchant, and Prince provided the instruction and Eirini Apostolopoulou, Antonio Fábregas, Joachim Kokkelmans, Martin Krämer, Jordina Nogués Graell, Anna Pilsbacher, Thor Håvard Solhaug, and Olga Urek participated in projects and discussion.

Aspects of this work have been presented by the authors at various conferences including OCP 11 (Leiden, 2014), OCP 12 (Barcelona, 2015), OCP 15 (London, 2018), the 25th Manchester Phonology Meeting (2017), Going Romance 34 (Paris, 2020), the Second Edinburgh Symposium on Historical Phonology, 2015, DgfS Annual Conference (Leipzig, 2015).

We are grateful for questions and comments from audiences at MIT (2013), UC Santa Cruz (2014, 2017), Stanford (2014), UC San Diego (2014), University of Trento, 2017, University of Bielefeld, 2017, Universität Frankfurt (2018), Uni Marburg, (2018). and the CNRS (Paris, 2017).

Additional support for Alber from the University of Verona (2017-19) and the Free University of Bozen-Bolzano (2020-22) and to Prince from the School of Arts & Sciences at Rutgers.

Preface

A *system* in Optimality Theory (OT) specifies the admitted candidates and the criteria ('constraints') that choose between them, and nothing more. The method of choice is fixed and sets OT apart from other theories that use the very same notions of structure and constraint.

Implicit within these commitments is the *typology* of the system — the set of languages (collections of optima) and grammars (collections of rankings or conditions on ranking) that the system admits. In OT, these can be calculated from a sufficient finite collection of candidate sets, themselves consisting of finite collections of possible optima, a *universal support*. Every linguistic theory has a typology in the broad sense of the term; they differ in the extent to which learners are deemed to have access to it and the extent to which analysts attend to it.

In theories configured like OT, a typology is distant from the premises that define it. From reviewing its defining commitments in isolation, interested parties typically cannot tell what patterns a system produces, or how they are produced. It's not even a certainty, in general, that theories of this character will produce what we'd recognize as patterns, or employ what we'd think of as higher-order mechanisms to do so.

From its first days, however, the ecological experience of OT is that its typologies are structured objects, and that its linguistic import lies in those emergent structures. We adopt this insight programmatically and take steps here to develop a theory of OT typologies that uncovers significant aspects of their structure. We hope to have expanded on what was implicit and explicit in prior studies and at the same time to have opened the way to further progress.

Property Theory, as we term it, is based on the idea that an entire typology can be generated from a set of *properties*, where each property is a pair of opposing ranking conditions, called *values*. The *value* generalizes both the notion of a single constraint dominating another single constraint and the notion of the ERC (Elementary Ranking Condition), in which at least one member of a set of constraints dominates all the members of another set. Key notions in Property Theory are the *constraint class*; the methods of choosing from a class based on ranking structure; and the notion of *scope*, whereby a given property may be assigned relevance to a limited set of other properties. A class is determined not from definitional characteristics, but from the role its members play in determining the structure of the typology. The over-arching goal is to explicate how the *traits* discerned in the languages are tied to specific values in the *property analysis* of a typology.

This document is the first chapter of our ms. monograph on the subject. It is meant to give a reasonably comprehensive and detailed grasp of the proposal and the methodology we have used to develop it. It falls into two parts.

- Examination of a trio of related typologies that bring out the basic ideas (sections 1.2-1.4). Further chapters of the book study more complicated typologies in detail.
- A focused view of the proposal; a presentation of OT as it understood here; and an overview of the progress of emergence in the context of OT (sections 1.5-1.6).

A bibliography of work in Property Theory is given in the final section (1.7). References to background and supporting literature are found throughout the text.

Chapter 1. Introduction to Property Theory

1.1 The Landscape
1.2 ERC, Value, Property: words of one syllable
    1.2.1 Analyzing nGo-1s
    1.2.2 Looking into the ERC
    1.2.3 Beyond the ERC: Property Theory
    1.2.4 The Property Analysis of nGo-1s
1.3 Public Classes, Treeoids: the system nGoX-2s
1.4 Sub, Mootness and Scope: the system nGo-2s
    1.4.1 The typology of nGo-2s
    1.4.2 The Property Analysis of nGo-2s: Being and Nothingness
    1.4.3 Mootness and Scope
1.5 Concise Outline of Property Theory
1.6 What is meant by *OT*?
    1.6.1 Gen, Con, Eval
    1.6.2 Language, Grammar, Typology: the Objects of OT
    1.6.3 From Micro to Macro
1.7 Appendix: Property Theory in the literature
References

1.1 The Landscape

An OT typology often presents as something of a jumble. Even when derived from clear and simple premises, its languages may offer a busy diversity of forms, and their grammars may seem to show only hints of shared and distinguishing ranking patterns. For this reason, perhaps, much discussion of formal typologies has been limited to assessing what's attested and what is not.

From the beginning, however, another way of thinking has been available, if not as often explored: that a typology is a richly structured object, demanding careful study, and that in its structure will be found the linguistic generalizations implied by the assumptions it emerges from. Our goal here is to develop an analysis of OT typologies from this perspective. In particular, we will propose that typologies resolve into choices between opposing ranking conditions of a certain determinate formal type, termed *properties*, which select the linguistic structures or *traits* manifest in the languages of the typology. We call the resulting theory of typological form Property Theory (PT) and we call the result of applying it to a given typology a *property analysis*.

As background, recall how Prince & Smolensky approach typologies of syllable structure (P&S: 1993/2004:ch. 6). They examine a basic theory —'basic' in the sense that only two types of segment are recognized and four types of syllable.[1] The P&S system seeks not only to match a generally-held understanding descended from Jakobson about what basic syllable canons exist —

[1] The explicit aim is to advance the work of Steriade (1982), Clements & Keyser (1983), Ito (1986, 1989).

in our terms, what *traits* languages display in forms and mappings — but also to discriminate the grammatical configurations (*properties*, for us) that give rise to them; and beyond that, to uncover patterns in the input-output map that the theory predicts.[2] More elaborately, Chapter 8 of P&S (p. 150-202) goes on to analyze in detail central aspects of a significantly richer syllabic typology, one with distinctions of sonority, pointing toward some key analytical concepts that reappear forcefully in this work.

But perhaps there is no real surprise in this historical asymmetry. OT typologies emerge from layer upon layer of formal construction, putting them at considerable distance from their defining commitments. Reasoning about them must respond to that layered structure. We cannot expect large-scale implied or emergent form to be immediately visible (or visible at all) in the small-scale actions and interactions that build it. The bee bustles: it does not think about hexagonal chambers; the goose eyes another goose and does not imagine or intend the flying V.

Theories differ markedly in the distance that must be traversed from abstract fundamentals to evident observables, and in how hard it is to make the journey. Planetary orbits pre-Newton were just items on a list; post-*Principia* they derive from a theory that not only makes no mention of them but requires special techniques — some still under development — to yield any information at all.[3] Along the same lines, early 'transformational grammar' tended to be notably construction-oriented. The 'passive transformation', for example, packaged and presented what came to be understood as a large amount of diverse information, much of it independently predictable or at least more broadly operative in the grammar. More recent evolutions of that theory require scrutiny of complex interactions between many elements, running within and across components, to determine how something like the passive is formed, if at all, in a given language. Careful, systematic reasoning becomes critical to grasping even the basics.

Analogously, phonological theories differ in the way information is packaged. For example, various structural inventories may be tabulated or they may be derived. In some theories, they are spelled out explicitly as part of a language's description, like Ptolemaic or Copernican orbits, and, as directly accessible objects, conditions may be placed upon them. In others, they only exist as emergent epiphenomena of the grammar and any generalizations about them must also be emergent. In SPE-type systems, for example, the inventory of lexical segments is fixed and subject to explicit restriction, but the collection of 'surface' segments is whatever happens to come out from application of the rule system to lexical entries. David Stampe (1969, 1973a,b), as well as others such as Dell (1973), Hudson (1974) and Myers (1991), observe that the input-output rule system itself indirectly imposes restrictions on the input inventory, particularly in the context of broad assumptions about lexical simplicity, often connected with learning. Prince & Smolensky (1993/2004) incorporate this insight, leading to a theory in which both lexical and surface inventories are emergent, as are therefore any patterns within them.

[2] Syllable canons, p.105 ex. (113). Ranking-to-I/O map relations, p. 116 ex. (140). Patterns of epenthesis, p. 116-118, exx. (141)-(145).

[3] See Newton (1687 *et seq.*). Montgomery (2019) provides an illuminating overview.

The Basic Syllable Theory (BST) of P&S: ch. 6 provides an illuminating example. Insertion of both C and V is freely allowed in candidates, yet is highly restricted in optima. Extending the P&S results slightly, it can be shown that under BST, an inserted C in optimal forms occurs only in V_V or #_V, and an inserted V only in C_C or C_#, where all cited environmental elements are present in the input.

Yet no such contextual restrictions are packaged together in the theory, nor can they be. The BST constraints themselves penalize certain output configurations — onsetless syllables, syllables with coda — as well as the context-free insertion of a consonant or vowel *anywhere*.[4] What a candidate is defined to be and what its competitors are (BST.Gen) will interact with the assumed constraints (BST.Con) through the theory-wide definition of optimality (EVAL) to produce the possible input-out maps. Analysis is required to ascertain what these turn out to be.

From this example, we hope it's clear that a shift in the direction of emergence has significant consequences for the practitioner, and is not merely an aesthetic choice to be appreciated by connoisseurs of fine theorizing. Scanning a rule that summarizes the allowed patterns of epenthesis rests on one set of skills; deducing them from interacting premises engages another.

What's required, at minimum, is a closer encounter with the inner workings of the theory. In the case at hand, the P&S argument depends on the notion of *harmonic bounding*, and applying it effectively requires control of all admitted structures and all posited constraints, as well as a clear sense of how the definition of optimality imposes order on competing candidates. Working within the theory at all requires some acquaintance with these notions, and a degree of success can be achieved, with fair probability, even when the conduct of analysis maintains a certain distance from validity. It's probably true that the typical consequence of faulty technique is incompleteness — failure to find all the successful rankings — and it may be less commonly the case that outright falsehoods are asserted, if we disregard unmotivated restrictions on those rankings that are found.[5] In the interests of upholding communal practice, some may regard it as a matter of taste whether or not an analyst is willing to live with optimism when, with further exertion, certainty is available.

But whatever the impact may be on linguistic inquiry in the narrow sense, the effects are sure to be drastic when an effort is being made to determine the global consequences of a theory, its emergent structure. Error, oversight, vagueness don't merely blur the picture: they disrupt and erase the very target of explanation. Without the optima, we do not have the languages in their fullness, and without the languages we do not have the typology. We cannot discern patterns in the languages if they are missing or distorted among the fragments we have managed to acquire. Similarly, without the rankings in their entirety, we do not have the grammars and cannot hope to

---

[4] The use of output constraints to regulate and simplify rules was a major theme of generative grammar, and a locus of strife, from the late '60's on. The immediate background of Prince & Smolensky (1993/2004) includes, as they note, work by Steriade, Ito, and others on the impact of syllable canons on phonological processes. (P&S attempt a census on p. 2.) The more distal background includes important work like Kisseberth (1970, 1972), roundly condemned by some at the time. The particularly *OT* move was to introduce the faithfulness constraint, so that interaction with output constraints could trigger as well as block IO maps (an issue noted in Kiparsky 1973).

[5] See Bane & Riggle (2012) for discussion.

determine the significant ranking patterns within them: those responsible for the traits that the languages contrast in and share. But that is the very goal of the present enterprise.

Achieving this goal requires a subtler, interactive process that entangles the analysis of language data and the analysis of ranking data. Neither is unambiguous by itself. Taking the two together, we have a much better chance of arriving at analytical decisions that mutually support each other. Error and incompleteness corrupt this prospect; we must therefore commit to the theory at a level of detail sufficient to reveal the features we wish to discern and collate.

To show how we intend to proceed and what we hope to accomplish, we will examine in this introductory chapter three related examples generated from the fundamentals of stress theory. Let's begin with what we might think of as the world's simplest interesting OT typology. With only two languages, each containing a single form, it sits within a broader family of stress systems that will provide the richer challenges addressed in chapters 4 through 6. More important for us, the analysis introduces most of the theoretical and conceptual motifs that recur throughout this work.

The examples also illustrate the methodology that guides the investigation. For us, *typology* means the languages and grammars that follow from a set of explicit assumptions about candidates and constraints: a *system*. An OT system *S* consists of two components: *S*.Gen, which defines the candidates and candidate sets — the admitted inputs, outputs, and the map between them — and *S*.Con, which defines every constraint of *S* as a function assigning a nonnegative integer to each candidate. The notion of optimality is part of OT per se and does not vary from system to system. This method of approach is required if we wish to make valid arguments: the bedrock notion of *optimality* becomes ill-defined without complete knowledge of the constraint set and the competitors. Specifying *S* gives that knowledge.

We can't hope, then, to be directly addressing the whole of universal grammar at every step in the research process. Advance toward it must be made through the study of defined systems ⟨*S*.Gen, *S*.Con⟩, deliberately narrowed, and work proceeds from system to system, related and not, as understanding accumulates. Since this strategy is compelled by logic, efforts to avoid it can only amount to pursuing it inexplicitly or defectively or under the impression that you are doing something other than what you are doing.

In light of this, we pursue here a somewhat abstract, incrementalist approach to theory-building, which is not ubiquitous in linguistics and indeed has more currency in some disciplines than in others. A particular system may be of interest as much for what it reveals about the theory, or more likely, some aspect of the theory, as for its match to some range of presumed facts. Other methodologies in the field start from top-down appeals to Kantian necessity, taking a stand in some version of νεφελοκοκκυγία, from which heights the direct path may be discerned, and run all the way to tight adherence to data modeling, from which (it is hoped) the empirical explicandum will emerge so clearly as to determine the eventual theory. Our goals are more modest. Since the linguistically significant patterns arise from interactions in OT, and are not perspicuous in *S*.Gen or *S*.Con for virtually any *S*, tools and concepts must be developed to expose them. The natural road into complexity makes its way through the simple, though of course these terms change meaning with progress. Our approach is to study typologies that connect with linguistic patterns of known interest, and we seek to gain a purchase on the way they emerge within OT systems along with the other not-always-obvious patterns that accompany them.

1.2 ERC, Value, Property: words of one syllable

Our start-up system recognizes one input, a single syllable, which is allowed to be either parsed into a foot (typically interpreted as *stressed*) or not. For reasons to be explained shortly, we name the system nGo-1s. Here is a sketch.

(1) nGo-1s.Gen
  a. *Candidate*: a candidate is a pair ⟨*in, out*⟩.
  b. *in*: a single syllable.
  c. *out*: a single syllable, either parsed into a foot, or unparsed.

From this vantage, 'syllable' is an atomic unit, as in purely quantity-insensitive stress phonology. We assume a familiar prosodic structure for an output, which consists of a single prosodic word that, in the general case we are not yet looking at, freely allows unfooted syllables as well as feet. For notational convenience we refer to that structure with the following condensed, linearized notation: X represents the head of a foot and o represents a unstressed syllable outside of a foot. Periods are used to demarcate prosodic units. In our system, then, we have the following entities.

(2) Structures of nGo-1s

Structure	Abbr.	Description
σ	o	input syllable
$[\ \sigma\ ]_{PrWd}$	.o.	prosodic word with single unfooted syllable
$[[\sigma]_F]_{PrWd}$	.X.	prosodic word with single monosyllabic foot

The broad linguistic sense behind this contrast has roots in both observation and theory. Substantively, across languages, it's a recurring observation that monosyllables often fall under tight restrictions and may only be allowed as clitics or the like. Languages which set a two-mora limit on content words show a related pattern (Prince 1980, McCarthy & Prince 1986, 1993a *et seq*.). This is interpretable as a consequence of intrinsic unstressability, which impacts the behavior of such forms in other components or strata of the overall grammar (P&S: 57ff). On the formal side, if prosodic theory allows unfooted syllables to appear *freely*, then letting everything be unfooted is an offer you can't refuse without special maneuvering, which we eschew. Thus, nGo-1s is positioned as one of the simplest within a hierarchy of increasingly ambitious stress typologies that embody well-founded (if not indisputable) assumptions about candidate footing patterns.

The constraint system is the one posited in *The Book of nGX* (Alber & Prince 2017), used in Chapters 4, 5, and (mostly) 6 of this work. Here's a concise tabulation.[6]

(3) **nGo-1s.Con**

Name	Abbr.	Definition	Verbose: *Returns a violation for*
a. Parse-σ	P-s	*o	each syllable not belonging to a foot
b. Trochee	Tr	*X.	each head-final foot .uX., .X.
c. Iamb	Ia	*.X	each head-initial foot .Xu., .X.
d. All Feet Left	AFL	*{σ, F}: σ...F	each pair {σ, F}, where σ precedes F
e. All Feet Right	AFR	*{σ, F}: F...σ	each pair {σ, F}, where F precedes σ

The familiar star-operator defines a constraint as a function from candidates to the non-negative integers ℕ = {0,1,…,}. It behaves exactly as in its use in the earliest OT literature. A constraint notated *K returns the number of matches to the pattern K in the candidate it applies to. In the case of faithfulness, not relevant here, the 'pattern' involves an input-output map. The formulation of the alignment constraints is based on Hyde (2012) and will be further explicated in Chapter 4. We will say that a constraint *penalizes* K, or *penalizes* a candidate for K, accumulating a *penalty* or *violation* for each instance of K.

Of interest in the present context are the first three constraints. We include the alignment constraints because they will become essential as soon as we step beyond this simplest system. The content of the constraints relevant to nGo-1s can be described verbosely as follows.

- Parse-σ (P-s) penalizes output syllables that lie outside a foot.
- Trochee (Tr) is so called, with an eye to conventional usage, because the only foot that does *not* provoke a violation is the bisyllabic trochee $[\sigma' \sigma]_F$. It penalizes every head-final foot.
- Iamb (Ia) follows the same naming convention. It penalizes every head-initial foot.

Here and in the stress systems we explore below, the monosyllabic foot .X. suffers the unique disadvantage of being both head-final (*Tr) and head-initial (*Ia). In much of the literature, it is assumed, or appears to be assumed, that the analogous foot-form constraints penalize only the *bisyllabic* foot of the disfavored headedness. Thus, classical 'Trochee' would be *.uX. in the standard notation, classical 'Iamb' would be *.Xu. Under these assumptions, which are not ours, the monosyllabic foot is effectively *favored* by both of these along with the (bisyllabic) foot after which the constraint is named. The monosyllable foot .X. is, on this view, *both* iambic and trochaic, rather than *neither*, as here. Penalizing unary .X. is left to an independent constraint, often called 'Foot Binarity'. In the constraints of (3), by contrast, the monosyllabic foot, since it is disfavored by both Ia and Tr, has nothing but Parse-σ to compel its appearance.

---

[6] For purposes of concord with the uses to come, we include the full definitions of Iamb and Trochee, which make use of the not-yet-relevant notation "u" for the nonhead of a Foot.

The name of the system 'nGo-1s' is mnemonically related to its founding premises. The 'n' of nGo alludes to the *new* anti-monopod definitions of Iamb and Trochee; the 'G' recalls 'generalized alignment', and the 'o' signals the fact that completely unparsed forms are admitted. The suffix '-1s' signals that the single admitted input has just one syllable.

1.2.1 Analyzing nGo-1s

The following violation profiles are obtained. The order of columns is not assumed to reflect any kind of ranking order, an aspect of OT 'best practices' that we will observe throughout this work.[7] As usual, an array of violation values will be called a *Violation Tableau* (VT).

(4) VT: nGo-1s (complete)

nGo-1s		P-s	Tr	Ia	AFL	AFR
/o/	.o.	1	0	0	0	0
	.X.	0	1	1	0	0

The two possible optima deliver the two languages of the typology.

(5) Languages of nGo-1s

Name	Optimum
L_o	o→.o.
L_X	o→.X.

From these data, we may construct the grammars G_o and G_X for the two languages, represented as ERCs presented in a *Comparative Tableau* (CT).

(6) CT: G_X for L_X

/o/	P-s	Tr	Ia	AFL	AFR
.X. ~ .o.	W	L	L	*e*	*e*

It is not difficult to scan the CT and derive its meaning.

• The constraint P-s favors the desired Winner (.X.) because it is parsed into a foot and its competitor (.o.) is not.

• The constraints Tr and Ia favor the desired Loser (.o.). The unparsed form .o. contains no feet of the wrong headedness by virtue of containing no feet at all, while its competitor (.X.) manages to offend both foot-type constraints, since it has both initial (*Tr) and final (*Ia) head location.

• Neither AFL nor AFR have anything to say, because both competitors are equally well aligned, indeed perfectly aligned, at both sides.

[7] See "Representing OT Grammars" (Prince 2017b) for the rationale.

Thus, in any linear order on the constraint set that selects .X. as optimal, the constraint P-s must dominate both of the foot-type constraints, lest one of them eject .X. in favor of .o., its competitor. The constraints AFL and AFR may appear anywhere. Any and all linear orders on nGo-1s.Con that meet these requirements lie in G_X and no others do.

It's instructive to look a little more closely at the calculations that lie behind this narrative. The competing pair [o→.X. ~ o→.o.] earns W on P-s because of the specific violation values of the two candidates as displayed in tableau (4). Writing C(**q**) for the value assigned by constraint C to candidate **q**, and C[**q**~**z**] for the comparative value rating **q** with respect to **z**, we have

P-s(.X.) = 0
P-s(.o.) = 1
∴ P-s[o→.X. ~ o→.o.] = W

Similarly, the constraints Tr and Ia assign L to the competition:

Tr(.X.) = 1 Ia(.X.) = 1
Tr(.o.) = 0 Ia(.o.) = 0
∴ Tr[o→.X. ~ o→.o.] = Ia[o→.X. ~ o→.o.] = L

Finally, neither candidate shows any alignment violations:

AFL(.X) = AFL(.o.) = 0
AFR(.X.) = AFR(.X.) = 0
∴ AFL[o→.X. ~ o→.o.] = AFR[o→.X. ~ o→.o.] = *e*

Parallel calculations produce the grammar of language L_o.

(7) G_o for L_o

/o/	P-s	Tr	Ia	AFL	AFR
.o. ~ .X.	L	W	W	*e*	*e*

Here the sense of the violation pattern is reversed: the targeted winner in L_o is exactly the loser in L_X. Thus, W and L are flipped, while *e* stays *e*.

1.2.2 Looking into the ERC

To read a CT in general, it's necessary to recognize that **q**~**z** always follows the order W~L, indicating that a desired Winner **q** is compared against a desired Loser **z**. Constraints construed comparatively evaluate not the individual contestants but the contest between them. Three outcomes are distinguished: W, *e,* L, according to which candidate betters the other, including neither.

(8) The comparative values

C[**q**∼**z**]	Numerics	Paraphrase
W	C(**q**) < C (**z**)	'C favors the desired winner **q** over desired loser **z**'
e	C(**q**) = C(**z**)	'C doesn't distinguish between **q** and **z**'
L	C(**z**) < C(**q**)	'C favors the desired loser **z** over the desired winner **q**'

The result of evaluating **q**∼**z** over all constraints is typically presented as an (arbitrarily ordered) list or 'vector' of values W, *e*, L, often displayed for convenience as the rows of a comparative tableau, as in (6) and (7). The value *e* is often omitted to avoid clutter.

The list or 'vector' of comparative values delivers the exact ranking requirements that must be satisfied by any total order on the constraint set if **q** is to survive comparison with **z** in the course of ordinary OT filtration. These requirements constitute an 'Elementary Ranking Condition' and the term is extended to the vector itself, abbreviated as ERC. It is a valuable feature of OT that valid ranking information is obtained from a minimal local comparison that ignores any and all other members of the candidate set that **q** and **z** belong to.[8]

The notion of a grammar is grounded in *optimality* and the conditions that ensure it. Understood in this way, the grammar of a language must denote *all* total orders on the constraint set that yield the same optima, not just one or some easily accessible subset. A grammar is not a single total order but an order *structure* of a certain formal type[9] and individual linear orders are consistent with it, or not. In recognition of this fact, we follow Merchant & Prince (2021) in referring to a single total order on the constraint set acronymically as a *leg*, abbreviating the expression ***l***inear ***e***xtension of a ***g***rammar, which we will typically symbolize by the letter λ, possibly with a subscript.[10]

It happens that the grammar G_o contains 40 legs while G_X contains 80. Though it's possible to construct something of interest from such numbers for extra-grammatical topics like learning and frequency of typological attestations, as in Riggle (2010) and Bane & Riggle (2008), or for historical change, as in Kiparsky (2014), what's important for the theory is that the grammar embodies the *requirements* for producing the optima of a language. The analysis of typological structure begins with those requirements.

[8] For discussion of this property in Social Choice Theory, see "Independence of Irrelevant Alternatives," Wikipedia.
[9] As Riggle (2010) reports and Merchant & Riggle (2016) prove, a grammar is an *antimatroid*, a structure that includes the partial order as a special subcase.
[10] The term 'linear extension' is borrowed from the theory of partial orders, where it refers to a total order that is consistent with the requirements of a given partial order.

For an ERC to express a ranking relation between 2 or more constraints, it must contain both W and L. The logical content of such an ERC reduces to the slogan "some W dominates every L." [11] Spelled out in detail, our examples look like this. We abbreviate the competition by omitting the shared, obvious input.

(9) ERC talk – you listen

ERC	P-s	Tr	Ia	AFL	AFR	Interpretation
.X. ~ .o.	W	L	L	*e*	*e*	P-s≫Tr *and* P-s≫Ia : P-s dominates *both* of {Tr, Ia}
.o. ~ .X.	L	W	W	*e*	*e*	Tr≫P-s *or* Ia≫P-s : *one of* {Tr, Ia} dominates P-s[12]

The reason behind this disjunctive/conjunctive logic is not far to seek. In any leg λ faced with competitors **q** and **z**, the highest-ranked constraint that distinguishes **q** from **z** decides the choice between them. Non-distinguishing constraints evaluate [**q**~**z**] to *e*.

For **q** to win against **z** in evaluation by some specific leg λ, the highest-ranked non-*e* constraint in λ has to evaluate **q**~**z** as W. Thus by linearity of ranking, in λ *all* L's must be subordinated to that highest-ranked W. Of the W's, any one will do, so long as it dominates every L. Given the OT definition of optimality, the array of W,L,*e* values determines a predicate of individual legs — *some W dominates every L* — which, when true of a given leg λ, ensures that q survives the competition with z adjudicated by λ, and when false, ensures that it doesn't.[13]

ERC use is an instance of the closer encounter with theory recommended above. As always, there's an overhead — incorporating something new into the skill set — but it comes with an immediate payoff for the practitioner: greatly increased visibility for ranking requirements imposed by the data. A more abstract consequence is that the relevant ranking conditions have actually been obtained, removing guesswork, intuition, and error.

Access to the right tools also plays a fundamental role in analysis. We will make use of the logical calculus based on manipulation of the W,L,*e* values, availing ourselves of its products rather than its inner details (Prince 2002a). By means of it, ERC representation supports methodical determination of entailment and contradiction relations among ranking conditions. Important also to the overall enterprise, as we will see, is the 'join' of grammars (Merchant 2008, 2011), an ERC-

[11] The core of the condition is "if there is a (constraint assessing) L in ERC α, then in any leg λ that satisfies it, that L is dominated by a (constraint assessing) W." Because legs are linear orders, in any given leg one W will dominate all the others. P&S introduce the idea of dealing with pairwise comparison in the notion of 'mark cancellation' pp. 153, 174, 258-262), used as a proof technique. Arithmetically construed, "mark cancellation" simplifies the values that a constraint assigns to each member of a pair of competitors by subtracting one of the values from each, to reduce at least one candidate's violations in each constraint to 0. When the violations differ, the smaller value is subtracted off. This leaves a pair of violation vectors. By contrast, the single ERC vector is arrived at arithmetically by subtracting the desired Winner's violations from the desired Loser's and then computing the sign of the difference. W indicates that the sign is positive, L negative, and *e* zero. This procedure arrives at the representation that is useful for developing the logic of OT, and the mark-cancelled vector pairs would also have to be subjected to it in order to arrive at that representation.

[12] I.e. *at least* one of; the implicit *or* is as always meant nonexclusively.

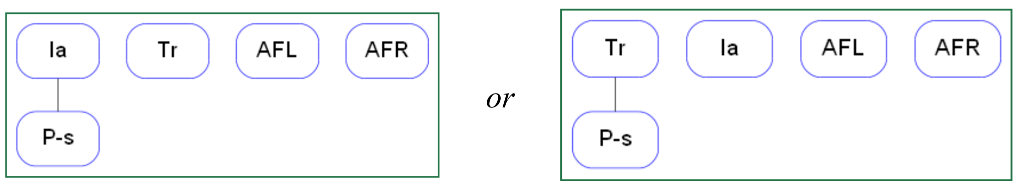
[13] Further aspects of ERC logic are taken up in §1.6 below, where OT is examined more closely.

combining operation that calculates the smallest formal grammar that contains the joined grammars, which can reveal crucial commonalities.

The ultimate virtue of the ERC is that it provides us with the means to represent grammars exactly and completely. Other familiar methods, like tableaux, even with dotted verticals, simply are not up to the job. The Hasse diagram works for some cases, not all, and as a diagram, is not suitable for analytical manipulation.[14] If the grammar is a focus of investigation, as it is in this work, the ERC set stands out as a primary object of interest.

The simple system nGo-1s exemplifies these assertions and leads beyond them. Observe first that the grammar of L_o — 'either Ia or Tr dominates P-s' — does not denote a partial order: the disjunction eliminates that possibility. It is also somewhere between risky and incorrect to think of a grammar broadly as a union of several partial orders. In the case of nGo-1s, this line of attack leads a representation like this:

(10) Caveat Lector: G_o portrayed as a union of partial orders



What dangers lie in this visual and apparently harmless portrayal?

First off, the constituent partial orders may overlap in the legs they denote, just as in the present case. A partial order on nGo-1s.Con that meets the sole requirement Ia$\gg$P-s also admits linear extensions in which Tr$\gg$P-s and even admits those in which Tr$\gg$Ia$\gg$P-s, as can easily be seen in the leftmost Hasse diagram. Thus it shares much with the other natural partial order, determined by Tr$\gg$P-s, displayed on the right, which unions with it to give the grammar G_o. Confusion is invited.[15]

Second, recovering the disjunctive set from a pair or more of Hasse diagrams requires a comparative visual scan that rapidly approaches real difficulty. But this data is readily accessible in the ERCs.[16]

Finally, it's just not the case that any two partial orders can be unioned to provide a grammar. The union must be describable as an ERC set. There's no getting away from it.

Any partial order can be represented by ERCs with just one W in each, but the general ERC is not so limited, either conceptually or ecologically. This is not unnatural within the kind of explanation provided by OT. Given the complexities of the real world, there can be no expectation that a given

[14] See Prince (2017b) for discussion of viable grammar representations.
[15] In particular, the confusion is to think of the left partial order as an abstractly "iambic" subgrammar, and the right hand partial order as abstractly "trochaic." *The Book of nGX* (Alber & Prince 2017) discusses the effects of adding extra rankings to render the disjuncts disjoint (Alber & Prince 2017: page indexed as "Mult.sp 1").
[16] The inutility of more elaborate schemes of Hasse-like notation, such as that employed in OTSoft (Hayes 2003-2017) is demonstrated in Prince 2006: 54-55. For recent discussion, see Prince 2017b.

effect must have a unique cause, or that distinct causal factors cannot overlap in their effects. In case at hand, the shared antipathy of Ia and Tr to monosyllabic feet means that either of them is sufficient to force non-parsing of monosyllabic input in L_o. A richer case emerges in more inclusive stress systems like nGX: the number of feet in a word can be limited by either of two distinct factors: a *word-level* structural requirement (alignment of feet to an edge), or by *foot-internal* considerations (the anti-iamb constraint Tr militates against iambs and therefore against multiple iambs, and vice versa). See Chapter 4 of this work, or *The Book of nGX* (sheetname: mult.sp (2)).

The system nGo-1s displays a subtler organizational feature, one that proves to be of pervasive importance in constraint systems: parallelism between constraints in the choice of optima. The constraints Ia and Tr are distinct elements in nGo-1s.Con, and they are freely ordered with respect to each other in the set of all orders, as are the other constraints in the system. Yet they function together: they form what we will call a *constraint class*. We can call the class 'FT', for 'foot-type'. Thus, FT = {Ia, Tr}.

In one sense, this is so obvious that its significance can easily be missed or myopically taken for granted. Do they not, in their formulations, refer to the same entities and mirror each other? But such observations are narrowly local to *S*.Con and do not guarantee that in the landscape of competitive choice, symmetry of definitions will be reflected in functional parallelism. Or that very different-looking definitions will not in the bypaths of filtration lead somewhere to the same or similar results. Optimality is determined not just by what stands out in the constraints-as-defined (*S*.Con) but also by what they have to choose among (*S*.Gen), and how the dynamics of choice proceed. In the case at hand, nGo-1s.Gen cooperates by treating iambic and trochaic feet with perfect symmetry. But more generally, a significant tour through the grammars may be required to establish parallelism or its lack. How *constraint classes* operate in the structure of typologies will be a main focus of our investigation.

Another feature revealed by this simple example is the binary oppositional structure of the ranking conditions — *properties* — that we will posit. The typology of nGo-1s is given by a single property, which consists of two mutually inconsistent *values*, each of which cashes out, in this case, to a single ERC that defines a grammar.

The negation of an ERC is obtained by switching W and L values, leaving *e* intact.[17] With more than one W or L in an ERC, de Morgan's Laws[18] lie in the background. Thus if α is WWL*ee*, its negation α* will be LLW*ee*, converting the disjunction of the first two W's in α into the conjunction of the same constraints, dominated, in α*. The logic runs like this, assuming a listing order of constraints in the cited ERCs as A, B, C.

---

[17] This identifies the native logic of OT as 3 valued, and 'paraconsistent', in that, for any length, there is an ERC such that the conjunction of that ERC and its negation is *not* a contradiction, namely the ERC containing neither W nor L, consisting entirely of *e*. Prince (2002a) shows that ERC logic is the 'relevance logic' RM3.

[18] Recall that de Morgan's Laws assert the following relations between negation, conjunction, and disjunction in propositional logic: (1) $\neg(P \,\&\, Q) \Leftrightarrow \neg P \vee \neg Q$, and dually, (2) $\neg(P \vee Q) \Leftrightarrow \neg P \,\&\neg Q$.

(11) Negating an ERC

	¬WWL = LLW	Justification
	¬(A≫C *or* B≫C)	Premise: ¬WWL
=	¬(A≫C) *and* ¬(B≫C)	de Morgan
=	C≫A *and* C≫B	Linearity of ranking

Observe that the *negation* of a ranking requirement like A≫C is, on this view, C≫A, 'C dominates A' rather than simply 'A does not dominate C'. The latter formulation leaves open the possibility that A and C are not ranked with respect to each other. But ranking requirements are predicates of individual *linear* orders. Thus, if A doesn't dominate C in a specific leg λ, it must be the case, due to linearity, that C dominates A in λ, exactly as shown in derivation (11).

1.2.3 Beyond the ERC: Property Theory

The ERC is indispensable for the representation of single grammars, but must be generalized to express other kinds of property values that delimit linguistically meaningful collections of grammars. A shift in perspective is required. An essential feature of the ERC that will be retained is its minimality as a ranking condition: putting aside the constraints about which it says nothing, it splits the constraints it controls into just *two* disjoint antagonistic classes: the dominating and the dominated.[19] But to determine how we interpret a nontrivial class of constraints in the context of a given leg, something further must be said: which member of the class is at play? For this, we look inside the classes themselves.

The key notion, we propose, involves the class-internal ranking relations within the leg. In particular, reference is made to the extremal positions in their ordering within the leg, as *dominant* in the class or as *subordinate* with respect to everything else in the class.

Consider a leg from G_o such as the following, in which the members of the class FT = {Tr, Ia} are bolded.

(12) λ_1 = AFR ≫ **Ia** ≫ AFL ≫ P-s ≫ **Tr**

In the leg λ_1, the constraint Ia is the *dominant* member of the class FT, in that it is ranked above *all* other members of the class (overkill: there's only one, but of course there could easily be more). To select this member from a leg, we introduce the operator ".dom" which we suffix to an expression denoting the class, writing

{Tr, Ia}.dom = FT.dom

[19] In principle, and in the literature, there exist more involved 'ranking schemata' which mention several conditions, some involving 3 or more levels of ranking. For example, in P&S:171 (232), the Possible Peak Condition is given as *P/☐ ≫ *P/α ≫ *M/α . This kind of condition will not be admitted as a property value in the present theory, but must represent some collocation of values.

This expression is meaningful only in the context of a given leg: within the constraint set nGo-1s.Con, neither Ia nor Tr can be identified as the dominator of the other. The same is true within the grammar G_o as a whole, which includes some legs in which Ia is the dominant member of FT and others in which Tr is dominant. Given a leg, however, 'FT.dom' unambiguously returns a single constraint. The expression 'FT.dom' therefore denotes a function from legs to constraints. For leg λ_1 in (12), we have

$$\text{FT.dom}(\lambda_1) = \text{Ia}$$

Similarly, as we proceed, we will find use for the dual operator ".sub" which attaches to a class designator and creates a function that, given a leg, returns the *lowest-ranked* member of the class in that leg. Thus:

$$\text{FT.sub}(\lambda_1) = \text{Tr}$$

In terms of the vocabulary of ordered sets, the function K.dom(λ) returns the *maximal* element of K as ordered in λ, and K.sub(λ) returns the *minimal* element. These notions arise in the analysis of the interaction of sonority levels and syllable structure in P&S, ch. 8, p. 182, where the expression *min*{PARSE, ONS, *M/□} is introduced to define "the least dominant of the three constraints." In our terms this is {PARSE, ONS, *M/□}.sub.[20] We use the dom/sub terminology because it accords transparently with the way that OT orders are described.

Our example grammar G_o shows an important feature of this notion. Consider a leg λ_2, in which Ia and Tr have swapped positions. But λ_2 is also a leg of G_o, satisfying the ERC [L.WW.*ee*] "Iamb or Trochee dominates P-σ" that defines it. The ERC is cited with the list order of constraints as in (9): P-s.Ia Tr.AFL AFR, with periods separating off the class FT = {Ia, Tr}, the members of which are bolded.

(13) λ_2 = AFR ≫ **Tr** ≫ AFL ≫ P-s ≫ **Ia**

Here, FT.dom(λ_2) = Tr. Seen from the dom/sub perspective, the generalization that unites both of these legs as elements of G_o is this: in every leg λ of G_o the dominant member of the class FT = {Tr, Ia}, *whatever it is*, dominates P-s in λ.

This new approach allows us to formulate the basic pattern to which a property value must, we claim, adhere: one constraint, determined by a class equipped with an operator, is compelled to dominate another constraint, also determined in the same way. A value will be notated concisely in the following manner, where K, J are classes, and op_1 and op_2 are, according to present understanding, either *dom* or *sub*.

(14) Format of Property Value. $K.op_1 > J.op_2$

---

[20] The operator max_s also appears, in an expression that returns the most sonorous segment s for which the constraint *M/s is dominated by the constraint C_{Ons},where C_{Ons} is the constraint *min*{PARSE, ONS, M/□}. Similarly for min_s.

For notational convenience, the operator is omitted in the trivial case where the class has a single element, so that we write the following for the value defining the grammar G_o, dropping the op on the singleton class {P-s}.

(15) Value defining G_o. FT.dom > P-s

What's meant is exactly the generalization suggested by our examples: for any grammar where this value holds true, in every leg of the grammar the constraint FT.dom dominates P-s.

- In leg $\lambda_1 \in G_o$, the value (15) holds because Ia dominates P-s.
- In leg $\lambda_2 \in G_o$, the value (15) holds because Tr dominates P-s.

More discursively, to spell out the details, we might write the value as a predicate of G, along these lines:

(16) $\forall\lambda\in G$ FT.dom $(\lambda) >_\lambda$ P-s (λ) .

This explicitly spells out the requirement that in every leg λ of G, the constraint FT.dom(λ) dominates P-s in the domination order $>_\lambda$ of the ordered set of constraints λ.

Since we've examined just 2 legs of the 80 in the grammar G_o, we need to justify the claim that value (15) truly and completely characterizes the grammar. To show this, we relate it to the ERC that is known to tell the whole story: L.WW.*ee*. (For readability, we insert periods setting off the members of FT.)

- Value ⇒ ERC. Observe first that if FT.dom dominates P-s in a leg λ, then it follows without complicated reasoning that *some* member of FT = {Tr, Ia} dominates P-s in λ, as required by the ERC.
- ERC ⇒Value. If, in accord with the ERC requirement, *some* member of FT dominates P-s in a leg λ, then surely FT.dom(λ) does. If a nondominant member is cited as a witness validating the some-member-dominates assertion, then because FT.dom(λ) is ranked above the witness, it too must dominate, by transitivity of ranking.

These considerations show that the value 'FT.dom> P-s' is entirely equivalent to the ERC defining G_o.

What then of G_X? We claim that the ranking-reversed value is exactly equivalent to the ERC W.LL.*ee* that defines it.

(17) Reversal of value. P-s > FT.dom

The argument goes like this:

- Value ⇒ERC. If P-s dominates FT.dom in λ, then since FT.dom is ranked above all other members of FT in λ, it must be that P-s also dominates them by transitivity of ranking.
- ERC⇒Value. If, as per the ERC, P-s dominates *all* members of FT, then by Aristotle it surely dominates FT.dom, which is one of them.

Here again, there is complete equivalence between the ERC and the .dom-based value.

1.2.4 The Property Analysis of nGo-1s

We can now spell out the full property analysis of nGo-1s, introducing the symbol '<>' to indicate that the two values of the property A<>B are A>B and B>A. We refer to A and B as the *antagonists* of the property. We will call the property 'Unarity', abbreviated to 'Un', in recognition of its focus on unary feet and their lack.

(18) Property analysis of nGo-1s

a. class FT = {Tr, Ia}
b. prop Un = FT.dom <> P-s traits: monosyllables are unfooted/footed

Since a principal goal of a property analysis is to explicate the traits observed in the languages of the typology in terms of the rankings that produce them, we list these as well. Because of the simplicity of the example, the value-associated traits are not far to seek.

- Value Un.o: The value FT.dom>P-s, which we name 'o', determines complete lack of feet in the output.
- Value Un.X: The value P-s > FT.dom, which we name 'X', determines full parsing into feet in the output.

When we wish to include explicit mention of the value names, we write the property as 'Un.o/X', ordering the value names in a way that correlates with the left-to-right/right-to-left reading of the expression FT.dom <> P-s.

As noted, the completeness and correctness of the claimed analysis have been confirmed by showing that each value is equivalent to the ERC that defines the denoted grammar.

We have seen that the general property form K.dom<>J.dom, for constraint classes K and J, is by our arguments equivalent to the ERC in which the constraints K∪J are exactly and exhaustively those receiving W and L. Recall that we are allowing ourselves to cite unit classes in a property expression without an attached (and trivial) operator.

- In the value K.dom>J.dom, K is the W-set and J the L-set of the corresponding ERC.
- In the value J.dom>K.dom, the W-set is J and the L-set is K.

This indicates that the .dom/.sub formulation is remarkably well-behaved with respect to 'negation' of values: the only thing that happens in the Property is the swapping of sides.

1.3 Public Classes, Treeoids: the system nGoX-2s

Following the incrementalist methodology outlined above, let's see what a mildly expanded version of our first system has to tell us about typological structure and the shape of Property Theory. This will expose the relation between properties and classes, useful modes of representing a property analysis, and the potential multiplicity of available analyses.

The tactical advance is to include both one and two syllable forms in an expanded system. We will therefore call the new system *nGoX-2s*, referring to the maximum length admitted.

A key decision in articulating systems richer than nGo-1s concerns the extent of footlessness allowed in supra-monosyllabic candidates. Here we will limit foot-lack to monosyllables, though later we will explore in detail what happens when it is allowed for all lengths (Chapters 5 and 6). The restriction to monosyllables echoes a common linguistic pattern where, broadly, as in nGo-1s, monosyllables are only allowed as clitics and similar morphosyntactic dependents but, within the component or stratum where stress is determined for unstressed inputs, polysyllables require the presence of stress. See, e.g., Prince (1980) and the extensive development in McCarthy & Prince (1986, 1993a *et seq*.) for the notion 'minimal word'.

(19) nGoX-2s.Gen

a. *Candidate*. A candidate is a pair ⟨*in, out*⟩.

b. *in*. A string of one or two syllables

c. *out*. A string of same syllable length as input: if 2 σ must contain a foot; if 1 σ, then either parsed or unparsed.

As before, 'syllable' is an atomic unit, and the same familiar prosodic structure for outputs is assumed. Each output consists of a single prosodic word but only monosyllables may lack a foot entirely. Words σσ must contain a foot and 'X' in the system name signals this limitation. To our condensed, linearized notation we add "u" to indicate the nonhead of a foot. In nGoX-2s, then, we have the following word-like entities.

(20) nGoX-2s.Gen:

	Structure		Abbr.	Description
inputs	1	σ	o	input syllable
	2	σσ	oo	input bisyllable
outputs	1.2	$[\ \sigma\ ]_{PrWd}$	.o.	monosyllabic prosodic word with single unfooted syllable
	1.2	$[[\sigma]_F]_{PrWd}$	.X.	monosyllabic prosodic word with single monosyllabic foot
	2.1	$[(\sigma'\ \sigma)_F]_{PrWd}$	.Xu.	trochaic bisyllable
	2.2	$[(\sigma\ \sigma')_F]_{PrWd}$	.uX.	iambic bisyllable
	2.3	$[(\sigma')_F\ \sigma]_{PrWd}$	.X.o.	bisyllable beginning with monosyllabic foot
	2.4	$[\sigma\ (\sigma')_F]_{PrWd}$	.o.X.	bisyllable ending with monosyllabic foot
	2.5	$[(\sigma')_F(\sigma')_F]_{PrWd}$	.X.X.	bisyllable with two monosyllabic feet

To repeat: "oo→.o.o." is *not* a candidate in this system. That's what 'X' in nGoX-2s refers to.

The constraints of the system nGoX-2s are the same as those used in nGo-1s, as in ex. (3).

(21) Defining *S*.Con. nGoX-2s.Con = nGo.1s.Con

Violations are assigned to the admitted candidates as follows:

(22) Violation Profiles of all nGoX-12s candidates

input	output	**P-s**	**Tr**	**Ia**	**AFL**	**AFR**
o	.o.	1	0	0	0	0
	.X.	0	1	1	0	0
oo	.Xu.	0	0	1	0	0
	.uX.	0	1	0	0	0
	.X.o.	1	1	1	0	1
	.o.X.	1	1	1	1	0
	.X.X.	0	2	2	1	1

The last three candidates from /oo/ are harmonically bounded by both of the first two, as may be easily checked. Observe that .Xu. and .uX. have but one violation (green), but each of the last 3 candidates incur those violations (pink) as well as others.

The optima from the two candidate sets combine freely, yielding a typology of 4 languages, named and tabulated here.

(23) The Extensional Typology of nGoX-2s

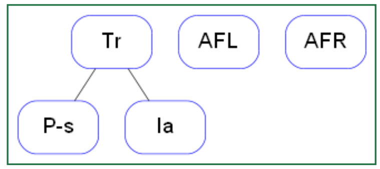
Lg. Name	/o/	/oo/
tr.o	.o.	. Xu.
ia.o	.o.	.uX.
tr.X	.X.	. Xu.
ia.X	.X.	.uX.

The grammars of these 4 languages are as follows. For visual clarity, *e*'s have been omitted. Since all grammars are partial orders, Hasse diagrams are shown adjacent to an ERC representation given in the form of a Skeletal Basis (SKB: Brasoveanu & Prince 2005/2011), which is maximally concise both in the number of ERCs it deploys and the number of L's it specifies, removing all L's that are deducible from transitivity of ranking. The Hasse diagram also displays no transitively-derivable information, so this brings the SKB as close in structure to a Hasse diagram as the facts of the theory will allow.[21]

[21] The SKB is the (transpose of the) incidence matrix of the directed hypergraph that describes the grammar, in which the constraints label the vertices and the ERCs label the hyperedges. For a recent overview of hypergraphs, see Ausiello & Laura (2017). Note that the "Primitive Ranking Condition" (PRC) of Prince 2007a, those in which an ERC has only one L, have the same structure as Horn Clauses, which are discussed in Ausiello & Laura, where the W-set of the PRC is hypergraphically the same as the antecedent of the Horn Clause, with the interpretive difference that elements of the antecedent are conjoined in the Horn Clause case. H/T to Paul Smolensky for the term "facts of the theory."

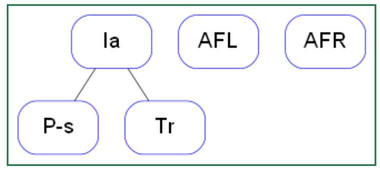
(24) tr.o

P-s	Tr	Ia	AFL	AFR
L	**W**	**L**		



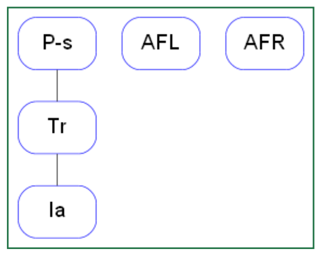
(25) ia.o

P-s	Tr	Ia	AFL	AFR
L	**L**	**W**		



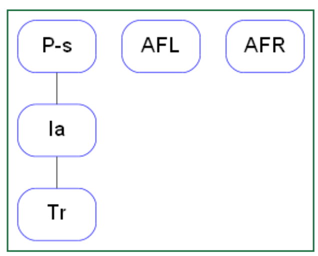
(26) tr.X

P-s	Tr	Ia	AFL	AFR
W	**L**			
	W	**L**		



(27) ia.X

P-s	Tr	Ia	AFL	AFR
W		**L**		
	L	**W**		



Two natural contrasts in extensional traits present themselves:

- Parsed vs. unparsed monosyllables (o/X)
- Trochaic foot vs. iambic foot in bisyllables (tr/ia)

Together they distinguish all the languages, and we've used this fact in naming them. We will see that they submit to explication via property analysis.

The first contrast is carried forward from nGo-1s: distinguishing the members of {tr.o, ia.o} from those of {tr.X, ia.X}. A quick scan of grammars (24) and (25) reveals that the o-value is identical to that of Un.o in nGo-1s: the highest-ranked member of FT = {Tr, Ia}, whatever it is, dominates P-s.

- val Un.o = FT.dom > P-s

An equally quick check shows that the opposite value is manifest in grammars (26) and (27), in which P-s conspicuously dominates both member of FT.

- val Un.X = P-s > FT.dom

The second trait contrast is determined within the constraint class FT itself. Every grammar has a two-syllable candidate set in which the choice between Ia>Tr and Tr>Ia must be made, and therefore the two constraints are the antagonists in a property of their own.

• prop FT = Tr <> Ia

In nGo-1s, the *class* FT stands by itself as an independent component of the analysis. In nGoX-2s, the FT constraints Ia and Tr play a richer role: thus, the class FT may be derived from the property FT: the class collects the antagonists of the property. To distinguish such a property from the constraint class consisting of its antagonists, we will always mark the property with a prefix 'p' and the class with a prefix 'c'. A constraint class like c.FT will be called 'public', since it is derived from a property p.FT. A 'private' class is an auxiliary structure referred to in the property analysis, but not itself derived directly from a property. For nGo-1s, the class c.FT is private, but for nGoX-2s, it is public. This demonstrates that both kinds of classes must be available to the theory.

The following gives a property analysis (PA) of nGoX-2s, based on these observations.

(28) PA for nGoX-2s

a. prop	p.FT.tr/ia	= Tr <> Ia	traits: feet are trochaic/iambic
b. prop	p.Un.o/X	= FT.dom <> P-s	traits: monosyllables are unfooted/footed
c. class	c.FT	= {Tr, Ia}	

A grammar is obtained by choosing a value for each property and conjoining the results. The typology is obtained by making every such choice, delivering the four grammars. For ease of reference, values will be named in a way that recalls their content and distinguishes them from other values. For example, the values of p.FT will be called 'tr' and 'ia'. When we wish to be fully informative, we will write p.FT.tr/ia as the complete descriptor. Along the same lines, we'll write p.Un.o/X. I

It is useful to be able to tabulate the results to see how the typology emerges. A table laying out the property analysis of nGoX-2s comes out like this:

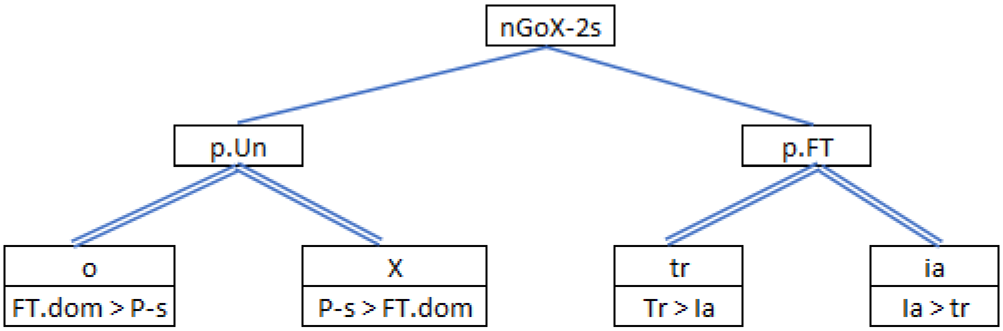
(29) PA table for nGoX-2s

nGoX-2s	**p.Un**	**p.FT**
tr.o	o	tr
ia.o	o	ia
tr.X	X	tr
ia.X	X	ia

In more generic circumstances, the values can less perspicuously be called from their formulations, using "a" for left-to-right and "b" for right-to-left, so that p.FT.a is Tr>Ia and p.FT.b for Ia>Tr, and the table filled accordingly. Throughout, the OTWorkplace property analysis tools will be used to check and tabulate our analyses.

A related notation, which will become more useful as we proceed, is the *property treeoid*.

(30) Property treeoid for PA of nGoX-2s



The intended interpretation is that the single lines terminate in properties of which values must be chosen and the double lines mark the inconsistent values of a single property, where one and only one is chosen. The choices combine freely with each other, yielding the 4 grammars.

This property analysis has been driven by the goal of explicating the traits of the languages in terms of the ranking structure that yields them. An immediate question, then, is whether the typology admits of other analyses, driven only by the possibilities of construing the ranking structure with Property Theory as developed so far. In fact, since an ERC forms a valid property with its negation, and since two of the four grammars are given by single ERCs, we can readily build an analysis from the grammars themselves, cutting across the o/X split we have presumed so far. Let us call the new analysis PA_2.

The first new property of PA_2 will split the grammar tr.o from all others by simply transmuting its grammar [L.WL.*ee*] (24) "Tr dominates Ia and P-s" into a property, accomplished by matching it with its negation [W.LW.*ee*] (25) "P-s or Ia dominates Tr." This requires recognizing the private class c.IP = {Ia, P-s}. We name the values in this case by a version of the grammar name for one of them, marking its complement with the prefix '¬'.

(31)	prop	p.TrIP.tro/¬tro	=	Tr < > IP.dom	traits: tr.o *vs*. all others
	class	c.IP	=	{Ia, P-s}	

Paralleling this, we add a property that splits the grammar of ia.o from all others.

(32)	prop	p.IaTP.iao/¬iao	=	Ia <>TrPs.dom	traits: ia.o *vs*. all others
	class	c.TP	=	{Tr, P-s}	

We must retain p.FT, though within this analysis, c.FT is no longer a recognized class, either public or private.

(33)	prop	p.FT.tr/ia	=	Tr < > Ia	traits: feet are trochaic/iambic

The analysis may be tabulated as follows:

(34) PA_2 table for nGoX-2s

PA_2	p.TrIP	p.IaTP	p.FT
tr.o	tro	¬iao	tr
ia.o	¬tro	iao	ia
tr.X	¬tro	¬iao	tr
ia.X	¬tro	¬iao	ia

This multiplicity of analyses reflects the generic situation. PA_2 could even be right — for another system, with a different trait structure and a different constraint content.

But if we accept that manifesting 'o' in monosyllables defines a linguistically significant grouping of languages — is an authentic trait — then the first PA of nGoX-2s gives a direct account, and PA_2 does not. Those grammars satisfying the value p.Un.o carry the trait— while PA_2 can only characterize the ranking condition that selects the trait as p.TrIP.o ∨ p.IaTP.o, a disjunction of values that identifies no commonality, because it is based on a linguistically meaningless class structure.

Deciding between alternative trait analyses and alternative property analyses requires an assessment of the viability of each as well as of the match between them. As noted above, this kind of global reckoning provides unusually strong motivation for attending to the theory. Slippage anywhere can distort not just the local analysis, but the overall understanding of emergent structure, where the linguistic claims reside.

1.4 Sub, Mootness and Scope: the system nGo-2s

To complete the overview of Property Theory, we examine a third abstract stress system, obtained by a different decision about how nGo-1s is expanded to encompass bisyllabic input. The analysis displays two new structural characteristics that round out the basic vocabulary of Property Theory: the use of the .sub operator, mentioned above; and, appearing for the first time but central to the overall enterprise, the limitation of a property's contrast to a subdomain of the typology, which involves what we will call its *scope*.

1.4.1 The typology of nGo-2s

To obtain the new system nGo-2s, we admit unfooted output in bisyllables as well as monosyllables. The full extension of this freedom to all lengths leads to the system nGo, studied in detail in Chapter 5. The assumed prosodic theory of feet is the same as before, but the Gen component of the system is modified in clause (c) below to allow unrestrained (non-)parsing.

(35) nGo-2s.Gen

a. *Candidate*: a candidate is a pair ⟨*in, out*⟩.

b. *in*: one or two syllables

c. *out*: a prosodic word of same syllable length as input, freely foot-parsed or unparsed

The admitted structures are the same as those of nGoX-2s, with the addition of a single new candidate involving the bisyllabic output .o.o., which we will call the 'null parse'.

(36) Structures of nGo-12s

	Structure		Abbr.	Description
inputs	1	σ	o	input syllable
	2	σσ	oo	input bisyllable
outputs	1.1	$[\ \sigma\ ]_{PrWd}$	.o.	monosyllabic prosodic word with single unfooted syllable
	1.2	$[[\sigma]_F]_{PrWd}$	.X.	monosyllabic prosodic word with single monosyllabic foot
	2.1	$[\sigma\ \sigma]_{PrWd}$	.o.o.	footless bisyllable (null parse)
	2.2	$[(\sigma'\ \sigma)_F]_{PrWd}$	.Xu.	trochaic bisyllable
	2.3	$[(\sigma\ \sigma')_F]_{PrWd}$	.uX.	iambic bisyllable
	2.4	$[(\sigma')_F\ \sigma]_{PrWd}$	.X.o.	bisyllable beginning with monosyllabic foot
	2.5	$[\sigma\ (\sigma')_F]_{PrWd}$	.o.X.	bisyllable ending with monosyllabic foot
	2.6	$[(\sigma')_F(\sigma')_F]_{PrWd}$	.X.X.	bisyllable with two monosyllabic feet

The new output .o.o. (2.1) engenders the possibility that *all* output forms of a language may lack feet. It is not uncommonly assumed, if tacitly, that a (natural) language under scrutiny lacks feet; the premises of nGo-2s.Gen make this a formal typological reality. In addition to such ready visibility, footlessness may also be posited in the internal strata of phonology in the manner of Lexical Phonology. (See McCarthy & Prince 1993 for stratified grammar under OT; subsequent development includes Kiparsky 2014 and Bermúdez-Otero 2010.) We thus regard it as a hypothesis anchored in recognizable linguistic considerations. Setting aside the question of unassailable truth, it is quite revelatory in the present context to see how the assumption shapes the typology nGo-2s that it sits within.

The constraint system nGo-2s.Con is exactly the same as the ones used above.

(37) nGo-2s.Con = nGoX-2s.Con (21) = nGo-1s.Con (3)

The following violation profiles are assigned to the candidates. As with nGoX-2s, bisyllabic candidates with monosyllabic feet .X. are harmonically bounded by those with bisyllabic feet.

(38) Violation Profiles of all nGoX-2s candidates

input	output	P-s	Tr	Ia	AFL	AFR
o	.o.	1	0	0	0	0
	.X.	0	1	1	0	0
oo	.o.o.	2	0	0	0	0
	.Xu.	0	0	1	0	0
	.uX.	0	1	0	0	0
	.X.o.	1	1	1	0	1
	.o.X.	1	1	1	1	0
	.X.X.	0	2	2	1	1

The typology admits one new language in addition to those of nGoX-2s: the footless language we call *nil*, with unparsed forms .o. and .o.o. as outputs. The 5 languages of nGo-2s are these:

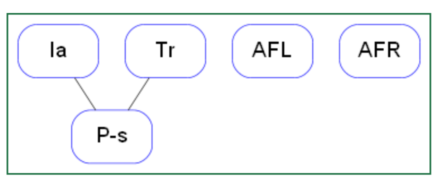
(39) Extensional Typology of nGoX-2s

Name	/o/	/oo/
nil	.o.	.o.o.
tr.o	.o.	. Xu.
ia.o	.o.	.uX.
tr.X	.X.	. Xu.
ia.X	.X.	.uX.

The following are the grammars. Because they are all partial orders, each can be given in both ERC form and as a Hasse diagram. To clarify the ranking structure in the comparative tableaux, we include the L's derived by transitivity but enclose them in parentheses. To clarify the class structure, we explicitly indicate the location of FT = {Ia, Tr}, a constraint class that carries over intact from nGoX-2s.

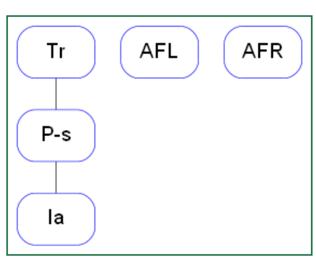
(40) nGo-2s: *nil*

	FT			
P-s	Tr	Ia	AFL	AFR
L		**W**		
L	**W**			



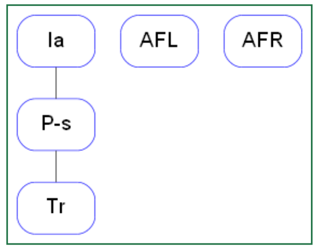
(41) nGo-2s: tr.o

	FT			
P-s	Tr	Ia	AFL	AFR
W		**L**		
L	**W**	**(L)**		



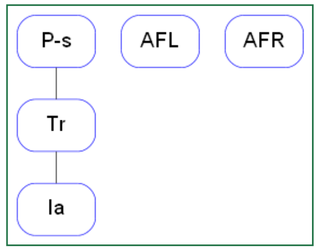
(42) nGo-2s: ia.o

	FT			
P-s	Tr	Ia	AFL	AFR
W	**L**			
L	**(L)**	**W**		



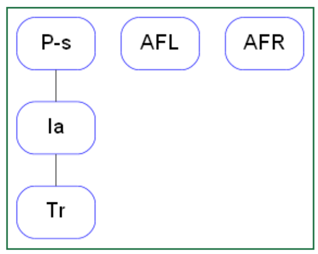
(43) nGo-2s: tr.X

	FT			
P-s	Tr	Ia	AFL	AFR
W	**L**	**(L)**		
	W	**L**		



(44) nGo-2s: ia.X

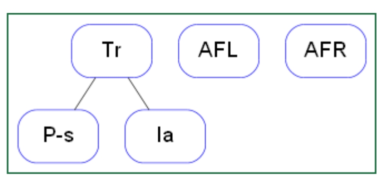
	FT			
P-s	Tr	Ia	AFL	AFR
W	**(L)**	**L**		
	L	**W**		



The new bisyllabic candidate leads to the new language *nil* and therefore impacts the grammatical structure of other languages in the typology. Putting aside footless *nil*, the four foot-bearing languages of nGo-2s are extensionally identical to those of nGoX-2s, consisting of either .o. or .X. in the monosyllables and a footed bisyllable, either iambic or trochaic. But the *grammars* of languages with optimal .o. differ in the two systems, because of the presence/ absence of .o.o. as a 2σ competitor. Here are the nGoX-2s grammars of tr.o and ia.o, repeated from (24) and (25). A foot is required in 2σ forms by nGoX-2s.Gen.

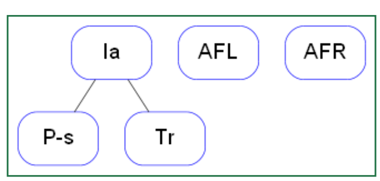
(45) nGoX-2s: tr.o

P-s	Tr	Ia	AFL	AFR
L	**W**	**L**		



(46) nGoX-2s: ia.o

P-s	Tr	Ia	AFL	AFR
L	**L**	**W**		



All that's required in nGoX-2s is that the dominant foot-type constraint be ranked above both P-s and the subordinate foot-type constraint. By contrast, in nGo-2s, the grammars of tr.o and ia.o require a total order on the constraints {P-s, Tr, Ia}. Some legs that yield tr.o in nGoX-2s, namely those containing Tr≫Ia≫P-s, end up within the grammar of *nil* in nGo-2s. The same is true, symmetrically, for ia.o.

(47)

System	Language	Grammar Req.	Ref.
nGoX-2s	tr.o	Tr ≫ Ia ≫ P-s Tr ≫ P-s ≫ Ia	ex. (45)
nGo-2s	*nil*	Tr ≫ Ia ≫ P-s	ex. (40)
	tr.o	Tr ≫ P-s ≫ Ia	ex. (41)

The reason is not far to seek. In nGo-2s, the grammar of tr.o must fend off the null parse .o.o., which is not even a candidate in nGoX-2s. In nGo-2s, Tr views both .o.o. and .Xu. as perfect, and the ranking of Ia and P-s is required to distinguish them. The same holds for Ia with respect to .o.o. and .uX. , swapping Ia for Tr in the argument. Thus, each of tr.o and ia.o surrenders a leg, as it were— the one with P-s at the bottom — to create the grammar of *nil*.

This effect illustrates a subtlety of emergence and the value of assessing all interacting components accurately.[22] Conditions in the 2σ candidate set — whether .o.o. is admitted as a candidate — determine aspects of the grammar of those languages which share a 1σ optimum.

1.4.2 The Property Analysis of nGo-2s: Being and Nothingness

An eyeball check of (41)-(44) shows that the property p.FT = Tr<>Ia splits the last four (somewhere foot-requiring) grammars, all but the grammar of *nil*, in exact accord with the way they are labeled for foot type. Similarly, p.Un = FT.dom<>P-s splits the footful four as labeled, along the o/X dimension. This establishes that the property structure found in nGoX-2s is carried forth into the grammars of the four languages it shares with nGo-2s.

The possible optimality of the null parse .o.o. is the hallmark of the new system. Its role may be discerned in the following VT, where the constraint order is rearranged for convenience of discussion. Harmonically bounded candidates have been removed, as they contribute nothing to ranking.[23]

[22] For related discussion of the deleterious effects of overlooking possible optima as competitors, even for grammars in which they are not optimal, see Bane & Riggle 2012.
[23] As first recognized in Samek-Lodovici & Prince (1998).

(48) Violation profiles of the 2σ optima, nGo-2s

		FT				
input	output	Tr	Ia	P-s	AFL	AFR
oo	.o.o.	0	0	2	0	0
	.Xu.	0	1	0	0	0
	.uX.	1	0	0	0	0

The fate of the null parse depends crucially on the relation between P-s and the class FT = {Tr, Ia}.[24] Observe that the violation profiles of the three constraints show the simple 'diagonal' pattern in which each candidate is penalized by only one constraint.[25] As noted in P&S:112, the optimum is determined by "whichever of the [three relevant constraints] is lowest" in the legs of the grammar.

We submit that the relation between the 3 distinguishing constraints should be understood as fundamentally binary and class-driven.

[1] When P-s is ranked beneath *both* members of FT, the parsed forms .uX. and .Xu are each eliminated by the contrary foot form constraint (Tr and Ia, respectively). The null parse is optimal.

[2] When ranked above at least one member of FT, P-s eliminates the null parse and a footed form is optimal.

These observations give the conditions under which the null parse attains optimality (or not) and thus exactly characterize the one new property that appears in nGo-2s. To spell it out, we appeal to the operator *sub,* which when attached to class designator K, defines a function that returns the lowest ranked ('subordinate') member of the class K in a leg to which it is applied.

We call the new property ExF, for 'Exist Foot'. Its two values are these, accompanied by an informal paraphrase:

(49) Prop ExF

name	value	informal	trait
nil	FT.sub > P-s	Ia & Tr ≫ P-s	all optima lack feet
hf	P-s > FT.sub	P-s ≫ Ia or Tr	some optimum has feet

The name *hf* is mnemonic for 'have a foot'. The value nil delivers the entire grammar of the nil language, as in (40), repeated here for convenience.

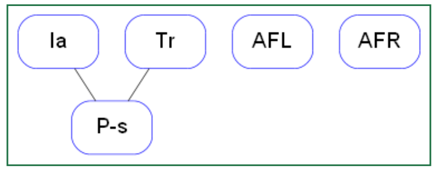
---

[24] With the null parse absent from the universe of the possible, as previously for nGoX-2s in (22), the choice is strictly between .uX. and .Xu., which differ only in their performance on Ia and Tr. The optimum is decided by whichever of those constraints dominates the other, according to the value p.FT.tr/ia, and P-s fits in anywhere.

[25] See P&S:ch. 6 for an example of this violation pattern in syllable theory, taken up again in Chapter 3 below.

(50) ExF.nil: FT.sub > P-s

	FT			
P-s	Tr	Ia	AFL	AFR
L		**W**		
L	**W**			



Neither of these representations refers explicitly to the subordinate member of FT = {Ia, Tr}. Rather, they both say "Ia and Tr dominate P-s." But this is equivalent to the value expression, as may be seen by examining any leg of which the value is true.

- When the subordinate member of the class FT = {Ia, Tr} dominates P-s in some leg λ, it must be the case that *any and all* other members of the class FT dominate P-s in that leg. This comes about because all members of a class dominate the subordinate member by definition, and so by transitivity of ranking, they must dominate anything that it dominates. So the value expression entails the ERC/Hasse representations of (50)
- Conversely, if all members of a class dominate some other constraint, then surely the subordinate member does. Thus, the ERC/Hasse representations entail the value expression.

Reversing and thereby negating the nil value yields the value hf = P-s> FT.sub. There is a similar translation into logic talk: "P-s dominates some member of FT." This may be understood directly from the definition of the operator .sub: when P-s dominates FT.sub in a leg, it is necessary (if unsurprising) that P-s dominates *some* member of FT. Conversely, if some member of a class is dominated by a given constraint, then the subordinate member is surely dominated by it as well. The logical interpretation may also be approached by applying de Morgan's Laws to the interpretation of the value ExF.nil. For example, from "¬(Tr *and* Ia dominate P-s)", we obtain "P-s dominates Tr *or* Ia," using the fact that ¬(A≫B) in λ ensures B≫A in λ, because λ is a linear order.[26]

These findings resemble those noted for the .dom operator, but they reverse the role of conjunction and disjunction.

- In the .dom case, as in the 'o' value of FT.dom<>P-s = Un.o/X, the expression FT.dom>P-s requires (disjunctively) that '*some* member of FT dominates P-s.' This shows the ERC pattern of disjunction among the dominators.
- The value FT.sub>P-s, by contrast, encountered above as ExF.nil (50), comes out (conjunctively) as '*every* member of FT dominates P-s'. This can be rendered as an ERC *set*, whose individual members conjoin to produce the global requirement defined by the set, but cannot be reduced to a single ERC.

Flipping P-s to the other side creates the opposing value in each case.

- The .dom-based value P-s>FT.dom = Un.X yields 'P-s dominates *every* member of FT, following the ERC pattern.

[26] See (11) above for a more detailed version of a similar argument.

• The .sub-based value P-s>FT.sub = ExF.hf yields 'P-s dominates Tr *or* Ia', which is expressible neither by ERC nor ERC set. In an ERC, the dominated are conjoined: some W dominates *every* L. And in an ERC set, for example (50), the ERCs are conjoined, not disjoined. Negating them creates, by de Morgan, a disjunction of (negated) ERCs.

The presence of the sub operator in a value expression thus takes us beyond the descriptive capacity of both the single ERC and the ERC set, which suffice to represent grammars but not the collocations of grammars that are denoted by Property values. A property with .dom on both sides is exactly an ERC; a property with .sub on either is not.[27] The vocabulary of Property Theory, as promised at the outset, thus properly includes and generalizes the ERC.

1.4.3 Mootness and Scope

To complete the basic analysis of nGo-2s, we recall that the four languages whose grammars satisfy the value ExF.hf — those which 'have a foot' in some optimal form — are exactly the four languages of bisyllabically-footed nGoX-2s. The grammar of the nil language consists of those legs that accord with ExF.nil, namely those in which both FT constraints dominate P-s, stripping them, as it were, from the grammars of ia.o and tr.o as they appear in nGoX-2s. But despite this difference in leg content, the same properties, aside from ExF.nil/hf, are at play.

(51) Prop p.ExF.nil/hf

value nil	FT.sub > P-s	trait: all optima lack feet
value hf	P-s > FT.sub	trait: some optimum has feet

(52) Prop p.Un.o/X

value o	FT.dom > P-s	trait: monosyllables are unfooted
value X	P-s > FT.dom	trait: monosyllables are footed

(53) Prop p.FT.tr/ia

value tr	Tr > Ia	trait: (binary) feet are trochaic
value ia	Ia> Tr	trait: (binary) feet are iambic

For a grammar to satisfy a value, *all* of its legs must individually satisfy it: this is definitional. For example, the nil language has 40 legs, and in every one of them both of the constraints Ia and Tr dominate P-s, meeting the requirement of the value ExF.nil, namely FT.sub>P-s.

A single leg satisfies one value of a property, or it satisfies the other: it cannot satisfy both: given the two constraints denoted by the antagonists of a value, one must dominate the other, simply because a leg is a linear order. For the same reason, a leg cannot satisfy *neither* value.

[27] Assuming of course nontrivial classes that contain more than one constraint.

But at the grammar level, it is entirely possible for some legs of a grammar to go one way while others go the other. In the nil language, for example, the relation between Ia and Tr is free, since there are no bisyllabic feet to force the choice of head position. Thus, in 20 of its legs, Tr>Ia, and in the other 20, Ia>Tr. Half of the nil legs are formally iambic, half formally trochaic, yet no extensional traits distinguish these orderings.

We call this logical situation *mootness*, and say that a property P.a/b is logically *moot* in a grammar if some of its legs satisfy P.a and others satisfy P.b.

(54) **Mootness**. For a property P.a/b and a grammar G, if there is $\lambda_1 \in G$, where P.a(λ_1) is true, and there is also $\lambda_2 \in G$, where P.b(λ_2) is true, then P.a/b is *moot* with respect to G.

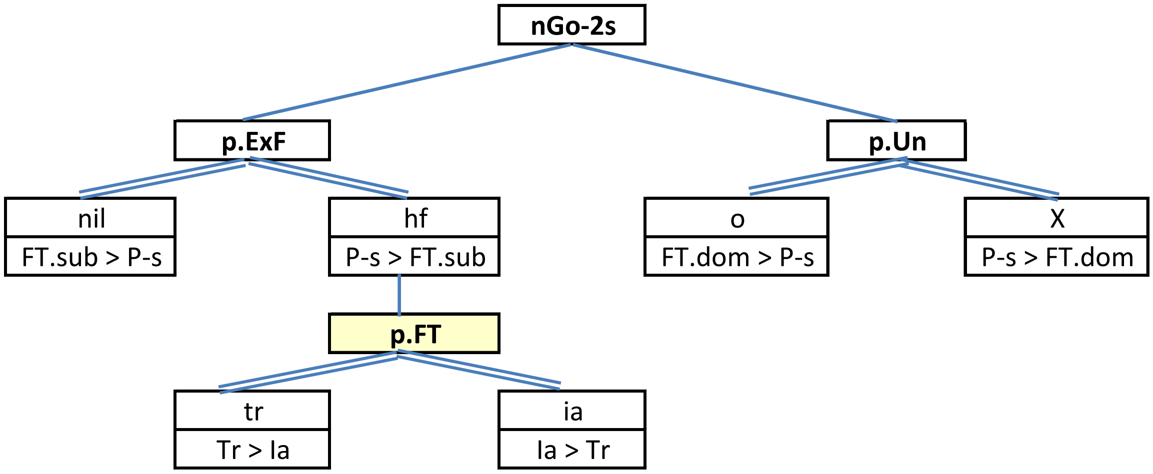
A property that is moot for some grammar must be excluded from its property analysis. For example, the property p.FT = Tr<>Ia can only be allowed to play a role in grammars that require feet in bisyllables: namely, those specified with the value ExF.hf = P-s>FT.sub. In this case, one value — hf — leads to further articulation, while grammars bearing the other value are immune to it.

If p.FT.tr/ia is allowed to play out freely over ExF.nil, two intensionally distinct grammars "nil.tr" and "nil.ia" will be generated, which are extensionally identical. This is directly contrary to the basic notion of a grammar as the set of all legs that have the same optima. We speak of the morning star and the evening star, but we do not base planetary astronomy on the assumption that they are distinct objects.

Mootness threatens to over-articulate grammars. How is this to be avoided? Let's identify the *scope* of a property as a set of other properties over which it has free play, thereby specifying grammars. The basic rule of combination, then, is that the values of a property conjoin only with the values *within its scope*. In the case of nGoX-2s, where bisyllables are unavoidably footed, each property is in the scope of every other and all values combine freely. In the case of nGo-2s, the property p.ExF cannot be in the scope of p.FT. The treeoid provides a natural way to represent the phenomenon of scope as it manifests in nGo-2s, through the dependence of a property on the choice of a specific value of another property.

Here's a treeoid that correctly delimits the scope of p.FT.

(55) Property Treeoid for nGo-2s



As a dependent of ExF.hf , the property p.FT is isolated from ExF.nil, and is not allowed to combine with it. Observe that p.Un has p.FT in its scope, allowing for the contrast in grammars between tr.o and tr.X, ia.o and ia.X.

Since the values of p.FT are available when and only when hf is chosen as the value of p.ExF, the property p.FT articulates only the grammars bearing the value hf. By contrast, a grammar bearing the nil value of p.ExF (far left) excludes the contradictory value hf and therefore its substructure; it is *outside the scope* of p.FT. Treeoid structure thus embodies a natural notion of scope that avoids a proliferation of value combinations denoting nonempty leg sets that are not full-fledged grammars.

The appearance of mootness is shown explicitly in the following property analysis table, calculated in OTWorkplace.

(56) PA table for nGo-2s

nGo-2s	p.ExF	p.Un	p.FT
nil.o	nil	o	*moot*
ia.o	hf	o	ia
tr.o	hf	o	tr
ia.X	hf	X	ia
tr.X	hf	X	tr

The table discloses further logical structure which is relevant to the notion of scope. As we've just seen, the absence of grammars "nil.ia" and "nil.tr" is due to mootness. *Both* values of p.FT are encountered in various legs delivering the single language nil = {o→.o., oo→.o.o.}, and therefore its grammar cannot be specified for either, since specification for a value requires that all legs of a grammar satisfy it. But the typology also manifests a lack of contrast between nil.o and nil.X: only *one* value of the pair Un.o/Un.X shows up attached to nil. For this, the reason is different: the bisyllabic optimum .o.o. is not allowed to co-occur with monosyllabic .X., because the value conjunction ExF.nil & Un.X is logically inconsistent, quite independent of any extensional considerations about languages.

In treeoid (55), the property p.Un.o/X is assigned *wide scope*, since it depends on no values. Consequently, its values are predicted by the rule of value combination to combine with p.ExF, which sits in its scope.

The nonexistence of nil.X follows not from the mootness of some property, but from the internal logic of the two values 'nil' and 'X'. The contradiction in their demands is not far to seek.

(57) A logical interaction of sub and dom

Value	Definition	Interpretation
p.ExF.nil	FT.sub > P-s	'both members of FT dominate P-s'
p.Un.X	P-s > FT.dom	'P-s dominates both members of FT'

Though a language 'nil.X' ={o→.X., oo→.o.o.} is entirely feasible as a collocation of optima from different candidate sets, it is ruled out by the logic of the property analysis, taken with the strictness of the ranking order. In no leg can the constraint P-s both dominate and be dominated by the same constraints.

Because a grammar is defined to be a nonempty set of legs, we can allow free combination of the values of p.ExF and p.Un, as the scopal arrangement of treeoid (55) requires. The collocation nil.X is allowed but denotes no grammar. Thus, the set of grammars denoted by the treeoid is exactly the set of grammars that forms the typology, as desired.

But there is no pressing need to allow p.Un to interact with p.ExF at all. The 'o' value in nil.o carries no information: it is already implied by 'nil'. This may be seen by reformulating the contradiction argument into an entailment argument.[28]

(58) Logic II: Entailment

Value	Definition	Interpretation
p.ExF.nil	FT.sub > P-s	'both members of FT dominate P-s'
p.Un.o	FT.dom > P-s	'some member of FT dominates P-s'

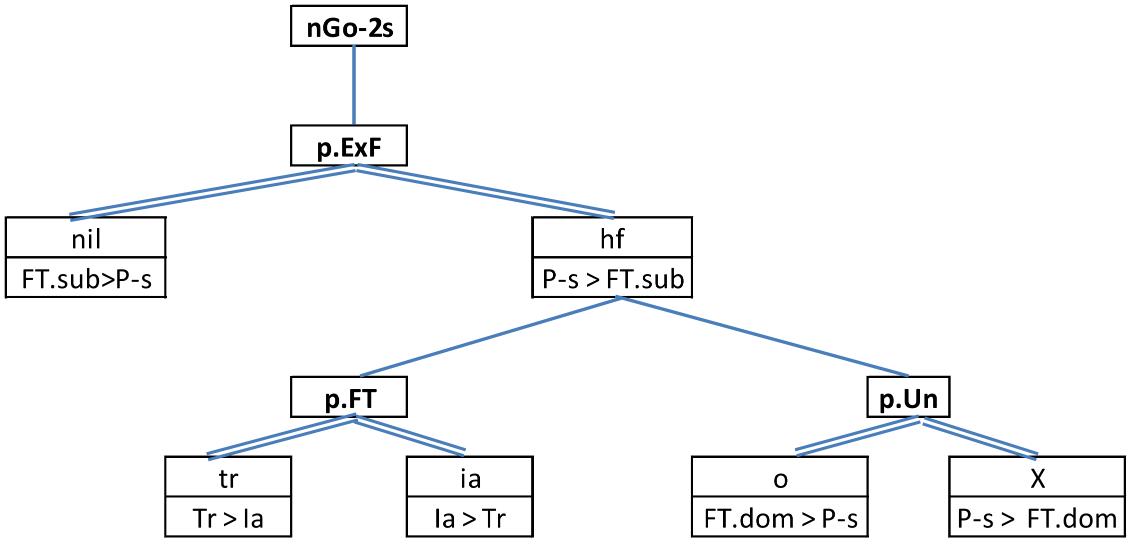
If something holds of every member of a nonempty set, it holds of *some* member of that set.

We are therefore free to take another view of the scopal relations, one which limits p.Un to the scope of ExF.hf. Under this assumption, p.Un .o/X supplies a value only where the opposite value

[28] This maneuver is, of course, always possible, if sometimes cognitively opaque. If A&B are inconsistent, then ¬(A&B) is true, and this is equivalent to A→¬B. Observe that p.Un.o = ¬ p.Un.X.

may be nontrivially supplied as well. A reconfigured property treeoid provides a direct account of this decision.

(59) Re-scoped PA of nGo-2s: scope of p.Un narrowed



In the re-scoped PA (59), both p.FT and p.Un are dependents of ExF.hf. They are in each other's scope, but the value ExF.nil lies outside the scope of both. We can rewrite the PA table to reflect this arrangement, should we wish to. We write ⟨*name*⟩ to indicate a value that is entailed and withdraw coloration from its cell.

(60) Renotated PA table for re-scoped PA

nGo-2s	p.ExF	p.Un	p.FT
nil	nil	<o>	*moot*
ia.o	hf	o	ia
tr.o	hf	o	tr
ia.X	hf	X	ia
tr.X	hf	X	tr

At this point, there is little reason to choose between these alternatives PAs, although as understanding of the organization of typologies advances, criteria may emerge to settle the question. It's worth noting that the narrow scoping of p.Un in the treeoid (59) and table (60) tracks the contrast structure of the traits quite closely, where its wide scope placement in the (55)/(56) version appeals more aggressively to logic. Thus, if the treeoid is tied closely to the expression of trait structure, giving narrow scope to p.Un will be motivated. But if it's regarded as a costly stipulation to impose narrow scope beyond what's absolutely required by mootness, then the wide scope version would be favored. It will become clear as we proceed that being able to adjust scope with respect to entailed values is essential to revealing a meaningful, trait-sensitive treeoid structure in typologies.

We note that *scope* as used here is represented in very much the way that scope is represented in other disciplines and subdisciplines that appeal to the notion. The scope of a property is the set of nodes that it c-commands in the treeoid. And the parent node of a narrow-scope property in treeoids like (55) and (59) defines the domain in which c-command is defined.

Scope is the analytic interpretation of mootness and entailment, and as such must be distinguished from them, since both are purely logical and inflexible, given the assumptions of the analysis. Patterns of mootness in an analysis vary with assumptions about the content of properties, as do patterns of entailment and contradiction. It is possible to find analyses in which mootness appears in ways that radically defy a tree-like assignment of scope; see Chapter 3 of this work. In examples in our own work not reported here (e.g. Alber 2017, Alber & Arndt-Lappe, submitted), and in cases studied in the literature, such as in Bennett & DelBusso (2018), the domain of scope has been found to be defined by more than one value, in Boolean combination, in a way that is representable by multi-domination in the treeoid.

For example, in their exploration of the typologies generated by different definitions of ABC(D) constraints, Bennett & DelBusso (2018) find that in several systems (T_{GG}, T_{GR}, T_{RG} and T_{RS}, pp. 19, 26, 28, 30 respectively), the scope of some properties is best described as a *disjunction* of values. Another example is provided by Danis (2014), which finds *conjunction* of values to be necessary. See DelBusso (2018, 12ff. and throughout) for further discussion. The typology nBo in Chapter 6 of this work, based on Alber (2005), provides particularly rich instance. Setting the limits on what a scope domain can be therefore plays an important role in restricting what a property analysis can be. The treeoid is starting-point for this investigation.

With scope in place as the final essential element in the vocabulary of Property Theory, we may write down a complete analysis of nGo-2s. We will write scope by citing the value for which a property must be decided, its domain: thus, A<>B /**v** indicates that the property defined by A<>B lies within the domain of value **v** (or more generally, the value combination **v**), taking scope over other properties in that domain. When there is no limitation on the scope, we describe it as 'wide' and may omit mention of it for conciseness.

(61) PA of nGo-2s: as (55), with p.Un wide

Status	Name	Antagonists	Scope	Traits
prop	p.ExF.nil/hf	FT.sub <>P-s	/wide	No optimum/some optimum has a foot
prop	p.Un.o/X	FT.dom<>P-s	/wide	1σ optimum lacks feet/is footed
prop	p.FT.tr/ia	Tr<>Ia	/hf	2σ feet are trochaic/iambic

(62) PA of nGo-2s: as in (59), with p.Un narrow

Status	Name	Antagonists	Scope	Traits
prop	p.ExF.nil/hf	FT.sub <>P-s	/wide	No optimum/some optimum has a foot
prop	p.Un.o/X	FT.dom<>P-s	/hf	1σ optimum lacks feet/is footed
prop	p.FT.tr/ia	Tr<>Ia	/hf	2σ feet are trochaic/iambic

Since the class c.FT is public, it is not explicitly included in the PA definition.

The first analysis (61) of nGo-2s, represented in treeoid (55), is identical the second (62), represented in treeoid (59) in all respects but the scope assigned p.Un.

Progress to date can be iconically summarized in the PA tables of the typologies we've seen.

(63) nGo-1s: ERC, class, value, Property, dom

nGo-1s	p.Un
G_o	o
G_X	X

(64) nGoX-2s: sub, public class, treeoid

nGoX-2s	p.Un	p.FT
tr.o	o	tr
ia.o	o	ia
tr.X	X	tr
ia.X	X	ia

(65) nGo-2s: mootness, scope

nGo-2s	p.ExF	p.Un	p.FT
nil.o	nil	o	*moot*
ia.o	hf	o	ia
tr.o	hf	o	tr
ia.X	hf	X	ia
tr.X	hf	X	tr

(66) nGo-2s: entailment, scope

nGo-2s	p.ExF	p.Un	p.FT
nil	nil	<o>	*moot*
ia.o	hf	o	ia
tr.o	hf	o	tr
ia.X	hf	X	ia
tr.X	hf	X	tr

1.5 Concise Outline of Property Theory

Property Theory offers a step toward understanding the macrostructure induced by the interactions of the atomic components of OT, those that when defined yield an OT *system*. The three related typologies just examined are not so complex as to present insuperable barriers to understanding, yet they suffice to bring out the formal elements of the theory. The goal of this section is to provide a focused account of the concepts introduced above.

To begin, we distinguish between a set of structures and mappings, a *language*, and the *grammar* generating them. These each sit within a *typology*,[29] which projects the totality of consequences following from the basic assumptions of a system. An *extensional* typology is a set of languages, organized by shared and distinguishing *traits*. An *intensional* typology is a set of grammars.

The principal goal of Property Theory is to explicate the extensional traits in terms of the intensional conditions that give rise to them, regarding these as macrostructure that a typology displays. The various components of the explanation, often ambiguous in themselves, shed light on each other, particularly when analyzed with full systematicity. We treat trait structure as accessible through the linguistic background, leaving open its details, which we expect to become clearer as the enterprise proceeds. Central to the work is a concrete proposal as to the character of the intensional macrostructural conditions that characterize typologies at this level.

The core analytical notion is the pairing of two mutually inconsistent ranking conditions or *values* into a *property*. A value is a Boolean predicate of a single leg. A *leg* is a linear order on the constraint set, so called because a grammar is an order structure that resembles and generalizes a partial order (an 'antimatroid') and we may speak of its linear extensions, the total orders that are consistent with it. Thus, for any leg λ and any value V, $V(\lambda)$ is either true or false, while the opposing value $\bar{V}(\lambda)$ is respectively false or true. For any λ, and for a property with values V and $\bar{V}$, either $V(\lambda)$ or $\bar{V}(\lambda)$ holds, but not both. The values are expected to correlate with traits and explain how they emerge.

Properties *combine* to generate all cross-property conjunctions of their values. A property analysis of a typology is a collection of *properties* sufficient, under their rules of combination, to generate every grammar in the typology and no grammar that does not belong to the typology.
Our proposal is that the value takes on a fixed, simple form which generalizes the irreducibly basic notion of one constraint being required to dominate another. An *elementary value* is an expression like Tr>Ia, which, given a leg λ, asserts that in the domination order $\gg_\lambda$ of that leg, the constraint Tr dominates the constraint Ia. The opposing value is Ia>Tr. Clearly, for any given leg, one of these is true and the other false, exactly as required. Together, they form an *elementary property*.

[29] This term is stripped down from the *factorial typology* of P&S. There is an implicit distinction of emphasis between our usage and theirs. While any construction of OT must recognize the 'factorial' totality of rankings on a given *S*.Con, just as any theory must live with the totality of its consequences, the original term tends to direct attention toward the languages and away from the organization of the ranking set into grammars. A grammar is the entire set of linear orders that deliver the same optima, or, equivalently and much more usefully, a set of conditions defining that set of orders. We hope the balance will be restored by use of the unadorned term *typology*. In the background is *linguistic typology*, used to refer to the study of attested patterns in natural languages, which provides an important component in the assessment of grammatical proposals. Here too, our use of the bare term *typology* is meant to address an implicit imbalance, emphasizing typology as the characterization of linguistic possibility, no matter how derived.

The key to generalization beyond the elementary is to recognize that constraints, individually elements of OT microstructure, function together in *classes* as components of the macrostructure. The chief analytical burden that this imposes is determining which element(s) of a multi-constraint class may function in a given circumstance.

In our examples, the constraints Tr and Ia behave as a class in every system discussed, which we call FT. In one usage, they function interchangeably. Thus, in p.Un, for the value o, we have FT.dom>P-s, requiring that either Tr or Ia dominates P-s in any leg that satisfies it; one constraint suffices and each is as good as the other, when in the right position in a given leg. In another usage, both are required to share a ranking relation: for the opposite value p.Un.X, we have P-s>FT.dom, requiring that Tr and Ia are both dominated by P-s.

This type of functional unity of diverse constraints is already implicit in the notion of the ERC, the fundamental ranking requirement imposed by pairwise comparison of candidates, one of which is asserted to be better than the other in the OT sense. The ERC recognizes a W-set of constraints which are locally interchangeable in that any one may do the domination work of any other in the pairwise comparison at hand; an L-set of constraints that must all be dominated (by a member of the W-set); and an *e*-set of constraints that view the competitors as equivalent. Property Theory recognizes the disjunctive and conjunctive roles, but extends them beyond their ERCwise correlation with domination and subordination respectively.

To achieve this, the theory shifts focus to ranking order *within* classes. Two dual operators are posited: *dom* and *sub*. (We acknowledge the possibility, indeed the certainty, that other ways of accessing the contents of a class will reveal themselves as knowledge of typological structure widens.) Given a class and a leg, these define functions that return the constraint in the leg which occupies one of the two extremal positions: for a class K, we define K.dom(λ) to the constraint of K that dominates all the other members of K in λ, and K.sub(λ) to be the constraint of K that is subordinate to (i.e. dominated by) all the other members of K in λ. From (12) and (13), we have:

(67) Let λ_1 = AFR ≫ **Ia** ≫ AFL ≫ P-s ≫ **Tr**
FT.dom (λ_1) = Ia
FT.sub (λ_1) = Tr

(68) Let λ_2 = AFR ≫ **Tr** ≫ AFL ≫ P-s ≫ **Ia**
FT.dom (λ_2) = Tr
FT.sub (λ_2) = Ia

A more exact account would run as follows.

(69) The dom operator. For λ a linear order on *S*.Con, and K⊆*S*.Con, K.dom(λ) is the unique C∈K such that for every D∈K, C≠D, $C \gg_\lambda D$.

(70) The sub operator. For λ a linear order on *S*.Con, and K⊆*S*.Con, K.sub(λ) is the unique C∈K such that for every D∈K, C≠D, $D \gg_\lambda C$.

An even more exact account would recognize that $K \subseteq S.Con$ is not quite the final story. We ultimately want a constraint class K to be allowed to contain expressions like H.dom and H.sub, which themselves are built from classes. Thus, a *class* is itself defined relative to a leg, and we should properly write K(λ) or something similar, to denote the subset of *S*.Con obtained by evaluating all expressions in K with respect to λ.

Using the class concept, a general *value* is denoted by a somewhat elliptical expression of the form $K.op_1 > J.op_2$, where op_1 and op_2 are either dom or sub. The expressions $K.op_1$ and $J.op_2$ are called the *antagonists* of the value.

(71) Value Expression: $op_i \in \{dom, sub\}$

$$K.op_1 > J.op_2$$

When K or J are singleton sets, as in an elementary value, the choice of operators is immaterial and mention of them is omitted. Values like those of p.Un.o and p.ExF.nil are stated like this:

FT.dom > P-s
FT.sub > P-s.

In the general value (71) just as in an elementary value like Tr>Ia, the expressions $K.op_1(\lambda)$ and $J.op_2(\lambda)$ each denote a single constraint. But in the general case, the constraint denoted varies with the leg, as in exx. (67) and (68).

Observe that when K and J consist only of constraints, so that $K,J \subseteq S.Con$, a value K.dom>J.dom is exactly an ERC (see §1.2.3, § 1.4.2). In all cases, the classes K and J are required to be disjoint, as discussed in Chapter 3 of this work. This is another respect in which the value displays an ERC-like character.

At the grammar level, a value is held to be true if and only if it is true of *every* leg in the grammar. This commitment also echoes the way an ERC is interpreted: an ERC holds of a grammar if and only if every leg of the grammar satisfies that ERC.[30] In more detail:

(72) A value $V = K.op_1 > J.op_2$ holds of a grammar G iff $\forall \lambda \in G, V(\lambda)$.

Putting these notions together, we arrive at the notion of a *property*, which pairs two opposing values related by reversal of the domination order of the antagonists. A generic property is written as follows, with the caveat noted above that trivial classes containing a single constraint are not supplied with operators.

(73) Property. $K.op_1 <> J.op_2$

[30] Comparison between an optimum of the grammar and a single competitor gives veridical information about the entire grammar, independent of the behavior of any and all other candidates. In an ERC representation of a grammar, the requirement is that every leg of the grammar satisfy every ERC in the set.

This compresses the two values into one expression. It may be expanded along these lines.

(74) Value Structure of a Property

prop P.a/b

value a $K.op_1 > J.op_2$

value b $J.op_2 > K.op_1$

Strikingly, a class may consist of the antagonists of a property: $\{K.op_1, J.op_2\}$. In this case, we call it a *public class*. An example is FT = {Ia, Tr} in nGoX-2s (28) and nGo-2s (61)-(62), which collects the antagonists of the property FT = Ia<>Tr. When not, we call it a *private class*. An example is FT in nGo-1s (18).

To distinguish a class from a property, which may be otherwise have the same name, we use classifying prefixes: p.K denote the property K and c.K denotes the class K.

Scope is the final crucial component of the theory. Scope provides a way to sequester properties and prevent them from over-articulating values where they are extensionally irrelevant.

A property combines freely with the properties within its scope; outside its scope, neither of its values can be specified. In nGo-2s, for example, the property p.FT does not combine with ExF.nil, so that no grammars nil.ia or nil.tr are generated.

When a property combines with every other property, we describe its scope as *wide*. Scope can be narrower than *wide*, and, as we will see as we proceed, much narrower and narrower in interesting ways.

To get a sense of how this works out, consider two properties P.a/a* and Q.c/c*. If they are both wide scope, i.e. both unrestricted, or if one is in the scope of the other, the following 4 conjunctions emerge ($2\times2 = 4$).

(75) Free Combination of P.a/a* and Q.c/c*

a & c

a & c*

a* & c

a* & c*

Now assume a scopal restriction that limits specification of Q to grammars where P.a*. Then we have only these 3:

(76) Licit value combinations when Q takes scope /P.a*

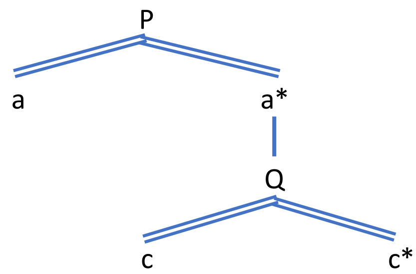
a

a* & c

a* & c*

We propose that scope restrictions are given in terms of property values. In the example, values Q.c and Q.c* always co-occur with P.a*, never with P.a. A tree structure can represent this relation as one of dependency. Positioning Q as a daughter of P.a*, we can see that Q.c/c* lies within the domain of a*.

(77) Property Q within the domain of value P.a*



This is a substructure of (what we call) a 'treeoid', a rooted, acyclic graph, with two kinds of nodes (excepting the root, which is merely a label for the whole structure), and two kinds of branches: one type (shown double) connecting a property to its mutually inconsistent values, and the other type (shown single) connecting a property to a dominating value or to the root. The treeoid defines a set of conjunctive choices from its value nodes, with the double lines descending from a property node indicating that one or the other of its value nodes must be chosen, and the single line indicating that a property must be decided and specified when its parent node is specified. Every licit combination of choices is admitted, and collectively they generate the entire property analysis. See (30), (55), and (59) for fully articulated examples.

In the cases examined here, the scope is restricted to the domain of a single value. As noted above, scope can involve more complicated logical conditions on values; in typologies that have been studied to date, one finds both disjunction and conjunction of values, as well yet more complex conjunctive and disjunctive dependencies, which can be represented by multi-domination. In this work, we will not be examining such cases, although in Chapter 6., we look at a particularly interesting case of complex determination of scope by values, which points to further directions of development.

Scope is indispensable to the theory because of the way that *satisfying a property value* is defined for legs and grammars. A leg satisfies a value when its order structure is consistent with the requirements of the value, and every leg either meets or fails those requirements: given a leg λ, and the two constraints A, B picked out in V(λ), either $A \gg_{\lambda} B$ or $B \gg_{\lambda} A$ and not both.

A grammar, by contrast, is defined to satisfy a value only when *all* of its legs meet the condition that the value imposes. A simple typology like nGo-2s suffices to show that a value may be true of some of the legs of a given grammar while its opposing value is true of the others. In this case, the grammar itself satisfies *neither* of the two values. Thus, satisfaction of properties by grammars is intrinsically tripolar. For a property P.v/v* and grammar G, we have these patterns:

(78) The three patterns of satisfaction

- v(G) = T and v*(G) = F. Consequence: G may be specified for v.
- v(G) = F and v*(G) = T. Consequence: G may be specified for v*.
- v(G) = F and v*(G) = F. Consequence: G *may not* be specified for either v or v*.

In nGo-2s, for example, the nil language lacks feet altogether and therefore its grammar includes legs in which Tr>Ia as well as otherwise identical legs in which Ia>Tr, with no extensional distinction in the optima for either class of legs. Neither of the values of p.FT.ia/tr is true of nil.

The property p.FT is therefore said to be *moot* in the nil grammar. Since all the grammars generated by a property analysis must belong to the typology, a property cannot be allowed to play out over the domain in which it is moot, for it will split grammars into distinction-without-difference twins, of which 'nil.tr' and 'nil.ia' would be examples. The value p.ExF.nil must reside *outside the scope* of p.FT to maintain the match between grammar and language.

Another logical characteristic of property analyses that interacts with scope possibilities is the existence of entailment and contradiction relations between values of different properties. As we have seen in the study of nGo-2s, the specification for a value of one property can entail specification for a certain value of another property, and exclude counter-specification. In that case, the effect is due to the inherent logic of the dom/sub operators.

(FT.sub>P-s) ⇒ (FT.dom>P-s) *i.e.* ExF.nil ⇒Un.o

The entailment relation is clear from the interpretations of the values.

- "If the *subordinate* constraint in the class FT dominates P-s in some leg, then the *dominant* constraint does as well."
- Equivalently, "if *every* constraint in the class FT dominates P-s, then *some* constraint in FT does."

The immediate consequences is that ExF.nil and Un.X cannot coexist. Only nil.o denotes a nonempty set of legs, a grammar, while nil.X denotes the empty set, which is not a grammar.

Schematically, if for some properties P.a/a* and Q.c/c*, we have the entailment relation

P.a ⇒ Q.c,

we must also have, by logic, that

P.a & Q.c*

is contradictory and describes nothing.[31]

In such cases, the analyst is free to entertain two distinct hypotheses about scopal structure.

- **Wide**. Let P and Q freely combine, with at least one in the scope of the other. Under this assumption, the nonexistent combination P.a & Q.c* is ruled out on grounds of logic by its failure to denote a grammar. Three combinations are admitted:

a & c
a* & c
a* & c*

[31] Recall that $p \Rightarrow q$ is equivalent to $\neg(p \wedge \neg q)$. Therefore, if $p \Rightarrow q$ is true, so is $\neg(p \wedge \neg q)$, and therefore $p \wedge \neg q$ is false.

• **Narrow**. Let the scope of Q be limited to P.a* so that the fruitless pairing P.a & Q.c* never arises. Under this assumption, the three admitted combinations are these:

a
a* & c
a* & c*

In the case of nGo-2s, there is little to favor one scopal alternative over the other. But the distinction is not without a difference in the broader context of scopal relations, their theoretical limits and empirical interpretation. Here's a broad analogy: we know that A>B and B>C entails A>C, by transitivity of ranking, but the logical fact that A>C is entailed leaves open the question of whether A>C is extensionally significant, motivated by candidate contrasts, which must be ascertained by examining *S*.Con rather than the theory of linear order. We shall see in richer cases that the freedom to adjust scope so as to include or exclude entailed values is crucial to being able to construct a tree-like analysis structure that is responsive to trait patterns.

Scope, then, is asserted as a construal of mootness and entailment relations among property values, crucial to construing the relation between intensional and extensional analyses of typological structure. Hypotheses about what scope can be will therefore play a significant role in limiting what a property analysis can be.

To conclude, we emphasize again that the fundamental goal of the property analysis of an individual typology is to explicate the emergence of trait patterns in a system *S* from the micro-structural assumptions about candidate structure (S.*Gen*) and constraint structure (S.*Con*). Our focus throughout is on the intensional structure given by the properties and their scope, but the trait structure, about which we have as yet less to say, plays an essential role throughout. Indeed, we expect that the traits correlated with property values will re-construct the languages with the completeness that combination of values generates the grammars.

1.6 What is meant by *OT* ?

LIKE PARENTS, WHOSE GREATEST INFLUENCE on their offspring occurs at the moment of conception (S. Pinker, p.c.), or Joyce's artist who "like the God of creation, remains … invisible, refined out of existence, indifferent, paring his [*or her* -BA & AP] fingernails," the theorist sets the theory on its unswerving course when the premises are laid down and, through study, must then endeavor to catch up with the consequences. It can take a while.

The basic premises and objects of OT are laid out here so that we may proceed with some confidence to retrieve the patterns and structures they entail. What Merchant (2015) calls "the tedium of specificity" need not however preclude insight or intuition. We begin with a reasonably careful presentation of the theory as it is understood in this work and proceed to a characterization of the various notions of language, grammar, typology that emerge from it. We conclude with a global overview of the transit from microstructure to macrostructure that motivates and frames the present work.

1.6.1 Gen, Con, Eval

OT is a theory of choice, and as such it must spell out [1] the alternatives to be chosen among, [2] the criteria that determine the choice, and [3] the method by which the diverse demands of multiple criteria are unambiguously adjudicated. Any instantiation of OT thus rests on these three pillars:

[1] Gen: defines the structures and mappings — *candidates*, in the jargon — that are to be evaluated, and how they are organized into *candidate sets*, from each of which the optimal candidates are selected.

[2] Con: spells out the criteria of decision between competing candidates

[3] Eval: defines what it means to be optimal.

These become more narrowly OT-specific as we proceed down the list. Every theory of linguistic form defines its assumed structures and their possible relations (Gen): in OT, these are the alternatives to be chosen among. The criteria that rate various structural and mapping possibilities reside in Con. Eval defines what it means to be optimal, given the alternatives and the criteria, and thereby distinguishes OT from other theories that utilize the very same versions of Gen and Con.

Even at this broad level of generality, certain basic methodological consequences are apparent. To be *optimal* means that there is nothing better. To establish optimality, then, *all* admitted alternatives must be evaluated.[32] Evaluation itself rests on the set of criteria: *all* of them must be considered, at the risk of invalidity; if any are irrelevant to the choice at hand, they must be shown to be so.

These considerations may seem harmlessly general, but they force us to be not only specific but also deliberate in the course of theory development. We cannot argue validly in a landscape

[32] Even those which can never be optimal because there is always something better must be detected and dismissed.

where the alternatives or the criteria are incompletely known. OT must therefore be studied within what we have called the *system*, a well-defined object given by spelling out Gen and Con — not for all of language, but for that system in itself. A system *S* is defined as ⟨S.Gen, S.Con⟩, and it is within such a system that 'optimal' becomes meaningful.

In this introductory chapter we have examined three *systems*: nGo-1s, nGoX-2s, nGo-2s. These share certain aspects of Gen and Con, diverge in others, and therefore provide a model in miniature of how investigation can proceed under the strictures of validity. Achieving a general theory with universal ambitions is approached incrementally and constructively, with simplifications and expansions and theoretical experiments driven by formal considerations and by theses at various levels of detail about the empirical domains under scrutiny.

S.Gen

For systems like those considered here, *S*.Gen associates an *input* structure with a set of *output* structures. To do this, it is necessary to spell out what an input and an output consist of: their structural elements, and how they combine. When faithfulness is at issue, as it usually is not in these pages, it is also necessary to indicate how individual elements of the input are related to individual elements of the output: typically, via *correspondence* (McCarthy & Prince 1995), which is taken to be a partial function from (certain elements of) an input to (certain elements of) output.[33] A *candidate* is then a triple ⟨*input*, *output*, *correspondence-between-them*⟩. We can refer to the totality of candidates defined by *S*.Gen as *S*.Cand.

It is critical to specify what elements stand in correspondence, particularly in circumstances where complex structures are being related. For example, in C,V-based syllable theories developed from those in P&S ch.6, it is typically the case that only segments C,V stand in correspondence, although syllables, features, and subsyllabic constituents of various types may be part of the representation. Thus, the Ident(F) class of feature faithfulness constraints introduced in McCarthy & Prince 1995 recognizes only segmental correspondence. By contrast, if features or other elements such as moras are held to be in correspondence, further articulation of correspondence theory is required. None of these articulations will play a role in the discussion here: we mention them to emphasize that they are part of a system *S* and must be spelled out in *S*.Gen, since general impressions retained from experience with other systems are not guaranteed to be reliable.

[33] A function f:A→B relates each element of its domain A to some single element of its codomain B. Under this definition, some elements of B may left unrelated to any elements of A. The more general *partial function* allows elements of A to be left out as well. See Merchant & Prince (2021, ch. 1), for use of this notion wrt correspondence.

S.Con

A constraint is, minimally, a device that assigns each candidate a penalty, starting at 0 for *nihil obstat*, all good. Therefore: a function from the set of all candidates, *S*.Cand, as defined by *S*.Gen, to $\mathbb{N} = \{0,1,2,\ldots\}$,[34] the set of non-negative integers.[35] The penalties issued by a well-defined constraint are given meaning in the definition of optimality (Eval), which is constant across systems.

In P&S, a constraint assigns a 'multiset' of *marks* to a candidate. This characterization usefully emphasizes that the theory turns not on counting (i.e. on numerical values *per se*) but on comparison (more, less, same); see P&S, ch.5, for development. A multiset, unlike a set per se, can have repeated elements, and is defined as a set with an accompanying function into the positive integers that records the *multiplicity* of each of its elements. The integers are thus formally inescapable, and we go directly to them without the intermediate 'mark' construct. What matters is not that the integers appear, but how they are calculated with.

The bottom line for a constraint definition, then, is that it provides an unambiguous recipe for mapping candidates to integers. Many locutions still in occasional use don't quite manage to reach this level of explicitness: calls to "ban" this, or to "avoid" that, or requirements like "all such-and-such are so-and-so" depend on a 'You Know What I Mean' (YKWIM) sense of things. Such turns of phrase perhaps tend to confound what a constraint is in itself with its unblemished expression or with its tendential effect in the context of a grammar. But behind them lies a fairly clear intuition, which an experienced reader can often decrypt with relative ease: a constraint detects the presence of some patch of structure in a candidate, and assesses a penalty (of 1) for each its occurrences. This is what's embodied in the familiar star notation: thus, Parse-s written as *o designates a function that returns the number of unfooted syllables in the output component of a candidate; and similarly, if more elaborately for a faithfulness constraint.

A slight generalization of the familiar gives us what we need: thus, when we write $*x{:}\mathrm{P}(x)$, we mean to designate the constraint function that returns the number of instances of x in a candidate satisfying the condition P. This is an elliptical but unambiguous formulation: elliptical, because as a function, its argument is a *candidate*, which is not mentioned; unambiguous, because it tells us for each and every candidate exactly how to assess the numerical penalty that the constraint assigns.

Much more can be said about what constitutes a constraint, committing to what x and P can be, in order to attempt a substantive characterization of the patterns that can be registered in grammar.

[34] The integers are convenient but not sacrosanct: what's necessary is that the codomain be 'well-ordered' in that every subset has a minimal element. Equivalent, one could talk of non-positive integers and maxima.

[35] A further restriction sometimes encountered limits the codomain of the candidate function to $\{0,1\}$. This is useful for establishing suggestive formal results of limited scope, but ecologically, constraints of this character are not thick on the ground. The reason is that constraints typically detect a patch of structure, which may occur many times within a single candidate. Thus, imposing this restriction provides a model of the incremental methodology. (See P&S, ch. 5.3, p. 97; Frank & Satta 1998; Mai & Bakovic 2020).

But the most important characteristic of the general notation, for our purposes, is that it delivers us from YKWIM and makes clear how the constraint is defined, leading us away from the temptations of vagueness.

S.Gen and *S*.Con

When a grave error invalidates mathematical reasoning, some practitioners say: *there is no proof.* A similar catastrophe awaits the linguist who slights *S*.Gen and *S*.Con: *there is no system*. From this, it follows that nothing can be soundly concluded. The basic considerations just reviewed therefore impose a significant methodological burden on the analyst: *S*.Gen and *S*.Con must be defined at the outset, in their entirety. We will follow this procedure throughout.

Eval: the definition of Optimality

Even with the alternatives and the criteria defined — with *S*.Gen and *S*.Con in hand — we are still short of being able to perform the basic act of a theory of choice: namely, to *choose*. The situation is enriched by the fact that different criteria typically detect different aspects of structure and mapping, or rate the same aspects differently, and thus cannot be expected to agree on the merits (or, more properly, deficits) of particular candidates. Full possession of *S*.Con still allows for many, many ways of reconciling the diverse judgments of the constraint set.

Nevertheless, there is one mode of judgment which all such multi-criteria decision-making protocols are likely to share. If we interpret the numerical values as *penalties* — 'violations', in the jargon—then a single constraint, confronted with a set of candidates, will select from it those which earn the minimal value that the constraint assigns to members of that set: the smallest penalty.

Let's take this bare-bones commitment as the foundational step in building a definition of optimality. We have it that a constraint is a function from candidates to $\mathbb{N} = \{0,1,2\ldots\}$. The value C assigns to a candidate **q** we write as C(**q**), for a constraint C and a candidate **q**.

From the base notion of constraint as an assigner of penalties to individual candidates, we can develop the derived notion of a constraint that selects from or filters sets of candidates, the result of which we'll write as C[K], for a nonempty set of candidates K⊆*S*.Cand. The idea is that C[K] is the *best* of K, as judged by C — namely, the set of all elements in K that are assigned the minimal value that C takes on the set K. Notice that this is well defined, since every set of nonnegative integers does indeed have a minimal element.[36] Crucially, then, C[K] is always non-empty: choice is forced.

[36] Things can fall apart if we expand to include the negative integers as well: there is no *least* negative integer, since according to the standard ordering $-1 > -2 > -3 > \ldots$. This becomes an issue if we want constraints to assign rewards as well as or instead of penalties, since a reward is a negative penalty. If we reward the presence of some element, and allow that element to be inserted, we'll arrive at a candidate set with candidates that get better without limit along some dimension, making it impossible to rate any of them as *best* in that respect. Kimper (2011, 2016)

(79) **Definition. Single constraint as filter**. For $C \in S$.Con and a nonempty set of candidates K given by *S*.Gen, the function C[ ]: $\wp(S.Cand) \rightarrow \wp(S.Cand)$, is given by

$$C[K] =_{df} \{q \in K: \forall k \in K, C(q) \leq C(k)\}$$

In standard technical terminology, C[K] would be called 'arg min C(k), k∈K'.[37] This accomplishes filtration in the desired sense, since it designates a nonempty subset of K, derived from performance on C. Notice that C[ ] is an entirely different function from C( ), since it has a different domain and codomain, as it maps from subsets of *S*.Cand to subsets of *S*.Cand rather than from members of *S*.Cand to nonnegative integers.

The next step puts us into OT proper. We define what it means for a set of constraints which have been organized into a strict linear order, a 'hierarchy', to select from a set of candidates. The underlying idea can be grasped intuitively as sequential filtration, under the slogan "take the best, ignore the rest," as Gigerenzer & Goldstein (1996) have sharply phrased it.

Suppose we have a total ordering on, for example, three constraints, written like this:

$$H = C_1 \gg C_2 \gg C_3$$

where the order relation is referred to as '(strict) domination' or 'ranking'. To arrive at the new object H[K], the result of filtering K by hierarchy H, we filter first by C_1, the 'highest ranked', yielding $C_1[K] \subseteq K$. This has been defined in (79), so we have not left the realm of the meaningful.

We repeat the process, filtering by the highest ranked constraint in the rest of the hierarchy. This is C_2 and it yields $C_2[C_1[K]]$, a nonempty subset of $C_1[K]$. Observe that since $C_1[K]$ is a set of candidates, C_2 applies to it via definition (79) without complication. We then conclude by another repetition of the process, terminating with

$$H[K] = C_3[C_2[C_1[K]]]$$

In the standard notation for function composition, this comes out as

$$H[K] = C_3 \circ C_2 \circ C_1[K]$$

Since each constraint filtration function $C_k[\ ]$ produces a nonempty set, we are guaranteed that H[K] is nonempty. OT filtration thus always produces a nonempty result from a nonempty input set of candidates, inheriting the forced choice character of single constraint filtration.

This process just sketched can be given a concise recursive definition. Let K be a set of candidates and P be a linear order '≫' on a set of constraints that contains at least one constraint. Using concatenation to represent domination, we write CP for the hierarchy in which C stands at the topmost constraint, with P denoting the rest. |H| designates the size of H in number of constraints.

---

notes that this debilitating effect can be forestalled in Harmonic Serialism, where it could lead to nonterminating derivations, if the notion of 'single change' is configured so that insertion of dummy structure (penalized) and assignment of featural values to the insertee (rewarded) are kept separate.

[37] See Arg Max, Wikipedia, for discussion.

(80) Definition. Filtration by a constraint hierarchy

1) $|H| = 1$. $C[K] = \{q \in K: \forall k \in K, C(q) \leq C(k)\}$

2) $|H| > 1$. $CP[K] = P[C[K]$

The first clause repeats the definition of filtration by a single constraint, which is shared with many systems of optimization. The second writes out the description of filtration by several constraints outlined above, which is proper to OT. With each application of clause 2), the unresolved part of the hierarchy ("CP") shrinks by a constraint (to "P"), so that with repeated application it is sure to vanish, at which point clause 1) may be invoked.

Repeated application of 2) thus unfolds $H = C_1 \gg \ldots \gg C_n$ to give the following:

$$H[K] = C_n[\ldots[C_1[K]\ldots]$$

For this, we may write:

$$H[K] = C_n \circ \ldots \circ C_1[K]$$

The definition of single constraint filtration in clause 1) of (80) may now be applied *n* times, inside out as it were, from the highest ranked constraint (C_1) to the lowest (C_n), to achieve a final result, a nonempty subset of K. This closely reflects the intuitive understanding that drives communal practice,

Upon this scaffolding we may construct the key notion: optimality. To do this, we must pay close attention to K and H. For a candidate to be *optimal*, we must start out with a K that is an entire candidate set, as specified by *S*.Gen. The hierarchy H must include all the constraints of *S*.Con. With these prior conditions met, the general definition of filtration of a set by a hierarchy (80) will be invoked: a candidate q is optimal in its candidate set K with respect to a given hierarchy H if it lies within the subset of K selected by H.[38]

(81) **Definition**. Optimality. Let K be a candidate set as defined by *S*.Gen for a system *S*. Let H be a linear order on all the constraints of *S*.Con. Then $q \in K$ is *optimal* in K with respect to H if $q \in H[K]$.

Although this has a familiar ring, it is rich in consequence. Particularly significant, as we have emphasized throughout, is its reliance on complete knowledge of *S*.Gen and *S*.Con: a candidate is optimal only if it survives filtration of a set that contains *all* of its competitors; and filtration must poll the judgment of *every* constraint. This is both obvious from the general perspective of rational choice and less than manifest in much communal practice. In addition to setting a requirement for the success of local arguments of the form '*this* candidate is optimal', there is a broad methodological entailment: the *system* is the locus of valid argument, and advance toward universal grammar proceeds as understanding of systems accumulates.

[38] Often this will consist of a single candidate — but whenever candidates share a violation profile, the evaluation mechanism sees them as identical. Hence, H[K] may contain more than one *candidate*, although there will be only one violation profile for all of its members, often called 'co-optima'.

It's worth noting how the notion of 'richness of the base' is preserved in this account. Its essential meaning is that the *input* set is the same across grammars. This means that all cross-linguistic differences follow from the one factor that varies freely to generate typologies from *S*.Gen and *S*.Con: the ranking of constraints with respect to each other. The idea that observed language-particular input patterns can be understood in this way, as emergent from the same grammar that maps input to output, without a separate and often redundant apparatus of input conditions (Kisseberth 1970, Kenstowicz and Kisseberth 1977)), is explicitly tied to the work of David Stampe in P&S (§4.31,§9.3). This commitment puts further pressure, if more is needed, on the requirement that all candidates and all candidate sets be considered, since the theory offers no hope that a clean-up detail of language-particular input restrictions can be called upon to sweep away unwanted detritus that the input-output grammar leaves behind. Richness of the base is as pertinent to the *system* as it is to the holistic endpoint of systematic investigation, Universal Grammar.

Deriving optimality from filtration is immediately useful when the analyst has in hand a linear order on the full constraint set, a hierarchy, often called a *ranking*, or less ambiguously, a *leg*. Defining filtration thus directly solves what Brasoveanu & Prince (2005/11) call the 'selection problem' — what does a given ranking *select* from a set of candidates? However, the ecologically most common situation has a very different character: with *S*.Gen and *S*.Con articulated, a certain candidate ―understood to represent an observed linguistic entity of some sort―is desired optimal. Now we face the 'ranking problem': which linear orders on *S*.Con will choose that candidate from among all its competitors?

The first solution that naively occurs is to enumerate the *n*! rankings of *S*.Con (assumed to hold *n* constraints) until at least one is found that filters as hoped, or more ambitiously, until all those are found that so filter. However, an indirect approach proves to be far more informative. Instead of seeking to eliminate all distinct competitors at once, as a full ranking will do, it is more instructive to deal with them one at a time. Because of the way choice is defined in OT, valid ranking information can be derived from the isolated comparison of a desired optimum and a single competitor. By accumulation, review of all viable competitors ― the finite collection of those that can win under some ranking ― fills in the picture completely. This fact, taken seriously, leads the way to developing a much improved understanding of the representation of OT grammars, and thus serves as a kind of natural start point for the project undertaken here.

The basic unit of this perspective is the comparison between a desired optimum and one of its competitors, which yields the Elementary Ranking Condition (ERC) discussed in §1.2 above. From the root definition of a constraint as function that attaches numerical penalties to candidates, we construct a new kind of constraint, perhaps better called a comparator, which recognizes only the *relation* between a designated candidate and a single alternative. For this purpose, it is not the numerical difference that counts, but only the *sign* of the difference, exactly echoing the sensitivities of OT filtration as applied to a pair of candidates. To emphasize the distinction, let us briefly embrace exactitude and depart from the standard convention of retaining the old name for the new entity: from C, let us derive dC. We have, for two competitors **q** and **z**, this array of possibilities:

(82) **Deriving the Comparative Constraint**

Pair	dC(q∼z)	C relation	σ(C(z)−C(q))	Filtration
q∼z	W	C(**q**) < C(**z**)	+	C[{**q**,**z**}] ={**q**}
	e	C(**q**) = C(**z**)	0	C[{**q**,**z**}] ={**q**,**z**}
	L	C(**q**) > C(**z**)	−	C[{**q**,**z**}] = {**z**}

Within an OT system, this notion of constraint as comparator yields arrays like that of ex. (6) of §1.2, repeated below. Recall that the columns are *not* given in ranking order; all ranking information is contained in the values W, L, *e*.

(83) VT

/o/	P-s	Tr	Ia	AFL	AFR
.o.	1	0	0	0	0
.X.	0	1	1	0	0

(84) CT

/o/	dP-s	dTr	dIa	dAFL	dAFR
.o. ∼ .X.	L	W	W	*e*	*e*

In the example, either Tr or Ia must dominate P-s, compelling binarity. From the definition of optimality (81), we have it that in a case like this, with W and L both present, *some W must dominate every L* in any ranking that selects the desired optimum, namely **q** from {**q**,**z**}. This follows because in the filtration sequence, the first constraint distinguishing **q** from **z** must select **q** and eject **z**. Thus, its comparative version must assign W to [q∼z]. CT (84) denotes some 80 rankings, any one of which is as good as any other for achieving the desired goal.

With every such comprehensive list of comparative values, then, there is an associated restriction on ranking that is satisfied in all rankings that select the desired winner. This is the Elementary Ranking Condition proper (ERC), though the term is extended to the list itself. Since the overall optimum for the entire candidate set is favored in *every* pairwise contest with its rivals, it's simply a matter of assembling the relevant ERCs to arrive at a complete description of the conditions that must be satisfied by any ranking that chooses the optimum.

With this, we have stepped across the border from the general possibility of optimization and advanced into OT itself. At the constraint level, OT begins with the ERC.

Thus far, we've examined the relatively obvious case of an ERC that contains both W and L, and therefore imposes a restriction on ranking. It's worth being aware of what happens when an ERC [**q**∼**z**] lacks W or lacks L. In such cases, ranking has no impact on the choice between **q** and **z**, because there is no constraint that could enable a contrary choice: all constraints either regard the two compared candidates as equivalent, or tilt their favor toward the same one. The resulting interpretations are analogous to tautology and contradiction. From the logical point of view, we may divide such ERCs into two fundamental types.

• Those that are always true: **No L**.

An ERC [**q**∼**z**] without L's reports that no constraint in the entirety of *S*.Con favors **z** over **q**. Thus **q** survives the competition no matter how the constraints are ordered.

An ERC with no L must be vacuously true of every ranking of S.Con.

- A special subtype: **No L and no W either**.

This ERC is all *e*. Every constraint regards **q** and **z** as equivalent and both candidates are treated identically by any $\lambda \in$Ord(S.Con), regardless of ranking. Lacking L, an all-*e* ERC is, again, vacuously true of every λ.

• Those that are always false: **No W and at least one L**.

An ERC [**q**∼**z**] *with* L but without W reports that no constraints in *S*.Con favor **q** over **z**: some constraints may view them as the same (*e*) but one or more favor **z** (L). Thus **q** lacks advocates while **z** has at least one, so that **q** cannot survive the competition with **z**, whatever the ranking.

The Elementary Ranking Condition — which looks for domination of L by W — must be *false* of every λ, there being no W to do the job. In this case, the desired Winner **q** is harmonically bounded by its competitor **z** and cannot be optimal under any ranking.

ERCs like these play somewhat different roles in OT practice. Those which have L but no W, said to belong the set L^+, spell the end of a desired optimum's hopes for optimality: no ranking can deliver it. Thus, they do not appear in grammars, although they play a crucial role in the course of analysis. Those lacking L, said to belong to W*, contribute no ranking information and, on grounds of redundancy, will not appear in any concise representation of a grammar.

Perhaps the most straightforward way to state the Elementary Ranking Condition explicitly as a predicate of legs is to put it into an $\forall\exists$ form that make the role of vacuous satisfaction obvious. Given a leg λ , a total order on *S*.Con, and given α, an assignment of values {W,L,*e*} to the constraints of *S*.Con, the predicate $\tilde{\alpha}(\lambda)$ says: for every constraint D assigned L by α, there is some constraint C assigned W by α such that $C \gg_\lambda D$ in the order $\gg_\lambda$ .To spell it out in symbols,

(85) **ERC**. For $\lambda \in$Ord(*S*.Con); C,D$\in$*S*.Con, and α:*S*.Con$\rightarrow${W,L,*e*}, the predicate $\tilde{\alpha}(\lambda)$ is

$$\forall D\ \exists C\ \ ([\alpha(D) = L] \Rightarrow [\alpha(C) = W\ \&\ C \gg_\lambda D]),$$

Remarks. The expression $\tilde{\alpha}(\lambda)$ is *false* for α = [**q**∼**z**] if and only if λ({**q**,**z**}) = {**z**} by the definition of OT filtration (80), when the desired optimum loses to z on all rankings. This happens when the antecedent is satisfied and the consequent is not. By contrast, if there are no L's in α, then there are no D's satisfying the antecedent in any λ, and it is false under all instantiations of D: therefore, the whole expression $\tilde{\alpha}(\lambda)$is (vacuously) true. Observe that since λ is a total order, when $\tilde{\alpha}(\lambda)$ is true, there is always one W in λ that dominates all the others, and the $\exists\forall$ slogan 'some W dominates all L's' turns out to be equivalent to the $\forall\exists$ ERC (85).

Because an ERC α corresponds to a logical expression $\tilde{\alpha}(\lambda)$, it submits to logical manipulation. Its intrinsic logic is 3-valued, responding to the 3 basic relations that explicate the intuitive notion 'better than' in OT. Chief among the manipulations are those which determine consistency and entailment (Prince 2002ab), which are used to detect *in*consistency and redundancy: these play a central role in the algorithmic compression of an arbitrary consistent ERC set to a minimal equivalent set (Brasoveanu & Prince 2005/2011).

Important as well for the present enterprise is an operation that uses the relevant 3-valued definition of the connective *or*, the Merchant *join* (Merchant 2008, 2011). The join of two sets of ERCs is the minimal ERC set entailed by both, rather as P∨Q is a minimal expression entailed by each of P and Q. The join produces an ERC set with special properties: it is the minimal super-grammar that contains what the joining grammars *share*, in the appropriate sense.[39] Since Property Theory is all about finding the ranking characteristics that unite and distinguish grammars, the join will have a significant role to play in the practical analysis of typologies. Details of these operations will be discussed as they become relevant.

Since the Elementary Ranking Condition $\tilde{\alpha}(\lambda)$ makes no reference to candidates, it can be put to use as the logical building block with which grammars are described as ranking structures. The significance of the ERC is thus at least twofold:

- concretely, it enables a solution to the ranking problem: given a desired optimum, what rankings select it?
- abstractly, it enables a solution to the problem of representing OT grammars: what ranking conditions are necessary and sufficient to obtain all and only the desired optima, and how can they be formulated?

What then is an *OT grammar*? We have assembled the elements from which OT grammars are built and the conceptual tools to build them. Let's now see how they emerge from the infrastructure that has just been put in place.

1.6.2 The Objects of OT: Language, Grammar, Typology

Given a specific OT system *S* = ⟨*S*.Gen and *S*.Con⟩, and given a commitment to what optimality means in OT, we can talk about the objects that emerge. Among the most basic are *language* and *grammar*.

Given a ranking on *S*.Con, a *language* is defined as the set of all optimal candidates it selects from *S*.Cand.

[39] The idea of *sharing* is natural when dealing with a pair of expressions like (P & R) and (¬P & R). These *share* R. If we disjoin then as (P & R) ∨ (¬P & R), it's clear that whatever truth value is assigned to letter P, the whole expression is equivalent to R. The unshared, intuitively put, cancel out in the disjunction. Note however that WeL and eWL join to WWL, in which case the resulting disjunction is not intuitively 'shared' by the disjuncts.

A *grammar* is, most elementally, the collection of all rankings that choose the same optima. More informatively, a grammar may be understood to be the set of ranking conditions that delimit that set. We can distinguish the two notions by referring to the first as a ranking grammar and the second as an ERC grammar.

Given these distinctions, several versions of the notion *typology* can be discriminated.

- The *extensional typology* of *S* is the set of all languages made available by *S*.
- The *intensional typology* of *S* is the set of all grammars provided by *S*.

The two senses of the term *grammar* yield two senses of (intensional) typology: a typology of grammars may be understood as a collection of sets of rankings; or it may be understood as a collection of ERCs denoting those rankings. The goal of Property Theory is to understand how higher-order conditions on ranking ('properties') define a typology in a way that illuminates the *traits*, or structural patterns, of the extensional language.

The value of attending to these notions can be appreciated from an example. Consider the Basic Syllable Theory (BST) of P&S:ch. 6, which will be analyzed in detail in Chapter 2 of this work. Here we sketch it sufficiently to recall it to mind, in support of noting some key interactions.

BST is a form of fundamental syllable theory that recognizes only syllables shaped CV, CVC, VC, V in its fully syllabified output; that allows correspondence only between C and C, V and V, where correspondence respects the linear order of the input but may fail in either direction; and which allows only the input/output disparities of deletion and insertion, where insertion discriminates between inserting V and inserting C

(86) BST.Gen

a. A candidate is a structure ⟨*in*, *out*, correspondence⟩.

b. A candidate set consists of all candidates that share a given input.

c. An input is a string from $\{C,V\}^+$.

d. An output is a string, possibly null, from σ*, where $\sigma \in (C)V(C)$.

e. Correspondence is a partial function from input to output relating C to C and V to V, respecting the order of the input.

BST.Con consists of the following familiar constraints, which we recall here for purposes of explicitness. We write x∼y for 'x corresponds to y'.

(87) BST.Con

a. Markedness

m.Ons	*.V	'penalize onsetless syllables'
m.NoCoda	*C.	'penalize syllables with coda'

b. Faithfulness

f.max	*x∈C: x∈in & ∄y∈out such that x∼y.	'penalize deletion'
f.depV	*y: y∈*out* & y=V & ∄x∈*in* such that x∼y.	'penalize V insertion'
f.depC	*y: y∈*out* & y=C & ∄x∈*in* such that x∼y.	'penalize C insertion'

The extensional typology of BST consists of 12 languages. Let's consider the language we'll call (C)V.f.ins,[40] in which the output syllable canon (C)V allows onsetless syllables (f) but no codas, with insertion (ins) used to avoid proscribed consonant clusters. Here's a valid support for the grammar, a collection of candidates sufficient to derive it, with its associated ERCs. We write 'e' for the epenthetic vowel, 't' for the epenthetic consonant, and '∅' for the result of deletion. Note that according to BST.Gen, every licit output syllable must contain a vowel, so any input C which does not sit next to an input vowel, like /C/ here, must either be deleted or supported by V-insertion.

(88) Support for (C)V.f.ins

Input	Winner	Loser	m.NoCoda	f.depC	f.max	m.Ons	f.depV
[1] CVC	.CV.Ce.	.CVC.	**W**				**L**
[2] V	.V.	. tV.		**W**		**L**	
[3] V	.V.	∅			**W**	**L**	
[4] C	.Ce.	∅			**W**		**L**

- ERC [1] gives m.NoCoda≫f.depV, so that codas are avoided by insertion.
- ERC [2] gives f.depC≫m.Ons, so that onsetless syllables are *not* avoided by insertion.
- ERC [3] gives f.max≫m.Ons., so that onsetless syllables are *not* avoided by deletion.
- ERC [4] gives f.max≫f.depV, so that otherwise unsyllabifiable C is syllabified via insertion.

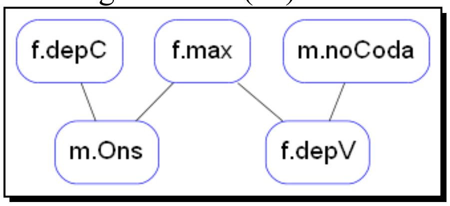
These concrete findings may be (mildly) compressed into the following ERC grammar, where the ERC is not treated as a report from the field but as an expression of ranking relations. Since it is a partial order, with one W per ERC, a Hasse diagram may also be given.

(89) (C)V.f.ins: ERC grammar

m.NoCoda	f.depC	f.max	m.Ons	f.depV
W				**L**
	W		**L**	
		W	**L**	**L**

[40] The naming convention lists first the syllable canon, then indicates the treatment of underlying vowels and (next) consonants in the two following fields.

(90) Hasse Diagram from (89)



To get a sense of the level of generalization that's achieved in the ERC grammar (89), consider the ranking grammar tabulated below, which lists all of its linear extensions, or legs.

(91) The 16 legs of (C)V.f.ins from BST

λ_1	f.depC	≫	f.max	≫	m.NoCoda	≫	m.Ons	≫	f.depV
λ_2	f.depC	≫	f.max	≫	m.NoCoda	≫	f.depV	≫	m.Ons
λ_3	f.depC	≫	f.max	≫	m.Ons	≫	m.NoCoda	≫	f.depV
λ_4	f.depC	≫	m.NoCoda	≫	f.max	≫	m.Ons	≫	f.depV
λ_5	f.depC	≫	m.NoCoda	≫	f.max	≫	f.depV	≫	m.Ons
λ_6	f.max	≫	f.depC	≫	m.NoCoda	≫	m.Ons	≫	f.depV
λ_7	f.max	≫	f.depC	≫	m.NoCoda	≫	f.depV	≫	m.Ons
λ_8	f.max	≫	f.depC	≫	m.Ons	≫	m.NoCoda	≫	f.depV
λ_9	f.max	≫	m.NoCoda	≫	f.depC	≫	m.Ons	≫	f.depV
λ_{10}	f.max	≫	m.NoCoda	≫	f.depC	≫	f.depV	≫	m.Ons
λ_{11}	f.max	≫	m.NoCoda	≫	f.depV	≫	f.depC	≫	m.Ons
λ_{12}	m.NoCoda	≫	f.depC	≫	f.max	≫	m.Ons	≫	f.depV
λ_{13}	m.NoCoda	≫	f.depC	≫	f.max	≫	f.depV	≫	m.Ons
λ_{14}	m.NoCoda	≫	f.max	≫	f.depC	≫	m.Ons	≫	f.depV
λ_{15}	m.NoCoda	≫	f.max	≫	f.depC	≫	f.depV	≫	m.Ons
λ_{16}	m.NoCoda	≫	f.max	≫	f.depV	≫	f.depC	≫	m.Ons

What's most noticeable, perhaps, is that the generalizations apprehended by the ERC grammar are dispersed to the point of invisibility in the ranking grammar (91). Yet each leg successfully delivers all the optima of the language, and one may even find occasionally in the literature references to a single leg as the 'grammar' of a language when there are many legs that deliver it.

Many of the relations in each leg are of no linguistic significance, artifacts of the formal requirement a leg be a linear order. By 'artifact' is meant some feature of the representation which is due to a formal commitment (for example, 'linear order' or 'conciseness' or even 'learnability') rather than to any direct motivation in the language or in the grammar as a leg set.[41] The following leg provides an extreme example in which *not a single* relation of adjacency in the ordering has any substantive interpretation.

[41] We omit discussion of analytical shortfall due to simple errors of methodology, like failure to define *S*.Gen and *S*.Con, or failure to respect those definitions by leaving out candidates and constraints in calculations of optimality. These have no formal motivation.

(92) The disjointed leg λ_{10} of (C)V.f.ins

f.max
|
m.NoCoda
|
f.depC
|
f.depV
|
m.Ons

Consulting the ERC grammar or its Hasse equivalent, or simply reviewing the aggregate ranking grammar, it is easily determined that f.max has no necessary relation to m.NoCoda, either dominated or dominating; that m.NoCoda has no meaningful relation to f.depC; and so down the order.

Generalizations of linguistic import emerge from analysis of the whole ranking grammar: that's why it is essential to amass the entire collection of legs that yield the optima, and to require of an ERC grammar that it denote all the members of that set, and of course no others.

Without this goal, other artifacts of omission can be introduced in the name of efficiency or other nonlinguistic virtues. The algorithm RCD, for example, efficiently renders the grammatical description of a language as a *stratified* partial order, in which the members of each stratum are all ranked below those of any higher stratum and above those of any lower stratum (Tesar & Smolensky 1993 *et seq*.; Prince 2002a, 2008; Brasoveanu & Prince 2005/2011). In the simple case at hand, two strata result:

(93) RCD Stratification of (C)V.f.ins

m.NoCoda, f.depC, f.max || m.Ons, f.depV

But this flattening is also fraught with artifacts: m.NoCoda has no grammatical relation with m.Ons; likewise, f.depC has no relation with f.depV; and so on. Nevertheless the set of legs denoted by the stratified order (93), a *proper* subset of the actual grammar, embodies these spurious relations.[42]

ERC theory supports an algorithm — Fusional Reduction, *FRed* — which produces a maximally concise equivalent of any set of ERCs.[43] It produces either the MIB — the 'Most Informative Basis' — which contains the maximum number of L's, displaying all transitively deducible information — or the 'Skeletal Basis', which contains the minimal number of L's, removing all

[42] RCD cracks the problem of obtaining *some* legs that deliver the data, which originally seemed difficult but turned out, via RCD, to be easy. But the cost associated with efficient analysis is loss of information: too high to pay when the goal is finding the actual structure of grammars in a typology.

[43] 'Concise' here means 'composed of the fewest number of ERCs'. Other measures of conciseness may then be imposed once this criterion is met, as e.g. 'minimal number of W's and L's", which gives the SKB.

information deducible from transitivity.[44] These provide veridical and complete representations of any OT grammar, in the sense that all and only the legs of the ranking grammar satisfy the SKB or the MIB. But the quest to remove artifacts is not over.

Even in the realm of the SKB, there is no sign indicating whether a relation deducible from transitivity is substantively motivated. If the SKB contains A≫B and B≫C, it also licenses the further inference that A≫C, from the definition of ≫ as an order. Yet A≫C may have no linguistic significance, in which case it is a formal artifact — or, contrariwise, it may stand on its own as a motivated relation. In either case, the representation is the same — the SKB, like the Hasse diagram, erases transitively-derived information. The MIB, by maximizing the number of L's, includes all the transitive inferences, neutralizing the substantive distinction in the other direction.

Property Theory then steps in to offer itself as a different take on what relations are the real ones, with the hope that artifacts of representation can be yet further erased, if not removed entirely.

1.6.3 From Micro to Macro

The practitioner specifies *S*.Gen and *S*.Con and the rest unfolds by logic, without the possibility of further intervention, and, often enough, with surprises. We are now in a position to trace the course of this unfolding, as higher order structures and patterns build from the basic commitments.

What counts as a 'higher-order' structure can differ radically from theory to theory: in a rule-package theory, like early Generative Grammar, the analyst was charged with framing 'rules' that directly encode many details of a mapping, and was given considerable resources to do so.[45] In later developments, and especially in a theory like OT, even the simplest mapping from input to output is derived from interactions and accumulated from the individual fate of a multitude of individual candidates. The analyst is granted no ability to steer the map toward a favored outcome without affecting the rest of the grammar and indeed the other grammars in the typology where it lives. The analyst is not even allowed to cross the threshold into OT proper, since the contents of *S*.Gen and *S*.Con can be used verbatim by many different theories of optimization.

Consider first the course of development on the constraint side, where the only point of user access is *S*.Con.

---

[44] The SKB is therefore closest to the Hasse Diagram, which is also 'transitively reduced' in the relevant sense. The SKB is an algebraic representation ('incidence matrix') of the hypergraph that represents an OT grammar in the general case. When there is one W per ERC in the SKB, an OT grammar is a partial order, and the SKB provides an incidence matrix for the associated Hasse Diagram. When there's an ERC in the SKB or MIB with 2 or more W's, the order structure that it represents is a proper 'antimatroid', that is, an antimatroid not equivalent to a partial order.
[45] This characterization is true of the communal practice associated with the theory, rather than with the in-principle version of it virtually never seen in practice. Platonically, in the idea of it, a rule system was meant to be emergent from a global optimization over grammar and lexicon with respect to the Evaluation Metric (for discussion, see Prince 2007b).

(94) A progression in levels of organization

► constraint	$\in f{:}S.\mathrm{Con}\to\mathbb{N}$	microstructure
• ranking	$\in \mathrm{Ord}(S.\mathrm{Con})$	
• sets of rankings	(R-) Grammar	
• sets of grammars	(R-) Typology	macrostructure

A parallel course of construction passes through the ERC, in which the *constraint* of *S*.Con, which assigns a penalty, is transformed into a 'comparator', which rates the relative status of a pair of candidates to which penalties have been assigned.

(95) ERC-based development

Object	Source	
► constraint	$\in C{:}\ S.\mathrm{Con}\to\mathbb{N}$	microstructure
• comparator	$\in dC{:}\ S.\mathrm{Con}\times S.\mathrm{Con}\to\{\mathrm{W},\mathrm{L},e\}$	
• ERC	$\in f{:}S.\mathrm{Con}\times S.\mathrm{Con}\to\{\mathrm{W},\mathrm{L},e\}^n$	
• set of ERCs	ERC-Grammar	
• set of grammars	(ERC-) Typology	macrostructure

On the extensional side, the concept of a language emerges from a similar tower of constructions. Here the user has access only to *S*.Gen.

(96) Extensional Side

Object	Source	
► candidates	$\in S.\mathrm{Cand}$ via *S*.Gen	microstructure
• optimum	$\in S.\mathrm{Cand}$ via Eval	
• set of optima	Language	
• set of languages	(Extensional-)Typology	macrostructure

It was recognized explicitly in P&S that the relations among these formal objects determine how analysis of linguistic patterns must proceed. Chapter 8 studies a system 'Universal Syllable Theory' (UST) in which sonority levels are introduced to amplify the simple binary C/V distinction of BST. It is shown that imposing a hierarchy of sonority-based affinities (more accurately, disaffinities) for 'peak' (generalized V) and 'margin' (generalized C) positions leads to a variety of large-scale results, typically implicational, about the segmental distribution of peaks and margins in the languages of the resulting typology. The analysis begins with careful consideration of what it takes to establish such facts. Consider the following assertion:

(97) "Cross-linguistically [i.e. within the posited system -BA & AP], the inventory of possible codas is a subset of the inventory of possible onsets, but not vice versa." (P&S: 154).

What does it take to establish such a claim about a derived object like a typology, which emerges only after layer-on-layer of formal construction from its primitives?

P&S give a full account in their terms, worth reproducing here (with minor updates to the present context).[46]

(98) An Onset/Coda asymmetry spelled out

> **For all** rankings λ of the […] constraints allowed [by UST.Con,] and
> > **for all** segments s,
> > > **if there exists** an input string I_s [allowed by UST.Gen]
> > > containing s
> > > > for which **there is [**an output] $B_{Cod/s}$ in which s is associated to [the σ node] Cod such that
> > > > **if** C is any other candidate parse of I,
> > > > **then** $B_{Cod/s}$ is more harmonic than C w.r.t the ranking λ ($B_{Cod/s} \succ C$),
> > >
> > > **then there exists** an input string I_s'
> > > containing s
> > > > for which **there is** a parse $B_{Ons/s}'$ in which s is associated to [the σ node] Ons such that
> > > > **if** C' is any other candidate parse of I_s'
> > > > **then** $B_{Ons/s}'$ is more harmonic than C' w.r.t. the ranking λ ($B_{Ons/s}' \succ C'$);
>
> but **not** vice versa.

This account of the logic could be condensed by referring to higher-level objects such as grammars, optima, and the like, but its virtue is exactly that it makes their content unavoidable, detailing the steps of the journey from UST = ⟨UST.Gen, UST.Con⟩ to conclusions about the patterns that emerge in the resulting typology.

The analysis of Ch. 8 targets sonority-induced implicational relations, such as "if [a segment] τ is a possible coda, then it is a possible onset." (P&S: 187), which stand at a considerable remove from the founding assumptions. Along the way, a variety of supporting conditions are established. For example, the whole-language trait of admitting syllable codas takes this form, retaining their notation, but translating their constraint names into its more recent form:

(99) "Codas Allowed" (P&S: ex. (250), p. 184)

$\{$f.max, f.depV$\} \gg$ m.NoCoda

This we recognize immediately as the value

$\{$f.max, f.depV$\}$.sub > m.NoCoda

where the constraint class is the set of faithfulness constraints relevant to Codas.

When sonority is involved, more complex conditions are found, many taking forms that can be rendered as values in the vocabulary of Property Theory, though some go beyond it by involving

[46] We also gloss over the (unremarkable) input-output correspondence needed for full modernization.

a three-level ranking statement. Part of a Possible Onset Condition (P&A: ex. (231), p. 171), for example, requires

$$\text{m.Ons \& f.depC} \gg \text{m.C}(\tau) \gg \text{m.V}(\tau)$$

where we write m.C(τ) for the constraint militating against treating segment τ as a 'margin', i.e. onset or coda element, and we write m.V(τ) for the constraint assessing it as a 'peak'.

In another point of contact, Ch. 8 introduces operators *min* and *max* dealing with fixed Peak and Margin hierarchies, primarily to pick out sonority levels, which figure as derived parameters that encapsulate aspects of segmental behavior. In one case, however, the *min* operator plays exactly the role of .sub, identifying the constraints that play a role in defining the sonority level π_{Ons} in ex. (247).

(100) 'Critical Constraints'.(P&S: ex. (246), p. 182)

$$C_{Ons} \equiv \min\{\text{f.max, f.depC, m.Ons}\};\ C_{Nuc} = \min\{\text{f.max, f.depV}\}$$

The P&S analysis is notable for establishing key characteristics of a defined typology, or rather class of typologies, before tools existed that could calculate the languages and grammars that follow from the basic commitments to *S*.Gen and *S*.Con. The problem it sets and solves is endemic to ambitious theories which allow user access only to primitives that combine autonomously to produce results distant from assumptions. In the case of OT, the transition from basal microstructure to the beginnings of macrostructure begins with the choice of optimum from an individual candidate set, one input to one output. Any notion of a broader-scale input-output mapping, at roughly the grain-size of a 'rule', follows (when it does) from the aggregation of these individual choices; on this basis, one may look yet further for the interaction of such mappings. The emergence of general, patterned behavior reaches to the level of the typology, where — we optimistically expect on the basis of results to date — groups of languages can be defined and distinguished by the linguistic characteristics or *traits* that they share and lack.

Parallel to extensional development is the emergence of intensional structure in grammars. Most accessible, perhaps, is the clustering of constraints into classes, forming a basic structural unit of Property Theory. Here, too, there is a micro-to-macro shift: we start from the individual constraint, but end up with classes of constraints. This is foreshadowed in the ERC and the ERC set, which reveal that dominating and dominated constraints come ecologically in cohesive sets that recur in a variety of circumstances. Property theory both generalizes and specializes this finding, identifying recurrent classes that are finer than those lumped together in ERCs, and allowing them to function analytically in ways that go beyond what ERC form allows.

Information about ranking structure is absent at the microlevel and appears only in dispersed form in the ranking grammar, which is just a set of legs united by a common set of optima. ERC representation brings us closer to the patterns, but is both elevated and encumbered by its connection with specific optimum-suboptimum candidate contrasts on the one hand and with formal notions of conciseness on the other. Property Theory aims to identify the fundamental unit of ranking as the *value*.

The explanatory question posed by the twinned development of macrostructure — among languages and among grammars — is how the two relate to each other. How do assumptions about *S*.Gen ('extensional') and *S*.Con ('intensional') combine over generations of structure to support large-scale generalizations about linguistic patterning? How do configurations of ranking generate configurations of traits?

This question essentially is forced upon us by the nature of the theory, which, unexamined, does not provide an answer: the sand grain does not announce the dune. Because each side is frequently ambiguous, so that several understandings of the patterns of traits are almost inevitably accompanied by several understandings of the patterns of ranking, the hope for a determinate answer rests on arguments from matching the two. Systematicity in the attack on the problem is mandated; without it, there is little hope of going beyond sporadic insights and isolated assertions. From this point of view, the typology becomes the central object of study, since it is only at that level that a proposed theory has a chance of revealing its consequences.

It's important to note that nothing in the broad scheme of things outlined here guarantees *a priori* the emergence of large-scale structure as opposed to further compounding of the sporadic, the irreducibly fine-grained, the inexplicable. This is a consequence of the surrender of control. Only the development of a theory of the emergent such as the one advocated here, probing the macro level to find what it contains and does not contain, can recover what the basal theory says about the reality that it hides or reveals.

1.7 Appendix: Property Theory in the literature

Property Theory has received significant exposure in classes, talks, pre-prints, non-prints, tech reports, articles and dissertations prior to the release of the present work. We provide here a coarsely classified list of many of these contributions.

Foundational

Alber, Birgit & Alan Prince. 2015-2017. Outline of Property Theory [Entwurf einer allgemeinerten Eigenschaftstheorie]. Ms. University of Verona and Rutgers University, New Brunswick.

Alber, Birgit, Natalie DelBusso & Alan Prince. 2016. From Intensional Properties to Universal Support. *Language* 92.2. e88-e116. ROA-1235.

Alber, Birgit & Alan Prince 2017. The Book of nGX. *Memoirs of the Society for Typological Analysis* 1.1. ROA-1312.

Alber, Birgit & Alan Prince. In prep. *The Structure of OT Typologies*. Ms. Free University of Bolzano-Bozen and Rutgers University, New Brunswick.

Applications to Linguistic Analysis

Alber, Birgit. 2017. The Book of BTT. *Memoirs of the Society for Typological Analysis* 1.2. ROA-1327.

Alber, Birgit & Sabine Arndt-Lappe. Submitted. Anchoring in Truncation: a typological analysis.

Alber, Birgit & Joachim Kokkelmans, submitted. Typology and Language Change: the Case of Truncation.

Alber, Birgit & Marta Meneguzzo. 2016. Germanic and Romance onset clusters - how to account for microvariation. In Ermenegildo Bidese, Federica Cognola & Manuela C. Moroni (eds.), *Theoretical Approaches to Linguistic Variation*, 25–51. Amsterdam: John Benjamins Publishing Company.

Apostolopoulou, Eirini. Submitted. Where *r* you going?

Bennett, William G. & Natalie DelBusso. 2017. Typological consequences of ABCD constraint forms. In C. Hammerly, A. Lamont, B. Prickett, & K.A. Teztloff (eds.), *NELS 47*: *Proceedings of the 47th Annual Meeting of the North East Linguistic Society*, 1:75-88. Amherst, MA: UMass GLSA. ROA-1307.

Bennett, William G. and Natalie DelBusso. 2018. Typological effects of ABC constraint definitions. *Phonology* 35.1:1-37. ROA-1326.

Bennett, William G. and Natalie DelBusso. 2020. Summing constraints in and across properties. *Phonology* 37/3. 367-381.

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

Danis, Nicholas. 2014. Deriving Interactions of Complex Stops. Ms. Rutgers University. ROA-1220.

Danis, Nicholas. 2017. *Complex Place and Place Identity*. Ph.D. dissertation Rutgers University, New Brunswick, N.J. ROA-1324.

DelBusso, Natalie. 2016. Deriving reduplicants in Prosodic-Morphology Typologies. In A. Albright & M. A. Fuller (eds.), *Supplemental Proceedings of the 2014 Annual Meeting on Phonology* (*AMP* 2014). Washington, DC: LSA. ROA 1251.

DelBusso, Natalie. 2018. *Typological Structure and Properties of Property Theory*. New Brunswick, NJ: Rutgers University Ph.D. dissertation.

DelBusso, Natalie. 2020. The Final-over-Final Condition, Stringency, and Typological Structure. *Linguistic Inquiry* 51.4. 765-784.

Kokkelmans, Joachim. 2021. *The Phonetics and Phonology of Sibilants. A Synchronic and Diachronic OT Typology of Sibilant Inventories*. PhD dissertation, University of Verona.

McManus, Hope. 2016. *Stress Parallels in Modern OT*. Ph.D. dissertation. Rutgers University, New Brunswick, N.J. ROA-1295.

Merchant, Nazarré. 2018. The Contours of nGY. ROA-1342, 1343.

Merchant, Nazarré & Martin Krämer. 2018. The Holographic Principle: Typological Analysis Using Lower Dimensions. ROA-1340.

In addition, we also recommend as relevant to the enterprise:

Merchant, Nazarré. & Alan Prince. 2021. *The Mother of All Tableaux: Order, equivalence, and geometry in the large-scale structure of OT.* (Revised Version). To appear in the series *Advances in Optimality Theory*. Equinox Publishing. ROA-1382.

References

Alber, B. 2005. Clash, Lapse and Directionality. *Natural Language & Linguistic Theory*,23.3. 485-542.

Alber, Birgit. 2015a. Microvariation and typological properties. 1st Workshop on the Formal Structure of OT Typologies (WoFT 1), Rutgers University, May 29-30.

Alber, Birgit. 2015b. Minimality and property analysis. Rutgers Optimality Research Group (RORG), Rutgers University, Oct. 7, 2015.

Alber, Birgit & Alan Prince 2017. The Book of nGX. *Memoirs of the Society for Typological Analysis* 1.1. ROA-1312.

Ausiello, Giorgio & Luigi Laura. 2017. Directed hypergraphs: Introduction and fundamental algorithms — A survey. Theoretical Computer Science, volume 658, Part B. 293-306. https://www.sciencedirect.com/science/article/pii/S0304397516002097.

Bane, Max & Jason Riggle. 2008. Three Correlates of the Typological Frequency of Quantity-Insensitive Stress Systems. *Proceedings of the Tenth Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, volume 10. Association for Computational Linguistics. 29–38. ROA-966.

Bane, Max & Jason Riggle. 2012. Consequences of Candidate Omission. *Linguistic Inquiry* 43.4. 695-706. As "When more choice means less freedom: a note on candidate sets and typologies," ROA-1108.

Bermudez-Otero, Ricardo. 2010. Stratal Optimality Theory (1910).

Bermúdez-Otero, Ricardo. 2018. Stratal Phonology. In S.J. Hannahs & Anna R. K. Bosch (eds.), *The Routledge Handbook of Phonological Theory*, 100-134. Abingdon: Routledge.

Bermúdez-Otero, Ricardo. In preparation. *Stratal Optimality Theory* (Oxford Studies in Theoretical Linguistics). Oxford: Oxford University Press.

Brasoveanu, Adrian & Alan Prince. 2005/2011. Ranking and Necessity: the Fusional Reduction Algorithm. Natural Language and Linguistic Theory ROA-794.

Clements, George N and Samuel J. Keyser. 1983. *CV Phonology*. Cambridge, MA: MIT Press.

Danis, Nick. 2014. Deriving interactions of complex stops. ROA-1220.

Dell, François. 1973. *Les Règles et les Sons*. Collection Savoir, Paris: Hermann.

Frank, Robert & Giorgio Satta, 1998, "Optimality theory and the generative complexity of constraint violability," *Computational Linguistics 24*, 307–315.

Gigerenzer, Gerd. and Daniel Goldstein. 1996. Reasoning the fast and frugal way. *Psychological Review*: 103.4, 650-69.

Hayes, Bruce. 2003-2017. OTSoft: Optimality Theory Software. UCLA.

Hudson, Grover. 1974. The Role of SPCs in Natural Generative Phonology. In A. Bruck, R.A. Fox, and M.W. LaGaly, eds., *Papers from the Parasession on Natural Phonology*, 171-83. Chicago: Chicago Linguistic Society.

Hyde, Brett. 2012. Alignment constraints. *Natural Language & Linguistic Theory* 30: 789-836.

Ito, Junko. 1986. *Syllable Theory in Prosodic Phonology*, Ph. D. Dissertation, UMass, Amherst.

Ito, Junko. 1989. A Prosodic Theory of Epenthesis, *Natural Language & Linguistic Theory* 7, 217-260.

Kenstowicz, Michael & Charles Kisseberth. 1977. *Topics in Phonological Theory*. New York: Academic Press.

Kimper, Wendell A. 2011. *Competing triggers: Transparency and opacity in vowel harmony*. Ph.D. Dissertation: University of Massachusetts at Amherst. Also available from Proquest. AAI3482711.

Kimper, Wendell A. 2016. Positive constraints and finite goodness in Harmonic Serialism. In Joe Pater and John McCarthy, eds., Harmonic Grammar and Harmonic Serialism (Equinox Press, Advances in Optimality Theory series). [Pre-publication version]

Kiparsky, Paul. 1973. Abstractness, opacity, and global rules. In O. Fujimura et al. (eds.), *Three Dimensions of Linguistic Theory*, 57-86. As Part 2 of "Phonological Representations," pp. 3-136, Tokyo: TEC Company, Ltd.

Kiparsky, Paul. 2014. New perspectives in Historical Linguistics. In C. Bowern & B. Evans (eds.), *The Routledge Handbook of Historical Linguistics*, pp. 64-102. Routledge, New York.

Kiparsky, Paul. 2014. Stratal OT. Ms. Stanford University.

Kisseberth, Charles. 1970. On the Functional Unity of Phonological Rules. *Linguistic Inquiry* 1, 291-306.

Kisseberth 1972. On Derivative Properties of Phonological Rules. In *Contributions to Generative Phonology*, M. Brame ed., 201-228. Austin: University of Texas Press.

Mai, Anna & Eric Bakovic. 2020. Cumulative constraint interaction and the equalizer of HG and OT. Proceedings of the Annual Meetings on Phonology 2019. DOI: 10.3765/amp.v8i0.4678. ROA-1366.

McCarthy, John, and Alan Prince. 1986. *Prosodic Morphology 1986.* Ms. University of Massachusetts and Brandeis University.

McCarthy John & Alan Prince. 1993a. *Prosodic Morphology: Constraint Interaction and Satisfaction*. ROA-482.

McCarthy, John & Alan Prince 1993b. Generalized Alignment. In G. Booij and J.v. Marle, *Yearbook of Morphology*: 79-153. Boston: Kluwer Academic Publishers. ROA-7.

McCarthy, John & Alan Prince. 1995. Faithfulness and Reduplicative Identity. ROA-60. Also in *University of Massachusetts Occasional Papers in Linguistics 18: Papers in Optimality Theory*. Ed. by J. Beckman, S. Urbanczyk and L. W. Dickey. pp. 249-384.

Merchant 2008. *Discovering underlying forms: contrast pairs and ranking*. Ph.D. dissertation, Rutgers University. ROA-964.

Merchant 2011. Learning ranking information from unspecified overt forms using the join. CLS & ROA-1146.

Merchant, Nazarré. 2015. The Typology of Phonological Analysis.

Merchant, Nazarré & 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.

Myers, Scott. 1991. Persistent Rules. *Linguistic Inquiry*, 22" 315-44.

Newton, Isaac. 1687. *Philosophiæ Naturalis Principia Mathematica*. Londini : Jussu Societatis Regiae ac typis Iosephi Streater, prostat apud plures bibliopolas, anno MDCLXXXVII.

Montgomery, Richard. 2019. The Three-Body Problem. *Scientific American*: vol 321.2.

Prince, Alan. 1980. A Metrical Theory for Estonian Quantity. *Linguistic Inquiry* 11, 511-62.

Prince, Alan. 2002a. *Entailed Ranking Arguments*. ROA-500.

Prince, Alan. 2002b. Arguing Optimality. ROA-562.

Prince, Alan. 2006. Implication & Impossibility in Grammatical Systems. ROA-880.
Prince, Alan. 2007a. No More than Necessary. ROA-882.
Prince, Alan. 2007b. The pursuit of theory. In P. Lacy (Ed.), *The Cambridge Handbook of Phonology* (Cambridge Handbooks in Language and Linguistics, pp. 33-60). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511486371.003.
Prince, Alan. 2008. RCD – The Movie. ROA-1057.
Prince, Alan. 2016. What is OT? ROA-1271.
Prince, Alan. 2017a. OT Checklist. ROA-1306.
Prince, Alan. 2017b. Representing OT Grammars. ROA-1309.
Prince, Alan, Nazarré Merchant & Bruce Tesar. 2007-2021. OTWorkplace.
Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory*: *Constraint Interaction in Generative Grammar*. Blackwell. ROA-537.
Riggle, Jason. 2010. Sampling Rankings. ROA-1075.
Samek-Lodovici, Vieri and Alan Prince. 1998. Optima. ROA-363.
Stampe, David. 1969. On the Acquisition of Phonological Representation. In *Papers from CLS 5*, R. Binnick, A. Davidson, G.M. Green, and J.L. Morgan, eds., pp. 433-444. Chicago: Chicago Linguistic Society.
Stampe, David. 1973a. On Chapter 9. in *Issues in Phonological Theory*, de Gruyter Mouton. DOI: https://doi.org/10.1515/9783110905137.44
Stampe, David. 1973b. *A Dissertation in Natural Phonology*. Ph.D. dissertation, University of Chicago. Garland Publishing (1979).
Steriade, Donca. 1982. Greek prosodies and the nature of syllabification. Doctoral Dissertation. MIT, Cambridge, MA.
Tesar, Bruce & Paul Smolensky. 1993. The Learnability of Optimality Theory:an Algorithm and some Basic Complexity Results. ROA-2.
Tesar, Bruce & Paul Smolensky. 1996a. Learnability in Optimality Theory (Short Version) ROA-155.
Tesar, Bruce & Paul Smolensky. 1996b. Learnability in Optimality Theory (Long Version). ROA-156.
Tesar, Bruce & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29:229-68.
Tesar, Bruce & Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge: MIT Press.

> De esas premisas incontrovertibles dedujo que la Biblioteca es total y que sus anaqueles registran todas las posibles combinaciones de los veintitantos símbolos ortográficos (número, aunque vastísimo, no infinito) o sea todo lo que es dable expresar: en todos los idiomas.
>
> — La Biblioteca de Babel