Representing OT Grammars

Alan Prince

Last Mod. April 19, 2017

Abstract

Analysis proceeds with varying degrees of faithfulness to the theory that it rests on. Within OT, a commonly encountered chokepoint is the representation of grammars, which not infrequently have a form that differs from expectation. A variety of table types and graphical structures are deployed in the literature to handle the eventualities; many are not up to the task and impose their own debilitating artifacts.

This note sets out to determine where the common representational schemes succeed and where they fail in representing the content of OT grammars. Discussion begins by settling on a notion of grammar that accords with the general meaning of the term within Generative Grammar. Definition in hand, we assess the VT (violation tableau), the dashed VT, the Hasse Diagram, the set of Hasse diagrams, the dotted Hassoid, and the CT (comparative tableau), arguing mostly from familiar examples. Concrete conclusions are drawn about best (and non-best) practices.

Contents

Acknowledgment. Thanks to Natalie DelBusso, Birgit Alber, Nazarré Merchant, and Jane Grimshaw for much discussion of key points in this and related material and for valuable comments on the ms., which have led to attempts at improvement.

Representing OT Grammars

Alan Prince

0 Introductory

IF we want to talk about the grammars provided by a given linguistic theory, we must represent them. If we want to be faithful to the theory and its rendition of the data, we must represent them accurately. In the OT literature, for example, a variety of tabular and graphical structures are offered as representations of ranking requirements. It's worthwhile, then, to sort out the options, with an eye to determining their 'carrying capacity' — what they *can* represent and what they can't — to discern when they give the grammar as defined and when they distort.

Preparatory to anything else, we need to brush off the unclarities that cling to the notion 'grammar'. An OT *system* is arrived at by specifying the candidate sets to be evaluated and the constraints that evaluate them. Where does the notion of a *grammar* emerge from these basics?

To see what's at stake, let's step back and consider the general situation, independent of OT. Typically, in dealing with a single linguistic analysis, we have a set of well-formed derivations, each a sequence of structures, admitted by a congeries of formal conditions. Some subset of the structures in a derivation, — in the simplest case, two: the first and last — are taken to constitute the essentials; their coupling is validated by the existence of a licit derivation between them. For example, under the SPE grammar of English (e.g. SPE:222), we have such input-output pairs as /ab=kēd/, [æksīyd] for 'accede', where the first leads to the second by following the required sequence of posited rules. Let's say that the aggregate of all such pairs constitutes the 'language'. The conditions delimiting all licit derivations define its 'grammar'.

In serialist theories, derivations typically consist of more than one step, and the grammar bears the burden of specifying the steps that can be taken — the legitimate transitions from one structure to another. For example, a grammar might contain 'rules', expressions written in a certain vocabulary and interpreted in a certain way; a derivation would consist of the results of applying those rules in a sequence, starting with a designated form. Some such sequences give you the language you're interested in, some do not; the serialist grammar must also therefore characterize all the desired sequences of rule applications.

Crucially, there is no assumption that only one sequence will work, and no reason to expect any such thing. A grammar delimits *all* successful sequences of rules. One simple way to do this would be to list all the admitted sequences. Since this list is structured, a more valuable line of attack is to cite the conditions that delimit the list. In SPE-like systems, these conditions are generally taken to require of certain pairs of rules that one precedes the other in any licit derivational sequence. If that is the correct way to characterize an SPE grammar, then rule application is governed by a partial order, in the sense that any total sequence of applications that accords with the requirements of the partial order —that is: any of its 'linear extensions'— will deliver the target language.

With this general background in mind, we return to OT as a specific instance of a generative linguistic theory. To jargonize in the familiar way, let's say that an OT system S is defined by GEN.S and CON.S. Optimality is defined by OT itself, as follows. We are given a *candidate set* K — a set of competitors, as defined by GEN.S — and we are given a linear order λ on the constraint set CON.S, a 'ranking', drawn from Ord(CON.S), the set of all linear orders on CON.S. An optimum of K, with respect to λ, is a candidate that survives filtration by λ. Filtration proceeds in the familiar 'take the best, ignore the rest' fashion through λ, whereby each constraint has its say — remain or leave — on the survivors of the previous constraint in the order λ. (For details, see e.g. Prince & Smolensky 1993/2004, henceforth P&S.)

If we apply λ to every candidate set admitted by GEN.S, we obtain a collection of optima — the 'language' delimited by λ. What then is its *grammar*?

The first temptation, and one that has not been staunchly resisted in the literature, is to speak of λ itself as a grammar. But a single linear order or 'ranking' is the beginning not the end of the story. More than one ranking, often, delivers the same optima, a fact recognized from the onset of the enterprise. In this (typical) case, any single ranking that delivers a language L will contain order relations that can be reversed without changing the choice of optima as well as those that cannot. The reversible relations are not linguistically significant. This effect parallels the ordering situation under serialism.

To see how this works in OT, let's consider an example. Suppose a system S_1 has three constraints and just one candidate set ('cset') containing three non-harmonically-bounded candidates, with violations arrayed as in the following display. We do not peer inside the candidates, which we view as opaque entities engaged in an abstract competition.

(1) System S_1 – Violation Assignments to Candidates

S_1	X	Y	Z
a	0	0	1
b	0	1	0
c	1	0	0

Lest it be thought that we have wandered off the true, twisting path into a world of congenial simplicities, recall from Basic Syllable Theory (P&S:§6, 111-114) the subsystem S_V that deals with the fate of problematic underlying vowels.[1] Its typology is defined by a cset of the following shape, which scrutinizes the canonically problematic input /V/.

(2) Vowel System – Violation Assignments to Candidates

S_V	f.max	f.dep	m.Ons
V→ .V.	0	0	1
V→ .*C* V.	0	1	0
V→ ε	1	0	0

We write ε for the empty string and italicize inserted elements for visibility.

[1] 'Problematic' = 'incurs a violation in the candidate no matter how it is parsed in optima'.

The same structure is found mutatis mutandis in the subsystem S_{Cod} dealing with consonants that can be faithfully syllabified as codas (P&S:§6,114-115). Their tale is told by the fate of the final C in /CVC/.

(3) Coda System – Violation Assignments to Candidates

S_{Cod}	f.max	f.dep	m.NoCoda
CVC→.CVC.	0	0	1
CVC→ .CV.C*V*.	0	1	0
CVC→ .CV.	1	0	0

Following the remark of P&S:112, we observe the following generalization: given a ranking in any of these systems, the winner is the candidate whose single violation is assessed by the bottom-ranked constraint. Although there are 6 = 3! linear orders on CON.S for each of the cited systems, there are only 3 languages. For S_1, these are {a}, {b}, and {c}. Each language is delivered by two different linear orders. Consider the language{a}, for example:

(4) System S_1. Ranking: X≫Y≫Z. Optimum: a.

S_1	X	Y	Z
☞ a			1
b		1	
c	1		

(5) System S_1. Ranking: Y≫X≫Z. Optimum: a.

S_1	Y	X	Z
☞ a			1
b	1		
c		1	

The pointing finger indicates the optimum. We omit zeroes for clarity. In this note, we use heavy verticals to indicate that a linear order is asserted.

The two distinct orders that produce language {a} fall together under one rubric: X and Y *both* dominate Z. No relation is required between X and Y: any will do. If we want to use the term *grammar* to refer to the linguistically significant conditions determining optima, or to the collection of rankings that those conditions denote, then the ranking X≫Y≫Z is *not* by itself a grammar, nor is Y≫X≫Z. Instead, these are the *linear extensions* of the grammar, when the grammar is construed as a set of defining conditions: for this reason, Merchant & Prince designate a ranking in a grammar by the acronym *leg* = *li*near *e*xtension of a *g*rammar. The object we want to be talking about is the *set* of all rankings, all legs, that yield the same optima, and the conditions that delimit that set.

At the intuitive level, this is not a startling new insight.[2] As the reference to P&S indicates, linguistic analysis under OT has always been understood to involve the hunt for the general conditions that delimit the *grammar* in just this sense. Once we are clear about the object of study, leaving behind the implicit and the plausible, we must ask what is needed to *represent* the grammars that emerge from the theory. And we can evaluate attempts, dismiss the failures, and embrace those that succeed.

1 The VT and its discontents

It was good for Paul and Silas.
It's good enough for me.
– Old Song

The original Violation Tableau (VT) display of P&S is designed to bring out key aspects of the filtration process.

(6) System S_1. Ranking: X≫Y≫Z. Optimum: **a**.

S_1	X	Y	Z
☞ **a**			*
b		*!	
c	*!		

The manicule points at the optimum. Shading delimits the set of competitors facing each constraint, up to the point where competition ceases, whereupon the entire column is engulfed.[3] The ! marks the point where a candidate is ejected, losing out to a set of unshaded survivors.

This leads us to our first, obvious conclusion: the VT is a fine and faithful way to represent the filtering action of a single leg, with the order of columns taken to be a ranking order, shown here typographically by heavy vertical lines.

But since a grammar may contain more than one leg, the VT in its simplest form does not provide a structure in which grammars can be generally represented.

Three ways of responding to this observation have emerged into common use.

(1) Enrich the conventions for structuring and interpreting tableaux.
(2) Use a non-tabular graphical structure based on the Hasse diagram.
(3) Use the Comparative Tableau (CT).

Let us begin with VT enrichment, which is easily seen to be a dead end, or worse.

---

[2] At the non-intuitive level, for those who find such things clearer, a characterization of the idea of a grammar is found in Appendix I. The idea of grammar as a set of legs is, as noted, implicit in works such as P&S and Prince 2002a, and has been taken as basic in works such a Riggle 2010, Yanovich 2012, Merchant & Prince 2016.

[3] A more straightforward convention, dropping the codicil, would simply and uniformly divide each column between the light (the set of live candidates) and the dark (the has-beens in the shade). An optimum row would, under this improvement, never carry shading. Examples of this usage may be found in the literature.

1.1 Dashed Hopes

For now we see through a glass, darkly… [4]

A natural move is to generalize the VT so that it can denote sets of rankings. This is conventionally accomplished by distinguishing between the solid vertical line, which signals order between the columns it separates, and the dashed line, marking lack of order. This works for the case at hand.

(7) System S_1. 2 Leg Grammar: {X≫Y≫Z, Y≫X≫Z}. Optimum: **a**.

S_1	X	Y	Z
☞ **a**			1
b		1	
c	1		

Such a tableau is divided into consecutive blocks set apart by the solid verticals. Within a block, any order is allowed; but the members of a block must dominate everything in the other blocks to the right. From (7) , under these conventions, we have X & Y ≫ Z, exactly as desired.

At this point, the shading notation is already compromised. It depends on having a unique survivor set at each column, but within a block every pattern of departure among the block's losers is going to appear in some linearization. One might contemplate notational maneuvers, but it's not worth trying to save shading, because the dashed, blocked tableau is not worth saving.

To see this, we ask the fundamental question: how does the carrying capacity of the blocked tableau measure up against what's needed for OT? It is easily seen that the blocked tableau does not even render all partial orders, much less the generic OT grammar.

Let's work from an example, so that the linguistic import of such failures is clear. Consider the grammar of the language (C)V.ins, drawn from the Basic Syllable Theory (BST) of P&S: §6. In this language, an optimal output contains only open syllables, and any 'problematic' input C — one that cannot be syllabified without incurring a violation: structurally, one that cannot be syllabified into onset position with a following input vowel— is preserved by vowel insertion. Thus, /CVC/→ .CV.C*V*., in which the second C is problematic and the epenthetic vowel (which incurs the problematizing violation) is italicized.

Here's a sketch of BST, offered as a quick reminder of its assumptions. See Merchant & Prince 2016:§1 and Alber & Prince (ms.) for recent investigation of the notions involved.

[4] "Because there were few ways to make a smooth piece of glass with a uniform thickness, these ancient glass-mirrors were made by blowing a glass bubble, and then cutting off a small, circular section, producing mirrors that were either concave or convex. These circular mirrors were typically small, from only a fraction of an inch to as much as eight inches in diameter.[11] These small mirrors produced distorted images, yet were prized objects of high value." Wikipedia, Mirror.

● CON.BST includes 5 constraints,

- 2 markedness: m.Ons (*[$_\sigma$V), m.NoCoda (*C$_\sigma$])
- 3 faithfulness: f.max (penalizing deletion); f.depV, f.depC (penalizing V-, C-insertion).

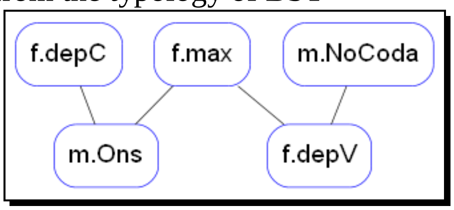
● GEN.BST recognizes any non-empty string over {C,V} as an input, and requires that outputs be fully parsed into non-overlapping *syllables*, which are constituents drawn from the repertory (C)V(C).

- A *candidate* is a triple ⟨input, output, corr.IO⟩, where corr.IO is a relation between input and output segments, treated as a partial function in Merchant & Prince: 42. Outputs preserve the segmental order of their corresponding inputs; an input segment may correspond to at most one segment in the output and vice versa.

- Correspondence may be properly partial, in the sense that input segments may lack output correspondents (deletion) or output segments may lack input correspondents (insertion).

These assumptions yield a typology of 12 languages (P&S: §6, augmented by Riggle, 2004:108ff.). The *grammar* of (C)V.ins consists of all 16 rankings of CON.BST that accord with the requirements of the following partial order: its *linear extensions*. See Appendix II for a list.

(8) (C)V.ins, from the typology of BST



It's not hard to parse this grammar qualitatively. Codas are disallowed — m.NoCoda is not crucially dominated (top right). Consonantal issues are resolved through insertion, because f.max≫f.depV (right side). Onsetlessness is allowed via faithful reproduction of certain input configurations, because m.Ons (lower left) is dominated by both relevant faithfulness constraints, f.depC and f.max.

These effects are achieved by requiring only 4 order relations, contrasting with a total of 10 (some redundant) that appear in a linear order on five elements, where the relationship of every constraint to every other is fixed.

A glance reveals, for example, that f.depC (upper left of diagram) is orderwise *incomparable* with every constraint except m.Ons (lower left). This means that although f.depC≫m.Ons in every leg, for any other constraint there are legs of (CV).ins in which f.depC dominates it and legs in which f.depC is subordinated to it. The reasons are clear: f.depC and f.depV cannot interact directly, because vowel epenthesis and consonant epenthesis do not supply competing alternatives; put broadly, they are solutions to different problems. In addition, f.depC has no required domination relations with f.max and m.NoCoda (top row), since all three are completely unviolated in optima of (C)V.ins, which entirely lack deletion, epenthesis, and codas.

The linear extensions of partial order (8) include every possible pairwise relation between f.depC and the members of its basket of incomparables, namely{f.max, m.NoCoda, f.depV}. But any given leg imposes a specific relation between f.depC and each of these: whatever it is, it has no effect on the outcome and is therefore an artifact of linearity, of no linguistic significance. The significance of this (local) ordering fact is that it is (globally) insignificant.

But these relations defy representation in a blocked tableau, in which incomparables must sit in the same block. And they must share the order relations with constraints outside the block. But f.depC cannot be blocked with both f.depV and m.NoCoda. That would place these two constraints in the same block, but they are not incomparable: m.NoCoda≫ f.depV.

If we even go so far as to put f.depC and m.NoCoda in one block, which does not lose any required order relations, we have nonetheless lost the ability to discriminate their distinct relations to m.Ons and f.depV. Consider, for example, m.Ons.

- f.depC ≫ m.Ons in *all* rankings of (C)V.ins.
- m.NoCoda ≫ m.Ons in some of those rankings and m.Ons ≫ m.NoCoda in others.

Similar remarks may be made with respect to m.NoCoda and{f.max, f.depC, m.Ons}.

It is easy to construct various blocked tableaux which yield some subset of the contents of the grammar of (C)V.ins. But none of them delimits the entire grammar. This flatly rules out the blocked tableau as a general representation scheme for grammars.

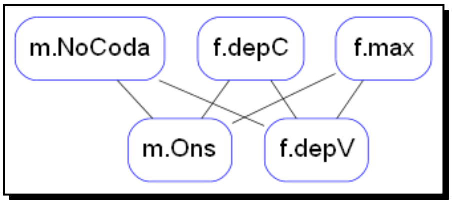
The blocked tableau is, of course, perfectly suited to the representation of the stratified hierarchy produced by RCD (Tesar &Smolensky 1993:8ff; Prince 2009). In this structure, each constraint is placed in the highest possible stratum that it can sit in without contradicting the ranking requirements on it. In the (C)V.ins case, we'd get the following as the output of RCD:

(9) (C)V.ins as rendered by RCD

(C)V.ins	f.depC	f.max	m.NoCoda	m.Ons	f.depV
…	…	…	…	…	…

This representation — or any which equivalently identifies the constraint blocks — is unfaithful to the data. It introduces 2 extra ranking relations, artifacts of the commitment to block structure, reducing the number of linear extensions to 12. Its Hasse diagram looks like this:

(10) Not the grammar of C(V).ins, from RCD



The output of RCD is an object of interest,[5] but it does not aim to be a grammar, and typically is not one. It can be viewed as a form of lossy compression of grammatical information, useful in certain circumstances, but seriously misleading in others.

Since RCD does not give us the set of legs which select L, its products have no immediate status in the theory of grammar. RCD remains important because its *failure* signals the sure and certain *non*existence of a grammar. This is crucial knowledge: whenever we want to know whether a given combination of desired optima can coexist grammatically, as we do when constructing a typology from a collection of csets, we have RCD to tell us. But determining existence and nonexistence does not require that any kind of ranking structure be recognized or built (Prince 2002b:18ff). That's an add-on that gives a quick, partial view of the grammar.

RCD and its less efficient modifications (Prince & Tesar 2009, Hayes 2009) play a continuing role in OT learning models. But in learning, the goal need not be to obtain the entire *grammar* as it is defined from the premises of the theory. It suffices to obtain a ranking or rankings that will generate the data, subject to restrictiveness requirements with respect to the unseen (Prince & Tesar 2004:1-2), and, possibly, concerns with efficiency that may lead to further departures and short-cuts. With the advent of an algorithm for determining the exact grammar motivated by data (Fusional Reduction, 'FRed', Brasoveanu & Prince 2005/11), the role of incomplete heuristics has necessarily diminished or even disappeared in what we might call 'woke' work.

Finally, we note that the dashed tableau provides what certain analysts of human misbehavior have termed 'the occasion of sin'.[6] As our example (10) shows, even a two-blocker can hide ranking claims that require more justification than an author might be willing or able to give. As the numbers of blocks rises, the number of implicit ranking claims rises steeply. If a dashed tableau is offered as an authentic representation of a grammar, the burden of justification swells accordingly, growing ever less likely to be confronted and discharged.

[5] For a look at its formal properties, see Prince 2002a:21ff.

[6] In this case, *sloth* and, though esteemed less deadly surely more troubling to rationality, the assertion of a 'thing which is not'.

2 A Gathering of Arcs

Having bumped up against two logically impenetrable barriers, let's now turn to a successful method of representing OT grammars based on the use of Hasse diagrams. We'll work this time from an example which is simple, abstract, and carries the features essential to showing the strengths and shortcomings of the method.

Consider an OT system S_2 that recognizes only two candidates and three constraints, with values assigned as shown.

(11) Violations in S_2

S_2	X	Y	Z
a	0	0	1
b	1	1	0

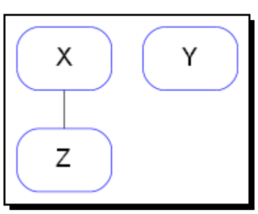
Choosing **a** as optimal, we obtain the following 4 legs as its grammar.

(12) Grammar of language {**a**} as the set of rankings delivering {a}

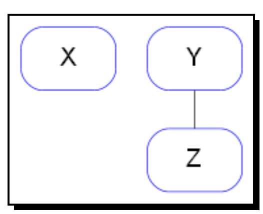
i. X ≫ Y ≫ Z
ii. X ≫ Z ≫ Y
iii. Y ≫ X ≫ Z
iv. Y ≫ Z ≫ X

From this we may extract the defining formulation: **a** is optimal iff X≫Z *or* Y≫Z. No conjunctive paraphrase exists in terms of atomic A≫B statements: these rankings therefore do not linearly extend a single partial order. But they can be rendered faithfully as the *union* of (the linear extensions of) two partial orders. The grammar may be faithfully represented as (the linear extensions of) two distinct Hasse diagrams.

(13) Grammar of {**a**} diagrammatically

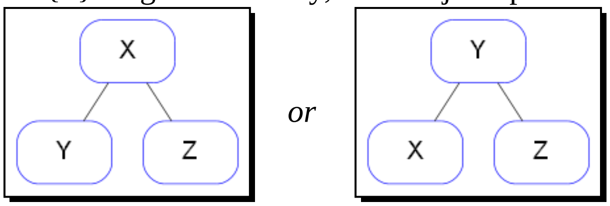


or



It is tempting to imagine that we have sliced the grammatical knot into two neat halves. But the two partial orders overlap: the legs X≫Y≫Z and Y≫X≫Z extend both. Should we wish to render the union disjoint, we can introduce new ranking relations solely for that purpose. Subjecting the left diagram to the condition X≫Y and the right to Y≫X does the trick, splitting the grammar between rankings {i, ii} and {iii, iv} of ex. (12) . The effect is shown immediately below:

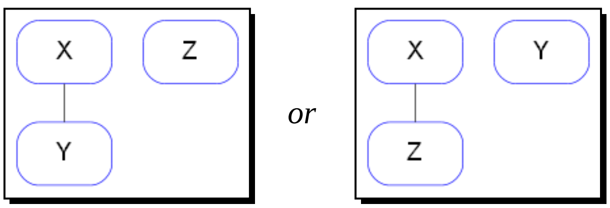
(14) Grammar of {**a**} diagrammatically, with disjoint partial orders



The cost (perhaps intolerable?) is that the onlooker must contend with two extra ranking relations that are motivated not by data or theory, but only by the desire for disjointness.

This is a hint, perhaps, of the broader fact that the notion 'union of partial orders', in its generosity, fails to tells us what an OT grammar can be. To see this, one need look no further than the following simple disjunction: X≫Y *or* X≫Z. On the face of it trivially different from the one we are examining, this condition fails to describe an OT grammar.

(15) Not a grammar



No VT or set of VTs can produce this collection of rankings. The heart of the matter is that OT grammars do not allow free disjunction of ranking conditions, as will quickly become apparent when we examine immediately below the conditions they are actually built from. In outline: a partial order can be delimited by conjoining conditions of the form A>B, where A and B are single elements being ordered. An OT grammar allows term A to involve disjunction, but not B.[7] Thus, 'X or Y dominates Z' is found all the time, but 'X dominates Y or Z' is not even possible. To see why this is so, we take the one final step that is required to achieve a fully adequate representation which denotes all and only the possible OT grammars.

3 A Grammar of ERCs

Let's return to our touchstone problem with the intention of solving it outright. We have a system S_2 that recognizes only two candidates and three constraints, with values as shown.

(16) Violations in S_2

S_2	X	Y	Z
a	0	0	1
b	1	1	0

[7] An OT grammar is an *antimatroid*, a combinatorial structure richer than a partial order. See Riggle 2004, and Merchant & Riggle 2012, 2016 for proof.

We want the conditions under which the language {**a**} is chosen. Any linear order on {X,Y, Z} which selects {**a**} must have the following property, which we'll call 'choice'.

Choice. The highest ranked constraint distinguishing **a** from **b** *favors* **a**.

To be explicit about the terminology: writing C(**k**) for the violation value assigned by C to the candidate **k**, we have

- A constraint C *distinguishes* **a** from **b** iff C(**a**) $\neq$ C(**b**).
- A constraint C *favors* **a** over **b** if C(**a**) $<$ C(**b**).

The property *Choice* depends on the local comparative behavior of each constraint in the system: Does C

- favor **a**?
- favor **b**? or
- fail to distinguish between them?

If we know which, we know everything about C, **a**, and **b** that is relevant to C's role in the grammar of S_2. If we know this about every C, then we have the grammar in our hands, because we have the key distinctions that are deployed in every ranking condition the data supports.

To pursue these observations, we introduce a data transformation that extracts the tripartite comparative judgments issued by OT constraints, improving on the 'mark cancellation' of P&S.[8] By this, each violation-assessing constraint C defined in S.CON, which looks at single candidates, is associated with a comparator that looks at a *pair* of candidates, one asserted optimal. It is a customary laxness in the literature to denote these related notions of 'constraint' by the same name, but they are clearly different beasts. In the interests of maintaining clarity, if only briefly, we denote the comparative congener of C by dC. [9]

While C takes on values in the nonnegative integers {0,1,2,….}, the comparator dC has only three possible outputs: better, worse, same. We denote these in the familiar way, as follows:

(17) Comparator Values

dC(**a**,**b**) = **W** iff C(**a**) $<$ C(**b**). '**a** better than **b**' = 'C favors **a** over **b**'
dC(**a**,**b**) = **L** iff C(**b**) $<$ C(**a**). '**b** better than **a**' = 'C favors **b** over **a**'
dC(**a**,**b**) = ***e*** iff C(**a**) = C(**b**). '**a** same as **b**' = 'C sees no difference btw. **a** and **b**'

[8] Mark cancellation (P&S:258, 261) recognizes the role of pairwise comparison, but works by reducing the number of violations in each constraint to 0 in either the desired optimum or its competitor, whichever has fewer. A reduced 0 in the desired optimum is ambiguous between W and *e*, while a reduced 0 in the competitor is ambiguous between *e* and L. Thus, the (adjusted) profiles of both are needed simultaneously for interpretation. To arrive at a single comparative representation, we must take leave of the non-negative integers. The comparative values are essentially the sign of the *difference* between the competitor and the desired optimum, marked as {1,–1, 0}.

[9] How different? The domain of C is the set of candidates, while the domain of dC is the set of *pairs* of candidates. An ERC results when the dC's are applied to a pair of competitors, the first asserted optimal (h/t N. Merchant). The codomain of C is the nonnegative integers {0,1,…}, while the codomain of dC is {W,L,*e*}, which can be naturally rendered numerically as {1,–1,0}. Curiously, C and dC are sometimes referred to as different 'formats' even though they give different information about different objects. It's roughly like referring to the derivative $f'(x)$ as another 'format' for f(x), so that e.g. 2x is a different 'format' for the data of x^2, or $x^2 + 7$, for that matter.

Applying this transformation to our example, we arrive at a comparative tableau (CT). We write **a** ~ **b** for the ordered pair (**a,b**). The ordering of the pair orients the distinction between W and L.

(18) Comparison in S_2

S_2	dX	dY	dZ
a ~ b	W	W	L

The property *Choice* declares that **a** will be chosen over **b** in every ranking in which X≫Z as well as in every ranking in which Y≫Z. Either of these conditions ensures that the highest ranked constraint distinguishing **a** from **b** will be one that favors **a**.

This is because dX(**a**,**b**) = W and dY(**a**,**b**) = W. By contrast, dZ(**a**,**b**) = L, absolutely barring Z from the decisive, highest-ranked-distinguisher position in any leg.

In moving to the comparative domain, we have liberated ourselves not only from the details of violation, but also from the candidates that give rise to them. Violation data is parochial to a candidate set; but from the pattern of W,L,*e* assignments, we gain a condition that holds of the entire grammar we seek. The interpretation of a W,L,*e* assessment follows directly from the way selection of optima is defined in OT, in terms of the highest ranked distinguishing constraint. It runs like this: given the presence of both W and L, it must be that *some* W dominates *every* L in every ranking that belongs to the grammar.

This the 'Elementary Ranking Condition' — the core statement out of which grammars are built. Let's abbreviate this as *ERC*. Given a fixed but arbitrary sequence of the constraints in CON.S, useful for keeping track of the constraints, we can represent any ERC as an 'ERC vector', a sequence of W,L, and *e* values, like WWL in example (18) above. Following common usage, we will typically refer to an 'ERC vector' simply as an 'ERC', tolerating the slight risk of confusion between the condition and its representation.

For any collection of violation data, transforming it into ERCs is guaranteed to provide us with a complete accounting of all the ranking conditions that are imposed by the data. If the data is sufficient to determine the grammar (a valid 'support' for the grammar, in the lingo), then we have what we have been seeking: a faithful representation of the grammar.

There are typically many such representations, logically equivalent in their consequences. The three-valued ERC carries a full logic with it (Prince 2002ab), and the operations of that logic may be used to simplify and reduce any ERC set down to a maximally concise representation, or 'basis' (Brasoveanu & Prince 2005/11). Of greatest interest are the MIB or 'Maximally Informative Basis', which displays, for each disjunctive set of dominators, all the constraints that they may or must dominate, and the SKB, or 'Skeletal Basis', which displays local domination relations only and eschews those derivable from the transitivity of the ranking relation.[10]

[10] The SKB is the (transposed) incidence matrix of the transitively-reduced directed hypergraph representing the grammar, the analog of the Hasse diagram for partial orders. A hypergraph allows edges to connect any number of vertices, so that a collection of (disjunctive) W constraints (qua vertices) may be connected by one 'edge' to a collection of L constraints (qua vertices). Each such edge is described by an ERC.

To get a sense of how this plays out, let's conclude by looking at a grammar drawn from one of the typologies that arises in the analytic study of stress prosody. The system nGX of Alber & Prince (ms.), further discussed in Alber, Delbusso, and Prince 2016, studies a mildly abstract typology of 'Quantity Insensitive' languages in which there is no distinction between main and secondary stress and in which every output form must have at least one stressed syllable. It may be defined as follows:

(19) **GEN.nGX:** definition of Candidate, Candidate set

- a. **Candidate** = ⟨ In, Out ⟩
- b. **In**. A string of syllables.
- c. **Out**. Given In of length *k* syllables, a parse (v. inf.) of *k* syllables in length.
- d. **Cset**. A cset consists of every candidate containing the same **In**.

GEN.nGX assumes the following prosody:

(20) **GEN.nGX**: definition of structures.

- a. A **syllable** is considered to be an atomic unit, the terminal node in prosodic structure.
- b. A **foot** consists of a string of one or two syllables.
- c. A syllable (σ) may belong to at most one foot.
- d. A foot has a unique **head**.
- e. A **Prosodic Word** (PrWd) consists of feet and syllables.
- f. A Prosodic Word contains at least one foot.
- g. **Parse**. A *parse* of a syllable string S is a PrWd, of which S constitutes the terminals.

The structure evaluated in nGX is thus an ordinary prosodic tree, rooted in PrWd, terminating in syllable nodes, in which PrWd dominates at least one foot, and in which each foot is headed. We allude to this structure with the following concise notation:

-	edge of foot or unparsed syllable
F	foot
X	head of foot
u	nonhead of foot
o	syllable not belonging to foot

With these conventions, the constraints of nGX come out like this:

(21) **CON.nGX**

Name	Definition	Accumulates a violation for:
a. Parse-σ.	*o	each unparsed syllable
b. Iamb	*-X	each head-initial foot
c. Trochee	* X-	each head-final foot
d. AFL	*(σ,F): σ…F	each pair (σ,F) where σ precedes F
e. AFR	*(σ,F): F…σ	each pair (σ,F) where σ follows F

Verbosely put: Parse-σ penalizes syllables outside feet; Iamb and Trochee penalize left and right headed feet, respectively, agreeing that monosyllable feet -X- are bad; AFL and AFR are the familiar Generalized Alignment constraints ('All Feet Left/Right', McCarthy & Prince 1993) that sum distance in syllables of each foot from the edge, formulated essentially as in Hyde 2012.

These definitions give rise to a typology of 12 languages, analyzed in Alber & Prince (ms.), and Alber, DelBusso, & Prince 2016). Of particular interest for our purposes are the grammars of the 'sparse' languages: those that admit only one foot per Prosodic Word. There are four of these, which show perfect left-right symmetry at both PrWd (AFL, AFR) and Foot (Iamb, Trochee) levels. Examining any one of these will therefore reveal everything about all of them. Let's look then at the sparse, left-aligned, trochaic instance, named sp.tr.L in the cited literature. The language has optima like these:

(22) Optimal Outputs of sp.tr.L, 2σ – 5σ

-Xu-	F
-Xu-o-	F-o
-Xu-o-o-	F-o-o
-Xu-o-o-o-	F-o-o-o

The Skeletal Basis of sp.tr.L is the following, using blanks for *e* to highlight the W, L distribution. Constraint names in the header are given bare, because the ERCs describe their ordering in legs, rather than their effect of data pairs.

(23) Grammar of sp.tr.L as SKB

nGX	Trochee	Iamb	AFL	AFR	Parse-σ
ERC #1	W	L			
ERC #2			W	L	
ERC #3		W	W		L

This is easily read.

ERC #1: Trochee ≫ Iamb 'Feet are trochaic'
ERC #2: AFL ≫ AFR 'Alignment is to the left'
ERC #3: Iamb ∨ AFL ≫ Parse-σ 'One foot per word'

As usual, the notation A∨B≫C is short for (A≫C) ∨ (B≫C).
The third ERC deserves comment. It tells us that either Iamb or AFL dominates Parse-σ in every leg in the grammar.

● In the first subcase, Iamb≫Parse-σ. But since Trochee≫ Iamb (ERC #1), it must also be the case that Trochee≫Parse-σ, by transitivity of ranking. Thus it is inevitable that both left-headed and right-headed feet are filtered out before Parse-σ is reached, up to the GEN.nGX requirement that a word must contain at least one foot. The number of feet is minimized (to one), on the grounds that all foot types are proscribed.

● In the second subcase, AFL≫Parse-σ. Since AFL penalizes all non-initial feet, it overrules the distaste for unfooted syllables, expressed in Parse-σ, that could compel their presence. Feet are minimized to avoid misalignment.

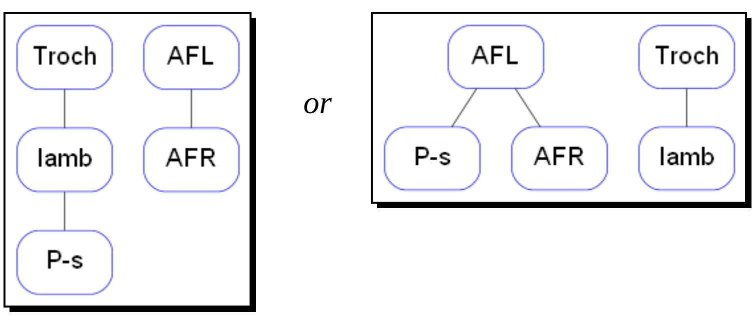
The constraint system contains two quite different explanations for the phenomenon of sparseness. The ERC accommodates this disjunctivity, and indeed renders it perspicuous.

Let's turn now to Hasse-based diagrammatic representations of the grammar. First, consider the one with overlapping disjuncts. This is arrived at by exploding the disjunctive ERC#3, which we can concisely write out as *e*W.W*e*.L, following the order of constraints in ex. (23) and separating the natural groups of constraints by periods, as *Foot-type***.***Alignment***.***Parse*. We replace the W's one at a time with *e*'s, so that *e*W.W*e*.L yields as disjuncts *e*W.*ee*.L (Iamb≫Parse-σ) and *ee*.W*e*.L (AFL≫Parse-σ).

(24) Exploding a disjunctive ERC: Erc#3 ≡ Erc#4a ∨ Erc#4b

nGX	Trochee	Iamb	AFL	AFR	Parse-σ
Erc #3		**W**	**W**		**L**
Erc #4a		**W**			**L**
Erc #4b			**W**		**L**

(25) A Hasse representation of sp.tr.L (nondisjoint)

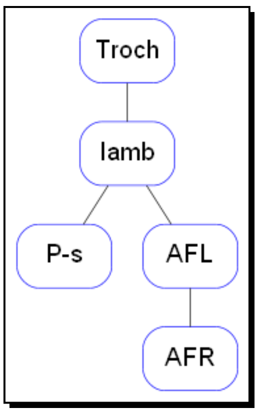


As an alternative, following the strategy ex. (14) above, we could ensure that the grammar's component partial orders are disjoint by imposing Iamb≫AFL on the left and AFL≫Iamb on the right as additional conditions. This move is arrived at by exploding the disjunctive ERC *e*.WW.*e*L by serially replacing the W's with L's, thus yielding *e*W.L*e*.L (Iamb≫AFL & Parse-σ) as one disjunct and *e*LW*e*.L (AFL≫ Iamb & Parse-σ) as the other.

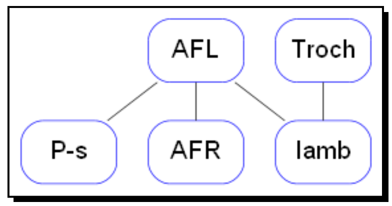
(26) Exploding a disjunctive ERC into disjoint disjuncts: Erc#3 ≡ Erc#5a ∨ Erc#5b

nGX	Trochee	Iamb	AFL	AFR	Parse-σ
Erc #3		**W**	**W**		**L**
Erc #5a		**W**	**L**		**L**
Erc #5b		**L**	**W**		**L**

(27) Another 2-Hasse representation of sp.tr.L, disjoint due to added Iamb/AFL relations



or



The disjoint formulation (27) segregates the legs of the grammar into its two different explanations for sparseness. The collaboration of Troch and Iamb on the left side ensures that a filtered candidate set exits Iamb in any left-side leg before any alignment constraint has been seen, ensuring that all its surviving members are sparse at that point by virtue of foot-type considerations. Similarly, the dominance of alignment on the right side ensures that in every right-side leg, at the point of exit from AFL, before the foot-type constraints have had a chance to collaborate, full sparseness is achieved throughout the surviving candidate set. Together, these two partial orders denote exactly the same total set of legs as in the first formulation in (25) , with the caveat, noted above, that we've inserted rankings derived from explanatory considerations (Iamb ≫AFL, AFL≫Iamb) into a portrayal of rankings motivated by data.

Hasse diagrams are powerful tools in that they engage the human visual system, and particularly so in the case where the grammar is a (single) partial order. When it is not, as with sp.tr.L, the profusion of diagrams leads to crucial opacities. Look back at (25) and (27) . Wherein lies the source of the disjunction? The Skeletal Basis , by contrast, lays it out for all to see: only ERC #3 — eW.We.L — contains multiple W's, and only these can be the source.

The Hasse diagram is a picture, albeit one that is precisely definable (Davey & Priestly 1990:7). The eye's plain version is a thing apart: even though the multiple Hasse diagram strategy denotes exactly the legs of the grammar, *analysis* needs an algebraic structure to operate on. The ERC set provides exactly that, even for partial orders; and beyond that, for all OT grammars, and for nothing else.

4 Make straight the way

In the search for an adequate representation of the OT grammar, we have reviewed the principal formal devices on offer, evaluating their ‘carrying capacity’ — the range of order structures they accommodate.

● The VT with ordered columns, and its generalization the blocked VT, both fall short, because a grammar is not guaranteed to be a single linear order, or even a set of orders that linearly extend a stratified hierarchy.

● The single Hasse diagram encompasses both species of VT, but still falls short, because an OT grammar is richer structure than a partial order (an *antimatroid*, see refs in fn. 7).

● The notion of a *collection* of Hasse diagrams suffices to include all OT grammars, and may therefore be used without fear of distorting the data. But the general set of Hasse diagrams overshoots the mark because it is capable of describing any collection of linear orders whatever (because a linear order is also a partial order), even those that do not qualify as OT grammars.

In stark contrast, the ERC set delimits exactly the OT grammars. This is readily seen, because of the fact, directly derivable from the basic definitions of the theory, that a grammar is delimited by a collection of candidate sets equipped with violation data. Any set of violation profiles with a desired optimum is transformable into an ERC set, per the definition of comparative values in (17) , thereby extracting all of its ranking information. Additional information-conserving logical manipulations (Prince 2002a) reduce any ERC set to its most concise forms (Brasoveanu & Prince 2005/11). Other logical transformations reveal further structure in grammars and typologies (Merchant 2008, 2011; Merchant & Prince 2016). It follows that the structure and properties of OT grammars are revealed in the study of ERC sets. If the goal of linguistic analysis is to arrive at a grammar, then the goal of OT analysis is to arrive at an ERC set.

5 Practices make perfect

Here error is all in the not done,
all in the diffidence that faltered …
- Canto LXXXI

We’ve got the logic. How then must we act? The study of ‘carrying capacity’ is driven by one imperative: *use it for what it’s good for*.

The goal is to advance with the theory as it is, rather than as seen through a haze of conjecture and half-measures. Let’s review the findings, to discern the ‘best practices’ that will put the analyst on a direct route to the grammar.

5.1 The VT

The primary functionality of the VT is to display the results of constraint evaluation, an absolute necessity for revealing the force of GEN.S and CON.S. If the column order is taken to be the ranking order, then the VT provides a perspicuous format for the display of filtration by one leg of the grammar.

Where the grammar has more than one leg, the VT typically cannot present the grammar and should not be called upon to do so. Notational efforts to compensate for this lack, for example the introduction of dashed vertical lines to represent blocking of incomparables, fall far short of what's needed to faithfully represent even the most ordinary grammars. In addition such maneuvers carry a moral hazard: they impose a burden of justification that can rarely be met and is just as rarely attempted.

The violation display provides the fodder for determining the ranking conditions imposed by data. But it does not immediately tell us what those conditions are: calculations must be made, exiting the vocabulary of violations. Make use of the VT to do what it does; then move on.

5.2 The CT

Calculations made, they can be displayed accurately in the CT ('comparative tableau'), which records the three-way distinction (better-than/worse-than/same-as) that governs ranking. The ERC, written on {W,L,*e*}or an equivalent such as {1,–1,0}, is subject to manipulations that reveal the grammatical structure of the data and allow for canonical, maximally concise representations that are completely faithful to the data (Prince 2002ab, Brasoveanu & Prince 2005/2011). The ERC set is co-extensive with the notion of an OT grammar.

5.3 The Hasse diagram

A single Hasse diagram gives an accessible, accurate, transitively reduced graphical representation of a partial order. Not all grammars are partial orders, but some are; for those, the Hasse diagram provides a faithful representation. When grammars fall outside the class of partial orders, they require a directed hypergraph, which (qua diagram) is by no means guaranteed to be as user friendly as its name might suggest.

Abandoning hope of portrayal in a single structure, a grammar may be represented, accurately, as a *set* of Hasse diagrams: the grammar is understood to be the union of the sets of linear extensions of each, with caveats about the potential for overlap in the leg sets denoted by the individual diagrams and the artifacts that can be introduced to render them disjoint.

For purposes of grammatical analysis, an algebraic representation is required, and the ERC set reappears. The Skeletal Basis is the (transposed) incidence matrix of the transitively reduced directed hypergraph given by the grammar, the Most Informative Basis is the (transposed) incidence matrix of the transitively closed version of the grammar, and either or both may serve.

5.4 Don't go there

Do the right thing of course, but best practices also involve prudently avoiding the likelihood and enticements of error. We conclude by noting a few familiar objects, calculations, and non-calculations that should be struck from the list of acceptable representations.

● **The products of RCD**. RCD is an invaluable tool for efficiently determining the viability of a set of desired optima and the consistency of its associated ERC set. RCD can be interpreted to yield a set of rankings, each of which delivers the desired optima: but it is not designed to, and generally does not, produce a grammar. Do not use it for that purpose.

● **The dashed VT**. As noted, the use of dashed verticals to stratify a list of constraints is adequate to represent the output of RCD and therefore fails with RCD in the representation of grammars. With the availability of the CT, where order information lies in the entries rather than the sequencing of columns, this heuristic is definitively superannuated and should be avoided.

● **The dotted Hassoid**. OTSoft (Hayes 2017) modifies the Hasse diagram to represent disjunction: using dotted instead of solid lines leading down to a node to represent the disjunction rather than conjunction of dominators (hence the suggested term: Hassoid.) This lacks the carrying capacity to represent the general case, which requires indexing of explicitly directed edges (Prince 2006:54-55), reflecting ERC structure. See Appendix III for examples.

● **The Hybrid Tableau** (HT). Holding a deserved position at the bottom of the list is the Hybrid Tableau (Prince 2002b:6) which displays violation and comparative information simultaneously in the cells of one array. A principal objection is that these are grossly different types of information, with different meanings attached to the constraint at the head of the column (see fn. 9 for details). Perhaps more damaging: while violations are parochial to a candidate set, ranking information is transportable, subject to logical manipulation, and relevant to the whole grammar, not just to the local status of a particular candidate. The HT is attractive in that its coordinated display promises a view of how one type of information converts into another. But simultaneous presentation of violation and comparison makes it difficult to grasp either, and requires that they be separated into VT and CT for purposes of analysis. Deprecated.

6 Conclusion

If the study of Generative Grammar is at least in part the study of grammars — their content, structure, and relations — then it falls to our lot to represent them in a way that allows us to carry out this study. We are fortunate in that the premises of OT are clear, so that we may be clear about what constitutes a *grammar* in the relevant sense. With faithful representations in hand, and with the inadequacies of others exposed to view, we are then free to pursue our good fortune with as much focus and ambition as we wish to bring to the task.

Appendix I Grammar and Inversion

Algebraically, for those who find clarity there, the notion of 'grammar' has a simple and general interpretation.

Given a system S, each linear order $\lambda \in \text{Ord}(\text{C}\textsc{on}.\text{S})$ picks out a unique language L consisting of the optima chosen by λ. If $\hat{K} = \{K_1, K_2, \ldots\}$ is the collection of all candidate sets admitted by GEN.S, then we may write $L = \lambda[\hat{K}] = \{\lambda(K_1), \lambda(K_2), \ldots\}$, the image of $\hat{K}$ under λ.

To put it another way: there is a function ω running from Ord(CON.S) to the collection T(S) of languages of S, termed its *extensional typology* in Alber & Prince (2015, ms.), which associates a unique language with each linear order on CON.S.

(28) Function $\omega: \lambda \mapsto L$ from rankings to languages

ω: Ord(CON.S) → T(S), where for $\lambda \in$ Ord(CON.S), $\omega(\lambda) = L \in T(S)$ s.t. $\lambda[\hat{K}] = L$.

The function ω adds no new concepts or premises to the theory but merely recognizes an extant state of affairs. With ω in hand, what we have called *the grammar* of $L \in T(S)$ is then just the *inverse image* of $L \in T(S)$ under ω.[11]

(29) Definition of 'grammar of L'

Grammar of $L \in T(S) =_{df} \omega^{-1}[L]$

The function $\omega^{-1}[\ldots]$ associates each language of T(S) with the set of linear orders on CON.S that select it. Thus, the domain of $\omega^{-1}[\ldots]$ is T(S) and its codomain is the set of subsets of Ord(CON.S).[12]

(30) Domain and codomain of inverse image function $\omega^{-1}[\ldots]$

$\omega^{-1}[\ldots]: T(S) \rightarrow \mathcal{P}(\text{Ord}(\text{CON.S}))$

The definition (29) gives the general sense of 'grammar' when ω is appropriately adjusted for the theory under consideration, substituting for 'linear order on CON.S' whatever is the cognate

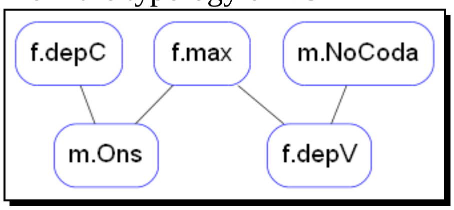
[11] The inverse image is more typically defined on *subsets* of the range of a function, rather than on individual elements, as is done here in the interests of notational simplicity. See Herstein 1964:12 for an instance of our usage. The inverse image of a function $f{:}A \rightarrow B$ is quite distinct from *the inverse function* $f^{-1}(\ldots){:}B \rightarrow A$, which will not even exist in the case of a many-one map (noninjective) or a map that does not provide a source for every element of its codomain (nonsurjective). Following a standard usage, we mark the difference typographically as a distinction between […] and (…). Other notations for the inverse image include $f^{\leftarrow}$, f*, and f "A, which invite less confusion than the familiar f^{-1}. The inverse image of L may also be termed the *preimage*, *level set*, or best of all, *fiber* of L. The more typical usage is also of linguistic interest, since $\omega^{-1}[S]$, for S some set of legs, is the smallest grammar that includes S. See Merchant & Prince 2016 for relevant discussion.

[12] Under the typical definition, which maps from sets to sets, the domain is $\mathcal{P}$(T(S)).

apparatus for selecting the well-formed. For example, in numerical optimization systems, the grammar $\omega^{-1}[L]$ consists not of one set of parameters that happen to yield L, but of every such set. Similarly, in rule-package serialism, by this definition the *grammar* contains (or denotes) every order on the rules that produces L, as indeed is commonly understood to be the case.

Appendix II Legs of (C)V.ins

(31) (C)V.ins, from the typology of BST



(32) The Legs of (C)V.ins

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

To count these without enumerating them, observe that the bottommost constraint must either be m.Ons or f.depV. Because of the symmetry of the diagram, each of these conditions must account for half of the legs. Consider the half with m.Ons at the bottom. Remove it and you have a 4 C poset to count. It takes the form of a loner (here f.depC) and 3 C's in the shape of a V. The V gives rise to two 3 C linear orders. Each of them allows for 4 spots for the loner to sit. So there are 8 legs with m.Ons at the bottom. And the same number with f.depV at the bottom. So, 16 legs in total.

To convince yourself that no stratified hierarchy exists that encompasses these legs, try to draw vertical lines separating the table into blocks of freely ordered constraints, where each block contains all instances of the constraints in the block.

Appendix III The Dotted Hassoid

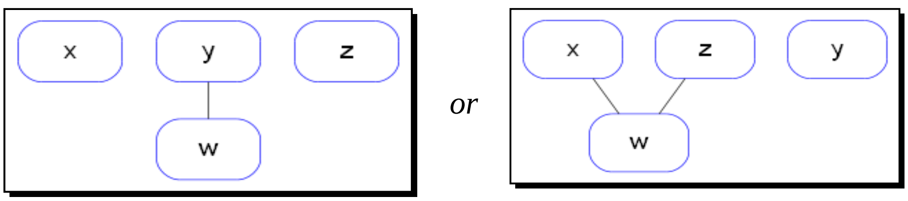
OTSoft (Hayes et al. 1999-2017) handles disjunction by using dotted lines, labeled with *or*, to indicate disjunctive dominators. As noted in Prince 2006:54-55, this notation lacks the carrying capacity to handle general OT. Consider the following grammar on 4 constraints (p.54).

(33) Test Grammar

x	y	z	w
W	W		L
	W	W	L

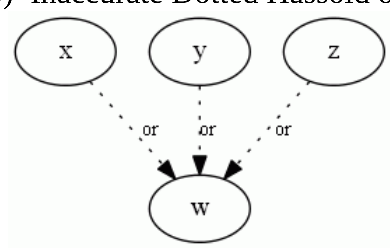
This disjunction can be given as two Hasse diagrams, which are the result of distributing out the disjunction over the conjunction that connects the two ERCs.

(34) A Hasse representation of the Test Grammar



OTSoft 2.5 produces the following Dotted Hassoid:

(35) Inaccurate Dotted Hassoid of the Test Grammar



This actually represents the 1-ERC grammar WWWL rather than the irreducible 2 ERCer (33) .

Both representations agree on the following:

- any leg in which y≫w belongs to the grammar.

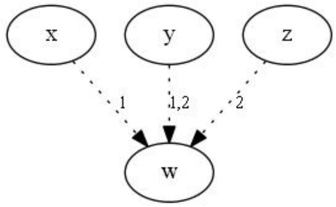
The difference is that the OTSoft version entails both of the following:

- any leg satisfying x≫w belongs to the grammar (false).
- any leg satisfying z≫w belongs to the grammar (false).

The reality is that x&z≫w is required when we don't have y≫w.

A solution would be to mark the first and second arcs of (35) with a shared index, and the second third arcs with another index. The indices would identify each pair as a separate disjunctive set, effectively tying each pair to the ERC that sponsors it. However, at this point it becomes uncertain what advantage lies in this particular form of diagrammatic representation.

(36) Dotted Hassoid with Indexed Arcs



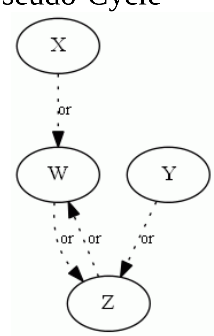
To appreciate the tangles that await, consider the following 2 ERC grammar on {X,Y,Z,W}.

(37) Pseudo-cyclic Grammar

X	Y	Z	W
W		W	L
	W	L	W

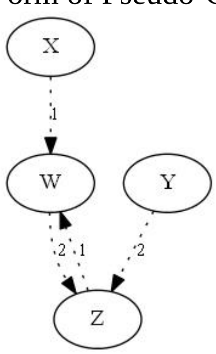
The OTSoft representation raises questions of interpretation:

(38) Dotted Pseudo-Cycle



The indexed form is coherent, if challenging, and shows that the direction of arcs must be explicitly indicated and cannot be reduced, Hasse-style, to a uniform downward flow. See Gallo et al. 1993 for discussion of directed hypergraphs.

(39) Indexed Form of Pseudo-Cyclic grammar



References

Alber, B., and A. Prince. 2015. Outline of Property Theory [Entwurf einer verallgemeinerten Eigenschaftstheorie]. Ms. University of Verona, Verona, Italy, and Rutgers University, New Brunswick, NJ.

Alber, B. and A. Prince. In prep. The structure of OT typologies. Ms. University of Verona and Typohedron.org.

Brasoveanu, A. & A. Prince. 2005/2011. Ranking and Necessity: the Fusional Reduction Algorithm. 2011: *Natural Language & Linguistic Theory* 34.29,3-70. 2005:ROA-794.

Davey, B.A., and H.A. Priestley. 1990. *Introduction to Lattices and Order*. CUP: Cambridge, New York, Melbourne.

Gallo, G., G. Longo, S. Nguyen, and S. Pallottino. 1993. Directed Hypergraphs and applications. *Discrete Applied Mathematics* 42:177-201.

Hayes, Bruce. 2009. Phonological acquisition in Optimality Theory: The early stages. In *Fixing Priorities: Constraints in Phonological Acquisition*: 158-203. Ed. René Kager. Joe Pater, and Wim Zonneveld. Cambridge: Cambridge University Press. ROA-327.

Hayes, B., B. Tesar, and K. Zuraw. 1999-2017. OTSoft 2.0-2.5. UCLA.

Herstein, I.N. 1964. *Topics in Algebra*. Blaisdell Publishing Company: Waltham, Toronto, London.

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

Joyce, J. 1922. *Ulysses*. Shakespeare & Company: Paris.

McCarthy, J. & A. Prince 1993. Generalized Alignment. In Geert Booij and Jaap van Marle, eds., *Yearbook of Morphology 1993*, 79-153. Kluwer: Boston. ROA-7.

Merchant, N. 2008. *Discovering underlying forms: contrast pairs and ranking*. Phd Dissertation, Rutgers University, New Brunswick. ROA-964.

Merchant , N. 2011. Learning ranking information from unspecified overt forms using the join. Proceedings from the Annual Meeting of the Chicago Linguistic Society 47.1: 261-275. ROA-1146.

Merchant & Riggle. 2012/2016. Erc Sets and Antimatroids. ROA 1158.
revised as: OT grammars, beyond partial orders: ERC sets and antimatroids
Natural Language & Linguistic Theory (2016), 34.1: 241-269.

Paul, Saint. ca. 53-54. First Epistle to the Corinthians.

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

Prince, A. 2002b. Arguing Optimality. ROA-562 (Revising Prince 2000, ROA-376).

Prince, A. 2006. Implication & Impossibility in Grammatical Systems: What it is & How to find it. ROA-880.

Prince, A. 2009. RCD –The Movie. ROA 1057.

Prince, A. 2016. What is OT? ROA-1271.

Prince, A., N. DelBusso, and N. Merchant. 2016. OTWorkplace Quick Start Guide. ROA-1291.

Prince, A. & P. Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. 2004:Blackwell. 1993: ROA-537.

Prince, A. and B. Tesar. 2009. Learning Phonotactic Distributions. In *Fixing Priorities: Constraints in Phonological Acquisition*: 245-291. Ed. René Kager. Joe Pater, and Wim Zonneveld. Cambridge: Cambridge University Press. ROA-353.

Prince, A., B. Tesar, and N. Merchant. 2008-17. *OTWorkplace*.

Riggle, J. 2004. *Generation, Recognition, and Learning in Finite State Optimality Theory*. PhD dissertation, UCLA.

Riggle, J., 2010. Sampling Rankings. ROA-1075.

Stevens, W. 1949. *An Ordinary Evening in New Haven*. Connecticut Academy of Arts & Sciences.

Tesar, B. and P. Smolensky. 1993. The Learnability of Optimality Theory: an Algorithm and some Basic Complexity Results. ROA-2.

Yanovich, I. 2012. On sets of OT rankings. ROA-1149.

Wikipedia.
Hypergraph
Inverse Image
Mirror