The Core of Optimality Theory

This chapter introduces the central premises of Optimality Theory. The chapter begins (§1.1) with the overall structure of OT, as proposed by Prince and Smolensky (1993). It continues with some general remarks about the nature of constraints (§1.2) and their modes of interaction through ranking (§1.3). These threads are joined to some practical suggestions for doing OT in §1.4. Readers encountering OT for the first time are advised not to read this chapter straight through; see “How to Use This Book” for a better plan of attack.

1.1 Basic Architecture

1.1.1 Candidate Comparison

Many theories of language can best be described as operational, rule based, or transformational: they take an input and apply some procedure that changes it into an output. But the primary action in OT is comparative: the actual output is the optimal member of a set of candidate output forms. Interesting analytic and theoretical results in OT come from understanding the details of how candidates are compared.

Candidates are compared by applying a hierarchy of violable constraints. The constraints assess the form of a candidate and its relationship to the input. Candidates inevitably differ in performance on various constraints. Of two candidates, the more harmonic is the one that performs better on the highest-ranking constraint that distinguishes between them. The actual output—the most harmonic or optimal candidate—is the one that is more harmonic in all its pairwise competitions with other candidates.

Because constraints are violable, the output typically disobeys at least some of the lower-ranking constraints. To draw an analogy from ethics, optimality is more like moral relativism or the Three Laws of Robotics than the Ten Commandments; it is about being the best among a choice of options, not about
Core of Optimality Theory

being objectively perfect. In the simplest situation, two candidates are under evaluation by a single constraint C. The optimal candidate is the one that incurs fewer violations of C. When there is more than one constraint, the ranking is strictly respected in comparing candidates; there is no global assessment of candidates based on their performance on the whole constraint gestalt. In fact, the optimal candidate may actually perform worse than its competitor on some constraint(s) ranked below the decisive one. So, if constraint C1 is ranked above C2 and C3 (that is, C1 dominates C2 and C3), then the output may perform worse than its competitor on both C2 and C3, as long as it performs better on C1. To cite an example from Prince and Smolensky (1993), “azzzzz” is alphabetized before “baaaaa” because alphabetical order is based on the leftmost distinguishing letter, regardless of how much the letters farther to the right might seem to encourage a different order.

This property, which Prince and Smolensky dub the strictness of strict domination, is somewhat counterintuitive, since it is quite unlike the more flexible system of priorities we apply in our everyday lives. For example, given a primary career goal of making lots of money and a secondary goal of living in an exciting city, few among us would stubbornly persist with these priorities when faced with offers of a job paying $61,000 in Paris, Texas, and a job paying $60,000 in Paris, France. Yet constraint ranking in OT has exactly that stubborn persistence. (Strict domination is the main difference between OT and connectionist models. See §2.4.)

Candidate comparison is often shown in a tableau, where an optimal candidate is compared with one or more of its competitors with respect to their performance on two or more constraints. A tableau therefore gives a perspicuous view of some of the constraints and rankings that are crucial in selecting a candidate as optimal. As in (1), constraints are given in domination order from left to right, and the rows contain the different candidates, one of which is optimal. The individual cells show the violation-marks (*) incurred by each candidate relative to each constraint. The optimal candidate is called out by the pointing hand.3

A ranking argument

<table>
<thead>
<tr>
<th>Constraint</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
</table>
| a. $\neq$ Cand\text{opt} | | *
| b. Cand\text{comp} | * | |

Readers wanting to see real tableaux now can take a look at §1.3, and in §1.4.1 I discuss the practical aspects of ranking constraints, introducing another tableau format that is particularly useful for discovering rankings.

In (1), C1 and C2 conflict in their evaluation of two candidates. C1 prefers Cand\text{opt}, but C2 prefers the competitor Cand\text{comp}. Since Cand\text{opt} is the observed output form, the conflict is resolved by ranking C1 above C2. A situation like this is a necessary condition for a valid ranking argument, a kind of proof that C1 dominates C2 in the hierarchy (written \([C1 \gg C2]\)). To ensure sufficient conditions for the validity of a ranking argument, it is also necessary to check that there is no constraint C3 with both of the following properties: C3 is ranked above C2, and C3 concurs with C1 by preferring Cand\text{opt}. In that situation, C3 invalidates the argument for \([C1 \gg C2]\) because C3 can also produce the effect of the ranking being argued for.4

Conflict is not the only possible relation between two constraints, but it is the only relation that can serve as the basis for a valid ranking argument. In the situations shown in (2a–c), there is no conflict between the constraints and so there is no basis for ranking them.

(2) a. C1 and C2 agree

<table>
<thead>
<tr>
<th>Constraint</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
</table>
| i. $\neq$ Cand\text{opt} | | *
| ii. Cand\text{comp} | * | *

b. C1 does not distinguish the candidates (both obey it)

<table>
<thead>
<tr>
<th>Constraint</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
</table>
| i. $\neq$ Cand\text{opt} | | *
| ii. Cand\text{comp} | * | *

c. C1 does not distinguish the candidates (both violate it)

<table>
<thead>
<tr>
<th>Constraint</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
</table>
| i. $\neq$ Cand\text{opt} | * | *
| ii. Cand\text{comp} | * | *

These tableaux separate the constraint columns with a dotted line to show that neither constraint provably dominates the other. These tableaux will not support a ranking argument because C1 and C2 concur in eliminating Cand\text{comp} (2a) or one of them assesses both candidates as equally good or bad (2b–c).

A constraint may assign more than one violation-mark to a candidate in one of two situations: either the constraint is violated at several different spots in the candidate under evaluation (e.g., the constraint assesses some aspect of syllable form, and a polysyllabic word contains several offenders, as in (14d)), or the constraint is violated gradiently, distinguishing noncompliant candidates by extent of violation (as is the case with edge Alignment constraints (§1.2.3)). As
OT is presently understood, multiple violations from either source are usually treated the same; they are just lumped together in the pile of violation-marks assigned to a candidate.

Candidate comparison is no different when there are multiple violations, and there is no need to count violation-marks, since better or worse performance is all that matters. Driving this point home, Prince and Smolensky introduce the method of mark cancellation. If and only if a tableau compares exactly two candidates, violation-marks that the two candidates share can be ignored or canceled, since those violation-marks contribute nothing to that particular comparison. For example, both candidates in (2c) share a violation-mark in the Cl column. These shared marks can be canceled, reducing (2c) to (2b). By reducing (2c) in this way, we can readily see that Cl contributes nothing to selecting the optimal candidate. Though this example involves single violations, mark cancellation is also useful when candidates incur multiple violations: if one candidate has three violation-marks from some constraint and another candidate has five, mark cancellation reduces this to zero and two, respectively. Comparison, rather than counting, is what matters.

Mark cancellation cannot be meaningfully applied to tableaux with more than two candidates since its purpose is to bring out the better and the worse in a pairwise comparison. With several candidates in play, it is better to use the comparative tableau format described in §1.4.1.

### 1.1.2 Ranked Constraints and EVAL

*Winning isn’t everything. It’s the only thing.*

—Attributed to Vince Lombardi

The grammar of a language is a specific constraint ranking. Language-particular ranking is the most important and perhaps only method in OT for explaining how and why languages differ from one another (§3.1.5). The ranking in a particular language is, in theory, a total ordering of a set of universal constraints.

In practice, though, it is not usually possible to discover a total ordering, and so the analyst must be satisfied with a partial ordering. There are just two legitimate ways of showing that C1 dominates C2: by a valid direct ranking argument like (1) or by a legitimate inference from valid direct ranking arguments. An example of the latter is a ranking argument based on transitivity of constraint domination, such as showing that C1 dominates C3 by establishing that C1 dominates C2 and that C2 dominates C3. When direct and inferred arguments for ranking are both present, they have to agree. Otherwise the analysis or the theory is just plain wrong. But when there is no evidence or inference available for ranking certain constraints, it is good analytic practice to report a partial order, as in (11) in §1.3.2. Partial ordering in the absence of constraint conflict is not the same thing as deliberate ties between conflicting constraints.

Tied rankings are a proposed extension of standard OT to account for within-language variation (§4.5).

Under the assumption that all constraints are universal (§1.2.1), the ranking is all that the learner must discover, and therefore some workable ranking turns to be a surprisingly easy task (§4.2.1). The analyst’s job is much harder than the learner’s: ranking arguments need to be discovered and their validity checked in a context where all hypotheses about universal constraints are necessarily tentative and mutable. Still, there are some useful heuristics to follow when positing or assessing proposed constraints (§1.4.4).

Suppose H is the constraint hierarchy for some language. To use H to select the most harmonic member of some candidate set, OT calls on the function EVAL, which gives meaning to the domination relation “>,” generalizing pairwise comparison to larger (possibly infinite) sets of candidates. The function EVAL returns the candidate set as a partial order, with its most harmonic member, the actual output form, standing at the top (§4.1.3).

In theory, there is no guarantee that EVAL will always return a single most harmonic member of the candidate set. Suppose two candidates incur identical violation-marks from all constraints. EVAL will be unable to decide between them, and if no other candidate is more harmonic, both will be optimal. In this case, within-language variation ought to be observed. In practice, though, this possibility might not be easy to realize; the universal constraint set is rich enough that EVAL usually returns a unique winner for any real-life H applied to any real-life candidate set. For this reason, within-language variation has usually been analyzed in other ways (§4.1.3, §4.5).

Although EVAL imposes a harmonic ordering on all the candidates, the standard approach assigns no interpretation to the details of the ordering below the topmost candidate. Suppose EVAL returns the harmonic ordering Cand\(_{opt} >\) Cand\(_{comp1} >\) Cand\(_{comp2}\), where > denotes the relation “is more harmonic than.” From this, we know that Cand\(_{opt}\) is the actual output form, but nothing can be concluded from the relative harmony of Cand\(_{comp}\) and Cand\(_{comp2}\) — only the optimum is given a linguistic interpretation. This is an important methodological point: valid ranking arguments like (1) must always involve an actual output form as one of the candidates being compared.

Samek-Lodovici and Prince (1999: 18) have a particularly clear and insightful way of describing EVAL. Think of a constraint as a function from sets of candidates to sets of candidates. Each constraint takes a set of candidates and returns the subset consisting of those candidates that perform best on that constraint. EVAL can then be understood in terms of function composition: a lower-ranking constraint takes as input the set of best performers on the higher-ranking constraint. For instance, if the set of candidates \{Cands\} and the hierarchy \([C1 \gg C2]\) are handed to EVAL, then the set of winners will be given by \((C2 \circ C1)(\{Cands\})\) or equivalently \(C2(C1(\{Cands\}))\). Since a constraint can never return less than one best performer, this formalization of EVAL correctly...
guarantees at least one winner. It also allows for the theoretical possibility of more than one winner when the outermost constraint returns a set containing two or more candidates. This formalization conforms rather well to the usual intuitive sense of how EVAL works: first it applies the highest-ranking (or innermost) constraint, then the next highest, and then the next, downward through the hierarchy (or outward through the composed functions) until there are no constraints left.

1.1.3 GEN

Thus far I have described two of the main components of OT, the language-particular constraint hierarchy H and the universal function EVAL, which applies H to a set of candidates. There are two others: a putatively universal set of constraints CON, discussed in §1.2, and the universal candidate generator GEN. The latter has two closely related functions: it constructs candidate output forms, such as words or sentences, and it specifies a relation between the candidate output forms and the input. Though details of the internal structure of GEN are still under development, the general principles underlying the theory of GEN are clear.\[11.5p1\]

GEN is universal, meaning that the candidate forms emitted by GEN for a given input are the same in every language. These candidates are also very diverse. This property of GEN has been called inclusivity or freedom of analysis. Precisely because GEN is universal, it must at a minimum supply candidates varied enough to fit all of the ways in which languages can differ. For example, languages disagree in how they syllabify a consonant cluster like br (cf. English alge.bra vs. Arabic jab.rī “algebraic”), so GEN will offer competing candidates that differ along this dimension, leaving the choice of the right one to the language-particular rankings in H. This freedom is limited only by primitive structural principles essential in every language, perhaps restricting GEN to a specific alphabet of distinctive features (in phonology) or to some version of X-bar theory (in syntax). Beyond this, the details of GEN are a matter for empirical investigation in the context of specific hypotheses about the nature of the input and the constraints. In phonology, there is a rough consensus about the properties of GEN (§1.1.3), but in syntax it is still more of an open question (§4.1).

Since GEN is the same in every language, it initially seems like a good place to deposit a wide variety of “hard” universals, beyond the bare structural principles just mentioned. For example, no known language syllabifies intervocalic br as *alge.bra, so why not incorporate this observation into the statement of GEN? This strategy is a natural continuation of several decades of linguistic theorizing that has sought to document various universal constraints and refine the statement of them. There is a flaw here, though. Hardwiring universals into GEN is inevitably a matter of brute-force stipulation, with no hope of explanation or connection to other matters – it is the end of discussion rather than the begin-
Recall that the ordering of operations is abstract, expressing postulated properties of the language faculty of the brain, with no temporal interpretation implied. (Chomsky 1995: 380 n. 3)

In short, a grammar is a function from some kind of input to some kind of output. A grammar is not an algorithm for computing that function nor is it a description of how speakers actually go about computing that function. Chomsky (1968: 117) sums up with "If these simple distinctions are overlooked, great confusion must result."

That confusion has sometimes led to skepticism about OT: how can EVAL sort an infinite set of candidates in finite time (cf. Bromberger and Halle 1997)? The error lies in asking how long EVAL takes to execute. It is entirely appropriate to ask whether EVAL, like Chomsky’s G, is well defined, captures linguistically significant generalizations, and so on. But questions about execution time or other aspects of (neural) computation are properly part of the performance model PM and must be addressed as such. And, not too surprisingly, there are computational models for OT that do not require infinite time to execute (see §4.3 and the references in §4.6 [11]).

1.1.4 Summary, with Possible Variations

The core universal elements of the OT architecture are summarized in (3).

(3) Basic OT architecture

\[ \text{input} \rightarrow \text{GEN} \rightarrow \text{candidates} \rightarrow \text{EVAL} \rightarrow \text{output} \]

GEN receives an input and emits a set of candidates that, in some precise way, depend upon the input. (There are also important things to say about the input itself – see §3.1.2.4.) EVAL applies the language-particular constraint hierarchy H to this candidate set, locating its most harmonic member. The most harmonic candidate is the output: it may be a phonological surface form, a syntactic S-Structure, or some other linguistic object.

The model in (3) is the simplest architecture compatible with OT’s basic assumptions. It maximally exploits OT’s capacity for global, parallel evaluation (§3.3). The output of an entire linguistic component, such as the phonology, is obtained from the input in a single pass through GEN and EVAL, which means that the candidates offered by GEN may show the effects of several notionally distinct processes simultaneously. The constraints applied by EVAL then rank these candidates for their global fitness, evaluating the effects of all of those processes in parallel. To see why it is described as global and parallel, compare this model to a theory like standard generative phonology (§2.1), where each rule applies in serial order and in isolation from all other rules coexisting in the grammar.

Some variations on this basic architecture reduce or eliminate the effects of global, parallel evaluation. Suppose that the output in (3) becomes the input for another pass through GEN, yielding a new set of candidates for evaluation. The most familiar version of this approach imposes a kind of modular or componential structure, treating the whole grammar of a language as a composite entity, as in Lexical Phonology or various instantiations of the Principles and Parameters (P&P) approach. Each module has its own distinct constraint hierarchy H, and perhaps even its own set of universal constraints CON. The output of the final module in the series is the observed surface form of the language (§3.3.3.4).

Another version of this approach is called harmonic serialism. It applies the same constraint hierarchy at each pass through GEN, continuing until there is convergence, when the output of one pass is identical to the output of the immediately preceding pass. Harmonic serialism unpacks some of the effects of globality and parallelism by imposing restrictions on GEN’s freedom of analysis. See §3.3.2.8 and §3.3.3.2 for further discussion.

Refinements or extensions like these still have the essential elements of OT: EVAL-mediated comparison of candidates by a hierarchy of violable constraints. No matter how the details are executed or in what overall context it is embedded, any model with these indispensable characteristics will express the central claim and insight of OT.

1.2 Theory of Constraints

1.2.1 The Universality of Constraints

Apart from the bare structural primitives embedded in GEN, all constraints in OT are in principle and in fact violable. This statement follows from the basic architecture of the theory: constraints have nowhere else to reside except in the language-particular hierarchy H, which means that any constraint could, in some language, be ranked below another constraint that compels it to be violated.

The null hypothesis is that all constraints are universal and universally present in the grammars of all languages (Prince and Smolensky 1993), and so UG incorporates a constraint component CON. What makes this the null hypothesis is a kind of Occamite reasoning: since language-particular ranking is in general able to account for languages where a putatively universal constraint does not hold true, it does not seem necessary to recognize a special class of language-particular constraints. (See §1.2.3 for some possible qualifications.) Differences between languages are no barrier to constraint universality when constraints are violable.

Constraint violability is a very different thing from parametrization. A parameter describes a requirement that is either reliably enforced or completely ignored: syllables must have onsets (yes/no); heads must precede/follow their complements. A constraint, no matter where it is ranked, always asserts its preference: Onset is violated by any syllable that lacks an onset in any language,
tou court. Whether it visibly asserts that preference depends on details of the
typical-particular ranking and the candidates under evaluation.

Suppose we say that a constraint is active if and only if it is the highest-ranking constraint that distinguishes some losing candidate from the winner. What we are talking about, then, is visible activity: every constraint, no matter where it is ranked, evaluates every candidate, but not every constraint will be visibly active. Whether a constraint is visibly active depends on the constraints that dominate it and the candidates it evaluates. Even within a language, when different candidate sets from different inputs are considered, a constraint might be active sometimes and inactive otherwise. This middle ground of partial activity follows from the interactional nature of OT (§1.3, §3.1, §3.2), but it is difficult or impossible to achieve in parametric models. (See the FAQ about parameters for a list of places in the text where this important difference between ranking and parameters is discussed.)

Universal constraints and language-particular ranking yield a factorial typology, another key notion from Prince and Smolensky (1993). Every permutation of the constraints in CON is predicted to be a possible human language, and the grammar of every observed human language must be one of those permutations. There are, however, some minor qualifications. There is no guarantee that every permutation will yield an observably distinct human language. For example, if two constraints in CON happen never to conflict on any candidate, then switching their ranking will have no effect. CON may also include universally fixed constraint hierarchies related to natural linguistic scales (§1.2.3). These fixed hierarchies limit the typological consequences of ranking permutation – in fact, they are themselves supported by typological considerations. In deference to this, the term “permutation” is implicitly qualified by “licit” throughout this book.

Here is an analogy to help clarify the notion of a factorial typology. Imagine a mode of psychotherapy based on the hypothesis that each type of human personality reflects a different prioritization of four universal desires (such as love, wealth, progeny, and power). Since these desires are universal and there are 4! = 24 different ways to rank them, there will be 24 distinct personality types. The goal of this psychotherapeutic modality is to determine how the analysand fits into this factorial typology. The “ranking arguments” consist of simple scenarios that involve clear choices between maximizing one desire or another – for instance, would you consider running for mayor if it meant giving up a better-paying but much less powerful job as a piano tuner, supposing that the choice had no effect either way on love or progeny?

Factorial typology makes a strong claim with important implications. It means, as a matter of simple methodological competence, that analysts must test every proposed constraint for its typological consequences under ranking permutation, and no phenomenon can be definitively analyzed in a particular language without considering cross-linguistic variation. OT’s inherently typological character thus places severe conditions on the adequacy of proposed analyses.

1.2 Theory of Constraints

1.2.2 Constraint Typology

Two basic types of constraints are distinguished in OT, faithfulness and markedness. Faithfulness constraints require identity between the input and the output candidate under evaluation, using the record of input/output disparity supplied by GEN (§1.1.3). Markedness constraints evaluate the form of the output candidate, favoring certain structural configurations (e.g., syllables with onsets, accusative objects) over others (e.g., syllables without onsets, dative objects). Constraints of both types are undoubtedly necessary. Without faithfulness constraints, all distinctions made by input forms would be reduced to some least-marked output (see the FAQ about unmarked form and ba). And without markedness constraints, there would be no way to account for languages differing systematically in the structures they permit (their inventories – §3.1.2). Interaction between faithfulness and markedness constraints is a key element of any OT analysis (§1.3).

In the earliest work on OT, markedness and faithfulness constraints were formally rather similar though notionally distinct. Faithfulness constraints were made to resemble markedness constraints by strictly limiting the kinds of mappings that GEN could perform. As in trace theory (Chomsky 1973) and some versions of autosegmental phonology (e.g., Selkirk 1981; Steriade 1982), surface forms were enriched to include covert structural indications of the unfaithful mappings that produced them. Phonological epenthesis involved a kind of overparsing: surface forms contained present-but-incomplete syllabic structures, as in Spanish /skwela/ → [Askwela] for escuela ‘school’. Phonological deletion involved underparsing: surface forms contained segments that were present but not syllabified, as in English /bamb/ → [bmb] bomb (cf. bombard, bombardier). These assumptions about GEN allowed the faithfulness constraints, like the markedness constraints, to evaluate surface structures alone. The faithfulness constraint FILL militated against empty segments like the [ŋ] in [Askwela], and its counterpart PARSE was violated by unsyllabified segments like the final [ŋ] in bomb. (See §3.3.3.5 for further developments along these lines.)

These simplifying assumptions about faithfulness are obviously not necessary elements of OT, and when it proved difficult to extend the early PARSE/FILL model to the full range of phonological generalizations, alternatives were sought. The correspondence theory of faithfulness posits a correspondence relation \( R \) from the input to each of its output candidates. For example, in the mapping /bati/ → bati, the candidate bati includes the information that b, a, and t correspond to segments of the input, but i does not. This is a violation of the constraint DEP, which says that \( R \) must be surjective (onto), so every element of the output stands in correspondence with the input. Analogously, MAX militates against deletion, requiring that the inverse relation \( R^{-1} \) be surjective, so every element of the input is in correspondence with the output. (The names of these constraints allude mnemonically to their functions: the output
constraints of correspondence theory prohibit one-to-many mappings, many-to-one mappings, and various other imaginable derangements of perfect identity between input and output. (On further extensions of faithfulness, see §3.2.1.2, and §3.3.3.5, and the references in §1.5 §3.)

Correspondence theory provides a general framework for stating constraints that demand faithfulness to linguistic objects. A candidate is unfaithful whenever its associated correspondence relation describes anything other than an order- and structure-preserving mapping that is one-to-one and onto. Research continues on the details of what the faithfulness constraints are, but the general outlines of the theory are fairly clear – in phonology at least.

In syntax, there is as yet no consensus on the form of faithfulness constraints: they might prohibit movement and other syntactic operations or require accurate surface spell-out of underlying distinctions (such as morphosyntactic features). A prohibition on movement is, of course, reminiscent of the Economy principles of the Minimalist Program. For example, any metric that prefers shorter derivations (as in Chomsky 1995: 138ff.) will roughly approximate the effects of faithfulness constraints. There are differences though: this Economy principle evaluates derivations, while faithfulness evaluates input → output mappings; faithfulness constraints are typically a good deal more specific than most proposed Economy principles; and Economy, unlike faithfulness, is seen as having a functional basis in minimization of effort. See §4.1 for more about faithfulness in syntax, §3.2.3 for some comparison of OT with Economy principles, and §4.4.2 for a proposed relation between minimization of effort and markedness constraints in OT.

Markedness constraints evaluate output structures. Like the phrase “faithfulness constraint,” the phrase “markedness constraint” is a term of art in OT: it refers to any constraint that assigns violation-marks to a candidate based solely on its output structure, without regard to its similarity to the input. A candidate is marked by or with respect to that constraint if it receives at least one violation-mark from it. For example, ONSET and SUBJECT are two markedness constraints that have been proposed in the OT literature (§1.3.1, §3.1.4.8). ONSET assigns a candidate one violation-mark for each vowel-initial syllable that it contains, demanding instead that syllables begin with a consonant (called the onset); SUBJECT assigns a candidate one violation-mark for each Spec-less IP (= subjectless sentence) that it contains. These are typical markedness constraints.

When they first encounter OT, many people share certain concerns about constraints. Most of these concerns have a common source: the projection of ideas about familiar constraints in other linguistic theories onto OT. Since faithfulness constraints are unique to OT, they do not bring this baggage with them; the problem mostly involves markedness constraints. Here are some possible misunderstandings and clarifications of the differences between OT constraints and the constraints of other theories.

The technical sense of markedness, as used in OT, is distinct from and a good deal more specific than the more familiar usage of this word in linguistics, dating back to the Prague School of the 1930s: “The concept of markedness in its most general characterization is concerned with the distinction between what is neutral, natural, or most expected (unmarked), and what departs from the neutral (marked) along some designated parameter” (Kean 1992: 390). A markedness constraint in OT may produce results related to this descriptive or typological sense of markedness (§3.1), but the formal constraint and the typological observation are two different things.

This terminological ambiguity can be a source of considerable confusion. I once received the following advice from an anonymous referee for a prominent journal:

My first comment addresses the discussion of segmental markedness: the primary evidence for markedness is implicational statements of the form: “If language L has structure A, it also has structure B”. . . . In the absence of such implicational relations between A and B, there is no consensus on what should count as marked and why.

This reviewer is assuming that OT markedness is exactly the same thing as Praguian markedness, leading to confusion of OT as a theory with a Prague-inspired methodology. In OT, because constraints are violable and one markedness constraint can conflict with another (see the next two paragraphs), an observed implicational relation “A only if B” is a sufficient but not a necessary condition for positing a markedness constraint that A violates and B does not (see §3.1.1 and §3.1.5.4). Implicational relations, then, are not the “primary evidence” for markedness constraints; they are just one clue.10 The real primary evidence for markedness constraints is the correctness of the typologies they predict under permuted ranking of the constraints in CON.

Mixing up these two different senses of markedness is also the source of another objection to OT: how is it possible for two markedness constraints to conflict with one another? The idea of markedness/faithfulness conflict is intuitively clear, but conflict among markedness constraints does not make sense from the Prague School perspective. Prague markedness is married to implicational relations like “A only if B,” so it is inherently unidimensional and non-conflicting: A is more marked than B in all languages under all circumstances. In OT, though, markedness is multidimensional – different constraints favor or disfavor different properties, and it would be astonishing if there were no conflicts. So, while CON may supply a constraint that A violates and B obeys, this itself does not entail the implicational relation “A only if B,” since there may be another markedness constraint in CON favoring B over A – perhaps under other conditions or even under exactly the same conditions. For examples of conflicting markedness constraints, see (13) and Chapter 3 passim.11 For “opposite” constraints, see §3.2.1.3.
At the other extreme is the occasionally voiced a priori insistence that every constraint in OT should conflict with every other constraint. This is of course not true. The source of this idea is harder to figure out, but it may stem from an assumption that OT constraints are really an elaborated system of parameters. There are basic differences between parameters and OT constraints and between parameter setting and constraint ranking. See the parameters FAQ for brief treatment of these differences and exhaustive references to discussion elsewhere in this book.

The word “constraint” itself is another source of terminological ambiguity. The constraints of more familiar theories are inviolable, whereas OT constraints can be violated under duress. It is tempting to import the inviolable constraints of other theories into Con, but this temptation should be resisted. The inviolable constraints of other theories are intended to state universals of human language; the violable constraints of OT do not state universals of human language, precisely because they are violable. Rather, OT requires that universals be derived from constraint interaction (§3.1.5).

There is another, more specific problem with importing constraints from other theories, where they are often surrounded by an apparatus of codicils necessary simply to assure inviolability. It is also common to find constraints in other theories that explicitly refer to other constraints in ways that mimic constraint domination in OT. Here are some examples (emphasis added throughout):

- In phonology, the Obligatory Contour Principle prohibits adjacent identical elements except across morpheme boundaries (McCarthy 1986).
- Hayes’s (1995: 95) Priority Clause involves implicit comparison of alternative outputs and explicit reference to another constraint prohibiting degenerate (e.g., monosyllabic) feet: “If at any stage in foot parsing the portion of the string being scanned would yield a degenerate foot, the parse scans further along the string to construct a proper foot where possible.”
- Halle and Vergnaud’s (1987: 10, 15) theory of metrical parsing implements several interdependent constraints. The Exhaustivity Condition says that parsing is exhaustive “subject to” a Recoverability Condition. And the Maximal Condition says that parsing constructs constituents that are as large as possible, “provided that other requirements on constituent structure are satisfied.”
- In syntax, “[m]ovement must be done after SPELL-OUT whenever it is possible to converge by doing so” (Poole 1998: 385 after Chomsky 1995: 398).
- From Roberts (1997: 426):
  a. Head movement is copying.
  b. *(x W1 W2), where Wa are morphological words.
  c. A head is spelled out in the highest position of its chain, subject to (b).

These hedges are descriptive necessities when constraints are inviolable and when there is no general theory of constraint interaction, but in OT they ought to follow from principled interaction of simple, violable constraints (§1.3). Constraints with hedges or codicils are not ready-made for importation into OT; they are research problems.

A final remark. First exposure to OT sometimes leads to insistence on a shortcut: “Just tell me what the constraints are.” This request is unreasonable. OT is a general framework for constraint interaction, and as such it does not entail a particular set of constraints in Con. Indeed, if OT is the right framework, and if all the constraints in Con were somehow known, then the profession of linguistics would be at an end. The constraints will be discovered gradually by time-honored methods of analysis, theorizing, further analysis, improved theorizing, and so on (see §1.4.4 for some research strategies). Constraints are specific empirical hypotheses about Universal Grammar (UG), and so it is inappropriate to demand a full accounting of them in advance of empirical research.

### 1.2 Theory of Constraints

Certain ideas about the form of constraints have proven useful in both phonology and syntax. This section gives a brief overview of three of these ideas, leaving more detailed explanation and exemplification for Chapter 3. If this material seems unfamiliar, it might be better to skip it for now and return to it when alerted in Chapter 3.

There is considerable internal structure to Con, making it much more than a mere deuteronomic list of what is forbidden and what is required. One source of structure is the constraint schema, an abstract formula for constructing all constraints of a certain type. The constraint schema that has been most extensively studied is edge Alignment, which supplies a template for constraints that refer to the edges of constituents. Local constraint conjunction is another, more controversial source of internal structure to Con. Local conjunction is a way of combining two constraints to get the force of both simultaneously. Harmonic alignment goes from substantive universal scales, like sonority or animacy, to universally fixed constraint rankings. It is the basis of many implicational universals. These three ideas are discussed in turn.

Edge Alignment constraints are subsumed under the schema shown in 

\[
\text{Catenarian constraint} \quad \text{[1.7.4]}
\]

\[(4) \quad \text{ALIGN(Cat}_1, \text{ Cat}_2, \text{ Edge)}
\]

The element standing at the Edge of any Cat also stands at the Edge of some Cat (where Cat and Cat are grammatical or prosodic constituents and Edge is left or right).

Alignment constraints demand that constituent-edges coincide. They quantify universally over their first argument and existentially over their second. For example, the constraint ALIGN (Foot, Word, Right) (often abbreviated ALIGN-R(Ft, Wd)) says that every stress foot is final in some word, while
ALIGN-L(Accusative, S) says that every instance of the morphosyntactic feature accusative must be realized initially in some clause. For examples and applications, see §3.1.2.4 (13), §3.1.4.6 (37), §3.1.5.3, §3.2.1.2 (59) and (61), §3.2.1.3, §3.2.3, §3.3.2.5, and §3.3.3.3.

Alignment constraints are usually construed gradually. Suppose some high-ranking constraint rules out the perfectly aligned candidate, and two candidates, neither of which is perfectly aligned, remain. The Alignment constraint favors the one that is closer to perfect alignment than the other. In such situations, it is of course necessary to be precise about how the extent of violation is to be translated into violation-marks. But no literal counting of violations is required, since it is enough for EVAL to distinguish better from worse performance (§1.1.2).\(^\text{14}\)

Local constraint conjunction is another source of internal structure in Con, originally proposed by Smolensky (1995b). The local conjunction of constraints C1 and C2 in domain D, written [C1 & C2]D, is violated if an only if both C1 and C2 are violated by the same instance of D. Suppose C1 and C2 are markedness constraints, each expressing some simple prohibition. The intuition behind local conjunction is to combine C1 and C2 to express some more complex prohibition, singling out “the worst of the worst” for special attention. For example, if first person objects are marked and null exponent of a morphological distinction is marked, then a fortiori null exponent of a first person object is marked (see §3.1.2.5, §3.1.5.4, and Aissen 1999). This basic idea has been extended to include the rather different notion of local self-conjunction, defining [C1 & C1]D to prohibit two distinct instances of C1 violation in D. This is one possible approach to dissimilation and similar processes.\(^\text{15}\)

Local conjunction is a powerful idea, but in the long run we also will need limits on which constraints can be combined in this way. The possibility of conjoining constraints somewhat mitigates the effects of the strictness of strict domination (§1.1). Suppose the ranking [C1 & C2 & C3]D has previously been established. Ranked above C1, the conjoined constraint [C2 & C3]D would allow the two otherwise low-ranking constraints to collude against the high-ranking one, approximately as in a connectionist model (§2.4). Nevertheless, there are important differences between local conjunction and numerical weighting: conjunction is categorical in its effects, locality is enforced by the domain argument D, and there is some potential for placing limitations on what constraints can be conjoined and in what domains (see the next paragraph). For further discussion and exemplification, see §3.1.2.5, §3.3.2.8 (93), §3.3.3.5 (103), and §4.4.2.

Because the Alignment schema and local conjunction are general techniques for constraint construction, they inevitably raise questions about the universal versus the language-particular.\(^\text{16}\) On the formal/substantive side, the main question is whether the Alignment and local conjunction schemata are enough, or whether there are also substantive limitations. Concretely, does UG contain constraints aligning either edge of every constituent with every other constituent, and does it contain the local conjunction of every pairing of simple constraints on every possible domain? Or are there substantive limitations on these schemata, so that UG contains only certain natural Alignment constraints or constraint conjunctions? It seems likely that there are indeed substantive limitations and hence that the space of imaginable Alignment or conjoined constraints is rather sparsely populated. Research on the interface between prosodic constituency and syntactic or morphological constituency has turned up certain specific natural pairings — for instance, the edges of phonological words are naturally aligned with the edges of lexical roots — and some bias toward a particular edge as well. Relatedly, there is evidence that Alignment constraints may be relativized to particular affixes, to allow prefix – infix – suffix alternations to be obtained through constraint interaction (§3.1.5.3, §3.2.2.1). Here again, the schema is narrowly limited (affixes are aligned at the left or right periphery of the stem), though its argument — the affected morpheme(s) — is obviously free. Substantive or formal limitations on constraint conjunction have also received some attention, though research is less far along.\(^\text{81},\text{93}\)

The issue of universal versus language-particular applications of constraint schemata has, as yet, not received as much attention (though see the references in §4.6 §14). Because they provide ways of constructing constraints from simpler elements, the Alignment and local conjunction schemata might seem to hark back to the rule-writing theories of early generative grammar (Chapter 2). Could learners, armed only with the schema (4), use their early experience to discover all of the Alignment constraints operative in their native language? Do learners have an innate but modest set of simple constraints, with part of learning being devoted to the discovery of how the universal constraints are conjoined in their native language?

The answer to both these questions could, in principle, be “no.” The schemata could play a purely passive role, giving structure to universal Con without being involved in learning. Since universal constraints must be supported by factorial typology, it might seem that there is an easy strategy for finding language-particular constraints and thereby falsifying the null hypothesis: check every dubious constraint for its typological consequences under ranking permutation. This strategy is not quite complete, though. A constraint for which there is no typological support cannot be universal, but this does not mean a language-particular constraint is required instead. Another possibility is to look more closely at the interaction of known constraints as the source of the phenomena that seem to motivate the dubious constraint. Because ranking is language particular and because interaction comes from ranking, interaction is by far the most common source of language-particular patterns and must always be considered as an alternative to a language-particular constraint (or a universal one, for that matter). Of course, interactional solutions will not always be self-evident.

The last organizing principle for Con to be discussed here bears on the analysis of multi-tiered implicational universals. These are observed universal
patterns of the form "... A only if B only if C ..." where some kind of scale of relative markedness is involved. Two techniques have been developed for analyzing implicational universals in OT; both involve imparting some internal structure to CON.

One idea is to define two constraints standing in a stringency relation, as in (5): if the violations of C1 are always a proper subset of the violations of C2, then C2 imposes a more stringent test than C1 does.\[^{17}\]

(5) A stringency relation

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrucA</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>StrucB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrucC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The constraints in (5) give the harmonic ordering \([\text{Struc}_A > \text{Struc}_B > \text{Struc}_C]\). That ordering holds regardless of how C1 and C2 are ranked with respect to one another. Through interaction with faithfulness constraints (§3.1.5.4), C1 and C2 define a system of implicational universals: any language that includes StrucC in its inventory of output forms must also include StrucB, but not vice versa; and any language that includes StrucB in its inventory must also include StrucA, but not vice versa — as long as no other constraints in CON favor StrucB over StrucA or StrucA over StrucB or StrucC (§1.2.2). Typical applications of the stringency idea involve a contextually restricted constraint as C1 and its context-free counterpart as C2. Some examples: be faithful to lexical forms versus be faithful to functional forms, a constraint that operates on all forms, lexical or functional (§3.1.4.3); or, nasals are prohibited before voiceless consonants versus nasals are prohibited everywhere (cf. §3.1.4.2).

Another approach to multi-tiered implicational universals involves universally fixed constraint rankings. Though the discussion so far has rightly emphasized the permutability of constraints, there are certain situations where a fixed universal hierarchy, as in (6), can prove useful.

(6) A fixed universal hierarchy

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrucA</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>StrucB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrucC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Like (5), (6) yields the harmonic ordering \([\text{Struc}_A > \text{Struc}_B > \text{Struc}_C]\) and thus would account for the same implicational universal under the same assumptions about the rest of CON. Observe that if the ranking were (wrongly) permutable, the same results would not be obtained, since if C2 dominates C1, the harmonic ordering is \([\text{Struc}_A > \text{Struc}_B > \text{Struc}_C]\).\[^{17}\]

Allowing free stipulation of fixed universal hierarchies involving arbitrary sets of constraints would greatly limit the interest and attractiveness of factorial typology. But there is some reason to think that all such hierarchies are derived by harmonic alignment of prominence scales.\[^{11.5}\] (Despite the terminological overlap, harmonic alignment and edge Alignment have nothing to do with one another.) Language is replete with natural scales, with one end more prominent, in an abstract sense, than the other: on the sonority scale, vowels are more prominent than liquids, which are more prominent than nasals, and so on; persons are numbered first, second, and third, in order of prominence; subject is more prominent than object, accusative more prominent than dative, and so on. Prominence scales are inferred orderings of linguistic objects; they are not the same thing as constraint hierarchies. Prominence scales, though, can be combined by harmonic alignment to form constraint hierarchies.

Harmonic alignment is defined as in (7) (after Prince and Smolensky 1993: 136).

(7) Harmonic alignment

Given a binary dimension D1 with a scale X > Y and another dimension D2 with a scale a > b > ... > z, the harmonic alignment of D1 and D2 is the following pair of harmony scales:

- \(H_x = X/a > X/b > ... > X/z\)
- \(H_y = Y/z > ... > Y/b > Y/a\)

The constraint alignment is the following pair of constraint hierarchies:

- \(C_x = *X/z > ... > *X/b > *X/a\)
- \(C_y = *Y/a > *Y/b > ... > *Y/z\)

The notation \(X/d\) describes a linguistic element that combines the properties d and X, such as a d that occurs in position or context X. The notation \(*X/d\) denotes a constraint that \(X/d\) violates.

Three different relations are symbolized in (7). As usual, \("\gg\)" means "dominates." It is a relation between constraints. The other two relations, \("\gg\)" and \("\gg\)," are relations between linguistic objects. The first, \("\gg\)," means "is more prominent than" on some natural linguistic scale. The relation \("\gg\) means "is more harmonic than" on a harmony scale derived by aligning two natural linguistic scales.

The idea in (7) is that two natural linguistic scales, one of which is binary, can be combined to form two harmony scales by aligning their most and least prominent elements. In one harmony scale, \(H_{x1}\), the more prominent member of the binary scale \(D_1\) is mapped onto the other scale \(D_2\) in \(D_2\)'s order. Hence, a prominent element on one scale combines most felicitously with a prominent element on the other scale, and so on down the line. The prominent X combines least felicitously with z, which is least prominent on \(D_2\). Conversely, in \(H_{y1}\), the
more prominent member of the binary scale $D_1$ is mapped onto $D_2$ in the opposite
of $D_2$'s order. The least prominent element on one scale combines most
felicitously with the least prominent element on the other scale, and so on up
the line. The constraint alignment, derived from the harmony scales, consists of
two fixed universal hierarchies. They are, in effect, the contrapositive of the
harmony scales. If, as in $H_x$, $X/a > X/b$, then, as in $C_x$, $X/a > X/b > X/a$. In other words, for $a$ to be a better instance of $X$ than $b$ is, the
constraint against $X/b$ must dominate the constraint against $X/a$.

Harmonic alignment of prominence scales establishes a preferred correla-
tion between two distinct but related dimensions; it is something like the height-
weight tables in diet books. For example, the syllable-position prominence scale
$[\text{Nucleus} > \text{Onset}]$ can be combined with the sonority scale $[\text{vowel} > \text{liquid} > \text{nasal} > \text{fricative} > \text{stop}]$ to form two harmony scales, $[\text{Nucleus/Onset} > \text{Nucleus/liquid} > \ldots]$ and $[\text{Onset/stop} > \text{Onset/fricative} > \ldots]$. By quasicontra-
position, these harmony scales are transformed into universally fixed constraint
hierarchies, $[\ldots > *\text{Nucleus/Liquid} > *\text{Nucleus/Vowel}]$ and $[\ldots > *\text{Onset/Fricative} > *\text{Onset/Stop}]$. Both hierarchies have important empirical
consequences: the nucleus hierarchy accounts for the implicational univer-
sal that some languages have only vowel nuclei (Italian) and some have both
liquid and vowel nuclei (English bottle), but no language has only liquid nuclei;
the effect of the onset hierarchy can be observed in early acquisition, when many
children avoid nasal or liquid onsets ($\S 4.2.2$). For further discussion and ex-
emplification, see §3.1.5.4.19

1.3 Constraint Interaction

Constraint interaction through ranking is the basis of description and explana-
tion in OT. The interaction of simple constraints can produce patterns of sur-
prising complexity. Permuting the ranking yields an array of typological
predictions. The discussion in this section, based on Prince and Smolensky
(1993: Chapter 3-4), gives a summary of the main kinds of interaction.18
The material in Chapter 3 shows how these simple interactions fit into the larger
picture.

1.3.1 Faithful and Unfaithful Mappings

In theory, the interaction of a few constraints should always be studied in the
broader context of the full constraint hierarchy. In practice, and in the current
expositional context, it is useful to look at a small number of constraints in iso-
lation. We begin with a constraint set that consists of one markedness constraint $M$ and one faithfulness constraint $F$. (Here and throughout this section, I assume
that the constraints under discussion can interact – that is, that they deal with
sufficiently similar matters as to make interaction possible.) Two rankings are
possible: if $F$ dominates $M$, then nothing happens, because violations of $M$ are

In (8a), there is no real competition for the faithful candidate. Since $pa.ta$
violates no constraints, it $\text{harmonically bounds}$ any candidate like $a.pa.ta$ or
$T.a.pa.ta$ that incurs violations of $\text{DEP}$ and/or $\text{Onset}$. Harmonic bounding is
defined as in (9) ($\S 3.1.5.3$).

(9) Harmonic bounding of $\text{DEP}$

The mapping $/A/ \rightarrow B$ harmonically bounds the mapping $/A/ \rightarrow C$ if and
only if the $/A/ \rightarrow B$ mapping incurs a proper subset of the constraint viola-
tions incurred by the $/A/ \rightarrow C$ mapping. (In other words, no constraint assigns
more violation-marks to the $/A/ \rightarrow B$ mapping than to the $/A/ \rightarrow C$ mapping,
and at least one constraint assigns more violation-marks to the $/A/ \rightarrow C$
mapping.)

In tableau (8a), given just the constraints and candidates shown, no language
can map $/pata/$ onto anything except $pa.ta$, since $pa.ta$ is fully faithful and has
no markedness violations, harmonically bounding its competitors. This situa-
tion evidences a kind of economy of derivation that follows from the definition
of $\text{Eval}$ in OT ($\S 3.1.5.2$, §3.2.3): the "economy" constraint $\text{DEP}$ is violable, but
violation is never gratuitous; it must always be compelled.

In tableau (8b), however, there is an interesting competing candidate – inter-
esting precisely because it is not harmonically bounded. In $a.pa.ta$, faithful
analysis leads to Onset violation, while in ?apa.ta, unfaithful analysis yields a candidate that obeys Onset. This sort of competition forms the basis of a valid ranking argument (§1.1.1) showing that Dep dominates Onset. With input /apata/ and with only the candidates shown, either Dep or Onset must be violated by the output form. The ranking [Dep > Onset] ensures that Dep is obeyed and Onset violated precisely in situations where obedience to both is impossible. 20

In a different language, with the ranking [Onset > Dep], input /apata/ is still faithfully analyzed as pata because pata harmonically bounds all its competitors. (By definition, harmonic bounding depends on what constraints are in Con but not on how they are ranked.) But the treatment of input /apata/ is different, as evidenced in (10).

(10) Different ranking: Input /apata/ → Output ?apata

<table>
<thead>
<tr>
<th>/apata/</th>
<th>Dep</th>
<th>Onset</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. ?apata</td>
<td>*</td>
<td>*</td>
<td>Epenthesis</td>
</tr>
<tr>
<td>ii. apata</td>
<td>*</td>
<td>*</td>
<td>Faithful</td>
</tr>
</tbody>
</table>

Because the markedness constraint Onset is top ranked, violation of the faithfulness constraint Dep is compelled. The result is an unfaithful mapping.

An [M > F] ranking like [Onset > Dep] is how OT approximates the effects of processes, rules, transformations, or operations in other linguistic theories (see §3.1.1 for the details). When faithful analysis of some input would violate M, a candidate that obeys M by violating F is chosen instead. In fact, there is no other reason why F would be violated: an unfaithful mapping is never possible unless it achieves improved performance on the markedness constraints of UG as they are ranked in some language-particular hierarchy (see Moreton 1996/1999 and §3.1.4.5). For this reason, OT can be described as a teleological or output-oriented theory. The motivation for a process is always to be found in the output configuration that it achieves or avoids.

Even when F is crucially dominated, violation of F is minimal because of the way that Eval selects the most harmonic candidate. If M dominates F, no candidate that violates F more than is necessary to obey M will ever be optimal. For this reason, mappings like /pat/ → ?apata or /apata/ → ?at?apa are sure losers if Con consists of just these two constraints. In short, Eval entails that mappings are unfaithful only when necessary and only to the extent necessary (another economy effect).

Sometimes, as in the previous paragraphs, the workings of OT constraints are described in teleological or functional language: “epenthesis is triggered by the need to satisfy Onset” or “the faithfulness constraint is violated no more than necessary to obey the markedness constraint.” Do not be misled by these external descriptions of the work that Eval accomplishes. Despite the expository usefulness of these paraphrases, OT has no literal triggering or blocking of constraints or processes, nor is there any directed progress toward the goal of satisfying some constraint. All Eval does is apply a constraint hierarchy to a set of candidates, proceeding in the same unintelligent, localistic way as function composition (§1.1.2). Triggering, blocking, and overall teleology are ways to understand the effects that Eval produces, but they are not overt or covert properties of Eval itself.

1.3.2 Homogeneity of Target/Heterogeneity of Process

The next level of interactional complexity involves a markedness constraint M in a hierarchy with two faithfulness constraints, F1 and F2. If both F1 and F2 dominate M, then all mappings are faithful, as in (8). But if at least one of F1 or F2 is ranked below M, then M compels violation of the lower-ranked one. Concretely, suppose M and F1 are Onset and Dep, as before, and F2 is Max, which prohibits deletion. Four permuted rankings, shown in (11), have Onset dominating at least one of the faithfulness constraints.

(11) Permuted rankings where Onset dominates Dep and/or Max

a. Onset > Dep > Max
   Dep > Onset > Max
   Onset, Dep > Max
   Onset, Max > Dep

The rankings in (11a) with Max at the bottom favor the mapping /apata/ → pata, as shown in (12).

(12) a. Input /apata/ → Output pata

<table>
<thead>
<tr>
<th>/apata/</th>
<th>Onset</th>
<th>Dep</th>
<th>Max</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. ?pata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Deletion</td>
</tr>
<tr>
<td>ii. apata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Faithful</td>
</tr>
<tr>
<td>iii. ?apata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Epenthesis</td>
</tr>
</tbody>
</table>

b. Input /apata/ → Output pata

<table>
<thead>
<tr>
<th>/apata/</th>
<th>Onset</th>
<th>Dep</th>
<th>Max</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. ?pata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Deletion</td>
</tr>
<tr>
<td>ii. apata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Faithful</td>
</tr>
<tr>
<td>iii. ?apata</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Epenthesis</td>
</tr>
</tbody>
</table>
The output is the same in (12a) and (12b), showing that the ranking of ONSET with respect to DEP is irrelevant to determining the outcome. (It is good analytic practice in such situations to report only those rankings for which there is evidence, using the format on the right in (11).) So, with the rankings [ONSET » MAX] and [DEP » MAX], inputs like /apata/ map to unfaithful candidates with deletion of the offending initial vowel. But if DEP is at the bottom of the hierarchy, as in (11b), then /apata/ maps to /apatə.

The partial factorial typology in (11) exemplifies OT's explanation for homogeneity of target/heterogeneity of process (§2.1, §3.1.4.2). There are many cases in phonology where the same target is reached in different ways in different languages. (The term "target," in the sense I am employing, is a loose metaphor for required or prohibited output configurations.) For example, certain consonant clusters are avoided by deletion in Diola Fogny (Niger-Congo, Senegal) but by vowel epenthesis in Ponapean (Austronesian, Micronesia).21 In OT, the markedness/faithfulness dichotomy, combined with factorial typology, predicts exactly this cross-linguistic variation. In contrast, rule-based theories typically do not (§2.1).

Homogeneity of target/heterogeneity of process is not a very familiar problem in syntactic typology, but the Romance clitics provide a striking example.22 The target is avoidance of clitic duplication. Three dispositions are observed: in some dialects of Spanish, one of the offending clitics is deleted (Se lava for *Se se lava 'one washes oneself'); in standard Italian, one of the clitics is changed to a non-duplicative form (Ci si lava for *Si si lava); and in the Conegliano Italian dialect, duplication is simply tolerated (Si si lava). This typology is typical of the effects of ranking permutation, where a single markedness constraint can be satisfied in diverse ways or even not at all, depending on its ranking with respect to faithfulness and other markedness constraints.

1.3.3 Blocking Effects

Because constraints are violable, a language may allow outputs that violate some markedness constraint M. Nevertheless, M may also be active in that same language under other conditions (§3.1, §3.2.2). For example, the basic [M » F] ranking can be modified by deploying a third constraint, C, that dominates M and sometimes compels violation of it. In this situation, C can be said to "block" the process characterized by the [M » F] ranking. There are two ways for this to happen, depending on whether C is another markedness constraint or another faithfulness constraint.

Suppose C is a markedness constraint, C_M. To produce a blocking pattern where M is only partially active, C_M must be violated by some, but not all, of the otherwise favored M-obeying candidates. Dutch supplies a concrete example.23 This language has the ranking [ONSET, MAX » DEP], with glottal-stop epenthesis to relieve onsetsless syllables after a: pa(ʔ)ella ‘paella’; a(ʔ)órta ‘aorta’. But there is no epenthesis, and consequently there is surface violation of ONSET, when the offending syllable is unstressed: chá.ós ‘chaos’; fára.o ‘Pharaoh’. The responsible markedness constraint C_M prohibits the weakest consonant, glottal stop, from serving as the onset to a weak (unstressed) syllable, as illustrated in (13a-b).

(13) a. Input /fára.o/ → Output fa.ra.o
   b. cf. Input /aórta/ → Output aʔ ör.ta

We know that ONSET is active in Dutch and that it crucially dominates DEP, because of forms like aʔjórtta. Nevertheless, the actual output in (13a.i) violates ONSET. Violation is required because the otherwise attractive alternative, (13a.ii), violates a higher-ranking markedness constraint, C_M. Alternative paths of unfaithfulness, as in (13a.iii), are ruled out by high-ranking MAX, leaving (13a.i) as optimal. (Other constraints militate against epenthesizing some consonant besides ?.)

This is a blocking interaction. By dominating ONSET, C_M blocks the process of glottal-stop epenthesis. In other words, ? is epenthesized into empty onsets except when it would produce a prohibited PV syllable. Interactions like these are abundant throughout grammar: mark the plural with -s except when there is a lexical plural form; wh must move except when movement is prohibited (say, because [Spec, CP] is already filled). Blocking indicates crucial domination, and hence partial activity, of a markedness constraint. Though the effect of [M » F] can be observed from some mappings, there are nonetheless M-violating surface forms. The precise conditions where M is violated are defined by its interaction with C_M.

Through factorial typology, the analysis proposed for Dutch also leads to predictions about other languages. Suppose that ONSET dominates C_M; it will be epenthesized into all onsetsless syllables, whether stressed or unstressed (as in Arabic). Or consider the ranking [ONSET, C_M » MAX » DEP]. This hierarchy will produce a language where all syllables have onsets, with stress determining whether epenthesis or deletion is used to achieve it: /aortal/ → aʔjórtta.
versus /fáraol → fára or faó. (Situations like this are known in the phonological literature as conspiracies – see §2.1 and §3.1.4.3 for examples). These observations emphasize the intrinsically typological nature of OT. It is simply impossible to analyze a single language without considering the broader typological context.

Another way to get the blocking pattern is to deploy a high-ranking faithfulness constraint, \( C_F \), above \( M \) in the \([M \gg F]\) ranking. To produce a blocking pattern where \( M \) is only partially active, \( C_F \) must rule out some, but not all, of the unfaithful mappings that \([M \gg F]\) would otherwise compel. One way for that to happen is if \( C_F \) is a positional faithfulness constraint, standing in a stringency relation (cf. (5)) to \( F \). Positional faithfulness constraints are identical to general faithfulness constraints, except that they only have force over the elements of some restricted domain: in phonology, some of the loci of positional faithfulness are roots or lexical items, word-initial syllables, and stressed syllables (§3.1.3.5). Suppose \( C_F \) is a positionally restricted version of \( DEP \), limited to word-initial syllables. Ranked above \( ONSET \), \( C_F \) will bar epenthesis into initial syllables but will say nothing about epenthesis elsewhere. The resulting pattern is frequently encountered in the world’s languages: onsetsless syllables are permitted initially but prohibited everywhere else. Tableau (14) illustrates this with data from Axininca Campa (Arawakan, Peru).24

(14) Input /iŋ-koma-i/ → Output iŋ. ko. ma. ti

<table>
<thead>
<tr>
<th></th>
<th>( C_F )</th>
<th>( ONSET )</th>
<th>( DEP )</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>X</td>
<td>X</td>
<td>*</td>
<td>Medial epenthesis only</td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td>X</td>
<td>X</td>
<td>Medial &amp; initial epenthesis</td>
</tr>
<tr>
<td>c.</td>
<td></td>
<td></td>
<td>X</td>
<td>Deletion</td>
</tr>
<tr>
<td>d.</td>
<td></td>
<td></td>
<td></td>
<td>Faithful</td>
</tr>
</tbody>
</table>

This example shows both activity by and blocking of \( ONSET \). It is visibly active on the medial /a-/ vowel sequence, forcing epenthesis of \( t \) in preference to the faithful form in (14d). Now compare (14a) and (14b). The latter has two epenthetic consonants so it incurs two \( DEP \) violations, but that should not matter because \( ONSET \) dominates \( DEP \) and (14a) has an \( ONSET \) violation that (14b) lacks. But \( ONSET \) is inactive on initial syllables, where it conflicts with the positional faithfulness constraint \( C_F \). In that position, then, the force of \( ONSET \) is blocked by this higher-ranking constraint.

The competition between (14a) and (14b) nicely illustrates Prince and Smolensky’s (1993: 130, 148, 221) Cancellation/Domination Lemma. \( \text{[(15)]} \) This lemma (see (15)) follows from the definition of constraint domination and hence from the nature of \( \text{EVAL} \), so it is not unfamiliar. It succinctly states what is necessary for one candidate to be more harmonic than another.

(15) Cancellation/Domination Lemma (C/D Lemma) (paraphrased)

Suppose two candidates \( A \) and \( B \) do not incur identical sets of violation-marks. Then \( A \) is more harmonic than \( B \) if and only if every uncanceled mark incurred by \( A \) is dominated by an uncanceled mark incurred by \( B \).

For \( A \) to beat \( B \), every constraint favoring \( B \) over \( A \) must be dominated by some constraint favoring \( A \) over \( B \). That’s it.

Recall from §1.1.1 how mark cancellation works. In a tableau that compares exactly two candidates, shared violation-marks can be safely deleted, since they make no contribution to the comparison. Suppose (14a) is compared with (14b), ignoring all other candidates.

(16) Mark cancellation

<table>
<thead>
<tr>
<th></th>
<th>( C_F )</th>
<th>( ONSET )</th>
<th>( DEP )</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

In (16), the canceled marks are overstruck with \( \times \). Now, according to the C/D Lemma, any (uncanceled) mark incurred by the winner must be dominated by some uncanceled mark incurred by the loser. After mark cancellation, the winner has just one violation-mark, located in the \( \text{ONSET} \) column. Because \( C_F \) dominates \( ONSET \), this mark is indeed dominated, and so (16a) is more harmonic. As always, the choice between candidates is made by the highest-ranking constraint on which they differ (§1.1.1), and in (16) that constraint is \( C_F \).

Factorial typology must also be considered. With the ranking \([M \gg C_F \gg F]\), \( M \) is active on the whole language, but with the ranking \([C_F \gg M \gg F]\) (as in Axininca Campa), \( M \)'s activity is limited to contexts not targeted by \( C_F \). The result is a well-documented difference in inventories (§3.1.2): the inventory of permitted linguistic objects may be richer in certain contexts than in others (§3.1.3.5). For example, Axininca Campa permits syllables with and without onsets in word-initial position, but only syllables with onsets elsewhere; Hindi allows verbs to select ergative-nominative or nominative-accusative case marking in perfective clauses, but only nominative-accusative case marking is permitted elsewhere.25

1.3.4 Summary

Permuted ranking supplies various ways to control the activity of markedness constraints and thereby to limit the unfaithful mappings they can compel. With the ranking \([F \gg M]\), all mappings are faithful, even if the output violates \( M \). The opposite ranking, \([M \gg F]\), yields unfaithful mappings and outputs that consistently obey \( M \). Adding a third constraint, markedness or faithfulness, that
is crucially ranked above \([M \gg F]\) restricts the unfaithful mappings to certain conditions, producing a bifurcation or nonuniformity in observed output forms (§3.2.1). Remarkably, even the \([F \gg M]\) ranking cannot guarantee that \(M\) is entirely inactive; it can only say that \(M\) will not compel unfaithful mappings. Even low-ranking \(M\) can have visible activity in situations where faithfulness is not at issue (§3.2.2).

This range of interactional possibilities is essential to explanation and analysis in OT. With it, the goal of positing a set of simple, universal constraints may be attainable. Through ranking permutation, interaction predicts a typology that follows directly from the hypothesized constraint set. As we will see in Chapter 3, interaction of simple constraints can produce complex surface patterns. Interaction also sharply distinguishes OT from parametric theories, since through interaction a constraint can be active but not always obeyed (§3.2.2).

### 1.4 How to Do OT

In the previous sections we have looked at OT from the top down: what are its basic premises, and how do they apply to language? Here we look at OT from the bottom up: how should an analyst informed about OT’s premises apply it to data, construct novel hypotheses, and generally proceed to explore the theory, critique it, and seek to improve it? This section is less about theory, then, and more about practice; the dicta here are not assumptions or deductions but rules of thumb derived from experience. Readers wondering when it starts to get easy can safely assume that everything that seems hard really is hard and that every potential mistake hinted at in this section is one that I have made, sometimes more than once, and not just when I was first learning about OT.

#### 1.4.1 Ranking Known Constraints

A ranking argument (§1.1.1) uses exactly two candidates to rank two constraints. For the argument to be valid, certain conditions must be met:

- The constraints must conflict; they must assess different members of the candidate pair as superior. Conflict is the only basis for a direct ranking argument.
- One of the candidates must be a winner and the other must be a loser. The whole point of constraint ranking is to select the right candidate as the winner. There is nothing to be learned by comparing one nonoptimal candidate with another.
- The ranking argument must be checked in the context of the full analysis. Suppose there is a tentative argument for \([C_1 \gg C_2]\) based on comparing \(\text{Cand}_{\text{winner}}\) with \(\text{Cand}_{\text{loser}}\). This argument is not solid until we have checked for the existence of a third constraint, \(C_3\), that meets the following conditions: it is ranked above \(C_2\), and it concurs with \(C_1\) by assigning fewer violation-marks to \(\text{Cand}_{\text{winner}}\) than \(\text{Cand}_{\text{loser}}\). If \(C_3\) with these properties exists, then the ranking argument for \([C_1 \gg C_2]\) is not valid.

In real life, the first two conditions for a valid ranking argument are of constant applicability, whereas the third turns out to be a problem only once in a while. The tableaux in (17a–h) schematize various situations that can arise in practice. Ask yourself whether each tableau presents a valid argument for \([B \gg C]\), then check your answers in the footnote.26

(17) Practice tableaux

<table>
<thead>
<tr>
<th></th>
<th>/in/</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td><img src="#" alt="Tableau a" /></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>b.</td>
<td><img src="#" alt="Tableau b" /></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>c.</td>
<td><img src="#" alt="Tableau c" /></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>d.</td>
<td><img src="#" alt="Tableau d" /></td>
<td>**********</td>
<td>*****</td>
</tr>
<tr>
<td>e.</td>
<td><img src="#" alt="Tableau e" /></td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Ranking arguments are often easier to develop and understand using the comparative tableau format introduced by Prince (2000). In a comparative tableau, each row shows the results of a direct comparison between the optimal candidate and one of its competitors. The cells show how each individual constraint evaluates each comparison: Does it favor the optimal candidate, in which case “W” (for winner) appears in the cell? Or does it favor the competitor, in which case “L” (for loser) appears in the cell? Or does it favor neither, in which case the cell is left blank? For an illustration, look at the tableau in (18), which translates the traditional violation tableau (14) into comparative format.27

(18) Tableau (14) in comparative format

<table>
<thead>
<tr>
<th>/in/</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ii.</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Remember: the rows compare the winner *iq.ko.ma.ti with each of the losing candidates, indicating whether *iq.ko.ma.ti (W) or a loser (L) fares better on each constraint. Look at row (18a) for an example. The winner *iq.ko.ma.ti fares worse than the loser *tiq.ko.ma.ti on ONSET, so there is an L in the cell referring to this comparison. Neither *iq.ko.ma.ti nor *tiq.ko.ma.ti violates MAX, so that cell is left blank. And because *tiq.ko.ma.ti has initial epenthesis but *iq.ko.ma.ti does not, the winner *iq.ko.ma.ti is favored by DEP_{par-a}.

Because the comparative tableau eponymously shifts the emphasis from accumulated violation-marks to direct candidate comparison, it eliminates the technical difficulties of mark cancellation and shows directly why a particular candidate is optimal. Succinctly, in any proper comparative tableau, the leftmost filled cell of any row must contain a W.28 This follows from the C/D Lemma (15), which says that any constraint favoring a loser must be dominated by some constraint favoring the winner. That is what we see in (18): every L, which indicates a loser-favoring comparison, is dominated by some W, which indicates a winner-favoring comparison.

Comparative tableaux make OT’s inherently comparative nature particularly apparent. They can also considerably simplify the problem of determining constraint rankings in situations where there are several failed candidates and several relevant constraints to juggle. Imagine that we are in the middle of analyzing Axininca Campa, with some hypotheses about what constraints are involved but little understanding of their ranking. The as-yet unranked comparative tableau in (19) illustrates that situation. (This tableau is like (18), but the columns have been deliberately scrambled to simulate ignorance of the correct ranking.)

(19) Comparative format prior to ranking

<table>
<thead>
<tr>
<th>/iq.koma-i/</th>
<th>ONSET</th>
<th>MAX</th>
<th>DEP</th>
<th>DEP_{par-a}</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. i.q.ko.ma.ti – *tiq.ko.ma.ti</td>
<td>L</td>
<td>W</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>b. i.q.ko.ma.ti – ko.ma.ti</td>
<td>L</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. i.q.ko.ma.ti – i.q.ko.ma.ti</td>
<td>W</td>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the C/D Lemma, we know that every constraint favoring the loser must be dominated by some constraint favoring the winner. This is equivalent to requiring, typographically, that every cell containing an L be preceded in the same row by a cell containing a W. So row (19a) is relatively uninformative: it tells us that DEP and/or DEP_{par-a} dominates ONSET but not which one. Row (19b) is more helpful: as long as all relevant constraints are on the table, we can be certain that MAX dominates ONSET, because MAX supplies the only W in that row to dominate ONSET’s L. Row (19c) offers equal certainty that ONSET dominates DEP. Together, rows (19b) and (19c) yield the ranking [MAX >> ONSET >> DEP], resolving the disjunction in (19a): since DEP is ranked below ONSET, only DEP_{par-a} can supply the W that overrules ONSET’s L in (19a). The final ranking, then, is [MAX, DEP_{par-a} >> ONSET >> DEP]. (In §4.2.1, this same example is used to illustrate a learning algorithm that bypasses the rather tricky disjunction of (19a).)
When it comes to figuring out constraint rankings, comparative tableaux have a big advantage over the traditional violation tableaux. In the traditional format, ranking arguments must be done with 2 x 2 tableaux to avoid massive confusion, but they also need to be checked with the full set of constraints and additional candidates to avoid potential invalidity. In the comparative format, it is possible to combine these two steps and get a much better picture of how several constraints assess a set of candidates.

Simple 2 x 2 ranking arguments are nonetheless essential for developing analyses and especially for presenting them in articles or lectures. It is a big mistake to take shortcuts. For instance, when I first drafted the Emai analysis in §3.1.4.3, I skipped the ranking arguments, wrote down some tableaux with the ranking that I thought was right, and proceeded from there. Fortunately, I discovered my blunder before circulating the draft. It is also wise not to lay down a giant constraint hierarchy at the beginning of a paper and then try to justify it retrospectively. Giant hierarchies are usually incomprehensible to everyone, the author no less than the reader, and even if they contain no outright errors, they often overspecify the rankings that can be proven with proper arguments.

### 1.4.2 Selecting Informative Candidates

Everyone who has ever presented a lecture about OT has had the unpleasant experience of suddenly hearing about a problematic candidate from an irritatingly astute member of the audience. Finding the right candidates to study may be the hardest but also the most useful practical skill in doing OT. This section includes some strategies for finding the right candidates to worry about and for eliminating uninformative candidates from further consideration. The next section describes the encounter with a problematic candidate and how to proceed from there.

For the analyst, figuring out which candidates not to worry about is the easiest task. If all the forms or mappings in a language obey some known constraint C, then it is safe to say that C is undominated in that language. And if C is undominated, then candidates violating C will not be instructive. For example, a first encounter with the Aixinica Campa example in (18) might lead to concerns about the failed candidate *gikomati*, where the initial ONSET violation is relieved by metathesis. But if metathesis is never observed in the language, then the faithfulness constraint militating against it (LINEARITY in correspondence theory) must be undominated. Under these circumstances, *gikomati* is just a distraction from the real business at hand – it deserves a footnote at best, and it definitely should not claim space in every tableau when the analysis is written up.

Incremental progress in developing an analysis can also help narrow the candidate set under active investigation. For instance, once it has been established that MAX dominates ONSET in Aixinica Campa, there is nothing to be gained by incorporating MAX-violating candidates in subsequent tableaux, and there can be a price to pay as readers’ attention flags.

Candidates that are harmonically bounded (§1.3.1) by other candidates can never win, and so they should not detain the analyst or the reader. Cand1 harmonically bounds Cand2 if and only if Cand1 incurs a proper subset of Cand2’s violation-marks, as in tableau (8a). In a comparative tableau (§1.4.1), candidates that are harmonically bounded by the winner are those that have a W but no L in their row.

The harder task is figuring out which candidates one should worry about. There is no certain procedure here, because OT’s highly interactive character can challenge the best analyst’s skills. But there are three basic strategies: one to use at the beginning of analysis, one in the middle, and one at the end.

At the outset of analysis, start with the fully faithful candidate, since it is guaranteed to be a member of every candidate set and to obey many constraints (all of the faithfulness constraints, at least). It is never harmonically bounded and so must always be dealt with by constraint ranking, if an unfaithful mapping is the intended outcome. Then proceed to deform the faithful candidate systematically, introducing individual unfaithful mappings and combinations of them to construct additional candidates. It is helpful to have a more or less explicit theory of GEN to aid in this systematic exploration of the candidate space. It is also helpful to have an overall sense of what the undominated constraints are to limit the task by setting aside hopeless candidates. Do not neglect to construct candidates that vary in properties not normally thought of as matters of faithfulness, such as differences in syllabification or perhaps other structural properties.

For example, suppose we are analyzing the Aixinica mapping /iŋ-koma-i/ → [iŋ.ko.ma.ti]. We start with the fully faithful candidate but take note of differences in syllabification that can distinguish otherwise identical candidates: [iŋ.ko.ma.i] and [iŋ.ko.ma.i] are different candidates. We then systematically deform the faithful candidate, perhaps starting with deletion – [iŋ.ko.ma], [iŋ.ko.mi], [ŋko.ma.i], [ko.ma], and so forth. Some of these candidates are readily disposed of. For example, [ŋko.ma.i] starts with a consonant cluster, unlike any actual word of Aixinica. This suggests that [ŋko.ma.i] violates an undominated markedness constraint, so it and other candidates like it can be safely set aside as footnote material at best. Similarly, once it has been established that MAX dominates ONSET on the basis of candidates like [iŋ.ko.ma], candidates with even more deletion, such as [ko.ma], do not merit further attention. Once we have learned what we can from candidates with deletion, we can proceed in the same fashion to candidates with epenthesis and other, less familiar unfaithful mappings.

The suite of candidates produced by the procedure just described will make a good starting point. Subsequent development of the analysis will target in on candidates showing particular properties. One important strategy in the middle of the analysis is to think logically about how the as-yet unranked constraints (such as MAX and $\text{DE}^\text{hier-o}$ in (19)) can be brought into conflict, so as to get a fuller picture of the constraint hierarchy. The results of this deduction can then
be brought to bear on the search for input-candidate pairs that can supply ranking arguments for these constraints. Another strategy is to seek confirmation for transitive rankings from direct arguments. If ranking arguments for \([A \gg B]\) and \([B \gg C]\) have already been established, it is wise to look for an input-candidate pair that brings A and C into direct conflict. Unwelcome proof of \([C \gg A]\) would call the whole analysis into question. Failure to take this necessary step "invites theoretical disaster, public embarrassment, and unintended enrichment of other people's careers" (McCarthy and Prince 1993b: 12).

When the work seems more or less done, the method of mark eliminability (Prince and Smolensky 1993: Chapter 7) can be used to check whether all potentially problematic candidates are under the control of the analysis. Start with the traditional violation tableau, like (20) from Axininca Campa.

(20) Violation tableau based on (14)

Now look at each violation-mark incurred by the winning candidate and consider a competitor that eliminates it, keeping all else equal. Has the analysis successfully disposed of that competitor, as it must? Well, \(\text{ijkomati}\) violates ONSET, and that mark could be avoided by deleting the initial syllable (*komati), epenthesisizing a consonant (*tiikomati), metathesizing (*ykomati), devocalizing (*VKomati), and so on. Using these candidates, determine whether ijkomati's ONSET mark is eliminable by violating some constraint ranked below ONSET or, worse yet, no constraint at all. If it is, the analysis has a problem and further study is needed.

None of these methods is perfect, since all rely on the analyst's ingenuity. Perhaps the hardest thing is realizing the diversity of ways that a constraint could in principle be satisfied, so that this diversity can be placed under analytic control. The space of what is logically possible is very broad, because GEN affords so many options. There is no easy way around this.

1.4.3 Diagnosing Problems

The analyst usually explores some empirical domain, attempting to gain insight by studying the interactional capabilities of some small set of simple constraints against some inputs and candidates. In this situation, it is not unusual to run up against problems of stinginess. For some inputs, the wrong candidate is selected by the hierarchy, and further ranking is impossible or unhelpful. The only possible conclusion (short of changing one’s underlying assumptions about GEN or the input) is that the constraint set being studied is too stingy and must be expanded.

The next section describes some heuristics for positing new constraints, but even without looking at an actual example, it is possible to draw some inferences about what the new constraint must do and how it must be ranked. There are three possible situations, simplified down to the bare bones in (21a–c). All three assume that the ranking \([A \gg B]\) has been established independently and that out1 is the intended output form.

(21) Situations requiring an additional constraint

In (21a) and (21b), the wrong candidate wins because the higher-ranking constraint favors out2 or both candidates tie on the higher-ranking constraint and the lower-ranking one favors out2. And in (21c), the candidates tie on both constraints, so there is insufficient discrimination between them.

The logic of the C/D Lemma (15) provides clues about the new constraint needed to solve these problems. The revised tableaux in (22a–c) illustrate.

(22) Like (21), but with additional constraint supplied
To ensure that \textit{out}1 wins in (21a), a constraint is needed that assigns fewer marks to \textit{out}1 than to \textit{out}2 and that dominates A. This constraint, C, has been added to (22a). Likewise, to select \textit{out}1 as optimal in (21b), we require a constraint that assigns fewer marks to \textit{out}1 than to \textit{out}2 and that dominates B, as shown in (22b). For (21c), a constraint is likewise necessary that assigns fewer marks to \textit{out}1 than to \textit{out}2, but its ranking cannot be determined on the basis of this evidence. (In general, constraints that only break ties between two candidates are unrankable, since they will correctly break the tie no matter where they appear in the hierarchy.) Once this bit of reasoning is out of the way, the search for the needed constraint is a lot simpler.

The process of inference just sketched allows the analyst to deduce some desiderata for a new constraint before that constraint has been formulated. This logic also allows the analyst to determine the indesiderata for any new constraints when a result derived from factorial typology is at stake. Many universals of language can be derived from factorial typology under specific assumptions about CON (§3.1.5.3): if CON contains only the constraints A, B, and C, then the only possible languages are those with grammars chosen from the six permutations of these constraints. In real life, of course, CON is not so small, but the logic of constraint interaction can help determine what constraints, if they existed, would invalidate results obtained from permuting a small set of constraints. For a concrete example, see the discussion of (51) in §3.1.5.3.

Factorial typology is also at stake when an existing constraint set is guilty of \textit{profligacy}, that is, if it yields nonoccurring languages under ranking permutation. Analysis and theorizing in OT combine the study of language-particular patterns with universal typology. A constraint set may be too rich in the typology it yields, even if it works for a particular language. Suppose study of some language has produced the tableau in (23), with the constraints ranked as shown.

(b) \begin{tabular}{|l|c|c|c|} \hline /in/ & A & C & B \hline i. & \texttt{out}1 & & * \hline ii. & \texttt{out}2 & * & \hline \hline \end{tabular}

If this tableau includes all of the relevant candidates and constraints, there are certain inferences that can be drawn with certainty:

(i) There is a language (the one shown in the tableau) where /\texttt{in}/ is mapped most harmonically to \textit{out}1.
(ii) There is a language where /\texttt{in}/ is mapped most harmonically to \textit{out}2 (if C dominates A and B).
(iii) There is a language where /\texttt{in}/ is mapped most harmonically to \textit{out}3 (if D dominates B and A dominates C).

Suppose, though, that diligent research in Harvard’s Widener Library has failed to uncover any languages of the predicted type (ii), where /\texttt{in}/ maps to \textit{out}3. The overgenerating constraint set needs to be reined in, and the only way to do that is to eliminate constraints or change their definitions so they assign different violations. In (23), the solution is to eliminate constraint D. Once that constraint is gone, \textit{out}1 harmonically bounds \textit{out}3, and so the /\texttt{in}/ \rightarrow \textit{out}3 mapping can never be optimal.

1.4.4 Positing New Constraints

Positing a new constraint is not to be undertaken lightly. Constraints in OT are not merely solutions to language-particular problems; they are claims about UG with rich typological consequences. Moreover, the need for a new constraint has to be established securely. The first and best place to look for the solutions to analytic problems is in interaction of known constraints, because that is where the most interesting results and explanations are to be found. The next place to look is in modification of a known constraint. Perhaps a subtle change in the formulation of a preexisting constraint will produce the desired result without adversely affecting other results attributed to that constraint. New constraints may seem to offer an easier solution, but they can bring a cost in typology, especially if the new constraint cannot eliminate an old one.

Suppose, though, that the need for a new constraint has been established conclusively. How does one proceed to formulate it? The learner is arguably supplied with an innate, universal constraint component \textit{CON}. But the analyst is in the more difficult position of attempting to determine the contents of \textit{CON} from indirect clues in the form of linguistic generalizations. The remarks in §1.2,
while not providing a comprehensive theory of possible constraints, suggest several heuristics to follow.

Descriptive universals rarely make good constraints, but descriptive tendencies often do. Indeed, the success of OT in incorporating phonetic or functional generalizations is largely a consequence of its ability to give a fully formal status to the otherwise fuzzy notion of a cross-linguistic tendency (§4.4). Tendencies, then, are a good place to start in theorizing about constraints, but they are not a good place to end up. Constraints need to apply in fully determinate ways; a constraint like “syllables tend to have onsets” is simply unintelligible as an instruction for how candidates are to be evaluated. Think of a constraint as a function that assigns a set of violation-marks to a candidate, and make sure that the function is well defined. Constraints formulated as “assign one violation-mark for every…” might seem wordy, but they are admirably explicit.

Another heuristic is to avoid slipping bits and pieces of EVAL into constraint definitions. For example, constraints should not make overt comparisons, since EVAL already has that job well in hand. A constraint like “* is a better (or less marked) syllable nucleus than *” is therefore inappropriate, as is “minimize the duration of a short vowel in an open syllable.” For the same reason, it is not necessary for constraint formulations to contain “should” or equivalent expressions, as in “syllable weight should not exceed two moras” or “avoid Pronoun.” Constraints prohibit or demand; they do not urge, cajole, or suggest. As a general strategy, constraint definitions requiring “except if” or “only if” clauses, such as “the head or specifier of a CP may be deleted only if that CP is a complement,” should be split into two constraints, leaving the contingency up to EVAL. Likewise, constraints should not paraphrase rewrite rules or transformations, like “form perfect iambs” or “move wh,” because OT attributes rewrite or transformational effects to markedness/faithfulness interaction (§1.3).

Constraints that merely describe a required or forbidden state of affairs are probably too superficial. Analyses, including some in this book, will occasionally invoke ad hoc state-of-affairs constraints, but this should always be seen as a temporary expedient to avoid a long diversion. State-of-affairs constraints typically have several problems. Since they derive from specific observations about a single language, they are unlikely to lead to a successful typology under ranking permutation. In addition, because OT explains things by constraint interaction, a complex descriptive constraint pretty much explains nothing. And taking the long view, state-of-affairs constraints are antithetical to developing an underlying theory of constraints along the lines of §1.2.3.

The final heuristic is to proceed cautiously, or at least consciously, when importing background assumptions from other theories. For example, research in phonology since the days of Trubetzkoy has often equated being more marked with having more structure. Contemporary versions of this representational theory of markedness include radical underspecification (Archangeli 1984; Kiparsky 1981, etc.) and privative features (Avery and Rice 1989; Steriade 1995, etc.). Representational markedness might make sense in OT, or it might not. This is an empirical question to be decided by the usual methods and not aprioristically. (See §3.2.1.4 for related discussion.)

Here is an example to illustrate many of the points made in this section. An analysis of the null pronoun Pro must account for contrasts like those given in (24).

(24) Distribution of Pro
a. Mary, hopes Pro, to see Bill.
   * Mary, hopes she/her, to see Bill.
b. * Mary, hopes Pro, will see Bill.
   Mary, hopes she, will see Bill.

Though there is much more to the problem than (24) lets on, I will as usual simplify the example and discussion down to just this contrast: a null pronoun in infinitival clauses versus an overt pronoun in finite clauses.

Observationally, Pro must be coindexed with an antecedent in the smallest XP that contains Pro and Tns (tense), which is why Pro subjects are not possible in tensed clauses like Pro will see Bill in (24b). This observation, though it accurately describes a state of affairs, would not make a good constraint. It has a composite structure that looks like several constraints rolled into one. Above all, the putative constraint makes an overt comparison, “smallest XP,” that is best left up to EVAL. Extremes like smallest or largest should be obtained by EVAL using simple constraints that describe prohibited or required configurations. (See Legendre, Smolensky, and Wilson 1998: 251–52 for an application of this reasoning in similar circumstances and §3.2.3 for related discussion.) What we want, then, is something more like the two constraints in (25) to characterize the observed state of affairs.

(25) Some tentative constraints for Pro
a. CONTROL
   Pro is coindexed with something. Assign one violation-mark * for every Pro that is not coindexed.
b. DOMAIN
   If Pro is coindexed with A, assign one violation-mark * for every XP, that contains Pro and Tns and does not contain A.

These constraints are not things of beauty (nor are they a complete analysis of (24)), but they have certain virtues. They decompose the observed state of affairs into smaller constraints whose interaction through factorial typology can be studied. They rely on EVAL itself to obtain the notion “smallest XP” and, not inconsequentially, they are defined clearly enough to be usable without ambiguity.

To sum up, here is a research strategy that has often proven to be productive. Take an intuition or observation about language and restate it as a con-
Coherence to Optimality Theory

**1.5 For Further Reading**

**1 The Basics**

The locus classicus of OT is Prince and Smolensky (1993). Their Chapter 5 would be appropriate (though challenging) reading at this stage. Other treatments of the fundamentals, in varying degrees of technical detail, include Archangeli and Langendoen (1997b), Kager (1999a), McCarthy and Prince (1993b: Chapter 2), Prince and Smolensky (1997), and Tesar, Grimshaw, and Prince (1999). Legendre's (to appear-a) introduction to OT focuses on syntax. Beckman, Walsh Dickey, and Urbanczyk (1995) is a useful compilation of papers on a variety of phonological and syntactic topics. The Rutgers Optimality Archive (http://roa.rutgers.edu) is indispensable.

**2 Mark Cancellation**

§6 Stringency Relations among Constraints

§7 Harmonic Alignment of Prominence Scales
Harmonic alignment of prominence scales is introduced in Prince and Smolensky (1993: Chapter 6, 8). Discussion and applications in phonology include Anttila (1997a), de Lacy (1999), Gnanadesikan (1997), Green (1993), Kenstowicz (1994b), and Lombardi (to appear). Aissen (1999), Artstein (1998), Grimshaw (to appear), and Lee (2000) discuss extensions and applications to morphosyntactic hierarchies (animacy, person/number/gender), and Burzio (1998) uses similar notions. Also see the references in §3.417.

§8 Basic Effects of Constraint Interaction
The treatment of constraint interaction in Prince and Smolensky (1993: Chapter 3, 4) is lucid. Some representative applications of permuted ranking to (morpho)syntactic typology include Bresnan (to appear-b), Grimshaw (to appear), Keer (1999b), Samek-Lodovici (1998), and Woolford (to appear). On homogeneity of target/heterogeneity of process and positional faithfulness, see the references in §3.4 8 and 7.

§9 Harmonic Bounding
Harmonic bounding was introduced by Samek-Lodovici (1992) and figures prominently in works like Keer (1999a), Morelli (to appear), Prince and Smolensky (1993: Chapter 9), and Samek-Lodovici and Prince (1999).

Notes
1. This formulation is based-on Grimshaw (1997b), Prince (2000), and, ultimately, the Cancellation/Domination Lemma of Prince and Smolensky (1993: 130, 148, 221) (see 15)). See Samek-Lodovici and Prince (1999: Appendix A) for further formal development.
2. The three laws of robotics are
   1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
   2. A robot must obey the orders given to it by human beings except where such orders would conflict with the First Law.
   3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.
3. Additional helpful annotations are often used in tableaux, including anclamation point to mark fatal violations (such as Candc's violation of C1 in (1)) and shading of cells that are irrelevant because higher-ranking constraints have been decisive.
4. Typos for typographic clarity. I do not use these annotations.
5. What if C3 concurs with C1 and there is no evidence that either one is ranked below C2? We are then stuck with a disjunction: we know that C1 or C3 dominates C2, but we do not know which one. Disjunctions like these are particularly problematic for learning theories, though an important idea in the learning of OT grammars avoids this difficulty (see §4.2.1).
6. Vieri Samek-Lodovici suggests the following as an example of a logical inference about ranking. Suppose there is a ranking argument establishing that A or B dominates C. (This can happen when the losing candidate violates both A and B, so it is unclear which is the fatal violation. See note 4.) Suppose another, independent ranking argument establishes that A or C dominates B. It is now legitimate to infer that A is top ranked, dominating both B and C, since that is the only way to combine the results of the two ranking arguments into a consistent constraint hierarchy. See Hayes (1997) for discussion of this and other situations where ranking inferences can be drawn.
7. Grimshaw (1997b: 411) exploits the possibility of multiple optimal outputs to account for syntactic optionality (§4.1.3), Hammond (2000) also discusses this point.
8. Also see Karttunen (1998) for a similar approach.
9. For the same reason, it is wrong to see GEN as somehow equivalent in its explanatory responsibilities to other theories' generative components, such as Chomsky's (1995) computational system for human language, CxM. For example, the dialogue "L" in Uraigereka (1998: 168) insinuates that "an optimality approach isn't particularly useful for combinatorial systems: what we want to understand is why Gen gives the structures it does. . . the bottom line is that something has to give us those structures, whether it's called CxM, Gen, or God. Personally, I'm interested in understanding the nature of the combinatorial function." L's pessimistic view of OT is unjustified since it is based on the entirely unreasonable requirement that GEN in OT should explain the same things that Chomsky's CxM does. Different theories impose different organizations on the world, and so individual components of those theories, taken in isolation, obviously cannot be compared in this way.
10. This formulation comes from Smolensky (1993).
11. Conflict among markedness constraints, though not anticipated by the Prague School, was also recognized in Natural Phonology (§2.1), as when Stampe (1973a: 23) speaks of the "contrary teleologies of contrary processes."
12. Grimshaw (1997b) and Speas (1997) make similar points. Jane Grimshaw supplied several valuable suggestions about how to cover the syntactic end of this discussion.
13. For the same reason, claims about specific constraints cannot be attributed to OT as a whole. This is a category error, exemplified by the following remark from Breen and Pensalfini (1999: 15 n): “However, these constraints [No-Onset and CODA – JMc] are quite explicitly ruled out in OT….” This statement confounds OT in general with a specific hypothesis about CON (i.e., that it contains the constraints Onset and No-CODA, but not their opposites). The confusion is particularly apparent once it is realized that nearly every syllable theory since Jakobson’s day adopts the same hypothesis, in one form or another. That hypothesis may be wrong, as Breen and Pensalfini maintain, but it is not some ineluctable idiosyncrasy of OT.

14. Alignment constraints have a somewhat ambiguous status in markedness/faithfulness

15. Alan Prince observes that local self-conjunction allows the finite set CON to be expanded into the denumerable set \{[Cl], [Cl&Cl], [Cl&Cl&Cl], \ldots\}. Prince has conjectured, and Paul Smolensky has proven, that a denumerable set of constraints yields 2\(^{\infty}\) possible grammars under ranking permutation. The proof goes like this. Assume two fixed constraint hierarchies of countably infinite length. Each hierarchy is by itself impermutable, but the two hierarchies can be intercalated in various ways. By substituting 0 for all the constraints in one hierarchy and 1 for all the constraints in the other, each intercalation can be mapped onto a countably infinite string of 0’s and 1’s: 00000111, 01100011, 11101000, etc. Prepose a decimal point, and you now have a representation of all the 2\(^{\infty}\) real numbers between 0 and 1. The implications for language learning and language typology of permitting uncountably many possible grammars have not yet been studied.

16. Discussions with Paul de Lacy shaped my understanding of this material.

17. The universality of [Cl \gg C2] means that no possible language has the ranking [C2 \gg C1]. It does not mean that C1 immediately dominates C2 (other constraints can intervene), nor does it mean that every language will supply evidence for [Cl \gg C2].

18. For a different approach to natural linguistic scales, based on ternary-valued distinctive features with their accompanying markedness and faithfulness constraints, see Gnanadesikan (1997).

19. The symbol “…” marks the end of one syllable and the beginning of another. There is also an implicit “…” at the beginning and end of each word. So a.p.a.ta consists of three syllables, a (which lacks an onset), pa, and ta.

20. What if the language does not permit inputs like /a.p.a.ta/? Richness of the base (§3.1.2) excludes the possibility of systematic, language-particular restrictions on inputs.

21. This example is based on Ito (1986, 1989).

22. This example is based on an OT analysis by Grimshaw (1997a), who attributes several of the observations to Bonet (1991).


25. The Hindi example is based on the OT analysis in Woolford (to appear).

26. Answers: (a, b, c, e) All no. There is no conflict between constraints B and C because they agree in their assessment of at least one candidate. (d) Yes: The absolute number of violations does not matter, but the relative number of violations does. (f) No. Because A dominates C and A concurs with B, these candidates cannot be used to prove that B dominates C. (g) Yes. A does not distinguish the two candidates, so it does not invalidate the argument for ranking B over C. (h) Yes. Because A is ranked below C, it cannot invalidate the argument for B over C.

27. Following the discussion in §1.3.3, I have replaced the dummy constraint C\(_r\) with an actual positional faithfulness constraint. The material here is based on discussions with Alan Prince and on Prince (2000).

28. Furthermore, every row of a comparative tableau must contain at least one filled cell, unless the intention is for the candidates compared in that row to both be optimal.

29. Overspecified rankings often lead to trouble further down the line: if one has, without justification, ranked A above B, the later discovery of a ranking argument proving that B dominates A will seem like a problem when in fact it is not.

30. In principle, CON could be universal but not innate if it could be reliably induced from the universally shared experiences of learners (§4.6 §14).


32. The temptation to describe the state of affairs rather than to seek an insightful analysis is not some new vice introduced by OT. Rather, Newmeyer (1996: Chapter 5) finds it to be a recurring problem in the history of generative grammar. For example, in much syntactic research of the 1960s, “[t]he author identified a construction, then wrote a transformational rule which came close to mimicking its surface characteristics” (p. 45). Recent research is no less immune to this vice, though it expresses itself in other forms, such as “the language-specificity of some parameters that have been proposed within GB” (p. 65). (I am indebted to Jane Grimshaw for discussion of this material and for suggesting the phrase “state-of-affairs constraint.”)

33. Prince and Smolensky (1993: Chapter 3, n. 13, citing a personal communication from Cheryl Zoll) discuss the idea that there is a broad family of constraints *STRUC (pronounced “star-struck”) that militate against all structure whatsoever, thereby implementing a very general representational markedness theory. They note that possible applications of this idea extend beyond phonology: witness Chomsky’s (1986: 4) prohibition on nonbranching N* or Grimshaw’s (1993, 1994) constraint MINIMAL-PROJECTION (though see Grimshaw 1997b: 381 for an alternative view). Cauley (1999a) pursues a representational approach to phonological markedness in OT.

34. Example (24) is based loosely on Speas (to appear).
In addition to the rankings in (79), PWdCON is dominated by the constraints responsible for higher-level phrasing, requiring IPh boundaries at the edges of clauses, around parentheticals, and so on.

By virtue of PWdCON, the normal or default condition for a function word is to be a clitic. Alignment constraints can impel a function word into PWd status but only under duress. This analysis not only works for English but also yields the right typology: in all languages, function words are typically cliticized unless special conditions like these obtain.

This theory of function-word phonology depends crucially on parallel evaluation. Candidate analyses differ in the prosodic structure assigned to function words, and so constraints can evaluate those differences in the wider prosodic context. Top-down effects are both expected and observed.

Bottom-up serialism has a difficult time with this phenomenon. Because the process of reduction is irreversible (e.g., a and her both neutralize to [a] in my speech), the standard serial analysis starts out by analyzing all words, both functional and lexical, as freestanding PWd's (cf. Selkirk 1972, 1984). Later in the derivation, as higher-level prosodic structure is erected, processes of citation or “destressing” apply, reducing function words in certain contexts. Derivations for the fragments to Boston and Where to? are given in (80).

<table>
<thead>
<tr>
<th>(80) Serial derivation of reduced and unreduced function words</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWd-level analysis</td>
</tr>
<tr>
<td>[[to]<em>{reduced} [Boston]</em>{reduced}]</td>
</tr>
<tr>
<td>Phrase-level analysis</td>
</tr>
<tr>
<td>[[to]<em>{reduced} [Boston]</em>{reduced}]</td>
</tr>
<tr>
<td>Destressing</td>
</tr>
<tr>
<td>[[to]<em>{reduced} [Boston]</em>{reduced}]</td>
</tr>
</tbody>
</table>

Destressing is blocked in the where to? case by the following IPh boundary.

Though the serial theory is basically bottom-up, the destressing rule is top-down in its effects. This example shows, then, that the serial, derivational theory must permit top-down, structure-changing rules. More seriously, the derivational theory has exactly the wrong take on the typological situation. According to this approach, the normal or default case is to analyze every syntactic “word,” functional or lexical, as an independent PWd. This structure is the default because it is imposed generally by the first relevant rule of the derivation, and its effects endure unless they are wiped out later by the special destressing process, which, moreover, makes the grammar more complex and therefore less highly valued under the Evaluation Metric (§2.1). This is surely the wrong prediction typologically. Across languages, the unmarked condition for function words is not to be freestanding PWd's, a generalization that PWdCON expresses. But out of descriptive necessity, the serial analysis must treat a highly marked system—where both lexical and function words are PWd's—as an unmarked default. That is simply backward.

There is a more general point here. Serial derivations bring with them various architectural imperatives, most of which are left tacit and unremarked on except in the “how to order rules” lecture of introductory phonology (and formerly syntax) courses. Bottom-up derivation is one imperative: for example, PWd's must be assigned before IPh's, since IPh dominates PWd in the prosodic hierarchy. Relatedly, bottom-up rules are structure building, but top-down rules can only be structure changing. Another architectural imperative of serialism involves the interaction of reversible and irreversible operations: as in the English example (80), the reversible operation must be ordered first. These imperatives sometimes force a particular analysis and, as we saw, that analysis may be implausible on typological grounds. Evaluating the empirical adequacy of serial derivation, like evaluating parallel derivation, must proceed on this basis.

A final remark. The Elsewhere Condition is often invoked in situations like this. (See §1.5 for references.) The idea is to apply destressing and PWd assignment together, disjunctively, with the more specific rule (i.e., destressing) taking precedence. This approach is quite workable—in fact, it is basically a parallel analysis. Destressing and PWd assignment compete to apply in the same contexts, and the more specific of them wins. But even with this refinement, the typological problem remains, since the typologically unjustifiable rule assigning PWd's to all function words remains as part of the grammar.

3.3.2.5 Consequences of Parallelism III: Structurally Remote Interaction

Because fully formed candidates are evaluated by the whole constraint hierarchy, obtaining some local harmonic advantage may have effects that are structurally or derivationally remote from the locus of that advantage.

This is pretty vague, so an analogy might help. I play chess in a fashion that locally and serially optimizes: when it is my turn, I look for a move that will let me attack another piece, with little or no thought for future consequences. Bobby Fischer at age 13 was a massively parallel chess-playing machine who, in a celebrated game against Donald Byrne, sacrificed a queen at move 18 to obtain a checkmate at move 41. Fischer’s skill allowed him to optimize globally over many futures considered in parallel, rather than to proceed locally and serially from his current position on the board.
Back to linguistics. The example of remote interactions between local and global structure comes from metrical phonology. In Yidiny (Pama-Nyungan, Australia) all words fall into two categories (see (81a–b)), those with trochaic (falling) rhythm throughout and those with iambic (rising) rhythm throughout:

(81) a. Trochaic rhythm
   (gálin) 'go (present)'
   (gúda)(gágu) 'dog (purposive)'
   (wúda)(bájíŋ) 'hunt (antipassive present)'
   (májíŋ)(dágal)(nínda) 'walk up (comitative subordinate dative)'

b. Iambic rhythm
   (galbfr) 'catfish'
   (bargán)(dajíŋ) 'pass by (antipassive past)'
   (mágí)(rínál)(dáñúmda) 'climb up (going comitative coming subordinate dative)'

The presumed rhythmic organization of these words – the metrical foot structure – is shown by the parentheses in (80), and the stresses are marked by acute accents.

The generalization that distinguishes these two word classes depends on vowel length. If an even-numbered syllable contains a long vowel (such as \i: or \u:), then that syllable and all other even-numbered syllables are stressed. Otherwise, odd-numbered syllables are stressed. More formally, the whole word has iambic feet if and only if an iambic parse will allow stress and length to coincide in some syllable of the word.

In this light, the generalization about Yidiny can be restated as follows: feet are iambic throughout the word if any foot has the optimal short-long iambic form; otherwise feet are trochaic. This generalization already suggests how Yidiny should be analyzed with interacting constraints. The constraint ALIGN-L(Ft, Hd(Ft)) requires every foot to have its head at the left edge. It therefore asserts the preference for or default status of trochaic feet in Yidiny. But ALIGN-L(Ft, Hd(Ft)) is crucially dominated by a constraint like LONG/STR “if long, then stressed,” as in (82).

(82) LONG/STR \(\succ\) ALIGN-L(Ft, Hd(Ft))

When a word contains no long vowels, or when LONG/STR and ALIGN-L concur in rejecting the iambic candidate, then the trochaic default emerges, see (83a–b).

In (83a), the trochaic analysis is favored by ALIGN-L because LONG/STR is simply irrelevant with words that contain no long vowels. In (83b), the trochaic analysis and LONG/STR agree in their assessment of iambic *wuŋábaːjíŋ, which is therefore rejected. (Compare the Kanakuru example in §3.2.1.3.)

The interaction of LONG/STR and ALIGN-L in (82)–(83) accounts for the local, bottom-up effect of vowel length on stress. But Yidiny also displays a remote effect: if LONG/STR forces one foot to be iambic, then all feet are iambic. This effect is apparent in iambic words of sufficient length, as in (84), where every foot except the one containing the long vowel is wrongly predicted to be trochaic, in conformity with ALIGN-L.

(84) A global effect

Candidate (84a) is the intended output form. But, according to this tableau, it is harmonically bounded by the competing candidates (84c–e). (That is, the competitors incur a proper subset of (84a)'s violation-marks – see §1.3.1, §3.1.5.3.) With the system developed so far, only a local effect of vowel length on stress is possible.

The defect in candidates (84c–e) is that they have a rhythmic lapse, a sequence of two unstressed syllables in a row. There is ample typological
justification for the rhythmic constraint *LAPSE and its counterpart, *CLASH, which prohibits the other way of disrupting rhythm, stress on adjacent syllables. By dominating ALIGN-L(Ft, Hd(Ft)), *LAPSE rules out all of the problematic candidates in (84), thereby globalizing the local effect of LONG/STR, as shown in (85).

(85) Global effect obtained with *LAPSE

<table>
<thead>
<tr>
<th></th>
<th>LONG/STR</th>
<th>*LAPSE</th>
<th>ALIGN-L(Ft, Hd(Ft))</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td></td>
<td></td>
<td>***</td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>c.</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>d.</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>e.</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

This analysis relies on the assumption that GEN emits complete output candidates, with all metrical feet in place. The grammar – [LONG/STR, *LAPSE ⇒ ALIGN-L] – selects the most harmonic of these candidates according to their overall structure. Because the competing candidates are complete, local well-formedness (LONG/STR) can have global effects.

When we look at how derivational theories have dealt with Yidiny, it becomes clear how problematic this interaction can prove to be. One approach is radically derivational (Hayes 1982): first iambic feet are assigned across the board, and later on feet are shifted to trochaic except in words with a stressed long vowel. But of course derivations like this do considerable violence to the claim of Hayes (1987) and others that the iambic/trochaic distinction is a parameter to be set on a language-by-language basis. Another approach is essentially parallel (Halle and Vergnaud 1987: 24): words are simultaneously parsed into trochaic and iambic feet, and subsequent rules select the preferred analysis and delete the other one. This idea of simultaneous, competing parses is strongly reminiscent of OT, but with a difference: the two parses are imposed on the same form, and consequently this difference in “candidates” is limited to competing prosodic analyses of a fixed segmental string (a little like the PARSE/FILL model of Prince and Smolensky 1993 (§1.2.2)).

3.3.2.6 Consequences of Parallelism IV: Derivationally Remote Interaction

Yidiny provides evidence of one kind of remote interaction that is possible in parallel OT. Another kind, derivational remoteness, has been extensively studied in research on the morphosyntax and morphophonology of lexical selection or allomorphy. Phenomena like these have been notoriously difficult to analyze in rule-based, derivational theories, as the following quotations emphasize:

What is most striking about [this analysis] is that the specific properties of the output form depend upon the other surface forms (both morphological and syntactic) that actively compete with it, and not on the details of the derivation of its formal structure, as in the classical generative approach to syntax. (Bresnan to appear-b)

This paper is concerned with a long-standing theoretical . . . problem. It is a theoretical problem because it regards the organization of the grammar; in particular it addresses the question of where in the grammar are lexical, unpredictable morphological alternations to be included, and where are phonological regularities to be expressed. (Mascaro 1996: 473)

The problem is that rules make no real contribution to analyzing these phenomena, which depend on notions like competition or selection. But competition and selection find ready expression in OT. We will look at two examples, one morphophonological and the other morphosyntactic.

The Catalan “personal article” has two forms, _en_ and _l’_. Like English _alan_ or French _beau/bel_, the choice is decided by the sound that begins the next word. The allomorph _en_ precedes consonants and the allomorph _l’_ precedes vowels, as illustrated in (86a–b).

(86) a. _en_ before consonants

b. _l’_ before vowels

l'Einstein

There is no regular phonological process relating the two allomorphs in Catalan (or English or French for that matter). There is no rule mapping some unique underlying form onto _en_ in one context and _l’_ in the other. So the existence of these two allomorphs and the relation between them is a matter for the lexicon (actually, the vocabulary – sec. 3.1.2.4) rather than for the grammar. But the choice between the allomorphs makes sense phonologically, and in fact it can be obtained through emergence of universal constraints (cf. §3.2.2).

Since the alternation itself is unpredictable, both alternants must be stored in the lexicon as a set rather than as a unique underlying representation: {_en/, /l’_}. Among the properties of GEN is the obligation to supply candidates with both alternants: _en Wittgenstein, l’Wittgenstein, en Einstein, l’Einstein_. Crucially, because both _en_ and _l’_ are present in underlying representation, candidate pairs like _en Wittgenstein_ and _l’Wittgenstein_ are both fully faithful to the input. Allomorph selection, then, brings no cost in faithfulness.

These faithfulness-free alternations provide an excellent opportunity for TETU, and that is precisely what is observed. In Catalan, onsetless syllables are abundant, showing that _ONSET_ is ranked below its antagonistic faithfulness constraints _DEP_ and _Max_. But _ONSET_, as we see in (87), emerges to decide which allomorph of the personal article to use with a vowel-initial name.

---

3.3 Consequences: Globality and Parallelism