1

An Introduction to Optimality Theory

1.1 How OT Began

Around 1990, Alan Prince and Paul Smolensky began collaborating on a new theory of human language. This collaboration led in fairly short order to a book-length manuscript, *Optimality Theory: Constraint Interaction in Generative Grammar*. Photocopies of the manuscript were widely distributed and had a terrific impact on the field of linguistics, even though it wasn't formally published until more than a decade later (as Prince and Smolensky 1993/2004). OT had and continues to have its greatest effect on phonology, but it has also led to important work in syntax, semantics, sociolinguistics, historical linguistics, and other areas. OT belongs on anyone's list of the top three developments in the history of generative grammar.

One of Prince and Smolensky's goals for OT was to solve a long-standing problem in phonology. Phonological theory in the tradition of Chomsky and Halle's (1968) *The Sound Pattern of English* (*SPE*) was based on rewrite rules. The rewrite rule A → B/C___D describes an input configuration CAD and an A → B transformation that applies to it. Rewrite rules can describe lots of phenomena, but they do a poor job of explaining how phonological systems fit together. (For a brief explanation of *SPE*'s main assumptions, see the boxed text at the end of this section.)

To illustrate, we'll look at some data from Yawelmani, a nearly extinct dialect of the California Penutian language Yokuts (Newman 1944).[1] In this language, syllables cannot be bigger than CVC (consonant-vowel-consonant). Various phonological processes are involved with this limit on syllable size. For instance, Yawelmani has a process that

deletes a vowel at the end of a word, as the data in (a) of (1) show. (The "." marks the boundary between two syllables.) But the data in (b) show that final vowels do not delete when they are preceded by a consonant cluster. The explanation for the difference between (a) and (b) is that deletion after a cluster would require a syllable that is too big or leave a consonant that cannot be syllabified: *[xatkʔ].[2]

(1) Yawelmani final vowel deletion

	Underlying	*Surface*	
a.	/taxaː-kʔa/	[ta.xakʔ]	'bring!'
	/taxaː-mi/	[ta.xam]	'having brought'
b.	/xat-kʔa/	[xat.kʔa]	'eat!'
	/xat-mi/	[xat.mi]	'having eaten'

Yawelmani also has a process of vowel epenthesis that applies to a cluster of three consonants in the middle of a word. (See (2). The data in (a) show epenthesis into triconsonantal clusters, and the data in (b) show that there is no epenthesis in smaller clusters.) If there were no epenthesis process, then the result would again be a syllable that is too big or a consonant that cannot be syllabified: *[ʔilk.hin].

(2) Yawelmani vowel epenthesis

	Underlying	*Surface*	
a.	/ʔilk-hin/	[ʔi.lik.hin]	'sing (nonfuture)'
	/lihm-hin/	[li.him.hin]	'run (nonfuture)'
b.	/ʔilk-al/	[ʔil.kal]	'sing (dubitative)'
	/lihm-al/	[lih.mal]	'run (dubitative)'

It is certainly possible to state *SPE*-style rewrite rules to account for these two processes in Yawelmani – V → Ø/VC___# and Ø → i/C___CC will do the job nicely. But, as Kisseberth (1970) first argued, those rewrite rules are missing an important generalization about the special role of surface-structure constraints in both rules. Final vowel deletion cannot create bad syllables in surface forms, and epenthesis exists to eliminate them. Adopting a suggestion from Haj Ross, Kisseberth called this situation a *conspiracy*.

When two or more rewrite rules are involved in a conspiracy, they directly or indirectly support some constraint on surface forms. In Yawelmani, the relevant constraints are a CVC limit on syllable size and a prohibition on unsyllabified consonants. Final vowel deletion is

blocked from applying when it would produce a surface form like *[xatkʔ], which cannot be fully parsed into maximally CVC syllables. Epenthesis is *triggered* to apply by the need to fix sequences that cannot be parsed into CVC syllables. In every conspiracy there is a constraint on surface forms, which we can refer to more succinctly as an output constraint, since it evaluates the output of the grammar. There is also some mixture of processes that are blocked by that output constraint and/or processes that are triggered by it.

Conspiracies are common in the languages of the world, and so it was a matter of some concern that the *SPE* theory of rewrite rules couldn't explain them. A rewrite rule, by its very nature, describes an input configuration and an operation that applies to it. A conspiracy is completely different: it refers to an output configuration, it involves several different operations, and those operations may participate in the conspiracy by applying or failing to apply, depending on the circumstances. When analysts try to describe conspiratorial behavior in terms of rewrite rules, they have to start using counterfactuals, as I did in the preceding paragraph: "blocked from applying when it *would* produce," "to fix sequences that *could* not be parsed." Statements like these show that the analyst understands what's really going on in the language, but counterfactual conditions have no place in *SPE*'s theory of how to apply rules. When a phonologist says something like "The epenthesis rule ensures that the language has only unmarked syllables," he or she is describing an intuition about how the system works. But that intuition has to be expressed formally, in the theory itself. Otherwise, we are just telling ourselves stories.

At around the same time that phonologists were beginning to grasp the importance of output constraints, syntacticians were having a similar revelation. For example, clitic movement in Spanish is triggered by an output constraint requiring that second person clitics precede first person clitics (Perlmutter 1971). That is why the direct and indirect objects appear in different orders in Te_{IO} me_{DO} *presento* 'I introduce myself to you' versus Te_{DO} me_{IO} *presentas* 'You introduce yourself to me.' Another example: in English, movement of *wh*-question words is blocked when it would leave the trace of *wh* immediately after the complementizer *that* (Chomsky and Lasnik 1977): **Who did you say that t left?* (cf. *Who did you say t left?*). These syntactic examples have parallels in Yawelmani phonology. The triggering of clitic movement by an output constraint in Spanish is like the triggering of epenthesis in Yawelmani. And the blocking of *wh*-movement in English is like the blocking of final vowel deletion in Yawelmani.

Chomsky and Lasnik (1977) proposed a theory of output constraints and their function that had (and continues to have) a great deal of influence in syntax. Their main idea is that all of the rewrite rules – that is, the syntactic transformations – are technically optional. An input to the grammar freely undergoes any, all, or none of the transformations. The result of freely applying transformations is a set of candidate surface structures. These candidate surface structures are checked by the output constraints, which are called filters, and some of them are marked as ungrammatical by the filters. For instance, the-*wh* movement transformation applies optionally, producing both *Who did you say that* t *left?* and *You said that who left?* as candidate surface structures. The *that*-trace filter marks the first of these as ungrammatical, so only the second is well-formed. Henceforth, I'll refer to Chomsky and Lasnik's proposal as the *filters model*.

The filters model does a good job of explaining how output constraints can seem to trigger or block transformations. Because the transformations are strictly optional, if there is a candidate surface structure that has undergone a transformation T, there is also a candidate derived from the same deep structure that has not undergone T. If a filter marks the result of applying T as ungrammatical, then the filter has in effect blocked T, since the derivation in which T has applied does not lead to a well-formed output. If a filter marks the result of *not* applying T as ungrammatical, then the filter has in effect triggered T, since the derivation in which T has failed to apply does not lead to a well-formed output. The filter isn't literally triggering or blocking T – it cannot, since the filter doesn't even apply in the same grammatical component as T – but the filter appears to be blocking or triggering T by ruling out the surface structure where T has or has not applied.

A goal of the filters model was to shift most of the burden of explaining syntactic patterns from the theory of transformations to the theory of filters. Transformations could be made much simpler and more general. In Government-Binding Theory (GB) (Chomsky 1981), the theory of transformations withered away almost entirely, leaving just the transformation Move α.

Although the filters model in syntax emerged not long after the discovery of the conspiracy problem in phonology, the filters model had surprisingly little influence on the field of phonology at that time. There are two main reasons for this, in my opinion. One of them, which I will explain in the next section, is that the filters model fails as an explanation for phonological conspiracies like Yawelmani's if output

constraints are inviolable, and constraints of that era were always assumed to be inviolable.

The other reason is that the field of phonology was so strongly influenced by *SPE*. *SPE*'s central hypothesis is that rules with simpler formulations are more natural, in the sense that they are more likely to occur in languages and to express linguistically significant generalizations. In accordance with this hypothesis, the *SPE* theory supplies abbreviatory devices that allow putatively natural rules to be formulated more simply. In a conspiracy, the output constraint is what makes the rules natural – the output constraint is the generalization that unites the disparate rules. Therefore, a theory of conspiracies embedded in the overall *SPE* research program would have to use the output constraint to simplify the statement of the rules that participate in the conspiracy.

Kisseberth (1970) proposed a theory of blocking effects along exactly these lines. He assumed an output constraint *CC{C, #} that is violated by medial clusters of three consonants or final clusters of two consonants. By assumption, a rule is blocked from applying if its immediate output violates this constraint. That assumption allows the formulation of the final vowel deletion rule to be simplified from V → Ø/VC___# to V → Ø/___#. And since simpler rules are more natural under *SPE*'s assumptions, the existence of the output constraint has in some sense explained why final vowel deletion is blocked after consonant clusters.

Kiparsky (1973b: 77–78) presents several criticisms of this proposal. One problem is that the rule V → Ø/___# is just as as simple, and therefore should be just as natural, in a language without the *CC{C, #} output constraint. But a language without the output constraint is a language without the conspiracy, and if conspiracies contribute to naturalness, then the language without one should be *less* natural. Another problem with this proposal is that it only works for blocking effects. Rules that are triggered by the output constraint won't receive simpler formulations. For instance, there is no way of using the output constraint to simplify the statement of the epenthesis rule, replacing Ø → i/C___CC with, say, Ø → i/___C. The problem with Ø → i/___C is that it would epenthesize [i] before every single consonant. The theory at that time lacked any sort of economy mechanism to ensure that epenthesis applies only when it's needed and not otherwise.

Starting in the mid-1970s, phonological research moved toward richer theories of representation that included syllables and other

structures. As phonological representations became more elaborated, it became possible to imagine an almost ruleless phonology in which automatic satisfaction of universal constraints on representations was all that mattered. Goldsmith (1976a, 1976b) and Prince (1983) worked on proposals along these lines for autosegmental and metrical phonology, respectively. This work ran headlong into another problem, however: the proposed universal constraints did not hold in every language all of the time. That is why the subsequent literature on autosegmental and metrical phonology, such as Pulleyblank (1986) and Hayes (1995), returned to language-particular rewrite rules as the central analytic mechanism.

By the end of the 1980s, there was certainly a consensus about the importance of output constraints, but there were also major unresolved questions about the nature and activity of these constraints. That "conceptual crisis at the center of phonological thought," as Prince and Smolensky (1993/2004: 2) refer to it, was not very widely acknowledged at the time, but in hindsight it's hard to miss. It's a major feature of the intellectual context in which OT was developed.

Explanation: The *SPE* theory and its relation to OT phonology

In *SPE*, every morpheme is assumed to have a unique underlying representation that is stored in the lexicon. The underlying representation includes all of the unpredictable phonological properties of a morpheme. For example, the Yawelmani imperative suffix has surface alternants [-kʔa] and [-kʔ], and the nonfuture suffix has alternants [-mi] and [-m] (see (1)). Their underlying representations are /-kʔa/ and /-mi/. (The underlying representations couldn't be /-kʔ/ and /-m/, because there would be no good way of explaining why [a] is epenthesized in one suffix and [i] in the other.)

The mapping from underlying to surface representations is accomplished by applying a series of ordered rewrite rules. For instance, the path from /taxaː-kʔa/ to [ta.xakʔ] requires two rules: first, the final vowel is deleted, yielding [ta.xaːkʔ], and then the vowel is shortened to produce the surface form [ta.xakʔ]. As the text mentions, a rewrite rule is an expression A → B/C___D that changes any CAD sequence into a CBD sequence. OT does not have rewrite rules or anything that resembles them.

SPE also includes a theory of representations. Every speech sound consists of a bundle of values for certain universal, binary distinctive features: [nasal], [round], and so on. In the 1970s and 1980s, *SPE*'s rather

simple representational theory was greatly enhanced. For instance, *SPE* does not include syllables in its representations, but later work would analyze the [ta.xaːkʔ] → [ta.xakʔ] mapping as a process of vowel shortening in a syllable that is closed by a consonant.

Most work in OT phonology presupposes *SPE*'s view of underlying representations, its theory of distinctive features, and many of the subsequent representational enhancements. It's important to realize, however, that OT itself does not require a commitment to any of these ideas.

QUESTIONS

1 How will the filters model work when several different transformations are applicable? What about when a transformation is applicable at several different places in a sentence? What about when a transformation is applicable to its own output?

2 The text promises that the next section will give an argument that the filters model cannot explain phonological conspiracies if constraints are inviolable. Try to figure out the argument before reading the section. (Hint: An output constraint is needed to ensure that final vowel deletion occurs in /taxaːkʔa/ → [ta.xakʔ].)

EXERCISES

3 Yawelmani has output constraints that limit syllables to a CVC maximum and require exhaustive syllabification. In Yawelmani, these constraints trigger epenthesis and block final vowel deletion. Can you imagine a different language that has the same output constraints but which block and/or trigger [0]other processes? Hypothetical examples are fine; it isn't necessary to identify actual languages.

1.2 Why Must Constraints Be Violable?

In the previous section, I alluded to a second reason why phonology did not develop an optional-rules-plus-output-constraints theory, similar to the filters model in syntax. The main impediment was the assumption, standard at the time, that output constraints are never violated.

Suppose we try to apply the filters model to Yawelmani. (It may be helpful to follow the chart in (3) as you read the rest of this paragraph.) Since there is epenthesis in the language, the transformational component must contain an optional epenthesis rule. Given /ʔilk-hin/ as the input to the transformational component, the output of that component will include [ʔi.lik.hin], with epenthesis, and various ways of syllabifying the word without epenthesis, such as *[ʔilk.hin] and *[ʔil.k.hin]. (I will use the notation ".k." to indicate that the [k] is outside the syllable on its left and right. It's unsyllabified.) These three forms are then checked by the filters. One filter, which I'll call *C^{unsyll}, prohibits unsyllabified consonants. It marks *[ʔil.k.hin] as ungrammatical. Another filter marks *[ʔilk.hin] as ungrammatical because it contains a syllable that exceeds the CVC limit. (I'll call this filter *Complex-Syllable.) Since *[ʔil.k.hin] and *[ʔilk.hin] are ruled out by the two filters, [ʔi.lik.hin] is the only grammatical output from this input. From the perspective of someone looking at the output of the grammar, it looks as if the filters *C^{unsyll} and *Complex-Syllable are triggering the epenthesis process. (For a brief explanation of the role of syllable structure in phonological processes, see the boxed text at the end of this section.)

(3) Filters model applied to Yawelmani – input /ʔilk-hin/ → [ʔi.lik.hin]

Input	Transformational component (all optional)	Output of transformational component	Filter component	Output of filter component
/ʔilk-hin/	syllabification epenthesis	[ʔi.lik.hin] [ʔil.k.hin] [ʔilk.hin]	*C^{unsyll} *Complex-Syllable	[ʔi.lik.hin] vs. *[ʔil.k.hin] *[ʔilk.hin]

Since Yawelmani also has final vowel deletion, the transformational component would also have to contain an optional rule that deletes final vowels. As a result of this rule, the output of the transformational component will include both [ta.xakʔ] and *[ta.xaː.kʔa]. Since *[ta.xaː.kʔa] is ungrammatical, some filter must rule it out. What filter? The obvious move is to posit a filter that forbids word-final vowels. We can call this filter *V#.

When we try to extend this analysis to the input /xat-kʔa/, however, we run into trouble. (Follow along in (4).) Among the outputs of the transformational component are [xat.kʔa] (which is correct), *[xat.kʔ] (with an unsyllabified consonant), and *[xatkʔ] (with a syllable that is too big). Unfortunately, all of these forms, including the correct one, violate some filter. The forms *[xat.kʔ] and *[xatkʔ] are marked as ungrammatical because they violate the filters *C^{unsyll} and *Complex-Syllable, respectively. The form that we want, [xat.kʔa], is marked as ungrammatical by the filter *V#, which was needed to make final vowel deletion obligatory in [ta.xakʔ]. The only form that isn't marked as ungrammatical is *[xa.tikʔ], which is wrong.

(4) Filters model applied to Yawelmani – input /xat-kʔa/ → wrong output

Input	Transformational component (all optional)	Output of transformational component	Filter component	Output of filter component
/xat-kʔa/	syllabification epenthesis final vowel deletion	[xat.kʔa] [xat.kʔ] [xatkʔ] [xa.tikʔ]	*C^{unsyll} *Complex-Syllable *V#	[xa.tikʔ] *vs.* *[xat.kʔa] *[xat.kʔ] *[xatkʔ]

This analysis fails because it's based on a wrong assumption, the assumption that filters are never violated. If filters are inviolable constraints on outputs, Yawelmani cannot possibly have a filter *V# – obviously, since it has vowel-final words like [xat.kʔa]! We could get around this problem by replacing *V# with a more specific filter, *VCV#, but this would be admitting defeat. The filter *VCV# stipulates something that our analysis really should explain: final vowel deletion is blocked in [xat.kʔa] because letting it apply would produce an unsyllabified consonant or a syllable that is too big. If we haven't explained that, then we haven't really accounted for Yawelmani's conspiracy.

A real explanation needs to derive the failure of final vowel deletion in [xat.kʔa] from the independently necessary filters *C^{unsyll} and *Complex-Syllable. The idea goes something like this. Even though [xat.kʔa] violates *V#, the alternative *[xat.kʔ] is even worse, since it violates *C^{unsyll}, and *C^{unsyll} *has a higher priority than* **V#*. To say the

same thing in a different way, *V# triggers final vowel deletion, but the constraint *C^{unsyll} sometimes blocks satisfaction of *V#. The same goes for *COMPLEX-SYLLABLE: it too has higher priority than *V#, so it too can block satisfaction of *V#. (You will be dealing with the *[xa.tikʔ] problem in exercise 17 in chapter 2.)

Although constraint priority relationships were occasionally mentioned in the pre-OT literature (e.g., Burzio 1994), the standard assumption was that all output constraints are inviolable and therefore unprioritized. The central thesis of OT, on the other hand, is that constraints are ranked and violable. Constraint prioritization is fundamental to the theory (Prince and Smolensky 1993/2004). The comparison between [xat.kʔa] and *[xat.kʔ] reveals a type of *constraint conflict* between *V# and *C^{unsyll}: a form that obeys one violates the other (see (5)). If *V# takes precedence, then the result is *[xat.kʔ], which obeys *V# at the expense of violating *C^{unsyll}. If *C^{unsyll} takes precedence, then the result is [xat.kʔa], which obeys *C^{unsyll} but violates *V#. Since [xat.kʔa] is what we want, priority goes to *C^{unsyll}.

(5) Constraint conflict with /xat-kʔa/

	*C^{unsyll}	*V#
[xat.kʔa]	*obeyed*	*violated*
*[xat.kʔ]	*violated*	*obeyed*

In OT terms, the higher-priority constraint *dominates* the lower-priority constraint. *C^{unsyll} must dominate *V# in the grammar of Yawelmani. We write this as *C^{unsyll} >> *V#. *COMPLEX-SYLLABLE also dominates *V#. This means that *V# will be satisfied only when this doesn't require an output with an unsyllabified consonant or a too-big syllable. With the input /xat-kʔa/, these constraints impose conflicting demands and the higher-ranking ones are decisive, blocking vowel deletion. With the input /taxaː-kʔa/, however, the final vowel can be deleted with no danger of leaving a consonant unsyllabified or creating a syllable that is too big (see (6)). In this case, there is no conflict between *C^{unsyll} and *V#, so both of them can and must be satisfied. Constraints are violable in OT, but violation is never gratuitous; it must always be compelled by some higher-ranking, conflicting constraint.

(6) No constraint conflict with /taxaː-kˀa/

	*C^{unsyll}	*V#
[ta.xakˀ]	*obeyed*	*obeyed*
*[ta.xaː.kˀa]	*obeyed*	*violated*

The goal in this discussion of Yawelmani was to explain away a conspiracy by deriving the failure of final vowel deletion in VCCV# words from independently necessary constraints on syllabification. The OT analysis that I've just sketched does exactly that: there is no final vowel deletion in [xat.kˀa] because the alternatives, *[xat.kˀ] and *[xatkˀ], leave a consonant unsyllabified or require a syllable that exceeds the language's limit on size. The most important and novel elements of this explanation are constraint ranking and violability, which allow *V# to be active in Yawelmani but not always satisfied.

This seemingly modest change in how to think about output constraints is in reality quite profound, with important implications that are still being explored more than a decade later. In the rest of this chapter we will see some of those implications.

Syllable structure and phonological processes

One of the most important developments in phonology during the 1970s and 1980s was the realization that syllable structure affects many phonological processes. Vowel epenthesis, for example, is often motivated by the need to fit consonants into restrictive syllable templates. Yawelmani /ʔilk-hin/ → [ʔi.lik.hin] is an example of this; because of epenthetic [i], the [k] can fit into Yawelmani's maximally CVC syllables, whereas without the [i] it couldn't (*[ʔil.k.hin] or *[ʔilk.hin]). Syllable structure requirements can also block processes, such as final vowel deletion in Yawelmani /xat-kˀa/ → [xat.kˀa].

Syllable structure offered some help with the conspiracy problem, but not enough. Selkirk (1981) proposed to solve the problem of how epenthesis is triggered by assuming that the initial pass of syllabification is able to create "degenerate" syllables that lack a vowel nucleus: [ʔi.lΔk.hin], with Δ standing for an empty nucleus constituent in the second syllable. In this way, the language's syllable structure template determines where and when epenthetic vowels are required.

The epenthesis process itself is just a matter of spelling out the empty nucleus as [i].

There were intractable problems in trying to extend this sort of approach to blocking effects, however. The /xat-kʔa/ → [xat.kʔa] mapping tells us that final vowel deletion is blocked because its output cannot be exhaustively syllabified. But when final vowel deletion is applied to /ta.xaː.kʔa/, the immediate output is [ta.xaː.kʔ], which also cannot be exhaustively syllabified. Presumably the difference is that Yawelmani also has a process of closed syllable shortening that changes [ta.xaː.kʔ] into the final output [ta.xakʔ], which can be exhaustively syllabified. The derivation, then, is /ta.xaː.kʔa/ → [ta.xaː.kʔ] → [ta.xakʔ]. By the same logic, then, what's wrong with the derivation /xat-kʔa/ → [xat.kʔ] → *[xa.tikʔ], since Yawelmani also has a process of vowel epenthesis? Clearly, there were difficult problems in sorting out when languages block processes and when they allow them to apply but fix up the results. (See Myers (1991), Paradis (1988a, 1988b), and Prince and Smolensky (1993/2004: 238–257) for discussion of this and related issues.)

The importance of syllable structure in phonology continues to be recognized in most OT work. There is nothing in OT *per se*, however, that requires a commitment to any particular theory of syllable structure or even to the existence of syllables.

QUESTIONS

4 "[T]he standard assumption was that all output constraints are inviolable and therefore unprioritized." Why "therefore"? Explain the connection between constraint violability and constraint prioritization.

5 "*C^{unsyll} must dominate the constraint *V# in the grammar of Yawelmani. . . . Likewise *Complex-Syllable dominates *V#. This means that *V# will be satisfied only when this doesn't require an output with an unsyllabified consonant or a too-big syllable." Why does it mean that?

EXERCISE

6 The following Three Laws of Robotics are cited by Asimov (1950) from the *Handbook of Robotics* (56th edition, published 2058). Restate the laws as ranked constraints.

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.

1.3 The Nature of Constraints in OT

In OT, constraints on output forms are called *markedness constraints* to distinguish them from constraints of a very different sort, *faithfulness constraints*. Faithfulness constraints prohibit differences between input and output. When underlying /taxaːk^{ʔ}a/ maps to surface [ta.xak^{ʔ}], faithfulness constraints against vowel deletion and vowel shortening are violated. When underlying /ʔilk-hin/ maps to surface [ʔi.lik.hin], there is a violation of a different faithfulness constraint, one that prohibits vowel epenthesis.

Faithfulness constraints are one of Prince and Smolensky's cleverest and least obvious ideas. No other theory of language has anything like OT's faithfulness constraints. Faithfulness constraints only make sense in a theory like OT that allows constraints to be violated. The reason: phonology and syntax are full of examples of unfaithful mappings like /taxaːk^{ʔ}a/ → [ta.xak^{ʔ}] and /ʔilk-hin/ → [ʔi.lik.hin], so faithfulness constraints have to be violable if they are going to be at all useful.

The job of a constraint is to assign *violation marks* to candidates. (Violation marks are conventionally written as asterisks.) Depending on how the constraint is defined and what the candidate is, a constraint can assign any number of marks from zero upwards. For example, *V# assigns no marks to [ta.xak^{ʔ}], since [ta.xak^{ʔ}] ends in a consonant. It assigns one mark to *[ta.xaː.k^{ʔ}a], however, since *[ta.xaː.k^{ʔ}a] ends in a vowel. The anti-epenthesis faithfulness constraint assigns one violation mark for every epenthesized segment. This constraint is called DEP, because it requires the output to DEPend on the input as the source of all its segments.[3] As (7) shows, DEP assigns no violation marks to *[ʔil.k.hin], one mark to [ʔi.lik.hin], two marks to *[ʔi.li.ki.hin], three to *[ʔi.li.ki.hi.ni], and so on. Each constraint's definition tells us how to determine the number of violation marks that it assigns to a given candidate.

(7) Violation marks assigned by DEP

	DEP
a. ʔil.k.hin	
b. ʔi.lk.hin	*
c. ʔi.li.ki.hin	**
d. ʔi.li.ki.hi.ni	***

DEP *favors* *[ʔil.k.hin] over [ʔi.lik.hin], *[ʔi.li.ki.hin], *[ʔi.li.ki.hi.ni], and so on (Samek-Lodovici and Prince 1999). Furthermore, DEP favors [ʔi.lik.hin] over *[ʔi.li.ki.hin], *[ʔi.li.ki.hi.ni], and so on. Likewise, DEP favors *[ʔi.li.ki.hin] over *[ʔi.li.ki.hi.ni], and so on. These preferences are DEP's *favoring relation* over this set of candidates. If a constraint assigns *n* violation marks to some candidate, then it favors that candidate over all of the candidates to which it assigns more than *n* marks. The candidates that totally obey a constraint are just one aspect of the constraint's favoring relation. Because constraints are violable in OT, it often happens that all viable candidates violate some constraint. In that case, it's important to know which candidates the constraint favors among those that violate it. For example, *[ʔil.k.hin] is ruled out because of its unsyllabified [k], so violation of DEP is unavoidable. The form [ʔi.lik.hin] is optimal because it is most favored among the DEP-violating candidates, as we can see from (7).

In general, the candidates that are most favored by some constraint C have two things in common: they receive the same number of violation marks from C, and no other candidate receives fewer violation marks from C. There is always at least one candidate that is most favored by C. At the other extreme, it's possible for all of the candidates to be most favored by C, if all candidates violate C equally.

Constraints are a major focus of research effort in OT, and that is why this book devotes an entire chapter (chapter 4) just to the problems of discovering, defining, and improving constraints. Furthermore, as we will see in chapter 5, most explanations and predictions in OT derive from specific assumptions about which constraints exist. The activities of modifying or rejecting old constraints and positing new ones are important research tools and important responsibilities of researchers working in OT.

Although research on constraints is central to OT, OT itself does not say much about the nature of constraints, beyond distinguishing between markedness and faithfulness. OT is a theory of how constraints interact with one another; it isn't a theory of what the constraints are, nor is it a theory of representations. For example, OT does not commit the analyst to any particular approach to syllable structure or phrase structure. Instead, OT supplies a framework for applying the constraints and evaluating the representations that are a necessary part of any theory of syllable structure or phrase structure. This is the reason why it has been possible to apply OT to phonology, syntax, and semantics, despite their different subject matter.

Prince and Smolensky put forward two very strong hypotheses about the universality of constraints. First, the constraints themselves are universal. Universal Grammar (UG) includes a constraint component Con that contains the entire repertoire of constraints. (There are separate Cons for phonology and syntax, with some overlap in their formal properties.) Second, all constraints are present in the grammars of all languages. These hypotheses follow from the more general assumption that constraint ranking is the *only* systematic difference between languages. (More about this in the next section.)

In actual practice, the hypothesis of absolute constraint universality is usually somewhat weakened. It may be necessary to admit language-particular limitations on the domains of constraints in the lexicon to deal with exceptions, loan words, and the like. There may also be formal schemas for constructing language-particular constraints, such as alignment or constraint conjunction. I will say more about these issues in chapter 4.

QUESTIONS

7 Chomsky (1995: 380) says this: "In Prince and Smolensky 1993, there seems to be no barrier to the conclusion that all lexical inputs yield a single phonetic output, namely, whatever the optimal syllable might be (perhaps /ba/)." This is sometimes known as the "*ba* objection" to OT. Respond to it.

8 Chomsky (1995: 380) criticizes faithfulness constraints on the grounds that identity between input and output is "a principle that is virtually never satisfied." Respond to this criticism.

9 "Because constraints are violable in OT, it often happens that all viable candidates violate some constraint. In that case, it's important to know which candidates the constraint favors among those that violate it." In light of this

statement, would you describe the presentation of constraint conflict in (5) as somewhat misleading? How would you correct this?

1.4 Candidate Sets: OT's GEN Component

In Chomsky and Lasnik's filters model, the transformations are all optional, so the transformational component produces a variety of possible outputs in which transformations have and have not applied. The filter component marks some of these possible outputs as ungrammatical. In OT, the equivalent of the transformational component is called the *Generator*, or GEN for short. The list of possible outputs supplied by GEN for a given input is called the *candidate set* for that input. The relationship among the input, GEN, and the candidate set is diagrammed in (8).

(8) Partial flowchart for OT
/input/ → GEN → {$cand_1$, $cand_2$, . . . }

Details of the input and of GEN, like details of the constraints, depend on our theory of representations and whether we are analyzing phonology, syntax, or semantics.

In phonology, where there is the widest agreement on such matters, the input is usually taken to be identical with the underlying representation of generative phonology. This is a level of representation in which every morpheme that alternates regularly has a unique form, such as plural /-z/ in /bæg-z/, /bʊk-z/, and /noːz-z/ (*bags*, *books*, and *noses*). The phonological GEN performs various operations on the input, deleting segments, epenthesizing them, and changing their feature values. These operations apply freely, optionally, and repeatedly to derive the members of the candidate set. For example, the candidate set from input /bʊk-z/ will include the results of rightward and leftward voice assimilation ([bʊks], [bʊgz]), epenthesis ([bʊkəz]), deletion ([bʊk]), and various combinations of these processes (e.g., [bʊkəs]). It will also include a faithful candidate, where nothing has happened: [bʊkz]. These diverse candidates, nearly all of which are ungrammatical, aren't the final output of the grammar; the final output is determined by how the constraint component filters the candidate set.

Candidates *compete* to be realizations of some input. For example, [bʊks], [bʊgz], [bʊkəz], [bʊk], [bʊkəs], [bʊkz], etc. all compete to be

the surface realization of the input /bʊk-z/. Candidates from different inputs do not compete; there is no comparison of the mapping /bʊk-z/ → [bʊks] with the mapping /noːz-z/ → [noːzəz]. Therefore, GEN defines the range of competitors for a given input. This range must include at least all of the ways that the input could be realized in any possible human language. In phonology, the candidate set typically contains much more than that – perhaps even every possible sequence of segments. In syntax, the nature of the candidate set is more of an open question, though see §2.9 and Legendre, Smolensky, and Wilson (1998) for discussion of how to go about answering this question, starting from OT's basic premises about competition.

It makes sense to assume that the operations in GEN are extremely general. The epenthesis operation, for example, does not specify certain contexts for epenthesis or certain segments to be epenthesized. Instead, it can insert any segment in any context. Of course, there are all sorts of limits on what can be epenthesized and where epenthesis can happen in actual output forms. But GEN isn't the place to impose these limits. Instead, an important goal of research in OT is to derive the language-particular and universal properties of linguistic processes from a specific theory of CON and the assumption that grammars are rankings of CON. A similar goal was articulated for the filters model: to show "that the consequences of ordering, obligatoriness, and contextual dependency can be captured in terms of surface filters . . . and further, that these properties can be expressed in a natural way at this level" (Chomsky and Lasnik 1977: 433). Government-Binding theory (Chomsky 1981) was an attempt to follow through on this goal by reducing the transformational component of the grammar to a single optional context-free operation, Move α. This highly general transformation is in the same spirit as OT's GEN.

If GEN is so unrestricted in its effects, then the candidate set is infinite. There are infinitely many candidates if GEN includes context-free structure-building operations like epenthesis in phonology or phrase-structure projection in syntax. These operations are allowed to apply indefinitely many times in candidate formation. For example, the candidates based on the input /noːz-z/ will include not only [noːzəz] but also [noːzəəz], [noːzəəəz], and so on.

The diversity and infinity of candidates is a source of worry to many people when they first encounter OT, and I will try to lay these worries to rest now.

The diversity of candidates can be troubling because it means that any candidate set will include forms that couldn't possibly be the output in any language. Presumably, no human language could possibly map underlying /noːz-z/ to surface [noːzəəəz]. But if [noːzəəəz] is never optimal, what is it doing in /noːz-z/'s candidate set? The answer to this worry is that the output of GEN isn't the final output of the grammar. The grammar as a whole does not overgenerate because the constraints filter the contents of the candidate set. Any decent theory of CON will explain why mappings like /noːz-z/ → [noːzəəəz] are impossible. That is where such explanations belong, in accordance with the overall goals of OT research that were discussed a couple of paragraphs above. This matter is the topic of chapter 5.

Another source of worry is mental or electronic computation: GEN will require infinite time to produce a candidate set, and the constraint component will require infinite time to evaluate the candidates. This worry starts from a wrong assumption: the formal definition of a theory of language is also its computational implementation. Since the very beginning, generative grammar has made a distinction between models of language competence and models of language processing or use. "If these simple distinctions are overlooked, great confusion must result," according to Chomsky (1968: 117). There is a lot of good work on computational modeling of OT, and none of this work stumbles over the infinity of candidates because all of it recognizes the distinction between theory and implementation. See the suggestions for further reading at the end of the chapter.

QUESTIONS

10 "Any decent theory of CON will explain why mappings like /noːz-z/ → [noːzəəəz] are impossible." How? [Hint: Think about markedness, since for [noːzəəəz] to win it must be less marked than its more faithful competitors [noːzz], [noːzəz], and [noːzəəz].]

11 Why not put a limit on the number of epenthesis operations that GEN can perform? Would this ensure that the phonological candidate set is finite, or does the phonological GEN include other potential sources of an infinity of candidates?

12 What are some hypotheses about the input in syntactic theory? How would we go about determining which hypothesis is best?

1.5 Candidate Evaluation: OT's EVAL Component

GEN produces a candidate set from an input, and that candidate set is submitted to OT's other main component, the *Evaluator*, or EVAL for short. The complete OT flowchart is given in (9). EVAL's job is to find the *optimal* candidate. EVAL does this by applying a language-particular constraint hierarchy to the set of candidates.

(9) Flowchart for OT
/input/ → GEN → {$cand_1$, $cand_2$, . . . } → EVAL → [output]

Since EVAL is so important in OT, I will describe it in a couple of different ways, first in formal terms and then in a more procedural fashion. (The procedural description is just an alternative way of thinking about the formalization. As I noted at the end of the previous section, this isn't a claim about some actual process of mental or electronic computation.)

The formal description of EVAL starts from the observation that any constraint can be defined as a function from a set of candidates {*cands*} to some subset of {*cands*} – specifically, to the subset consisting of those candidates that the constraint most favors. Then EVAL is the function defined by composing all of the constraints in the order in which they are ranked (Karttunen 1998, Samek-Lodovici and Prince 1999). For instance, the constraint hierarchy $*C^{unsyll}$ >> DEP in functional form looks like DEP($*C^{unsyll}$({*cands*})) or, in the other notation for function composition, DEP ∘ $*C^{unsyll}$({*cands*}).

In more procedural terms, EVAL starts with the constraint that is ranked highest, CONST1, and extracts the subset of {*cands*} that is most favored by CONST1. This subset is passed along to the next constraint in the ranking, CONST2, which does the same thing: it locates the subset of candidates that it most favors and discards the rest. This process continues until the set has been reduced to just one candidate. This is the optimal candidate. It does better on the constraints as ranked than any other candidate in the original candidate set.

The workings of EVAL are illustrated in (10). To keep things simple, we start with the assumptions in (a) about the candidate set and the constraints that evaluate it. In (b), the top-ranked constraint $*C^{unsyll}$ is applied. It favors three of the candidates over the fourth. Those favored candidates are kept, and the disfavored one is discarded. In (c), this set of three candidates is submitted to the next constraint in the ranking, DEP. It favors one of the candidates over the other two.

Since we have now reduced the candidate set to just one candidate, we have found the optimal candidate. This is the output of the grammar.

(10) EVAL at work
   a. Assume:
      Candidate set = {[ʔil.k.hin], [ʔi.lik.hin], [ʔi.li.ki.hin], [ʔi.li.ki.hi.ni]}
      Constraint hierarchy = *C^{unsyll} >> DEP
   b. Apply *C^{unsyll}
      Favors {[ʔi.lik.hin], [ʔi.li.ki.hin], [ʔi.li.ki.hi.ni]} (no marks) over {[ʔil.k.hin]} (one mark).
   c. Apply DEP
      Favors {[ʔi.lik.hin]} (one mark) over {[ʔi.li.ki.hin]} (two marks) and {[ʔi.li.ki.hi.ni]} (three marks).
   d. Output = [ʔi.lik.hin]

In theory, EVAL could run out of constraints before the candidate set has been reduced to a single member. This can only happen if two or more candidates receive exactly the same number of violation marks from all of the constraints. In other words, there is a tie. This kind of tie has occasionally been used to account for language variation or optionality, but often it's unwelcome and requires an additional constraint. (See §2.4 on the resolution of ties and §6.2 on analyzing variation in OT.)

To return to a point made earlier, EVAL never looks for candidates that *obey* a constraint; it only asks for candidates that *are most favored by* a constraint. Being favored by a constraint isn't the same as obeying it. One or more candidates are always favored, but it will sometimes happen that no candidate obeys a given constraint. As a result, there is always some optimal candidate (unless, absurdly, the initial candidate set is empty).

From the perspective of other linguistic theories, this is probably the most surprising thing about EVAL. EVAL maps every input to some output. In other theories, some inputs have no well-formed output because of inviolable constraints. In those other theories, for example, inviolable constraints mark *[bnæg] and **Who did he say that left?* as ungrammatical in English. Since OT has only violable constraints, how can it account for ungrammaticality?

In OT, a candidate's ungrammaticality is a consequence of its inferiority to other candidates rather than violating an inviolable constraint.

For instance, *[bnæg] isn't a possible word of English because the phonological grammar of English does not select *[bnæg] as the optimal candidate for any input. To show this, we naturally want to look at the input /bnæg/. Since every faithfulness constraint favors the mapping /bnæg/ → *[bnæg], some higher-ranking markedness constraint must rule it out. This constraint is perhaps a prohibition on onset clusters containing two (nasal or oral) stops. If this constraint dominates DEP, then EVAL will select [bənæg] rather than *[bnæg] as the output for the input /bnæg/. ([bənæg] isn't a real word of English, but unlike *[bnæg] it's pronounceable, and that is the point of the example.) This isn't quite enough to guarantee *[bnæg]'s ungrammaticality, however; that requires showing that *[bnæg] isn't optimal for *any* input. It's similar to studying language typology (see chapter 5).

This discussion of ungrammaticality in OT emphasizes a key point about this theory: it's *inherently comparative*. No candidate is good or bad in itself; it's only good or bad in relation to other candidates from the same input. A candidate set defines the limits of the comparison. Every member of a candidate set is in competition with every other member to be the output realization of that candidate set's input. For this reason, when we construct analyses we need to be sure to consider candidates that might give the desired winner some serious competition. For instance, it would be wrong to neglect candidates with final consonant epenthesis (*[ta.xaː.kˀaʔ], *[xat.kˀaʔ]) as competing ways of satisfying *V# in Yawelmani. I will have more to say about this important point in §2.5.

Some final remarks on terminology. Sometimes, we will need to say that one candidate is better than another without necessarily asserting that the better candidate is optimal. The phrase "*cand1* is more optimal than *cand2*" is very awkward; it's better to say that *cand1* is *more harmonic* than *cand2*. *Harmony* is the property that EVAL selects for. If *cand1* is more harmonic than *cand2*, then the highest ranking constraint that distinguishes between *cand1* and *cand2* is a constraint that favors *cand1*. The expressions *optimal* and *most harmonic* mean exactly the same thing when the full candidate set is under discussion.

QUESTION

13 "In theory, EVAL could run out of constraints before the candidate set has been reduced to a single member. This can only happen if two or more candidates receive exactly the same number of violation marks from all of the constraints. In other words, there is a tie. This kind of tie has occasionally been

used to account for language variation or optionality . . ." This approach to variation in OT is almost never used because it almost never produces multiple winning candidates. Why is that? (Hint: Think about the potential effects of low-ranking constraints.)

1.6 Constraint Activity

A constraint is *active* on some candidate set if it's the highest-ranking constraint that favors the winner over some loser. In other words, an active constraint knocks some loser out of the competition, accomplishing something that no higher-ranking constraint has managed to do.

For example, the constraint *C^{unsyll} is active in the /xat-kʔa/ → [xat.kʔa] mapping because it favors the winner [xat.kʔa] over the loser *[xat.kʔ], and no higher-ranking constraint does the same thing. (In fact, there is no constraint ranked higher than *C^{unsyll}.) In (11), the active role of *C^{unsyll} is signaled by adding "!" next to the violation mark that it assigns to *[xat.kʔ]. This is sometimes referred to as a *fatal violation*, since it knocks a candidate out of the competition for optimality.

(11) Active *C^{unsyll}

	*C^{unsyll}
a. → xat.kʔa	
b. xat.kʔ	*!

The constraint *V# is active in the /taxaː-kʔa/ → [ta.xakʔ] mapping because it favors the winner [ta.xakʔ] over the loser *[ta.xaː.kʔa] (see (12)). There is a higher-ranking constraint, *C^{unsyll}, but it isn't active on this pair of candidates.

(12) Active *V#

	*C^{unsyll}	*V#
a. → ta.xakʔ		
b. ta.xaː.kʔa		*!

On the other hand, *V# isn't active in the choice between [xat.kʔa] and *[xat.kʔ], since higher-ranking *C^{unsyll} does deprive *V# of the chance to be active in this evaluation (see (13)). Lower-ranking constraints are potentially active only when the winner and one or more losers tie on all of the higher-ranking constraints.

(13) Active *C^{unsyll}, but inactive*V#

	*C^{unsyll}	*V#
a. → xat.kʔa		*
b. xat.kʔ	*!	

A constraint can still be active even when the winner violates it. In Yawelmani, *C^{unsyll} has to dominate DEP to account for epenthesis in /ʔilk-hin/ → [ʔi.lik.hin] (vs. *[ʔil.k.hin]). As (14) shows, the optimal candidate violates DEP once, but losers like *[ʔi.li.ki.hin] and *[ʔi.li.ki.hi.ni] violate it even more. When candidates violate a constraint by different amounts, the severity of the violation matters, and the constraint favors the candidate that violates it the least.

(14) Active but violated DEP

	*C^{unsyll}	DEP
a. → ʔi.lik.hin		*
b. ʔil.k.hin	*!	
c. ʔi.li.ki.hin		**
d. ʔi.li.ki.hi.ni		***!

Example (14) illustrates a property of EVAL called *minimal violation*. Although the winner violates DEP, it violates DEP less than any other candidate except the one ruled out by higher-ranking *C^{unsyll}. Constraints are violable in OT, but violation is minimal.

Example (14) also shows that minimal violation of faithfulness constraints produces a kind of economy of derivation, in something like Chomsky's (1991) sense. Because faithfulness constraints are violated

minimally, the winning output candidate can differ from the input only as much as necessary to do better on any higher-ranking constraints. With the input /ʔilk-hin/, DEP must be violated in order to satisfy *C^{unsyll}, so some discrepancy between input and output is unavoidable. But the discrepancy is still minimal because DEP is violated minimally.

Markedness constraints can also be active when they are dominated. Some observations about syllable structure illustrate this. The markedness constraint ONSET is violated by onsetless (i.e., vowel-initial) syllables (Ito 1989: 222 and others). In the Malaysian Austronesian language Timugon Murut, ONSET must be crucially dominated because onsetless syllables occur in surface forms, such as [am.bi.lu.o] 'soul'. (The [u] and [o] are in "two distinct phonetic syllables," according to Prentice (1971: 24).) Onsetless syllables could be avoided by epenthesizing a consonant, as in *[ʔam.bi.lu.ʔo], so DEP has to be ranked above ONSET to prevent this from happening (see (15)). And since onsetless syllable could also be avoided by deleting the problematic segments ((c) in (15)), ONSET has to be dominated by the anti-deletion faithfulness constraint MAX. (It is called MAX because it requires the input segments to be MAXimally expressed in the output.[4])

(15) Active but violated ONSET

	DEP	MAX	ONSET
a. → am.bi.lu.o			**
b. ʔam.bi.lu.ʔo	**!		
c. bi.lu		***!	
d. am.bil.u.o			***!

Now look at candidate (d) in (15). Because of how [l] is syllabified, this candidate has one more onsetless syllable than the winner has, and so it's disfavored by ONSET. Even though the winner violates ONSET, this constraint still actively eliminates candidate (d). When a markedness constraint is active in a language but also violated by some winners in that language, the situation is known as *the emergence of the unmarked*, sometimes abbreviated TETU (McCarthy and Prince 1994a). The idea is that a preference for some universally unmarked structure, such as syllables with onsets, can emerge under the right circumstances even if the language as a whole permits the corresponding

marked structure. Candidate (d) loses because ONSET emerges to disfavor it, even though ONSET is violated elsewhere in the language (and even in this very word). Emergence of the unmarked is an important difference between OT and parametric theories of language, as we will see in §1.7.

The idea that markedness constraints can be active but violated is hard to absorb and exploit fully. When I first learned about OT, I brought with me the belief that legitimate linguistic constraints had to state absolute truths about surface forms. I was uncomfortable with saying that ONSET actively favors [am.bi.lu.o] over *[am.bil.u.o]. I would have been happier with a specific constraint against, say, *[VC.V] syllabification, where a syllable-final consonant is followed by syllable-initial vowel. This constraint is categorically true in Timugon Murut, but only because it stipulates additional conditions that allow it to be categorically true. (In that respect, it's like the rejected constraint *VCV# in Yawelmani.)

It requires some effort to get past these prejudices inherited from other theories. The best practice in OT is to state constraints in very general ways and then try to limit their activity through interaction with higher-ranking constraints. Formulating constraints that refer to highly specific surface configurations, such as *[VC.V], isn't a very successful analytic strategy in OT.

QUESTIONS

14 Explain how the minimal violation property follows from the definition of EVAL in §1.5.

15 The emergence of the unmarked is relevant to the choice of *which* segment to epenthesize when other constraints have determined that *some* segment must be epenthesized. Can you figure out why?

EXERCISES

16 From the information given in this section, can you determine the relative ranking of MAX and DEP in Timugon Murut? If so, what is the ranking? If not, what sort of additional evidence would you need?

17 Imagine you have joined an internet dating site. To find your compatible mate, you are required to rank five desirable qualities in a mate according to the importance you place on them. The qualities are physical attractiveness,

intelligence, sense of humor, good hygiene, and wealth. How would you go about figuring out your personal priority system for these attributes using OT style ranking methods? Could you have a problem determining the relative priority of good hygiene and wealth if all of the wealthy people you know also practice good hygiene?

1.7 Differences between Languages

Different languages have different rankings of CON. In Timugon Murut, DEP and MAX dominate ONSET, so there are onsetless syllables. In Arabic, ONSET dominates DEP, so a consonant is epenthesized: /al-walad/ → [ʔal.wa.lad], *[al.wa.lad] 'the boy'.

The strongest hypothesis is that constraint ranking is the *only* way that languages differ. In other words, all systematic differences between languages should be accounted for by permuting the ranking of a set of universal constraints. This hypothesis means, among other things, that every constraint in CON is in the grammar of every language. Even when a language seems to completely ignore some constraint C, C remains in the language's constraint hierarchy. In this situation, C is inactive because of other constraints that dominate it and not because it has been removed from the grammar.

In other linguistic theories, differences between languages are often attributed to *parameters*. A parameter is a constraint that can be turned off. For instance, the [Onset] parameter would be turned off in Timugon Murut, which allows onsetless syllables, and turned on in Arabic, which forbids them. Parametric theories have problems with emergence of the unmarked effects. If [Onset] is off in Timugon Murut, then why is [am.bi.lu.o] preferred to *[am.bil.u.o] and *[amb.il.u.o]? In pre-OT days, Ito (1989: 223) addressed this problem by parameterizing [Onset] as strong/weak rather than on/off. [Strong Onset] says "Onsetless syllables are forbidden." [Weak Onset] says "Avoid onsetless syllables." The word "avoid" tells us that [Weak Onset] is really just a version of [Strong Onset] that can be violated minimally. In OT, minimal violation is a general property of all constraints, so it isn't necessary to build it into the definition of this or any other specific constraint.

Language differences will be a particular focus of our attention in chapter 5. Chapters 2 and 4 lay the foundation for studying this important topic.

QUESTION

18 What would it take to prove that some markedness constraint was literally absent from the grammars of some languages, rather than merely low-ranking? When answering this question, feel free to make any necessary assumptions about the other constraints in CON.

EXERCISE

19 Show that even low-ranking faithfulness constraints are universally present in the grammars of all languages. The material in §1.6 offers a hint about how to make this argument.

1.8 The Version of OT Discussed in This Book

In this and subsequent chapters, I am describing a version of OT that can be called "standard" or "classic." Standard or classic OT incorporates almost all of Prince and Smolensky's (1993/2004) main ideas. There is only one systematic difference between this standard theory and what Prince and Smolensky say: how faithfulness is implemented. The standard theory formulates faithfulness constraints like MAX and DEP using correspondence theory (McCarthy and Prince 1995, 1999). These constraints have replaced Prince and Smolensky's original faithfulness constraints PARSE and FILL, which were formulated somewhat differently. (Correspondence theory, PARSE, and FILL will be explained in §4.6.)

As I noted in §1.3, OT itself does not say anything specific about the constraints in CON, particularly the markedness constraints. Markedness constraints embody substantive claims about phonology, syntax, or some other linguistic domain. OT is a formal system in which notions like constraint priority are rigorously defined, but it does not say what the constraints are. Likewise, OT itself does not say anything about the nature of representations, though it provides a framework in which the well-formedness of representations can be evaluated using violable constraints.

Because OT itself does not specify what the constraints are, research in OT is primarily focused on developing and improving hypotheses about the constraints in CON in order to understand and eventually solve specific empirical problems. Exploring the results of ranking permutation, improving or rejecting old constraints, and positing new constraints are familiar activities to anyone working in this theory.

This book, particularly in chapters 4 and 5, offers plenty of guidance about how to do these things with maximal effectiveness.

Another type of OT research explores the effects of various possible changes in OT's basic assumptions. What if OT had derivations? Can a language have more than one constraint ranking? Work that addresses questions like these will be introduced in chapter 6, along with pointers to the literature.

A third type of research deals in formal analysis of OT, including learnability, logic, and computation. Some of this work is discussed in §2.11 and §2.12.

1.9 Suggestions for Further Reading

Among the article-length overviews of OT are Archangeli (1997), Legendre (2001), McCarthy (2003b, 2007c), Prince and Smolensky (1997, 2003), Smolensky, Legendre, and Tesar (2006), and Tesar, Grimshaw, and Prince (1999). Kager (1999) is a textbook that focuses on applications of OT to several phonological phenomena: syllabification, stress, reduplication, and cyclicity. Yip (2002) is a textbook about tone with information about how OT can be applied to tonal phenomena. McCarthy (2002) is a guide to OT's main concepts and the results that follow from them. It also includes an extensive bibliography, with references organized by topics at the end of each chapter.

Anyone who works through *Doing Optimality Theory* is ready for more advanced reading, starting with Prince and Smolensky (1993/2004). The next step after that depends on the individual reader's interests. If they tend toward phonology, then the papers collected in McCarthy (2003a) are probably the best place to start. Two other useful anthologies, Lombardi (2001) and Féry and van de Vijver (2003), are focused on segmental and syllabic phonology, respectively. Readers of a syntactic bent could not do better than to consult two anthologies of papers on OT syntax, Legendre, Grimshaw, and Vikner (2001) and Sells et al. (2001). In addition, there are now several anthologies on OT semantics and pragmatics (Blutner et al. 2005, Blutner and Zeevat 2004, de Hoop and de Swart 1999), and one on historical linguistics (Holt 2003). The roots of OT in cognitive science, as well as applications to phonology, syntax, and other areas, are the topic of another anthology, Smolensky and Legendre (2006).

Some of the most important work on OT is available for free on the Rutgers Optimality Archive (http://roa.rutgers.edu). ROA, which

was created by Alan Prince in 1993, is an electronic repository of "work in, on, or about OT." It's a fabulous resource for the student as well as the veteran scholar. To find ROA papers on specific topics, you can use ROA's built-in function for searching abstracts, but it's better to use Google, which searches the body of papers as well. Use the Google directive *site:roa.rutgers.edu* in the search string – e.g., *metathesis site:roa.rutgers.edu* will locate all of the ROA postings that mention metathesis anywhere in the text.

Notes

1 Nowadays, the preferred name for this Yokuts dialect is Yowlumne. I retain the earlier name since it is much more familiar to most linguists.
2 According to Newman (1944: 29) and most subsequent analysts, final vowel deletion is limited to CV suffixes like /-kʔa/ and /-mi/. I believe it is more accurate to say that overt alternations are limited to these suffixes, since longer or shorter suffixes do not present opportunities for alternations.
3 Kathryn Flack informs me that "DON'T EPENTHESIZE" is in use as a mnemonic for DEP.
4 A somewhat forced mnemonic for MAX: "MAKE expressed."