Part I

The Basics

Chapter 1 | Alan Prince and Paul Smolensky

Optimality Theory: Constraint Interaction in Generative Grammar

Editor's Note

Optimality Theory first gained wide exposure from a course taught by Prince and Smolensky at the 1991 Summer Institute of the Linguistic Society of America. The earliest and still the most detailed exposition of the theory is their 1993 manuscript, an excerpt from which is here published for the first time. There has been much interest in this emerging theory; it has been the subject of a large and growing professional literature, an extensive electronic archive (http://roa.rutgers.edu), many courses and conference papers, and several textbooks. Although it was originally applied to phonology, the relevance of OT to topics in phonetics, morphology, syntax, sociolinguistics, psycholinguistics, and semantics has become increasingly apparent.

This chapter includes these excerpts: introductory material and motivation for the theory, including an analysis of Berber syllabification, drawn from sections 1 and 2 of Prince and Smolensky (P&S) (1993); an explanation of how constraints and constraint hierarchies evaluate candidates (section 5 of P&S 1993); the basic CV syllable theory with elaborations (section 6 and part of section 8 in P&S 1993); the theory of inventories and the lexicon (most of section 9 in P&S 1993). Readers may encounter sporadic references to other parts of P&S (1993): sections 3 and 4 on blocking and triggering (exemplified with Tongan stress, Tagalog infixation, Hindi stress, and Latin foot and word structure); section 7 on Lardil phonology; and section 10 on OT's relationships with functionalism, computation, Connectionism, Harmony Theory, and constraint-and-repair theories.

Readers approaching OT for the first time should begin with sections 1.2 and 2 of this chapter, followed by section 6, and then section 5. Readers can then go on to read the other parts of this chapter or other chapters in this book. Some natural pairings: the constraint HNUC in section 2 of this chapter re-emerges in stress theory in chapter 9; the CV syllable theory in section 6 of this chapter is studied from the perspectives of parsing and learning in chapters 4 and 5, respectively; the idea of faithfulness constraints (section 6.2.1) is generalized in chapter 3; emergence of

Excerpt (with minor revisions by the authors) from:
Prince, Alan and Smolensky, Paul (1993) *Optimality Theory: Constraint Interaction in Generative Grammar*. New Brunswick, NJ: Rutgers University Center for Cognitive Science. Technical report RuCCS-TR-2. [Available on Rutgers Optimality Archive, ROA-537.]

the unmarked is discussed briefly at the end of section 6.1 in this chapter and is the subject of chapter 26; lexicon optimization, which is discussed in section 9.3 of this chapter, is the topic of chapter 32.

[. . .]

1.2 Optimality

The standard phonological rule aims to encode grammatical generalizations in this format:

(1) A → B / C—D

The rule scans potential inputs for structures CAD and performs the change on them that is explicitly spelled out in the rule: the unit denoted by A takes on property B. For this format to be worth pursuing, there must be an interesting theory which defines the class of possible predicates CAD (Structural Descriptions) and another theory which defines the class of possible operations A → B (Structural Changes). If these theories are loose and uninformative, as indeed they have proved to be in reality, we must entertain one of two conclusions:

(i) phonology itself simply doesn't have much content, is mostly 'periphery' rather than 'core', is just a technique for data-compression, with aspirations to depth subverted by the inevitable idiosyncrasies of history and lexicon; or
(ii) the locus of explanatory action is elsewhere.

We suspect the latter.

The explanatory burden can of course be distributed quite differently than in the re-write rule theory. Suppose that the input–output relation is governed by conditions on the well-formedness of the *output*, 'markedness constraints', and by conditions asking for the *exact preservation of the input* in the output along various dimensions, 'faithfulness constraints'. In this case, the inputs falling under the influence of a constraint need share no input-specifiable structure (CAD), nor need there be a single determinate transformation (A→B) that affects them. Rather, we generate (or admit) a set of candidate outputs, perhaps by very general conditions indeed, and then we assess the candidates, seeking the one that best satisfies the relevant constraints. Many possibilities are open to contemplation, but some well-defined measure of value excludes all but the best.[1] The process can be schematically represented like this [the function H-eval, 'Harmonic Evaluation', determines the relative Harmony of the candidates]:

(2) Structure of Optimality-theoretic Grammar
a. Gen (In_k) → $\{Out_1, Out_2, \ldots\}$
b. H-eval (Out_i, $1 \leq i \leq \infty$) → Out_{real}

The grammar must define a pairing of underlying and surface forms, (input$_i$, output$_j$). Each input is associated with a candidate set of possible analyses by the function Gen (short for 'generator'), a fixed part of Universal Grammar. In the rich representational system employed below, an output form retains its input as a subrepresentation, so that departures from faithfulness may be detected by scrutiny of output forms alone. A 'candidate' is an input–output pair, here formally encoded in what is called 'Out$_i$' in (2).

Gen contains information about the representational primitives and their universally irrevocable relations: for example, that the node σ may dominate a node *Onset* or a node μ (implementing some theory of syllable structure), but never vice versa. Gen will also determine such matters as whether every segment must be syllabified – we assume not, below, following McCarthy 1979 and others – and whether every node of syllable structure must dominate segmental material – again, we will assume not, following Itô 1986, 1989.

The function H-eval determines the relative Harmony of the candidates, imposing an order on the entire set. An optimal output is at the top of the harmonic order on the candidate set; by definition, it best satisfies the constraint system. Though Gen has a role to play, the burden of explanation falls principally on the function H-eval, a construction built from well-formedness constraints, and the account of interlinguistic differences is entirely tied to the different ways the constraint-system H-eval can be put together, given UG.

H-eval must be constructible in a general way if the theory is to be worth pursuing. There are really two notions of generality involved here: general with respect to UG, and therefore cross-linguistically; and general with respect to the language at hand, and therefore across constructions, categories, descriptive generalizations, etc. These are logically independent, and success along either dimension of generality would count as an argument in favor of the optimality approach. But the strongest argument, the one that is most consonant with the work in the area, and the one that will be pursued here, broaches the distinction, seeking a formulation of H-eval that is built from maximally universal constraints which apply with maximal breadth over an entire language.

Optimality Theory, in common with much recent work, shifts the burden from the theory of operations (Gen) to the theory of well-formedness (H-eval). To the degree that the theory of well-formedness can be put generally, the theory will fulfill the basic goals of generative grammar. To the extent that operation-based theories cannot be so put, they must be rejected.

Among possible developments of the optimality idea, we need to distinguish some basic architectural variants. Perhaps nearest to the familiar derivational conceptions of grammar is what we might call 'harmonic serialism', by which Gen provides a set of candidate analyses for an input, which are harmonically evaluated; the optimal form is then fed back into Gen, which produces another set of analyses, which are then evaluated; and so on until no further improvement in representational Harmony is possible. Here Gen might mean: 'do any *one* thing: advance all candidates which differ in one respect from the input.' The Gen ⇆ H-eval loop would iterate until there was nothing left to be done or, better, until nothing that could be done would result in increased Harmony. A significant proposal of roughly this character is the *Theory of Constraints and Repair Strategies* of Paradis 1988a, 1988b, with a couple of caveats: the *constraints* involved are a set of parochial level-true phonotactic statements, rather than being universal and violable, as we insist; and the *repair strategies* are quite narrowly defined in terms of structural description and structural change rather than being of the 'do-unto-α' variety. A key aspect of Paradis's work is that it confronts the problem of well-definition of the notion 'repair': what to do when applying a repair strategy to satisfy one constraint results in violation of another constraint (at an intermediate level of derivation). Paradis refers to such situations as 'constraint conflicts' and although these are not conflicts in our sense of the term – they cannot be, since all of her constraints are surface- or level-true and therefore never disagree among themselves in the assessment of output well-formedness – her work is of unique importance in addressing and shedding light on fundamental complexities in the idea of wellformedness-driven rule-application. The 'persistent rule' theory of Myers 1991 can similarly be related to the notion of Harmony-governed serialism. The program for *Harmonic Phonology* in Goldsmith 1991, 1993 is even more strongly of this character; within its lexical levels, all rules are constrained to apply harmonically. Here again, however, the rules are conceived of as being pretty much of the familiar sort, *triggered* if they increase Harmony, and Harmony itself is to be defined in specifically phonotactic terms. A subtheory which is very much in the mold of harmonic serialism, using a general procedure to produce candidates, is the 'Move-x' theory of rhythmic adjustment (Prince 1983, Hayes 1991).[2]

A contrasting view would hold that the Input → Output map has no internal structure: all possible variants are produced by Gen in one step and evaluated in parallel. In the course of this paper, we will see instances of both kinds of analysis, though we will focus predominantly on developing the parallel idea, finding strong support for it, as do McCarthy & Prince 1993. Definitive adjudication between parallel and serial conceptions, not to mention hybrids of various kinds, is a challenge of considerable subtlety, as indeed the debate over the necessity of serial Move-α illustrates plentifully (e.g., Aoun 1986, Browning 1991, Chomsky 1981), and the matter can be sensibly addressed only after much well-founded analytical work and theoretical exploration.

Optimality Theory abandons two key presuppositions of earlier work. First, that it is possible for a grammar to narrowly and parochially specify the Structural Description and Structural Change of rules. In place of this is Gen, which generates for any given input a large space of candidate analyses by freely exercising the basic structural resources of the representational theory. The idea is that the desired output lies somewhere in this space, and the constraint system of the grammar is strong enough to find it. Second, Optimality Theory abandons the widely held view that constraints are language-particular statements of phonotactic truth. In its place is the assertion that constraints are essentially universal and of very general formulation, with great potential for disagreement over the well-formedness of analyses; an individual grammar consists of a ranking of these constraints, which resolves any conflict in favor of the higher-ranked constraint. The constraints provided by Universal Grammar are simple and general; interlinguistic differences arise from the permutations of constraint-ranking; typology is the study of the range of systems that re-ranking permits. Because they are ranked, constraints are regularly violated in the grammatical forms of a language. Violability has significant consequences not only for the mechanics of description, but also for the process of theory construction:

a new class of predicates becomes usable in the formal theory, with a concomitant shift in what we can think the actual generalizations are. We cannot expect the world to stay the same when we change our way of describing it.
[. . .]

2 Optimality in Grammar: Core Syllabification in Imdlawn Tashlhiyt Berber

Here we argue that certain grammatical processes can only be properly understood as selecting the *optimal output* from among a set of possibilities, where the notion *optimal* is defined in terms of the constraints bearing on the grammatical domain at issue.

2.1 The heart of Dell & Elmedlaoui

The Imdlawn Tashlhiyt dialect of Berber (ITB) has been the object of a series of remarkable studies by François Dell and Mohamed Elmedlaoui (Dell & Elmedlaoui 1985, 1988, 1989). Perhaps their most surprising empirical finding is that in this language any segment – consonant or vowel, obstruent or sonorant – can form the nucleus of a syllable. One regularly encounters syllables of the shape *tK*, *rB*, *xZ*, *wL*, for example. (Capitalization represents nucleus-hood of consonants.) Table 1 provides illustrative examples, with periods used to mark syllable edges.[3]

Table 1

Nucleus type	Example	Morphology	Reference
voiceless stop	.ra.t**K**.ti.	ra-t-kti	1985: 113
voiced stop	.b**D**.dL. .ma.ra.t**G**t.	bddl ma=ra-t-g-t	1988: 1 1985: 113
voiceless fricative	.t**F**.tKt. .t**X**.zNt.	t-ftk-t t-xzn-t	1985: 113 1985: 106
voiced fricative	.tx**Z**.nakk^{w}.	t-xzn#nakk^{w}	1985: 113
nasal	.tz**M**t. .t**M**.zħ.	t-zmt t-mzħ	1985: 112 1985: 112
liquid	.t**R**.gLt.	t-rgl-t	1985: 106
high vowel	.il.d**i**. .rat.l**u**lt.	i-ldi ra-t-lul-t	1985: 106 1985: 108
low vowel	.tR.b**a**.	t-rba	1985: 106

Dell and Elmedlaoui marshall a compelling range of evidence in support of the claimed patterns of syllabification. In addition to native speaker intuition, they adduce effects from segmental phonology (emphasis spread), intonation, versification practice, and prosodic morphology, all of which agree in respecting their syllabic analysis.

The domain of syllabification is the phonological phrase. All syllables must have onsets except when they occur in absolute phrase-initial position. There, syllables may begin with vowels, either with or without glottal striction (Dell & Elmedlaoui 1985: 127 fn. 20), evidently a matter of phonetic implementation. Since any segment at all can form the nucleus of a syllable, there is massive potential ambiguity in syllabification, and even when the onset requirement is satisfied, a number of distinct syllabifications will often be potentially available. But the actual syllabification of any given string is almost always unique. Dell & Elmedlaoui discovered that assignment of nuclear status is determined by the relative sonority of the elements in the string. Thus we find the following typical contrasts:

(3) Sonority Effects on Nuclear Status
 a. tz**M**t — *tZmt ‘*m* beats *z* as a nucleus’
 b. rat.l**u**lt — *ra.tL.wL.t ‘*u* beats *l* as a nucleus’

Orthography: we write *u* for the nuclear version, *w* for the marginal version of the high back vocoid, and similarly for *i* and *y*: as with every other margin/nucleus pair, we assume featural identity.

All the structures in (3), including the ill-formed ones, are locally well-formed, composed of licit substructures. In particular, there is nothing wrong with syllables *tZ*, *tL*, or *wL* nor with word-final sequences *mt* – but the more sonorous nucleus is chosen in each case. By examining the full range of such contrasts, Dell and Elmedlaoui establish the relevance of the following familiar kind of 8-point hierarchy:

(4) Sonority Scale
 |Low V| > |High V| > |Liquid| > |Nasal| > |Voiced Fric.| > |Voiceless Fric.| > |Voiced Stop| > |Voiceless Stop|

We write $|\alpha|$ for the sonority or intrinsic prominence of α.

With the sonority scale in hand, Dell and Elmedlaoui then propose an iterative syllable-construction procedure that is designed to select the correct nuclei. Their algorithm can be stated in the following way, modified slightly from Dell & Elmedlaoui 1985: 111(15):

(5) Dell–Elmedlaoui Algorithm for Core Syllabification (DEA)
 Build a core syllable (“CV”) over each substring of the form XY, where
  X is any segment (except [*a*]), and
  Y is a matrix of features describing a step of the sonority scale.
 Start Y at the top of the sonority scale and replace it successively with the matrix of features appropriate to the next lower step of the scale.
 (Iterate from Left to Right for each fixing of the nuclear variable Y.)

Like all such procedures, the DEA is subject to the Free Element Condition (FEC: Prince 1985), which holds that rules establishing a level of prosodic structure apply only to elements that are not already supplied with the relevant structure. By the FEC, the positions analyzed by the terms X,Y must be free of syllabic affiliation. Effectively, this means that any element seized as an onset is no longer eligible to be a nucleus, and that a segment recruited to nucleate a syllable is not then available to serve as an onset.

There are other syllabification phenomena in ITB that require additional rules beyond the DEA; we will abstract away from these and focus on the sense of DEA itself.[4] We will also put aside some wrinkles in the DEA which are related to parenthesized expressions in (5) – the lack of a glide counterpart for /a/, the phrase-initial loosening of the onset requirement, and the claimed left-to-rightness of the procedure.[5]

The DEA is a rule, or rather a schema for rules, of exactly the classical type A → B / C—D. Each rule generated by the schema has a Structural Description specified in featural terms and a Structural Change ('construct a core syllable'). To see how it works, consider the following derivations:

(6) DEA in Action

Steps of the DEA	/ratlult/ 'you will be born'
Seek [X][+low,–cns] *& Build*	(**ra**)tlult
Seek [X][–low,–cns] *& Build*	(ra)t(**lu**)lt
Seek [X][+cns,+son,–nas]	*–blocked by FEC–*
Seek [X][+cns,+son,+nas]	—
Seek [X][–son,+cnt,+voi]	—
Seek [X][–son,+cnt,–voi]	—
Seek [X][–son,–cnt,+voi]	—
Seek [X][–son,–cnt,–voi] *& Build*	(ra)t(lu)(**lT**)[6]

(7) DEA in Action

Steps of the DEA	/txznt/ 'you sg. stored'
Seek [X][+low,–cns]	—
Seek [X][–low,–cns]	—
Seek [X][+cns,+son,–nas]	—
Seek [X][+cns,+son,+nas] *& Build*	tx(**zN**)t
Seek [X][–son,+cnt,+voi]	—
Seek [X][–son,+cnt,–voi] *& Build*	(**tX**)(zN)t
Seek [X][–son,–cnt,+voi]	—
Seek [X][–son,–cnt,–voi]	—

(8) DEA in Action

Steps of the DEA	/txznas/ 'she stored for him'
Seek [X][+low,–cns] *& Build*	txz(**na**)s
Seek [X][–low,–cns]	—
Seek [X][+cns,+son,–nas]	—
Seek [X][+cns,+son,+nas]	–*blocked by FEC*–
Seek [X][–son,+cnt,+voi] *& Build*	t(**xZ**)(na)s
Seek [X][–son,+cnt,–voi]	—
Seek [X][–son,–cnt,+voi]	—
Seek [X][–son,–cnt,–voi]	–*blocked by FEC*–

The DEA provides an elegant and straightforward account of the selection of syllable nuclei in the language. But it suffers from the formal arbitrariness characteristic of re-writing rules when they are put to the task of dealing locally with problems that fall under general principles, particularly principles of output shape. (By 'formal arbitrariness', we mean that a formal system rich enough to allow expression of the desired rule will also allow expression of many undesired variations of the rule, so that the rule itself appears to be an arbitrary random choice among the universe of possibilities.) The key to the success of the DEA is the way that the variable Y scans the input, starting at the top of the sonority scale and descending it step by step as the iterative process unfolds. We must ask, why start at the top? why *descend* the scale? why not use it in some more elaborate or context-dependent fashion? why apply the scale to the nucleus rather than the onset?[7]

The answers are to be found in the theory of syllable structure markedness, which is part of Universal Grammar. The more sonorous a segment is, the more satisfactory it is as a nucleus. Conversely, a nucleus is more satisfactory to the degree that it contains a more sonorous segment. It is clear that the DEA is designed to produce syllables with optimal nuclei; to ensure that the syllables it forms are the most *harmonic* that are available, to use the term introduced in §1. Dell and Elmedlaoui clearly understand the role of sonority in choosing between competing analyses of a given input string; they write:

> When a string . . . PQ . . . could conceivably be syllabified as . . . Pq . . . or as . . . pQ . . . (i.e. when either syllabification would involve only syllable types which, when taken individually, are possible in ITB), the only syllabification allowed by ITB is the one that takes as a syllabic peak the more sonorous of the two segments. (Dell & Elmedlaoui 1985: 109)

But if phonology is couched in re-writing rules, this insight cannot be cashed in as part of the function that assigns structural analyses. It remains formally inert.

Dell and Elmedlaoui refer to it as an ‘empirical observation’, emphasizing its extra-grammatical status.

The DEA itself makes no contact with any principles of well-formedness; it merely scans the input for certain specific configurations, and acts when it finds them. That it descends the sonority scale, for example, can have no formal explanation. But the insight behind the DEA can be made active if we re-conceive the process of syllabification as one of choosing the optimal output from among the possible analyses rather than algorithmic structure-building. Let us first suppose, with Dell and Elmedlaoui, that the process of syllabification is serial, affecting one syllable at a time (thus, that it operates like Move-α or more exactly, Move-x of grid theory). At each stage of the process, let all possible single syllabic augmentations of the input be presented for evaluation. This set of candidates is evaluated by principles of syllable well-formedness and the most harmonic structure in the set is selected as the output. We can state the process informally as follows:

(9) Serial Harmonic Syllabification (informal)
Form the optimal syllable in the domain.
Iterate until nothing more can be done.

This approach depends directly on the principles of well-formedness which define the notion ‘optimal’. No instructions are issued to the construction process to contemplate only one featurally specified niche of the sonority scale. Indeed, the Harmonic Syllabification algorithm has no access to any information at all about absolute sonority level or the specific featural composition of vowels, which are essential to the DEA; it needs to know whether segment α is *more* sonorous than segment β, not what their sonorities or features actually are. All possibilities are entertained simultaneously and the choice among them is made on grounds of general principle. That you start at the top of the scale, that you descend the scale rather than ascending it or touring it in some more interesting fashion, all this follows from the principles that define relative well-formedness of nucleus–segment pairings. The formal arbitrariness of the DEA syllable-constructing procedure disappears because the procedure itself (‘make a syllable’) has been stripped of intricacies.[8]

This is an instance of Harmony-increasing processing (Smolensky 1983, 1986; Goldsmith 1991, 1993). The general rubric is this:

(10) Harmonic Processing
Go to the most harmonic available state.

We speak not of ‘relative well-formedness’ but rather of *relative Harmony*. Harmony is a well-formedness scale along which a maximal Harmony structure is well-formed and all other structures are ill-formed.

We conclude that the Dell–Elmedlaoui results establish clearly that harmonic processing is a grammatical mechanism; and that optimality-based analysis gives results in complex cases. Let us now establish a formal platform that can support this finding.

2.2 Optimality Theory

What, then, is the *optimal* syllable that Harmonic Syllabification seeks? In the core process that we are focusing on, two constraints are at play, one ensuring onsets, the other evaluating nuclei. The onset constraint can be stated like this (Itô 1986, 1989):

(11) The Onset Constraint (Ons)
Syllables must have onsets (except phrase initially).

As promised, we are not going to explicate the parenthesized caveat, which is not really part of the basic constraint (McCarthy & Prince 1993: §4). The nuclear constraint looks like this:[9]

(12) The Nuclear Harmony Constraint (Hnuc)
A higher sonority nucleus is more harmonic than one of lower sonority.
i.e. If $|x| > |y|$ then Nuc/x $\succ$ Nuc/y.

The formalizing restatement appended to the constraint uses some notation that will prove useful:

For 'x is more harmonic than y' we write x $\succ$ y.
For 'the intrinsic prominence of x' we write $|x|$.
'A/x' means 'x belongs to category A, x is the constituent-structure child of A'.

The two kinds of order $\succ$ and $>$ are distinguished notationally to emphasize their conceptual distinctness. Segments of high sonority are not more harmonic than those of lower sonority. It is only when segments are contemplated in a structural context that the issue of well-formedness arises.

It is necessary to specify not only the relevant constraints, but also the set of candidates to be evaluated. To do this we need to spell out the function Gen that admits to candidacy a specific range of structurings or parses of the input. In the case at hand, we want something roughly like this:

(13) Gen ($input_i$)
The set of (partial) syllabifications of $input_i$ which differ from $input_i$ in no more than one syllabic adjunction.

For any form $input_i$ to undergo Serial Harmonic Syllabification, the candidate set Gen($input_i$) must be evaluated with respect to the constraints Ons and Hnuc. There would be little to say if evaluation were simply a matter of choosing the candidate that satisfies both constraints. Crucially, and typically, this straightforward approach cannot work. Conflict between the constraints Ons and Hnuc is unavoidable; there are candidate sets in which no candidate satisfies both constraints.

Consider, for example, the syllabification of the form /ħaul-tn/ 'make them (m.) plentiful' (Dell & Elmedlaoui 1985: 110). Both Ons and Hnuc agree that the core syllable *ħa* should be formed: it has an onset as well as the best possible nucleus. Similarly, we must have a final syllable *tN*. But what of the rest of the string? We have two choices for the sequence /ul/: a superior nucleus lacking an onset, as in ***ul***; or an onsetted syllable with an inferior nucleus, as in *wL*. This situation can be perspicuously displayed in tabular form:[10]

(14) Constraint Inconsistency

Candidates /ħaul-tn/	Ons	Hnuc
~.wL.~		|l|
~.**u**l.~	*	|u|

The cells contain information about how each candidate fares on the relevant constraint. A blank cell indicates that the constraint is satisfied; a star indicates violation. (In the case of a scalar constraint like Hnuc we mention the contents of the evaluated element.) The first form succeeds on Ons, while the second form violates the constraint. The relative performance is exactly the opposite on Hnuc: because |u| > |l|, the second, onsetless form has the better nucleus. The actual output is, of course, *.ħa.wL.tN*. The onset requirement, in short, takes priority.

Such conflict is ubiquitous, and to deal with it, we propose that a relation of *domination*, or priority-ranking, can be specified to hold between constraints. When we say that one constraint *dominates* another, we mean that when they disagree on the relative status of a pair of candidates, the dominating constraint makes the decision. If the dominating constraint does not decide between the candidates – as when both satisfy or both violate the constraint equally – then the comparison is passed to the subordinate constraint. (In the case of a more extensive hierarchy, the same method of evaluation can be applied repeatedly.)

In the case at hand, it is clear that Ons must dominate Hnuc. The top priority is to provide syllables with onsets; the relative Harmony of nuclei is a subordinate concern whose force is felt only when the Ons issue is out of the way. We will write this relation as Ons ≫ Hnuc. Given such a hierarchy, an optimality calculation can be usefully presented in an augmented version of display (14) that we will call a *constraint tableau*:

(15) Constraint Tableau for Partial Comparison of Candidates from /ħaultn/

Candidates	Ons	Hnuc
☞ ~.wL.~		|l|
~.**u**l.~	* !	|u|

Constraints are arrayed across the top of the tableau in domination order. As above, constraint violations are recorded with the mark *, and blankness indicates total success on the constraint. These are the theoretically important conventions; in addition, there is some clarificatory typography. The symbol ☞ draws the eye to the optimal candidate; the **!** marks the *crucial* failure for each suboptimal candidate, the exact point where it loses out to other candidates. Cells that do not participate in the decision are shaded. In the case at hand, the contest is decided by the dominant constraint Ons; Hnuc plays no role in the comparison of *.wL.* and *.ul.* Hnuc is literally irrelevant to this particular evaluation, as a consequence of its dominated position – and to emphasize this, we shade its cells. Of course, Hnuc is not irrelevant to the analysis of *every* input; but a precondition for relevance is that there be a set of candidates that tie on Ons, all passing it or all failing it to the same extent.

If we were to reverse the domination ranking of the two constraints, the predicted outcome would be changed: now .**u**l. would be superior to .wL. by virtue of its relative success on Hnuc, and the Ons criterion would be submerged. Because of this, the ranking Ons ≫ Hnuc is *crucial*; it must obtain in the grammar of Berber if the actual language is to be generated.

The notion of domination shows up from time to time in one form or another in the literature, sometimes informally, sometimes as a clause clarifying how a set of constraints is to be interpreted. For example, Dell and Elmedlaoui write, "The prohibition of hiatus . . . *overrides*" the nuclear sonority comparison (Dell & Elmedlaoui 1985: 109, emphasis added). For them, this is an extra-grammatical observation, with the real work done by the Structural Descriptions provided by the DEA and the ordering of application of the subrules. Obviously, though, the insight is clearly present. Our claim is that the notion of domination, or 'overriding', is the truly fundamental one. What deserves extra-grammatical status is the machinery for constructing elaborately specific Structural Descriptions and modes of rule application.

To see how Serial Harmonic Syllabification (9) proceeds, let us examine the first stage of syllabifying the input /txznt/ 'you sg. stored, pf.'. It is evident that the first syllable constructed must be *.zN.* – it has an onset, and has the highest sonority nucleus available, so no competing candidate can surpass or even equal it. A more discursive examination of possibilities might be valuable; the larger-scale comparisons are laid out in the constraint tableau below.

Here are (some of the) leading candidates in the first round of the process:

(16) Constraint Tableau for Serial Syllabification of /txznt/ (partial, first step)

Candidates	Ons	Hnuc	Comments
☞ tx(z**N**)t		n	optimal: onsetted, best available nucleus
txz(**N**)t	* !	n	no onset, Hnuc irrelevant
t(x**Z**)nt		z !	|z| < |n|
(t**X**)znt		x !	|x| < |n|
txz(n**T**)		t !	|t| < |n|

Syllabic parsing is conceived here as a step-by-step serial process, just as in the DEA. A candidate set is generated, each produced by a single licit change from the input; the relative status of the candidates is evaluated, yielding an optimal candidate (the output of the first step); and that output will then be subject to a variety of further single changes, generating a new candidate set to be evaluated; and so on, until there are no bettering changes to be made: the final output has then been determined.

This step-by-step Harmony evaluation is not intrinsic to the method of evaluation, though, and, in the more general context, when we discard the restricted definition of Gen in (13), it proves necessary to extend the procedure so that it is capable of evaluating entire parsed strings, and not just single (new) units of analysis. To do this, we apply the same sort of reasoning used to define domination, but *within* the constraint categories. To proceed by example, consider the analysis of /txznt/ taking for candidates all syllabified strings. We present a sampling of the candidate space.

(17) Parallel Analysis of Complete Syllabification of /txznt/

Candidates	Ons	Hnuc		Comments
☞ .tX.zNt.		n	x	optimal
.Tx.zNt.		n	t !	|n| = |n|, |t| < |x|
.tXz.nT.		x !	t	|x| < |n|, t irrelevant
.txZ.Nt.	* !	z	n	Hnuc irrelevant
.T.X.Z.N.T.	* ! ****	n z x t t		Hnuc irrelevant

In evaluating the candidates we have kept to the specific assumptions mentioned above: the onset requirement is suspended phrase-initially, and the nonnuclear status of peripheral obstruents is, as in the DEA itself, put aside.

In this tableau, all the relevant information for harmonic evaluation of the parse of the whole string is present. We start by examining the first column, corresponding to the dominant constraint Ons. Only the candidates which fare best on this constraint survive for further consideration. The first three candidates all have syllables with onsets; the last two do not (to varying degrees). Lack of onset in even a single non-initial syllable is immediately fatal, because of the competing candidates which satisfy Ons.

The remaining three parses are not distinguished by Ons, and so Hnuc, the next constraint down the hierarchy, becomes relevant. These three parses are compared by Hnuc as follows. The most sonorous nucleus of each parse is examined: these are the most harmonic nuclei according to Hnuc. For each of the first two candidates the most sonorous nucleus is *n*. For the last candidate, the most sonorous nucleus is *x*, and it drops out of the competition since *n* is more sonorous than *x*. We are left with the first two candidates, so far tied on all comparisons. The Hnuc evaluation continues now to the next-most-harmonic nuclei, where the competition is finally settled in favor of the first candidate .tX.zNt.

What we have done, in essence, is to replace the iterative procedure (act/evaluate, act/evaluate, . . .) with a recursive scheme: collect the results of all possible actions, then sort recursively. Rather than producing and pruning a candidate set at each step of sequential processing, striving to select at each step the action which will take us eventually to the correct output, the whole set of possible parses is defined and harmonically evaluated. The correct output is the candidate whose complete structure best satisfies the constraint hierarchy. And 'best satisfies' can be recursively defined by descending the hierarchy, discarding all but the best possibilities according to each constraint before moving on to consider lower-ranked constraints.

The great majority of analyses presented here will use the parallel method of evaluation. A distinctive prediction of the parallel approach is that there can be significant interactions of the top-down variety between aspects of structure that are present in the final parse. In §4 and §7 [omitted here – Ed.] we will see a number of cases where this is borne out, so that parallelism is demonstrably crucial; further evidence is presented in McCarthy & Prince 1993. 'Harmonic serialism' is worthy of exploration as well, and many hybrid theories can and should be imagined; but we will have little more to say about it. (But see fn. 49 below on Berber syllabification. [omitted here – Ed.])

The notion of parallel analysis of complete parses in the discussion of constraint tableau (17) is the crucial technical idea on which many of our arguments will rest. It is a means for determining the relative harmonies of entire candidate parses from a set of conflicting constraints. This technique has some subtleties, and is subject to a number of variant developments, so it is worth setting out with some formal precision exactly what we have in mind. A certain level of complexity arises because there are two dimensions of structure to keep track of. On the one hand, each individual constraint typically applies to several substructures in any complete parse, generating a *set* of evaluations. (Ons, for example, examines every syllable, and there are often several of them to examine.) On the other hand, every grammar has multiple constraints, generating multiple sets of evaluations. Regulating the way these two dimensions of multiplicity interact is a key theoretical commitment.

Our proposal is that evaluation proceeds by constraint. In the case of the minigrammar of Ons and Hnuc, entire syllabifications are first compared via Ons alone, which examines each syllable for an onset; should this fail to decide the matter, the entire syllabifications are compared via Hnuc alone, which examines each syllable's nucleus.

Another way to use the two constraints would be to examine each (completely parsed) candidate syllable-by-syllable, assessing each syllable on the basis of the syllabic mini-grammar. The fact that Ons dominates Hnuc would then manifest itself in the Harmony assessment of each individual syllable. This is also the approach most closely tied to continuous Harmony evaluation during a step-by-step constructive derivation. Here again, we do not wish to dismiss this conception, which is surely worthy of development. Crucially, however, this is not how Harmony evaluation works in the present conception.

In order to characterize harmonic comparison of candidate parses with full generality and clarity, we need to specify two things: first, a means of comparing entire candidates on the basis of a single constraint; then, a means of combining the evaluation of these constraints. The result is a general definition of *Harmonic Ordering*

of Forms; this is, in its formal essence, our theory of constraint interaction in generative grammar. It is the main topic of §5.
[. . .]

5 The Construction of Grammar in Optimality Theory

Phonological theory contains two parts: a theory of substantive universals of phonological well-formedness and a theory of formal universals of constraint interaction. These two components are respectively the topics of §5.1 and §5.2. Since much of this work concerns the first topic, the discussion here will be limited to a few brief remarks. In §5.3, we give Pāṇini's Theorem, a theorem about the priority of the specific which follows from the basic operation of Optimality Theory as set out in §5.2.

5.1 Construction of harmonic orderings from phonetic and structural scales

To define grammars from hierarchies of well-formedness constraints, we need two distinct constructions: one that takes given constraints and defines their interactions, the other that pertains to the constraints themselves. The first will be discussed at some length in §5.2; we now take up the second briefly.

Construction of constraints amounts in many ways to a theory of contextual markedness (Chomsky & Halle 1968: ch. 9, Kean 1974, Cairns & Feinstein 1982, Cairns 1988, Archangeli & Pulleyblank 1992). Linguistic phonetics gives a set of scales on phonetic dimensions; these are not well-formedness ratings, but simply the analyses of phonetic space that are primitive from the viewpoint of linguistic theory. (We use the term 'scale' in the loosest possible sense, to encompass everything from unary features to n-ary orderings.)

Issues of relative well-formedness, or markedness, arise principally when elements from the different dimensions are combined into interpretable representations. High sonority, for example, does not by itself entail high (or low) Harmony; but when a segment occurs in a structural position such as nucleus, onset, or coda, its intrinsic sonority in combination with the character of its position gives rise to markedness-evaluating constraints such as HNUC above. Similarly, tongue-height in vowels is neither harmonic nor disharmonic in isolation, but when the dimension of ATR (Advanced Tongue Root) is brought in, clear patterns of relative well-formedness or Harmony emerge, as has been emphasized in the work of Archangeli & Pulleyblank (1992). These *Harmony scales* are intimately tied to the repertory of constraints that grammars draw on. Inasmuch as there are principled harmonic concomitants of dimensional combination, we need ways of deriving Harmony scales from phonetic scales. Symbolically, we have

(94) Harmony Scale from Interaction of Phonetic Scales
$\{a > b \ldots\} \otimes \{x > y > \ldots\} = ax > \ldots$

The goal of contextual markedness theory is to give content to the operator ⊗. Below in §8 we introduce a formal mechanism of *Prominence Alignment* which generates constraint rankings from paired phonetic scales, yielding a Harmony scale on their combination. In the syllable structure application of §8, the two phonetic scales which are aligned are segmental prominence (the sonority dimension) and syllable position prominence (Peak is a more prominent position than Margin). The result is a Harmony scale on associations of segments to syllable positions.

It is important to distinguish the three kinds of scales or hierarchies which figure in Optimality Theory. To minimize confusions, we have given each its own distinctive comparison symbol. Two of these figure in (94): elements are ordered on a phonetic scale by the relation '>', and on a Harmony scale according to '≻'. The third type of hierarchy in the theory is the domination hierarchy, along which constraints are ranked by the relation '≫'. These different types of scales are enumerated and exemplified in the following table:

(95) Three Different Scales in Optimality Theory

Type of scale or hierarchy	Relates	Symbol	Example	Meaning
Phonetic scale	Points along elementary representational dimensions	$>$	$a > l$	*a* is more sonorous than *l*
Harmony scale	Well-formedness of structural configurations built from elementary dimensions	$\succ$	$\acute{a} \succ \acute{l}$	a nucleus filled by *a* is more harmonic than a nucleus filled by *l*
Domination hierarchy	Relative priority of well-formedness	$\gg$	ONS ≫ HNUC	the constraint ONS strictly dominates the constraint HNUC

5.2 The theory of constraint interaction

In order to define harmonic comparison of candidates consisting of entire parses, we will proceed in two steps. First, we get clear about comparing entire candidates on the basis of a single constraint, using ONS and HNUC from the Berber analysis in §2 as our examples. Then we show how to combine the evaluation of these constraints using a domination hierarchy.

5.2.1 Comparison of entire candidates by a single constraint

The first order of business is a precise definition of how a single constraint ranks entire parses. We start with the simpler case of a single binary constraint, and then generalize the definition to non-binary constraints.

5.2.1.1 Ons: Binary constraints

It is useful to think of Ons as examining a syllable to see if it has an onset; if it does not, we think of Ons as assessing a **mark** of violation, *Ons. Ons is an example of a *binary constraint*; a given syllable either satisfies or violates the constraint entirely. The marks Ons generates are all of the same type: *Ons. For the moment, all the marks under consideration are identical. Later, when we consider the interaction of multiple binary constraints, there will be different types of marks to distinguish; each binary constraint $\mathbb{C}$ generates marks of its own characteristic type, *$\mathbb{C}$. Furthermore, some constraints will be non-binary, and will generate marks of different types representing different degrees of violation of the constraint: the next constraint we examine, Hnuc, will illustrate this.

When assessing the entire parse of a string, Ons examines each σ node in the parse and assesses one mark *Ons for each such node which lacks an onset. Introducing a bit of useful notation, let A be a prosodic parse of an input string, and let Ons(A) = (*Ons, *Ons, . . .) be a list containing one mark *Ons for each onsetless syllable in A. Thus for example Ons(*.txź.ńt.*) = (*Ons): the second, onsetless, syllable earns the parse *.txź.ńt.* its sole *Ons mark. (Here we use *ź* to indicate that *z* is parsed as a nucleus.)

Ons provides a criterion for comparing the Harmony of two parses A and B; we determine which of A or B is more harmonic ('less marked') by comparing Ons(A) and Ons(B) to see which contains fewer *Ons marks. We can notate this as follows:

$$\text{A} \succ_{\text{Ons}}^{\text{parse}} \text{B iff Ons(A)} \succ^{(*)} \text{Ons(B)}$$

where '$\succ_{\text{Ons}}^{\text{parse}}$' denotes comparison of entire parses and 'Ons(A) $\succ^{(*)}$ Ons(B)' means 'the list Ons(A) contains fewer marks *Ons than the list Ons(B)'. (We will use the notation '(*)' as a mnemonic for 'list of marks'.) If the lists are the same length, then we write[11]

$$\text{A} \approx_{\text{Ons}}^{\text{parse}} \text{B iff Ons(A)} \approx^{(*)} \text{Ons(B)}.$$

It is extremely important to realize that what is crucial to $\succ^{(*)}$ is not numerical counting, but simply comparisons of more or less. This can be emphasized through a recursive definition of $\succ^{(*)}$, a definition which turns out to provide the basis for the entire Optimality Theory formalism for Harmony evaluation. The intuition behind this recursive definition is very simple.

Suppose we are given two lists of identical marks *$\mathbb{C}$; we need to determine which list is shorter, and we can't count. Here's what we do. First, we check to see if either list is empty. If both are, the conclusion is that neither list is shorter. If one list is empty and the other isn't, the empty one is shorter. If neither is empty, then we remove one mark *$\mathbb{C}$ from each list, and start all over. The process will eventually terminate with a correct conclusion about which list is the shorter – but with no information about the numerical lengths of the lists.

Formalizing this recursive definition is straightforward; it is also worthwhile, since the definition will be needed anyway to characterize the full means of evaluating the relative harmonies of two candidate parses.

We assume two simple operations for manipulating lists. The operation we'll call **FM** extracts the First Member (or ForeMost element) of a list; this is what we use to extract the First Mark $*\mathbb{C}$ from each list. The other operation **Rest** takes a list, throws away its First Member, and returns the rest of the list; we use this for the recursive step of 'starting over', asking which list is shorter after the first * has been thrown out of each.

Since we keep throwing out marks until none are left, it's also important to deal with the case of empty lists. We let ( ) denote an empty list, and we define FM so that when it operates on ( ), its value is ø, the null element.

Now let α and β be two lists of marks. We write $\alpha \succ^{(*)} \beta$ for 'α is more harmonic than β', which in the current context means 'α is *shorter* than β', since marks are anti-harmonic. To express the fact that an empty list of marks is more harmonic than a non-empty list, or equivalently that a null first element indicates a more harmonic list than does a non-null first element $*\mathbb{C}$, we adopt the following relation between single marks:

(96) Marks are Anti-harmonic

$$ø \succ^{*} *\mathbb{C}$$

Remembering that $\approx$ denotes 'equally harmonic', we also note the obvious facts about identical single marks:

$ø \approx^{*} ø$ and $*\mathbb{C} \approx^{*} *\mathbb{C}$

Our recursive definition of $\succ^{(*)}$ can now be given as follows, where α and β denote two lists of identical marks:

(97) Harmonic Ordering – Lists of Identical Marks

$\alpha \succ^{(*)} \beta$ iff either:

(i) $\mathrm{FM}(\alpha) \succ^{*} \mathrm{FM}(\beta)$

or

(ii) $\mathrm{FM}(\alpha) \approx^{*} \mathrm{FM}(\beta)$ and $\mathrm{Rest}(\alpha) \succ^{(*)} \mathrm{Rest}(\beta)$

'$\beta \prec^{(*)} \alpha$' is equivalent to '$\alpha \succ^{(*)} \beta$'; '$\alpha \approx^{(*)} \beta$' is equivalent to 'neither $\alpha \succ^{(*)} \beta$ nor $\beta \succ^{(*)} \alpha$'. (In subsequent order definitions, we will omit the obvious counterparts of the final sentence defining $\prec^{(*)}$ and $\approx^{(*)}$ in terms of $\succ^{(*)}$.)

To repeat the basic idea of the definition one more time in English: α is shorter than β iff (if and only if) one of the following is true: (i) the first member of α is null and the first member of β is not (i.e., α is empty and β is not), or (ii) the list left over after removing the first member of α is shorter than the list left over after removing the first member of β.[12]

Now we can say precisely how Ons assesses the relative Harmony of two candidate parses, say *.tx́.zńt.* and *.txź.ńt.* Ons assesses the first as more harmonic than the second, because the second has an onsetless syllable and the first does not. We write this as follows:

.tx́.zńt. $\succ_{\text{Ons}}^{\text{parse}}$ *.txź.ńt.* because Ons(*.tx́.zńt.*) = ( ) $\succ^{(*)}$ (*Ons) = Ons(*.txź.ńt.*)

where $\succ^{(*)}$ is defined in (97).

As another example:

.tx́.zńt. $\approx_{\text{Ons}}^{\text{parse}}$ *.txź.nt́.* because Ons(*.tx́.zńt.*) = ( ) $\approx^{(*)}$ ( ) = Ons(*.txź.nt́.*)

In general, for any binary constraint $\mathbb{C}$, the harmonic ordering of entire parses which it determines, $\succ_{\mathbb{C}}^{\text{parse}}$, is defined as follows, where A and B are candidate parses:

(98) Harmonic Ordering of Forms – Entire Parses, Single Constraint $\mathbb{C}$
$\quad$ A $\succ_{\mathbb{C}}^{\text{parse}}$ B iff $\mathbb{C}$(A) $\succ^{(*)}$ $\mathbb{C}$(B)

with $\succ^{(*)}$ as defined in (97).

It turns out that these definitions of $\succ^{(*)}$ (97) and $\succ_{\mathbb{C}}^{\text{parse}}$ (98), which we have developed for binary constraints (like Ons), apply equally to non-binary constraints (like Hnuc); in the general case, a constraint's definition includes a harmonic ordering of the various types of marks it generates. The importance of the definition justifies bringing it all together in self-contained form:

(99) Harmonic Ordering of Forms – Entire Parse, Single Constraint
Let $\mathbb{C}$ denote a constraint. Let A,B be two candidate parses, and let α,β be the lists of marks assigned them by $\mathbb{C}$:
$\quad \alpha \equiv \mathbb{C}(A), \quad \beta \equiv \mathbb{C}(B)$
$\mathbb{C}$ by definition provides a Harmony order $\succ^{*}$ of the marks it generates. This order is extended to a Harmony order $\succ^{(*)}$ over lists of marks as follows:
$\quad \alpha \succ^{(*)} \beta$ iff either:
$\qquad$ (i) $\text{FM}(\alpha) \succ^{*} \text{FM}(\beta)$
$\quad$ or
$\qquad$ (ii) $\text{FM}(\alpha) \approx^{*} \text{FM}(\beta)$ and $\text{Rest}(\alpha) \succ^{(*)} \text{Rest}(\beta)$
This order $\succ^{(*)}$ is in turn extended to a Harmony order over candidate parses (with respect to $\mathbb{C}$), $\succ_{\mathbb{C}}^{\text{parse}}$, as follows:
$\quad$ A $\succ_{\mathbb{C}}^{\text{parse}}$ B iff $\mathbb{C}(A) \equiv \alpha \succ^{(*)} \beta \equiv \mathbb{C}(B)$

The case we have so far considered, when $\mathbb{C}$ is binary, is the simplest precisely because the Harmony order over marks which gets the whole definition going, $\succ^{*}$, is so trivial:

$\varnothing \succ^{*} {*}\mathbb{C}$

'a mark absent is more harmonic than one present' (96). In the case we consider next, however, the ordering of the marks provided by $\mathbb{C}$, $\succ^{*}$, is more interesting.

5.2.1.2 Hnuc: Non-binary constraints

Turn now to Hnuc. When it examines a single syllable, Hnuc can usefully be thought of as generating a symbol designating the nucleus of that syllable; if the nucleus is *n*, then Hnuc generates *ń*. Hnuc arranges these nucleus symbols in a Harmony order, in which x́ $\succ_{\text{Hnuc}}$ ý if and only if x is more sonorous than y: |x| > |y|.

If A is an entire prosodic parse, HNUC generates a list of all the nuclei in A. For reasons soon to be apparent, it will be convenient to think of HNUC as generating a list of nuclei *sorted from most to least harmonic, according to HNUC* – i.e., from most to least sonorous. So, for example, HNUC(*.txź.ńt.*) = (*ń*, *ź*).

When HNUC evaluates the relative harmonies of two entire syllabifications A and B, it first compares the most harmonic nucleus of A with the most harmonic nucleus of B: if that of A is more sonorous, then A is the winner without further ado. Since the lists of nuclei HNUC(A) and HNUC(B) are assumed sorted from most to least harmonic, this process is simply to compare the First Member of HNUC(A) with the First Member of HNUC(B): if one is more harmonic than the other, according to HNUC, the more harmonic nucleus wins the competition for its entire parse. If, on the other hand, the two First Members of HNUC(A) and HNUC(B) are equally harmonic according to HNUC (i.e., equally sonorous), then we eject these two First Members from their respective lists and start over, comparing the Rest of the nuclei in exactly the same fashion.

This procedure is exactly the one formalized above in (99). We illustrate the formal definition by examining how HNUC determines the relative harmonies of

A ≡ *.tx́.zńt.* and B ≡ *.t́x.zńt.*

First, $\mathbb{C}$ ≡ HNUC assigns the following:

$\alpha \equiv \mathbb{C}(\text{A}) = (\acute{n}, \acute{x})$ $\beta \equiv \mathbb{C}(\text{B}) = (\acute{n}, \acute{t})$

To rank the parses A and B, i.e. to determine whether

$\text{A} \succ_{\mathbb{C}}^{\text{parse}} \text{B}$,

we must rank their list of marks according to $\mathbb{C}$, i.e. determine whether

$\mathbb{C}(\text{A}) \equiv \alpha \succ^{(*)} \beta \equiv \mathbb{C}(\text{B})$.

To do this, we examine the First Marks of each list, and determine whether

$\text{FM}(\alpha) \succ^{*} \text{FM}(\beta)$.

As it happens,

$\text{FM}(\alpha) \approx \text{FM}(\beta)$,

since both First Marks are *ń*, so we must discard the First Marks and examine the Rest, to determine whether

$\alpha' \equiv \text{Rest}(\alpha) \succ^{(*)} \text{Rest}(\beta) \equiv \beta'$.

Here,

$\alpha' = (\acute{x})$; $\beta' = (\acute{t})$.

So again we consider First Marks, to determine whether

$\mathrm{FM}(\alpha') \succ^{*} \mathrm{FM}(\beta')$.

Indeed this is the case:

$\mathrm{FM}(\alpha') = \acute{x} \succ^{*} \acute{t} = \mathrm{FM}(\beta')$

since $|x| > |t|$. Thus we finally conclude that

.tx́.zńt. $\succ_{\text{Hnuc}}{}^{\text{parse}}$ *.t́x.zńt.*

Hnuc assesses nuclei x́ from most to least harmonic, and that is how they are ordered in the lists Hnuc generates for Harmony evaluation. Hnuc is an unusual constraint in this regard; the other non-binary constraints we consider in this work will compare their *worst* marks first; the mark lists they generate are ordered from least- to most-harmonic. Both kinds of constraints are treated by the same definition (99). The issue of whether mark lists should be generated worst- or best-first will often not arise, for one of two reasons. First, if a constraint $\mathbb{C}$ is binary, the question is meaningless because all the marks it generates are identical: *$\mathbb{C}$. Alternatively, if a constraint applies only once to an entire parse, then it will generate only one mark per candidate, and the issue of ordering multiple marks does not even arise. (Several examples of such constraints, including edgemostness of main stress, or edgemostness of an infix, are discussed in §4 [omitted here – Ed.].) But for constraints like Hnuc which are non-binary and which apply multiply in a candidate parse, part of the definition of the constraint must be whether it lists worst- or best-marks first.

5.2.2 Comparison of entire candidates by an entire constraint hierarchy

We have now defined how a single constraint evaluates the relative Harmonies of entire candidate parses (99). It remains to show how a collection of such constraints, arranged in a strict domination hierarchy [$\mathbb{C}_1 \gg \mathbb{C}_2 \gg \ldots$], *together* perform such an evaluation: that is, how constraints interact.

Consider the part of the Berber constraint hierarchy we have so far developed: [Ons $\gg$ Hnuc]. The entire hierarchy can be regarded as assigning to a complete parse such as *.txź.ńt.* the following list of lists of marks:

(100a) [Ons $\gg$ Hnuc](*.txź.ńt.*) = [Ons(*.txź.ńt.*), Hnuc(*.txź.ńt.*)] = [(*Ons), (*ń, ź*)]

The First Member here is the list of marks assigned by the dominant constraint: (*Ons). Following are the lists produced by successive constraints down the domination hierarchy; in this case, there is just the one other list assigned by Hnuc. As always, the nuclei are ordered from most- to least-harmonic by Hnuc.

We use square brackets to delimit this list of lists, but this is only to aid the eye, and to suggest the connection with constraint hierarchies, which we also enclose in square brackets. Square and round brackets are formally equivalent here, in the sense that they are treated identically by the list-manipulating operations FM and Rest.

The general definition of the list of lists of marks assigned by a constraint hierarchy is simply:

(101) Marks Assigned by an Entire Constraint Hierarchy
The marks assigned to an entire parse A by a constraint hierarchy [$\mathbb{C}1 \gg \mathbb{C}2 \gg \ldots$] is the following list of lists of marks:
$[\mathbb{C}1 \gg \mathbb{C}2 \gg \ldots](\mathrm{A}) \equiv [\mathbb{C}1(\mathrm{A}), \mathbb{C}2(\mathrm{A}), \ldots]$

Consider a second example, *.tx́.zńt.*:

(100b) [Ons ≫ Hnuc](*.tx́.zńt.*) = [Ons(*.tx́.zńt.*), Hnuc(*.tx́.zńt.*)] = [( ), (*ń*, *x́*)]

Since there are no onsetless syllables in this parse, Ons(*.tx́.zńt.*) = ( ), the empty list. A third example is:

(100c) [Ons ≫ Hnuc](*.t́x.zńt.*) = [( ), (*ń*, *t́*)]

As always in Berber, the Ons constraint is lifted phrase-initially, so this parse incurs no marks *Ons.

Now we are ready to harmonically rank these three parses. Corresponding directly to the example tableau (17) of §2, repeated here:

(102) Constraint Tableau for Three Parses of /txznt/

Candidates	Ons	Hnuc
☞ .tx́.zńt.		ń x́
.t́x.zńt.		ń t́ !
.txź.ńt.	* !	ń ź

we have, from (100a–c):

(103) Marks Assessed by the Constraint Hierarchy on Three Parses of /txznt/

A	[Ons ≫ Hnuc] (A)
.tx́.zńt.	[( ) , (ń x́)]
.t́x.zńt.	[( ) , (ń t́)]
.txź.ńt.	[(*Ons) , (ń ź)]

To see how to define the Harmony order $\succ_{[\text{Ons} \gg \text{Hnuc}]}$ that the constraint hierarchy imposes on the candidate parses, let's first review how Harmony comparisons are performed with the tableau (102). We start by examining the marks in the first, ONS, column. Only the candidates which fare best by these marks survive for further consideration. In this case, one candidate, *.txź.ńt.*, is ruled out because it has a mark *ONS while the other two do not. That is, this candidate is less harmonic than the other two with respect to the hierarchy [ONS ≫ HNUC] because it is less harmonic than the other two with respect to the dominant individual constraint ONS. The remaining two parses *.tx́.źńt.* and *.t́x.źńt.* are equally harmonic with respect to ONS, and so to determine their relative Harmonies with respect to [ONS ≫ HNUC] we must continue by comparing them with respect to the next constraint down the hierarchy, HNUC. These two parses are compared by the individual constraint HNUC in just the way we have already defined: the most harmonic nuclei are compared first, and since this fails to determine a winner, the next-most harmonic nuclei are compared, yielding the final determination that *.tx́.źńt.* $\succ_{[\text{ONS} \gg \text{HNUC}]}$ *.t́x.źńt.*

For the case of [ONS ≫ HNUC], the definition should now be clear:

(104) Harmonic Ordering of Forms – Entire Parses by [ONS ≫ HNUC]
$A \succ_{[\text{ONS} \gg \text{HNUC}]} B$ iff either
(i) $A \succ_{\text{ONS}} B$
or
(ii) $A \approx_{\text{ONS}} B$ and $A \succ_{\text{HNUC}} B$

For a general constraint hierarchy, we have the following recursive definition:

(105) Harmonic Ordering of Forms – Entire Parse, Entire Constraint Hierarchy
$A \succ_{[\mathbb{C}1 \gg \mathbb{C}2 \gg \ldots]} B$ iff either
(i) $A \succ_{\mathbb{C}1} B$
or
(ii) $A \approx_{\mathbb{C}1} B$ and $A \succ_{[\mathbb{C}2 \gg \ldots]} B$

All the orderings in (104) and (105) are of complete parses, and we have therefore omitted the superscript $^{\text{parse}}$. The Harmony order presupposed by this definition, $\succ_{\mathbb{C}1}^{\text{parse}}$, the order on entire parses determined by the single constraint $\mathbb{C}1$, is defined in (99).

It is worth showing that the definitions of whole-parse Harmony orderings by a single constraint $\succ_{\mathbb{C}}$ (99) and by a constraint hierarchy $\succ_{[\mathbb{C}1 \gg \mathbb{C}2 \ldots]}$ (105) are essentially identical. To see this, we need only bring in FM and Rest explicitly, and insert them into (105); the result is the following:

(106) Harmonic Ordering of Forms – Entire Parses, Entire Constraint Hierarchy (Opaque Version)
Let $\mathbb{CH} \equiv [\mathbb{C}1 \gg \mathbb{C}2 \gg \ldots]$ be a constraint hierarchy and let A,B be two candidate parses. Let α,β be the two lists of lists of marks assigned to these parses by the hierarchy:
$\alpha \equiv \mathbb{CH}(A), \quad \beta \equiv \mathbb{CH}(B)$
It follows that:
$\text{FM}(\alpha) = \mathbb{C}1(A), \quad \text{Rest}(\alpha) = [\mathbb{C}2 \gg \ldots](A);$
$\text{FM}(\beta) = \mathbb{C}1(B), \quad \text{Rest}(\beta) = [\mathbb{C}2 \gg \ldots](B)$

The hierarchy $\mathbb{CH}$ determines a harmonic ordering over lists of lists of marks as follows:

$\alpha \succ^{[(*)]} \beta$ iff either

(i) $\mathrm{FM}(\alpha) \succ^{(*)} \mathrm{FM}(\beta)$ (i.e., $\mathbb{C}1(\mathrm{A}) \succ^{(*)} \mathbb{C}1(\mathrm{B})$, i.e., $\mathrm{A} \succ_{\mathbb{C}1} \mathrm{B}$)

or

(ii) $\mathrm{FM}(\alpha) \approx^{(*)} \mathrm{FM}(\beta)$ (i.e., $\mathrm{A} \approx_{\mathbb{C}1} \mathrm{B}$)

and

$\mathrm{Rest}(\alpha) \succ^{[(*)]} \mathrm{Rest}(\beta)$

The harmonic ordering over candidate parses determined by $\mathbb{CH}$ is then defined by:

$\mathrm{A} \succ_{\mathbb{CH}}{}^{\mathrm{parse}}\ \mathrm{B}$ iff $\mathbb{CH}(\mathrm{A}) \equiv \alpha \succ^{[(*)]} \beta \equiv \mathbb{CH}(\mathrm{B})$

This definition of $\succ_{\mathbb{CH}}{}^{\mathrm{parse}}$ is identical to the definition of $\succ_{\mathbb{C}}{}^{\mathrm{parse}}$ (99) except for the inevitable substitutions: the single constraint $\mathbb{C}$ of (99) has been replaced with a constraint hierarchy $\mathbb{CH}$ in (106), and, accordingly, one additional level has been added to the collections of marks.

The conclusion, then, is that whole-parse Harmony ordering by constraint hierarchies is defined just like whole-parse Harmony ordering by individual constraints. To compare parses, we compare the marks assigned them by the constraint hierarchy. This we do by first examining the First Marks – those assigned by the dominant constraint. If this fails to decide the matter, we discard the First Marks, take the Rest of the marks (those assigned by the remaining constraints in the hierarchy) and start over with them.

Thus, there is really only one definition for harmonic ordering in Optimality Theory; we can take it to be (99). The case of binary marks (§5.2.1.1) is a simple special case, where 'less marked' reduces to 'fewer (identical) marks'; the case of constraint hierarchies (106) is a mechanical generalization gotten by making obvious substitutions.

5.2.3 Discussion

5.2.3.1 Non-locality of interaction

As mentioned at the end of §2, the way that constraints interact to determine the Harmony ordering of an entire parse is somewhat counter-intuitive. In the Berber hierarchy [Ons ≫ Hnuc], for example, perhaps the most obvious way of ordering two parses is to compare the parses syllable-by-syllable, assessing each syllable independently first on whether it meets Ons, and then on how well it fares with Hnuc. As it happens, this can be made to work for the special case of [Ons ≫ Hnuc] if we evaluate syllables in the correct order: from most- to least-harmonic. This procedure can be shown to more-or-less determine the same optimal parses as the different harmonic ordering procedure we have defined above, but only because some very special conditions obtain: first, there are only two constraints, and second, the dominant one is never violated in optimal parses. [note omitted.] Failing such special conditions, however, harmonic ordering as defined above and as used in the remainder of this work gives results which, as far as we know, cannot be duplicated or even approximated using the more obvious scheme of syllable-by-syllable evaluation. Indeed, when we extend our analysis of Berber even one step beyond the simple pair of constraints Ons and Hnuc (see §8), harmonic ordering clearly becomes required to get the correct results.

It is important to note also that harmonic ordering completely finesses a nasty conceptual problem which faces a syllable-by-syllable approach as soon as we expand our horizons even slightly. For in general we need to rank complex parses which contain much more structure than mere syllables. The 'syllable-by-syllable' approach is conceptually really a 'constituent-by-constituent' approach, and in the general case there are many kinds and levels of constituents in the parse. Harmonic ordering completely avoids the need to decide in the general case how to correctly break structures into parts for Harmony evaluation so that all the relevant constraints have the proper domains for their evaluation. In harmonic ordering, each constraint $\mathbb{C}$ independently generates its own list of marks $\mathbb{C}(A)$ for evaluating a parse A, considering whatever domains within A are appropriate to that constraint. In comparing A with parse B, the marks $\mathbb{C}(A)$ are compared with the marks $\mathbb{C}(B)$; implicitly, this amounts to comparing A and B with respect to the domain structure peculiar to $\mathbb{C}$. This comparison is decoupled from that based on other constraints which may have quite different domain structure.

The interaction of constraints in a constituent-by-constituent approach is in a sense limited to interactions within a constituent: for ultimately the comparison of competing parses rests on the assessment of the Harmony of individual constituents as evaluated by the set of constraints. Optimality Theory is not limited to constraint interactions which are local in this sense, as a number of the subsequent analyses will illustrate.

5.2.3.2 Strictness of domination

Our expository example [Ons $\gg$ Hnuc] in Berber may fail to convey just how strong a theory of constraint interaction is embodied in harmonic ordering. In determining the correct – optimal – parse of an input, as the constraint hierarchy is descended, each constraint acts to disqualify remaining competitors with absolute independence from all other constraints. A parse found wanting on one constraint has absolutely no hope of redeeming itself by faring well on any or even all lower-ranking constraints. It is remarkable that such an extremely severe theory of constraint interaction has the descriptive power it turns out to possess.

Such strict domination of constraints is less striking in the Berber example we have considered than it will be in most subsequent examples. This is because the dominant constraint is never violated in the forms of the language; it is hardly surprising then that it has strict veto power over the lower constraint. In the general case, however, most of the constraints in the hierarchy will *not* be unviolated like Ons is in Berber. Nonetheless, *all* constraints in Optimality Theory, whether violated or not in the forms of the language, have the same strict veto power over lower constraints that Ons has in Berber.

5.2.3.3 Serial vs. Parallel Harmony Evaluation and Gen

Universal Grammar must also provide a function Gen that admits the candidates to be evaluated. In the discussion above we have entertained two different conceptions of Gen. The first, closer to standard generative theory, is based on serial or derivational processing; some general procedure (Do-α) is allowed to make a certain single modification to the input, producing the candidate set of all possible outcomes of such modification. This is then evaluated; and the process continues with the output so determined. In this serial version of grammar, the theory of rules is narrowly circumscribed, but it is inaccurate to think of it as trivial. There are constraints inherent in the limitation to a single operation; and in the requirement that each individual operation in the sequence improve Harmony. (An example that springs to mind is the Move-x theory of rhythmic adjustments in Prince 1983; it is argued for precisely on the basis of entailments that follow from these two conditions, pp. 31–43.)

In the second, parallel-processing conception of Gen, all possible ultimate outputs are contemplated at once. Here the theory of operations is indeed rendered trivial; all that matters is what structures are admitted. Much of the analysis given in this work will be in the parallel mode, and some of the results will absolutely require it. But it is important to keep in mind that the serial/parallel distinction pertains to Gen and not to the issue of harmonic evaluation *per se*. It is an empirical question of no little interest how Gen is to be construed, and one to which the answer will become clear only as the characteristics of harmonic evaluation emerge in the context of detailed, full-scale, depth-plumbing, scholarly, and responsible analyses.

Many different theories of the structure of phonological outputs can be equally well accommodated in Gen, and the framework of Optimality Theory *per se* involves no commitment to any set of such assumptions. Of course, different structural assumptions can suggest or force different formal approaches to the way that Optimality theoretic constraints work. In this work, to implement faithfulness straightforwardly, we entertain a non-obvious assumption about Gen which will be useful in implementing the parallel conception of the theory: we will assume, following the lead of McCarthy 1979 and Itô 1986, 1989, that every output for an input *In* – every member of Gen(*In*) – includes *In* as a substructure. In the theory of syllable structure developed in Part II, Gen(/txznt/) will be a set of possible syllabifications of /txznt/ all of which contain the input string /txznt/, with each underlying segment either associated to syllable structure or left unassociated. We will interpret unassociated underlying segments as phonetically unrealized (*cf.* 'Stray Erasure'). On this conception, input segments are never 'deleted' in the sense of disappearing from the structural description; rather, they may simply be left free – unparsed. Our discussion of Berber in this section has focused on a fairly restricted subset of the full candidate set we will subsequently consider; we have considered only syllabifications in which underlying segments are in one-to-one correspondence with syllable positions. In following sections, we turn to languages which, unlike Berber, exhibit syllabifications manifesting deletion and/or epenthesis.

5.2.3.4 Binary vs. non-binary constraints

As might be suspected, it will turn out that the work done by a single non-binary constraint like Hnuc can also be done by a *set* (indeed a sub-hierarchy) of binary constraints. This will prove fundamental for the construction of the Basic Segmental Syllable Theory in §8, and we postpone treatment of the issue until then. For now it suffices simply to remark that the division of constraints into those which are binary and those which are not, a division which we have adopted earlier in this section, is not in fact as theoretically fundamental as it may at this point appear.

5.3 Pāṇini's Theorem on Constraint-ranking

One consequence of the definition of harmonic ordering is that there are conditions under which the presence of a more general constraint in a superordinate position in a hierarchy will eliminate all opportunities for a more specialized constraint in a subordinate position to have any effects in the grammar. The theorem states, roughly, that if one constraint is more general than another in the sense that the set of inputs to which one constraint applies non-vacuously includes the other's non-vacuous input set, and if the two constraints *conflict* on inputs to which the more specific applies non-vacuously, then the more specific constraint must dominate the more general one in order for its effects to be visible in the grammar. (This is an oversimplified first cut at the true result; such claims must be stated carefully.) Intuitively, the idea is that if the more specific constraint were lower-ranked, then for any input to which it applies non-vacuously, its effects would be over-ruled by the higher-ranked constraint with which it conflicts. The utility of the result is that it allows the analyst to spot certain easy ranking arguments.

We call this Pāṇini's Theorem on Constraint-ranking, in honor of the first known investigator in the area; in §7.2.1 [omitted here – Ed.], we discuss some relations to the Elsewhere Condition of Anderson 1969 and Kiparsky 1973. In this section we introduce some concepts necessary to develop a result; the proof is relegated to the Appendix [omitted here – Ed.]. The result we state is undoubtedly but one of a family of related theorems which cover cases in which one constraint hides another.

Due to the complexities surrounding this issue, we will formally state and prove the result only in the case of constraints which are Boolean at the whole-parse level: constraints which assign a single mark to an entire parse when they are violated, and no mark when they are satisfied.

(107) Dfn. Separation
A constraint $\mathbb{C}$ *separates* a set of structures if it is satisfied by some members of the set and violated by others.

(108) Dfn. Non-vacuous Application
A constraint $\mathbb{C}$ *applies non-vacuously* to an input *i* if it separates Gen(*i*), the set of candidate parses of *i* admitted by Universal Grammar.

A constraint may sometimes apply vacuously to an input, in that every possible parse of *i* satisfies the constraint. For example, in §7 [omitted here – Ed.] we will introduce a constraint Free-V which requires that stem-final vowels *not* be parsed into syllable structure. (E.g., Free-V favors *yi.li.yil.⟨i⟩* over faithful *yi.li.yi.li* 'oyster sp.' in Lardil.) Clearly, this constraint is vacuously satisfied for a stem which is not vowel-final (e.g., *kentapal* 'dugong'); all the parses of such an input meet the constraint since none of them have a stem-final vowel which is parsed!

(109) Dfn. Accepts
A constraint hierarchy $\mathbb{CH}$ *accepts* a parse P of an input *i* if P is an optimal parse of *i*.

When $\mathbb{CH}$ is the entire constraint hierarchy of a grammar, it is normally the case that only one parse P of an input *i* is optimal: the constraint set is sufficient to winnow the candidate set down to a single output. In this section we will need to consider, more generally, initial portions of the constraint hierarchy of a grammar, i.e., all the constraints from the highest-ranked down to some constraint which may not be the lowest-ranked. In these cases, $\mathbb{CH}$ will often consist of just a few constraints, insufficient to winnow the candidate set down to a single parse; in that case, $\mathbb{CH}$ will accept an entire set of parses, all equally harmonic, and all more harmonic than the competitors filtered out by $\mathbb{CH}$.

(110) Dfn. Active
Let $\mathbb{C}$ be a constraint in a constraint hierarchy $\mathbb{CH}$ and let *i* be an input. $\mathbb{C}$ is *active on i in* $\mathbb{CH}$ if $\mathbb{C}$ separates the candidates in Gen(*i*) which are admitted by the portion of $\mathbb{CH}$ which dominates $\mathbb{C}$.

In other words, the portion of $\mathbb{CH}$ which dominates $\mathbb{C}$ filters the set of candidate parses of *i* to some degree, and then $\mathbb{C}$ filters it further. When $\mathbb{C}$ is not active for an input *i* in $\mathbb{CH}$, the result of parsing *i* is not at all affected by the presence of $\mathbb{C}$ in the hierarchy.

(111) Pāṇinian Constraint Relation
Let $\mathbb{S}$ and $\mathbb{G}$ be two constraints. $\mathbb{S}$ *stands to* $\mathbb{G}$ *as special to general in a Pāṇinian relation* if, for any input *i* to which $\mathbb{S}$ applies non-vacuously, any parse of *i* which satisfies $\mathbb{S}$ fails $\mathbb{G}$.

For example, the constraint Free-V stands to Parse as special to general in a Pāṇinian relation: for any input to which Free-V applies non-vacuously (that is, to any input with a stem-final vowel V), any parse which satisfies Free-V (that is, which leaves V unparsed) must violate Parse (in virtue of leaving V unparsed). For inputs to which the more specialized constraint Free-V does *not* apply non-vacuously (C-final stems), the more general constraint Parse need not conflict with the more specific one (for C-final stems, Free-V is vacuously satisfied, but Parse is violated in some parses and satisfied in others).

Now we are finally set to state the theorem:

(112) Pāṇini's Theorem on Constraint-ranking
Let $\mathbb{S}$ and $\mathbb{G}$ stand as specific to general in a Pāṇinian constraint relation. Suppose these constraints are part of a constraint hierarchy $\mathbb{CH}$, and that $\mathbb{G}$ is active in $\mathbb{CH}$ on some input *i*. If G≫S, then $\mathbb{S}$ is not active on *i*.

In §7 [omitted here – Ed.], we will use this theorem to conclude that in the grammar of Lardil, the more specific constraint Free-V must dominate the more general constraint Parse with which it conflicts, since otherwise Free-V could not be active on an input like /yiliyili/.
[. . .]

6 Syllable Structure Typology I: The C/V Theory

6.1 The Jakobson typology

It is well known that every language admits consonant-initial syllables .CV~., and that some languages allow no others; that every language admits open syllables .~V. and that some admit only those. Jakobson puts it this way: "There are languages lacking syllables with initial vowels and/or syllables with final consonants, but there are no languages devoid of syllables with initial consonants or of syllables with final vowels" (Jakobson 1962: 526; Clements & Keyser 1983: 29).

As noted in the fundamental work of Clements & Keyser 1983, whence the quotation was cadged, these observations yield exactly four possible inventories. With the notation Σ^{XYZ} to denote the language whose syllables fit the pattern XYZ, the Jakobson typology can be laid out as follows, in terms of whether onsets and codas are obligatory, forbidden, or neither:

(113) CV Syllable Structure Typology

		Onsets	
		required	not required
Codas	forbidden	Σ^{CV}	$\Sigma^{(C)V}$
	allowed	$\Sigma^{CV(C)}$	$\Sigma^{(C)V(C)}$

There are two independent dimensions of choice: whether onsets are required (first column) or not (second column); whether codas are forbidden (row one) or allowed (row two).

The *Basic Syllable Structure Constraints*, which generate this typology, divide notionally into two groups. First, the structural or 'markedness' constraints – those that enforce the universally unmarked characteristics of the structures involved:

(114) Ons
A syllable must have an onset.

(115) –Coda
A syllable must *not* have a coda.

Second, those that constrain the relation between output structure and input:

(116) Parse
Underlying segments must be parsed into syllable structure.

(117) Fill
Syllable positions must be filled with underlying segments.

Parse and Fill are Faithfulness constraints: they declare that perfectly well-formed syllable structures are those in which input segments are in one-to-one correspondence with syllable positions.[13] Given an interpretive phonetic component that omits unparsed material and supplies segmental values for empty nodes, the ultimate force of Parse is to forbid deletion; of Fill, to forbid insertion.

It is relatively straightforward to show that the Factorial Typology on the Basic Syllable Structure Constraints produces just the Jakobson Typology. Suppose Faithfulness dominates *both* structural constraints. Then the primacy of respecting the input will be able to force violations of both Ons and –Coda. The string /V/ will be parsed as an onsetless syllable, violating Ons; the string /CVC/ will be parsed as a closed syllable, violating –Coda: this gives the language $\Sigma^{(C)V(C)}$.

When a member of the Faithfulness family is dominated by one or the other or both of the structural constraints, a more aggressive parsing of the input will result. In those rankings where Ons dominates a Faithfulness constraint, every syllable must absolutely have an onset. Input /V/ cannot be given its faithful parse as an onsetless syllable; it can either remain completely unsyllabified, violating Parse, or it can be parsed as .□V., where '□' refers to an empty structural position, violating Fill.

Those rankings in which –Coda dominates a Faithfulness constraint correspond to languages in which codas are forbidden. The imperative to avoid codas must be honored, even at the cost of expanding upon the input (*Fill) or leaving part of it outside of prosodic structure (*Parse).

In the next section, we will explore these observations in detail. The resulting Factorial construal of the Jakobson Typology looks like this (with '$\mathcal{F}$' denoting the Faithfulness set and 'F_i' a member of it):

(118) Factorial Jakobson Typology

		Onsets	
		Ons $\gg F_j$	$\mathcal{F} \gg$ Ons
Codas	–Coda $\gg F_i$	Σ^{CV}	$\Sigma^{(C)V}$
	$\mathcal{F} \gg$ –Coda	$\Sigma^{CV(C)}$	$\Sigma^{(C)V(C)}$

At this point, it is reasonable to ask whether there is any interesting difference between our claim that constraints like Ons and –Coda can be violated under domination and the more familiar claim that constraints can be *turned off* – simply omitted from consideration. The Factorial Jakobson Typology, as simple as it is, contains a clear case that highlights the distinction. Consider the language $\Sigma^{(C)V(C)}$. Since onsets are not required and codas are not forbidden, the Boolean temptation would be to hold that both Ons and –Coda are merely absent. Even in such a language, however, one can find certain circumstances in which the force of the supposedly nonexistent structural constraints is felt. The string CVCV, for example, would always be parsed .CV.CV. and never .CVC.V. Yet both parses consist of licit syllables; both are entirely faithful to the input. The difference is that .CV.CV.

satisfies Ons and –Coda while .CVC.V. violates both of them. We are forced to conclude that (at least) one of them is still active in the language, even though roundly violated in many circumstances. This is the basic prediction of ranking theory: when all else is equal, a subordinate constraint can emerge decisively. In the end, summary global statements about *inventory*, like Jakobson's, emerge through the cumulative effects of the actual parsing of individual items.

6.2 The Faithfulness interactions

Faithfulness involves more than one type of constraint. Ranking members of the Faithfulness family with respect to each other and with respect to the structural constraints Ons and –Coda yields a typology of the ways that languages can enforce (and fail to enforce) those constraints. We will consider only the Faithfulness constraints Parse and Fill (the latter to be distinguished by sensitivity to Nucleus or Ons); these are the bare minimum required to obtain a contentful, usable theory, and we will accordingly abstract away from distinctions that they do not make, such as between deleting the first or second element of a cluster, or between forms involving metathesis, vocalization of consonants, de-vocalization of vowels, and so on, all of which involve further Faithfulness constraints, whose interactions with each other and with the markedness constraints will be entirely parallel to those discussed here.

6.2.1 Groundwork

To make clear the content of the Basic Syllable Structure Constraints Ons, –Coda, Parse, and Fill, it is useful to lay out the Galilean arena in which they play. The inputs we will be considering are C/V sequences like CVVCC; that is, any and all strings of the language {C,V}*. The grammar must be able to contend with any input from this set: we do not assume an additional component of language-particular input-defining conditions; the universal constraints and their ranking must do all the work (see §9.3 for further discussion). The possible structures which may be assigned to an input are all those which parse it into syllables; more precisely, into zero or more syllables. There is no insertion or deletion of *segments* C, V.

What is a syllable? To avoid irrelevant distractions, we adopt the simple analysis that the syllable node σ must have a daughter *Nuc* and *may* have as leftmost and rightmost daughters the nodes *Ons* and *Cod*.[14] The nodes Ons, Nuc, and Cod, in turn, may each dominate C's and V's, or they may be empty. Each Ons, Nuc, or Cod node may dominate at most one terminal element C or V.

These assumptions delimit the set of candidate analyses. Here we list and name some of the more salient of the mentioned constraints. By our simplifying assumptions, they will stand at the top of the hierarchy and will be therefore unviolated in every system under discussion:

Syllable form:

(119) Nuc

Syllables must have nuclei.

(120) *Complex
No more than one C or V may associate to any syllable position node.[15]

Definition of C and V, using M(argin) for Ons and Cod and P(eak) for Nuc:
(121) *M/V
V may not associate to Margin nodes (Ons and Cod).

(122) *P/C
C may not associate to Peak (Nuc) nodes.

The theory we examine is this:

(123) Basic CV Syllable Theory
- Syllable structure is governed by the Basic Syllable Structure Constraints Ons, –Coda, Nuc; *Complex, *M/V, *P/C; Parse, and Fill.
- Of these, Ons, –Coda, Parse, and Fill may be relatively ranked in any domination order in a particular language, while the others are fixed in superordinate position.
- The Basic Syllable Structure Constraints, ranked in a language-particular hierarchy, will assign to each input its optimal structure, which is the output of the phonology.

The output of the phonology is subject to phonetic interpretation, about which we will here make two assumptions, following familiar proposals in the literature:

(124) Underparsing Phonetically Realized as Deletion
An input segment unassociated to a syllable position ('underparsing') is not phonetically realized.

This amounts to 'Stray Erasure' (McCarthy 1979, Steriade 1982, Itô 1986, 1989). Epenthesis is handled in the inverse fashion:

(125) Overparsing Phonetically Realized as Epenthesis
A syllable position node unassociated to an input segment ('overparsing') is phonetically realized through some process of filling in default featural values.

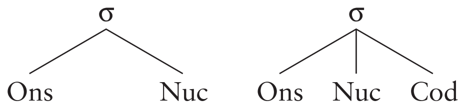
This is the treatment of epenthesis established in such works as Selkirk 1981, LaPointe & Feinstein 1982, Broselow 1982, Archangeli 1984, Kaye & Lowenstamm 1984, Piggott & Singh 1985, and Itô 1986, 1989.

The terms 'underparsing' and 'overparsing' are convenient for referring to parses that violate Faithfulness. If an input segment is not parsed in a given structure (not associated to any syllable position nodes), we will often describe this as 'underparsing' rather than 'deletion' to emphasize the character of our assumptions. For the same reason, if a structure contains an empty syllable structure node (one not associated to an input segment), we will usually speak of 'overparsing' the input rather than 'epenthesis'.

Suppose the phonology assigns to the input /CVVCC/ the following bisyllabic structure, which we write in three equivalent notations:

(126) Transcription of Syllabic Constituency Relations, from /CVVCC/

a.



b. $[_{\sigma}\ [_{\mathrm{Ons}}\ \mathrm{C}]\ [_{\mathrm{Nuc}}\ \mathrm{V}]]\ [_{\sigma}\ [_{\mathrm{Ons}}]\ [_{\mathrm{Nuc}}\ \mathrm{V}]\ [_{\mathrm{Cod}}\ \mathrm{C}]]\ \mathrm{C}$

c. .CV́.□V́C.⟨C⟩

Phonetic interpretation ignores the final C and supplies featural structure for a consonant to fill the onset of the second syllable.

The dot notation (126c) is the most concise and readable; we will use it throughout. The interpretation is as follows:

(127) Notation

a.	.X.	'the string X *is a* syllable'
b.	⟨x⟩	'the element x has no parent node; is free (unparsed)'
c.	□	'a node *Ons*, *Nuc*, or *Cod* is empty'
d.	x́	'the element x is a Nuc'

In the CV theory, we will drop the redundant nucleus-marking accent on V́. Observe that this is a 'notation' in the most inert and de-ontologized sense of the term: a set of typographical conventions used to refer to well-defined formal objects. The objects of linguistic theory – syllables here – are not to be confused with the literal characters that depict them. Linguistic operations and assessments apply to structure, not to typography.

We will say a syllable 'has an onset' if, like both syllables in the example (126), it has an Ons node, whether or not that node is associated to an underlying C; similarly with nuclei and codas.

The technical content of the Basic Syllable Structure Constraints (114–15) above can now be specified. The constraint Ons (114) requires that a syllable node σ have as its leftmost child an Ons node; the presence of the Ons node satisfies Ons whether empty or filled. The constraint –Coda (115) requires that syllable nodes have no Cod child; the presence of a Cod node violates –Coda whether or not that node is filled. Equivalently, any syllable which does not contain an onset in this sense earns its structure a mark of violation *Ons; a syllable which does contain a coda earns the mark *–Coda.

The Parse constraint is met by structures in which all underlying segments are associated to syllable positions; *each* unassociated or free segment earns a mark *Parse. This is the penalty for deletion. Fill provides the penalty for epenthesis: each unfilled syllable position node earns a mark *Fill, penalizing insertion. Together, Parse and Fill urge that the assigned syllable structure be faithful to the input string, in the sense of a one-to-one correspondence between syllable positions and segments. This is Faithfulness in the basic theory.

6.2.2 Basic CV syllable theory

We now pursue the consequences of our assumptions. One important aspect of the Jakobson Typology (113) follows immediately:

(128) THM. Universally Optimal Syllables
No language may prohibit the syllable .CV. Thus, no language prohibits onsets or requires codas.

To see this, consider the input /CV/. The obvious analysis .CV. (i.e., [$_{\sigma}$ [$_{\text{Ons}}$ C] [$_{\text{Nuc}}$ V]]) is *universally optimal* in that it violates *none* of the universal constraints of the Basic CV Syllable Theory (123). No alternative analysis, therefore, can be more harmonic. At worst, another analysis can be equally good, but inspection of the alternatives quickly rules out this possibility.

For example, the analysis .CV□. violates –CODA and FILL. The analysis .C□́.V. violates ONS in the second syllable and FILL in the first. And so on, through the infinite set of possible analyses – [.⟨C⟩V.], [.C□́.⟨V⟩.], [.□́.C □́.□V.], etc. *ad inf*. No matter what the ranking of constraints is, a form that violates even one of them can never be better than a form, like .CV., with no violations at all.

Because every language has /CV/ input, according to our assumption that every language has the same set of possible inputs, it follows that .CV. can never be prohibited under the Basic Theory.

6.2.2.1 Onsets

Our major goal is to explicate the interaction of the structural constraints ONS and –CODA with Faithfulness. We begin with onsets, studying the interaction of ONS with PARSE and FILL, ignoring –CODA for the moment. The simplest interesting input is /V/. All analyses will contain violations; there are three possible one-mark analyses:

(129) /V/ →

(1) .V. i.e., [$_{\sigma}$ [$_{\text{Nuc}}$ V]]
(2) ⟨V⟩ i.e., no syllable structure
(3) .□V. i.e., [$_{\sigma}$ [$_{\text{Ons}}$] [$_{\text{Nuc}}$ V]]

Each of these alternatives violates exactly one of the Basic Syllable Structure Constraints (114–17).

(130) Best Analyses of /V/

Analysis	Interpretation	Violation	Remarks
.V.	σ lacks Ons	*ONS	satisfies FILL, PARSE
⟨V⟩	null parse	*PARSE	satisfies ONS, FILL
.□V.	Ons is empty	*FILL	satisfies ONS, PARSE

Every language must evaluate all three analyses. Since the three candidates violate one constraint each, any comparison between them will involve weighing the importance of different violations. The optimal analysis for a given language is determined precisely by whichever of the constraints Ons, Parse, and Fill is *lowest* in the constraint hierarchy of that language. The lowest constraint incurs the least important violation.

Suppose .V. is the optimal parse of /V/. We have the following tableau:

(131) Onset Not Required

/V/	Fill	Parse	Ons
☞ .V.			*
⟨V⟩		* !	
.□V.	* !		

The relative ranking of Fill and Parse has no effect on the outcome. The violations of Parse and Fill are fatal because the alternative candidate .V. satisfies both constraints.

Of interest here is the fact that the analysis .V. involves an onsetless syllable. When this analysis is optimal, then the language at hand, by this very fact, does not absolutely require onsets. The other two inferior analyses do succeed in satisfying Ons: ⟨V⟩ achieves this vacuously, creating no syllable at all; .□V. creates an onsetful syllable by positing an empty Ons node, leading to epenthesis. So if .V. is best, it is because Ons is the lowest of the three constraints, and we conclude that the language does not require onsets. We already know from the previous section, Thm. (128), that onsets can never be forbidden. This means the following condition holds:

(132) If Parse, Fill ≫ Ons, then onsets are not required.

(The comma'd grouping indicates that Parse and Fill each dominate Ons, but that there is no implication about their own relative ranking.)

On the other hand, if Ons is not the lowest ranking constraint – if either Parse or Fill is lowest – then the structure assigned to /V/ will be consistent with the language requiring onsets. The following two tableaux lay this out:

(133) Enforcement by Overparsing (Epenthesis)

/V/	Ons	Parse	Fill
.V.	* !		
⟨V⟩		* !	
☞ .□V.			*

(134) Enforcement by Underparsing (Deletion)

/V/	FILL	ONS	PARSE
.V.		* !	
☞ ⟨V⟩			*
.□V.	* !		

These lucubrations lead to the converse of (132):

(135) If ONS dominates either PARSE or FILL, then onsets are required.

There is an important difference in status between the two ONS-related implications. To prove that something is *optional*, in the sense of 'not forbidden' or 'not required' in the inventory, one need merely exhibit one case in which it is observed and one in which it isn't. To prove that something is *required*, one must show that everything in the universe observes it. Thus, formal proof of (135) requires considering not just one trial input, as we have done, but the whole (infinite) class of strings on {C,V}* which we are taking to define the universal set of possible inputs for the Basic Theory. We postpone this exercise until the appendix [omitted here – Ed.]; in §8 we will develop general techniques which will enable us to extend the above analysis to arbitrary strings, showing that what is true of /V/ and /CV/ is true of all inputs.

The results of this discussion can be summarized as follows:

(136) Onset Theorem
Onsets are not required in a language if ONS is dominated by both PARSE and FILL.
Otherwise, onsets are required.

In the latter case, ONS is enforced by underparsing (phonetic deletion) if PARSE is the lowest-ranking of the three constraints; and by overparsing (phonetic epenthesis) if FILL is lowest.

If FILL is to be articulated into a family of node-specific constraints, then the version of FILL that is relevant here is FILL^{Ons}. With this in mind, the onset finding may be recorded as follows:

Lowest constraint	Onsets are . . .	Enforced by . . .
ONS	Not required	N/A
PARSE	Required	V 'Deletion'
FILL^{Ons}	Required	C 'Epenthesis'

6.2.2.2 *Codas*

The analysis of onsets has a direct parallel for codas. We consider the input /CVC/ this time; the initial CV provides an onset and nucleus to meet the ONS and NUC constraints, thereby avoiding any extraneous constraint violations. The final C induces the conflict between –CODA, which prohibits the Cod node, and Faithfulness, which has the effect of requiring just such a node. As in the corresponding onset situation (130), the parses which violate only one of the Basic Syllable Structure Constraints are three in number:

(137) Best Analyses of /CVC/

Analysis	Interpretation	Violation	Remarks
.CVC.	σ has Cod	*–CODA	satisfies FILL, PARSE
.CV⟨C⟩.	No parse of 2nd C	*PARSE	satisfies –CODA, FILL
.CV.C□́.	2nd Nuc is empty	*FILL	satisfies –CODA, PARSE

The optimal analysis of /CVC/ in a given language depends on which of the three constraints is lowest in the domination hierarchy. If .CVC. wins, then the language must allow codas; –CODA ranks lowest and violation can be compelled. If .CVC. loses, the optimal analysis must involve open (codaless) syllables; in this case –CODA is enforced through empty nuclear structure (phonetic V-epenthesis) if FILL is lowest, and through non-parsing (phonetic deletion of C) if PARSE is the lowest, most violable constraint. In either case, the result is that open syllables are *required*. This is a claim about the optimal parse of every string in the language, and not just about /CVC/, and formal proof is necessary; see the appendix.

The conclusion, parallel to (136), is this:

(138) Coda Theorem
Codas are allowed in a language if –CODA is dominated by both PARSE and FILL^{Nuc}.
Otherwise, codas are forbidden.

In the latter case, –CODA is enforced by underparsing (phonetic deletion) if PARSE is the lowest-ranking of the three constraints; and by overparsing (epenthesis) if FILL^{Nuc} is the lowest.

The result can be tabulated like this:

Lowest constraint	Codas are . . .	Enforced by . . .
–CODA	Allowed	N/A
PARSE	Forbidden	C ‘Deletion’
FILL^{Nuc}	Forbidden	V ‘Epenthesis’

Motivation for distinguishing the constraints FILL^{Ons} and FILL^{Nuc} is now available. Consider the languages Σ^{CV} in which only CV syllables are allowed. Here ONS and –CODA each dominate a member of the Faithfulness group. Enforcement of the dominant constraints will be required. Suppose there is only one FILL constraint, holding over all kinds of nodes. If Fill is the lowest-ranked of the three constraints, we have the following situation:

(139) Triumph of Epenthesis

Input	Optimal analysis	Phonetic
/V/	.□V.	.ℂV.
/CVC/	.CV.C□́.	.CV.C𝕍.

The single uniform Fill constraint yokes together the methods of enforcing the onset requirement ('C-epenthesis') and the coda prohibition ('V-epenthesis'). There is no reason to believe that languages Σ^{CV} are obligated to behave in this way; nothing that we know of in the linguistic literature suggests that the appearance of epenthetic onsets requires the appearance of epenthetic nuclei in other circumstances. This infelicitous yoking is avoided by the natural assumption that Fill takes individual node-classes as an argument, yielding Fill$^{\mathrm{Nuc}}$ and Fill$^{\mathrm{Ons}}$ as the actual constraints. In this way, the priority assigned to filling Ons nodes may be different from that for filling Nuc nodes.[16]

It is important to note that onset and coda distributions are completely independent in this theory. Any ranking of the onset-governing constraints {Ons, Fill$^{\mathrm{Ons}}$, Parse} may coexist with any ranking of coda-governing constraints {–Coda, Fill$^{\mathrm{Nuc}}$, Parse}, because they have only one constraint, Parse, in common. The universal factorial typology allows all nine combinations of the three onset patterns given in (136) and the three coda patterns in (138). The full typology of interactions is portrayed in the table below. We use subscripted *del* and *ep* to indicate the phonetic consequences of enforcement; when both are involved, the onset-relevant mode comes first.

(140) Extended CV Syllable Structure Typology

			Onsets		
			required		not required
			Ons, Fill$^{\mathrm{Ons}}$ ≫ Parse	Ons, Parse ≫ Fill$^{\mathrm{Ons}}$	Parse, Fill$^{\mathrm{Ons}}$ ≫ Ons
Codas	forbidden	–Coda, Fill$^{\mathrm{Nuc}}$ ≫ Parse	$\Sigma^{CV}{}_{del,del}$	$\Sigma^{CV}{}_{ep,del}$	$\Sigma^{(C)V}{}_{del}$
		–Coda, Parse ≫ Fill$^{\mathrm{Nuc}}$	$\Sigma^{CV}{}_{del,ep}$	$\Sigma^{CV}{}_{ep,ep}$	$\Sigma^{(C)V}{}_{ep}$
	allowed	Parse, Fill$^{\mathrm{Nuc}}$ ≫ –Coda	$\Sigma^{CV(C)}{}_{del}$	$\Sigma^{CV(C)}{}_{ep}$	$\Sigma^{(C)V(C)}$

If we decline to distinguish between the Faithfulness constraints, this simplifies to the Jakobson Typology of (118).

6.2.3 The theory of epenthesis sites

The chief goal of syllabification-driven theories of epenthesis is to provide a principled account of the location of epenthetic elements (Selkirk 1981, Broselow 1982, LaPointe and Feinstein 1982, Itô 1986, 1989). Theories based on manipulation of the segmental string are capable of little more than summary stipulation on this point (e.g., Levin 1985: 331; see Itô 1986: 159, 1989 for discussion). The theory developed here entails tight restrictions on the distribution of empty nodes in optimal syllabic parses, and therefore meets this goal. We confine attention to the premises of the Basic CV Syllable Structure Theory, which serves as the foundation for investigation of the theory of epenthesis, which ultimately involves segmental and prosodic factors as well.

There are a few fundamental observations to make, from which a full positive characterization of syllabically motivated epenthesis emerges straightaway:

(141) Prop. 1. $*[\ ]_{\text{Cod}}$
Coda nodes are never empty in any optimal parse.

Structures with unfilled Cod can never be optimal; there is always something better. To see this, take a candidate with an unfilled Cod and simply remove that one node. This gives another candidate which has one less violation of –Coda and one less violation of Fill. Since removing the node has no other effects on the evaluation, the second candidate must be superior to the first. (To show that something is *non*-optimal, we need merely find something better: we don't have to display the best.)

We know from the earlier discussion that Ons and Nuc must be optimally unfilled in certain parses under certain grammars. So the remaining task is to determine the conditions under which these nodes must be posited and left empty.

(142) Prop. 2. $*.(\square)\acute{\square}.$
A whole syllable is never empty in any optimal parse.

The same style of argument applies. Consider a parse that has an entirely empty syllable. Remove that syllable. The alternative candidate thereby generated is superior to the original because it has (at least) one less Fill^{Nuc} violation and no new marks. The empty syllable parse can always be bested and is therefore never optimal.

Of course, in the larger scheme of things, whole syllables can be epenthesized, the canonical examples being Lardil and Axininca Campa (Hale 1973, Klokeid 1976, Itô 1986, Wilkinson 1988, Kirchner 1992a; Payne 1981, Payne et al. 1982, Spring 1990, Black 1991, McCarthy & Prince 1993). In all such cases, it is the impact of additional constraints that forces whole-syllable epenthesis. In particular, when the prosody/morphology interface constraints like Lx≈Pr ["every lexical word corresponds to a prosodic word" – Ed.] are taken into account, prosodic minimality requirements can force syllabic epenthesis, as we will see for Lardil in §7 [omitted here – Ed.].

(143) Prop. 3. *.(□)□́C.
No syllable can have *Cod* as its only filled position.

Any analysis containing such a syllable is bested by the alternative in which the content of this one syllable (namely 'C') is parsed instead as .C□́.. This alternative incurs only the single mark *FILL^{Nuc}, but the closed-syllable parse .(□)□́C. shares this mark and violates –CODA as well. (In addition, the closed-syllable parse must also violate either ONS or FILL^{Ons}.)

Such epentheses are not unknown: think of Spanish /slavo/ → *e*slavo and Arabic /ħmarar/ → *ʔi*ħmarar. We must argue, as indeed must all syllable theorists, that other constraints are involved (for Arabic, see McCarthy & Prince 1990).

(144) Prop. 4. *[ ][ ]
Adjacent empty nodes cannot occur in an optimal parse.

Propositions 1, 2, and 3 entail that [ ][ ] cannot occur inside a syllable. This leaves only the intersyllabic environment .C □́.□V~. This bisyllabic string incurs two marks, *FILL^{Nuc} and *FILL^{Ons}. Consider the alternative parse in which the substring /CV/ is analyzed as tautosyllabic .CV~. This eliminates both marks and incurs no others. It follows that two adjacent epentheses are impossible.

We now pull these results together into an omnibus characterization of where empty nodes can be found in optimal parses.

(145) FILL Violation THM. Location of Possible Epenthesis Sites
Under the Basic Syllable Structure Constraints, epenthesis is limited to the following environments:
a. Onset, when Nucleus is filled:
.□V.
.□VC.
b. Nucleus, when Onset is filled:
.C □́.
.C □́C.
Furthermore, two adjacent epentheses are impossible, even across syllable boundaries.

The last clause rules out, for example, *.C □́.□V. We note that the Fill Violation Theorem will carry through in the more complex theory developed below in §8, in which the primitive C/V distinction is replaced by a graded sonority-dependent scale. [. . .]

8 Universal Syllable Theory II: Ordinal Construction of C/V and Onset/Coda Licensing Asymmetry

[Limitations of space do not permit inclusion of all of §8 of Prince and Smolensky (1993). But because the results in §8 are important and far-reaching, I have included the introduction to §8, which summarizes those results. – Ed.]

Syllabification must reconcile two conflicting sources of constraint: from the bottom up, each segment's inherent featural suitability for syllable peak or margin; and from top down, the requirements that syllables have certain structures and not others. The core conflict can be addressed in its most naked form through the idealization provided by CV theory. Input C's need to be parsed as margins; input V's need to be parsed as peaks. Syllables need to be structured as Onset-Peak-Coda; ideally, with an onset present and a coda absent. In the Basic Theory, only one input segment is allowed per syllable position. Problematic inputs like /CCVV/ are ones which bring the bottom-up and top-down pressures into conflict. These conflicts are resolved differently in different languages, the possible resolutions forming the typology explored in §6.

The CV theory gives some articulation to the top-down pressures: syllable shapes deviate from the Onset-Peak ideal in the face of bottom-up pressure to parse the input. By contrast, the bottom-up is construed relatively rigidly: C and V either go into their determined positions, or they remain unparsed. In real syllabification, of course, a richer set of possibilities exists. A segment ideally parsed as a peak may actually be parsed as a margin, or vice versa, in response to top-down constraints on syllable shape. One of the most striking examples of the role of optimality principles in syllabification, Tashlhiyt Berber (§2), exploits this possibility with maximal thoroughness. Berber syllabification on the one hand and CV syllabification on the other constitute extremes in the flexibility with which input segments may be parsed into different syllable positions in response to top-down pressure. In between the extremes lies the majority of languages, in which some segments can appear only as margins (like C in the CV theory), other segments only as peaks (like V), and the remaining segments, while ideally parsed into just one of the structural positions, can under sufficient top-down pressure be parsed into others.

In this section we will seek to unify the treatments of the two extremes of syllabification, Berber and the CV theory. Like the CV theory, the theory developed here will deal with an abstract inventory of input segments, but instead of just two abstract segments, each committed to a structural position, the inventory will consist of abstract elements distinguished solely by the property of *sonority*, taken to define a strict order on the set of elements. For mnemonic value we denote these elements *a, i, . . . , d, t*; but it should be remembered that all dimensions other than sonority are idealized away. In the CV theory, the universally superordinate constraints *M/V and *P/C prohibit parsing V as a margin or C as a peak. In the more realistic theory we now turn to, the corresponding constraints are not universally superordinate: the constraints against parsing any segment α as a margin (*M/α) or as a peak (*P/α) may vary cross-linguistically in their rankings. What Universal Grammar requires is only that more sonorous segments make more harmonic peaks and less harmonic margins.

From these simple assumptions there will emerge a universal typology of inventories of possible onsets, peaks, and codas. The inventories will turn out to be describable in terms of derived parameters π_{Ons}, π_{Nuc}, and π_{Cod}, each with values ranging over the sonority order. The margin inventories are the sets of segments *less* sonorous than the corresponding parameter values π_{Ons} or π_{Cod}, and the peak inventory is the set of segments *more* sonorous than the value of π_{Nuc}, Languages in which $\pi_{Ons} > \pi_{Nuc}$ are therefore languages with ambidextrous segments, which can be parsed as either onset or nucleus. The following diagram pictures the situation; the double line marks the zone of overlap.

(185) Languages with Ambidextrous Segments

← ← *onsets* ← $\boldsymbol{\pi}_{\mathrm{Ons}}$

→ greater sonority

$\boldsymbol{\pi}_{\mathrm{Nuc}}$ → *nuclei* → →

The theory entails a universal licensing asymmetry between onsets and codas: codas can contain only a subset, possibly strict, of the segments appearing in onsets. This fundamental licensing asymmetry will be shown to follow from the asymmetry between Onset and Coda in the Basic Syllable Structure Constraints. From the fact that Onsets should be present and Codas absent, it will follow in the theory that Coda is a weaker licenser.[17] To our knowledge, no other approach has been able to connect the structural propensities of syllables with the licensing properties of syllabic positions, much less to derive one from the other. This is surely a significant result, one that indicates that the theory is on the right track in a fundamental way. The exact nature of the obtained licensing asymmetry has some empirical imperfections which can be traced to the oversimplified analysis of codas in the internal structure of the syllable, and we suggest possible refinements.

The present section constitutes a larger-scale exploration of our general line of attack on the problem of universal typology. Universal Grammar provides a fixed set of constraints, which individual languages rank differently in domination hierarchies; UG also provides certain universal conditions on these hierarchies, which all languages must respect. The results obtained here involve a further development of the basic idea: *parametrization by ranking*. The parameters π_{Ons}, π_{Nuc}, and π_{Cod} are epiphenomenal, in that they do not appear at all in Universal Grammar, or indeed, in particular grammars: they are not, for example, mentioned in any constraint. These parameters are not explicitly set by individual languages. Rather, individual languages simply rank the universal constraints, and it is a *consequence* of this ranking that the (derived, descriptive) parameters have the values they do in that language. The procedures for reading off these parameter values from a language's constraint domination hierarchy are not, in fact, entirely obvious.

The analysis developed here introduces or elaborates several general concepts of the theory:

(186) Push/Pull Parsing

The parsing problem is analyzed in terms of the direct conflict between two sets of constraints:

a. *ASSOCIATE* constraints
   PARSE, FILL, ONS, and the like, which penalize parses in which input segments or structural nodes *lack* structural associations to a parent or child;
b. *DON'T-ASSOCIATE* constraints
   *M/V, *P/C, and –CODA and their like, which penalize parses which *contain* structural associations of various kinds.

(187) Universal Constraint Sub-Hierarchies

The DON'T-ASSOCIATE constraints *M/V, *P/C, superordinate in the CV theory, are replaced by an articulated set of anti-association constraints *M/*a*, *M/*i*, . . . , *M/*d*, *M/*t*; *P/*a*, *P/*i*, . . . , *P/*d*, *P/*t* which penalize associations between Margin or Peak nodes on the one hand and particular input segments on the other. Universal Grammar requires that the domination hierarchy of each language rank these constraints *M/α, *P/α relative to one another in conformity with the following universal domination conditions:

*M/*a* ≫ *M/*i* ≫ . . . ≫ *M/*d* ≫ *M/*t* (Margin Hierarchy)
*P/*t* ≫ *P/*d* ≫ . . . ≫ *P/*i* ≫ *P/*a* (Peak Hierarchy)

The Margin Hierarchy states that it's less harmonic to parse *a* as a margin than to parse *i* as margin, less harmonic to parse *i* as a margin than *r*, and so on down the sonority ordering. The Peak hierarchy states that it's less harmonic to parse *t* as a peak than *d*, and so on up the sonority order.

(188) Associational Harmony

The universal Margin and Peak Hierarchies ensure the following universal ordering of the Harmony of possible associations:

M/*t* ≻ M/*d* ≻ . . . ≻ M/*i* ≻ M/*a*
P/*a* ≻ P/*i* ≻ . . . ≻ P/*d* ≻ P/*t*

These represent the basic assumption that the less sonorous an element is, the more harmonic it is as a margin; the more sonorous, the more harmonic it is as a Peak.

(189) Prominence Alignment

These universal rankings of constraints (187) and orderings of associational Harmonies (188) exemplify a general operation, Prominence Alignment, in which scales of prominence along two phonological dimensions are harmonically aligned. In this case, the first scale concerns prominence of structural positions within the syllable:

Peak > Margin

while the second concerns inherent prominence of the segments as registered by sonority:

a > *i* > . . . > *d* > *t*

(190) Encapsulation

It is possible to greatly reduce the number of constraints in the theory by encapsulating sets of associational constraints *M/α, *P/α into defined constraints which explicitly refer to *ranges* of sonority. This corresponds to using a coarse-grained sonority scale, obtained by collapsing distinctions. This must be done on a language-specific basis, however, in a way sensitive to the language's total constraint hierarchy: which sets of associational constraints can be successfully encapsulated into composite constraints depends on how the language inserts other constraints such as PARSE, FILL, ONS, and so on, into the Margin and Peak Hierarchies, and how these two Hierarchies are interdigitated in the language. Encapsulation opens the way to developing a substantive theory of the sonority classes operative in syllable structure phenomena.

Along with these conceptual developments, this section introduces a collection of useful techniques for reasoning about constraint domination hierarchies in complex arenas such as that defined by the segmental syllable theory. A few of these techniques are:

(191) Harmonic Bounding for Inventory Analysis
In order to show that a particular kind of structure φ is not part of a universal or language-particular inventory, we consider any possible parse containing φ and show constructively that there is some competing parse (of the same input) which is more harmonic; thus no structure containing φ can ever be optimal, as it is always bounded above by at least one more-harmonic competitor. (This form of argument is used to establish the distribution of epenthesis sites in §6.2.3.)

(192) Cancellation/Domination Lemma
In order to show that one parse B is more harmonic than a competitor A which does not incur an identical set of marks, it suffices to show that every mark incurred by B is either (i) cancelled by an identical mark incurred by A, or (ii) dominated by a higher-ranking mark incurred by A. That is, for every constraint violated by the more harmonic form B, the losing competitor A either (i) matches the violation exactly, or (ii) violates a constraint ranked higher.

(193) The Method of Universal Constraint Tableaux
A generalization of the method of language-specific constraint tableaux is developed; it yields a systematic means for using the Cancellation/Domination Lemma to determine which parse is optimal, not in a specific language with a given constraint hierarchy, but in a typological class of languages whose hierarchies meet certain domination conditions but are otherwise unspecified.

[. . .]

9 Inventory Theory and the Lexicon

All grammatical constraints are violable, in principle. A constraint such as Ons, 'syllables have onsets', in and of itself and prior to its interaction with other constraints, does not assert that syllables lacking onsets are impossible, but rather that they are simply less harmonic than competitors possessing onsets. Its function is to sort a candidate set by measuring adherence to (equivalently: divergence from) a formal criterion. Constraints therefore define relative rather than absolute conditions of ill-formedness, and it may not be immediately obvious how the theory can account for the absolute impossibility of certain structures, either within a given language or universally. Yet in the course of the preceding analyses we have seen many examples of how Optimality Theory explains language-particular and universal limits to the possible. In this section, we identify the general explanatory strategy that these examples instantiate, and briefly illustrate how this strategy can be applied to explaining segmental inventories. We then consider implications for the lexicon, proposing a general induction principle which entails that the structure of the constraints in a language's grammar is strongly reflected in the content of its lexicon. This principle, Lexicon Optimization, asserts that when a learner must choose among candidate underlying forms which are equivalent in that they all produce the same phonetic output and in that they all subserve the morphophonemic relations of the language equally well, the underlying form chosen is the one whose output parse is most harmonic.

9.1 Language-particular inventories

We begin by examining a simple argument which illustrates the central challenge of accounting for absolute ill-formedness in a theory of relative well-formedness:

> For Optimality Theory, syllables without onsets are not absolutely ill-formed, but only relatively. The syllable .VC. (for example) is *more* ill-formed than the syllable .CV., but .VC. is not *absolutely* ill-formed. How can Optimality Theory bar .VC. from any language's syllable inventory?
>
> What Optimality Theory would need in order to outlaw such syllables is some additional mechanism, like a *threshold* on ill-formedness, so that when the graded ill-formedness of syllables passes this threshold, the degree of ill-formedness becomes absolutely unacceptable.

The fallacy buried in this argument has two facets: a failure to distinguish the inputs from the outputs of the grammar, coupled with an inappropriate model of grammar in which the ill-formed are those *inputs* which are rejected by the grammar. In Optimality Theory, the job of the grammar is not to accept or reject *inputs*, but rather to assign the best possible structure to every input. The place to look for a definition of ill-formedness is in the set of *outputs* of the grammar. These outputs are, by definition, well-formed; so what is ill-formed – absolutely ill-formed – is any structure which is never found among the outputs of the grammar. To say that .VC. syllables are not part of the inventory of a given language is not to say that the grammar rejects /VC/ and the like as input, but rather that no *output* of the grammar ever contains .VC. syllables.

We record this observation in the following remark:

(277) Absolute Ill-formedness
A structure φ is (absolutely) ill-formed with respect to a given grammar iff there is no input which when given to the grammar leads to an output that contains φ.

Note further that in a demonstration that .VC. syllables are ill-formed according to a given grammar, the input /VC/ has no *a priori* distinguished status. We need to consider every possible input in order to see whether its output parse contains a syllable .VC. Of course, /VC/ is a promising place to start the search for an input which would lead to such a parse, but, before concluding that .VC. syllables are barred by the grammar, we must consider all other inputs as well. Perhaps the optimal parse of /C/ will turn out to be .□́C., providing the elusive .VC. syllable. It may well be possible to show that if *any* input leads to .VC. syllables, then /VC/ will – but in the end such an argument needs to be made.

If indeed .VC. syllables are ill-formed according to a given grammar, then the input /VC/ must receive a parse other than the perfectly faithful one: .VC. At least one of the Faithfulness constraints Parse and Fill must be violated in the optimal parse. We can therefore generally distinguish two paths that the grammar can follow in order to parse such problematic inputs: violation of Parse, or violation of Fill. The former we have called 'underparsing' the input, and in some other accounts would correspond to a 'deletion repair strategy'; the latter, overparsing, corresponds to an 'epenthesis repair strategy' (cf. §1.2). (In §10.3 [omitted here – Ed.] we explicitly compare Optimality Theory to some repair theories.) These two means by which a grammar may deal with problematic inputs were explicitly explored in the Basic CV Syllable Structure Theory of §6. There we found that .VC. syllables were barred by either

(i) requiring onsets: ranking either Parse or Fill^{Ons} lower than Ons; or
(ii) forbidding codas: ranking either Parse or Fill^{Nuc} lower than –Coda.

One particularly aggressive instantiation of the underparsing strategy occurs when the optimal structure assigned by a grammar to an input is the *null* structure: no structure at all. This input is then grammatically completely unrealizable, as discussed in §4.3.1 [omitted here – Ed.] . There is some subtlety to be reckoned with here, which turns on what kinds of structure are asserted to be absent in the null output. In one sense, the *null* means 'lacking in realized phonological content', with maximal violation of Parse, a possibility that can hardly be avoided in the candidate set if underparsing is admitted at all. In another sense, the null form will fail to provide the morphological structure required for syntactic and semantic interpretation, violating M-Parse [a constraint that requires the structural realization of input morphological properties – Ed.]. To achieve full explicitness, the second move requires further development of the morphological apparatus; the first requires analogous care in formulating the phonetic interpretation function, which will be undefined in the face of completely unparsed phonological material. In this discussion, we will gloss over such matters, focusing on the broader architectural issues.

It would be a conceptual misstep to characterize this as *rejection of the input* and to appeal to such rejection as the basis of a theory of absolute ill-formedness. For example, it would be wrong to assert that a given grammar prohibits .VC. syllables because the input /VC/ is assigned the null structure; this is a good hint that the grammar may bar .VC. syllables, but what needs to be demonstrated is that *no* input leads to such syllables. In addition, a grammar which assigns some non-null structure to /VC/, for example .□V.⟨C⟩, might nonetheless prohibit .VC. syllables.

Subject to these caveats, it is clear that assigning null structure to an input is one means a grammar may use to prevent certain structures from appearing in the output. The Null Parse is a possible candidate which must always be considered and which may well be optimal for certain particularly problematic inputs. We have already seen two types of examples where null structures can be optimal. The first example emerged in the analysis of Latin minimal word phenomenon in §4.3.1, where, given a certain interpretation of the data, under the pressure of FtBin ["feet are binary at some level of analysis, mora (μ) or syllable (σ)" – Ed.] and Lx≈Pr, the optimal parse of the monomoraic input is null (but see Mester 1992: 19–23). The second was in the CV Syllable Structure Theory of §6, where it was shown that the structure assigned to /V/ is null in any language requiring onsets and enforcing Ons by underparsing: that is, where Parse is the least significant violation, with {Ons, Fill^{Ons}} ≫ Parse.

9.1.1 Harmonic Bounding and nucleus, syllable, and word inventories

Absolute ill-formedness, explicated in (277), is an emergent property of the interactions in a grammar. Showing that a structure φ is ill-formed in a given language requires examination of the system. One useful strategy of proof is to proceed as follows. First we let A denote an arbitrary candidate parse which contains the (undesirable) structure φ. Then we show how to modify *any such analysis* A to produce a particular (better) competing candidate parse B of the same input, where B does not contain φ and where B is provably more harmonic than A. This is sufficient to establish that no structure containing φ can ever be optimal. The structure φ can never occur in any output of the grammar, and is thus absolutely ill-formed. We can call this method of proof "Harmonic Bounding" – it establishes that every parse containing the structure φ is bettered by, bounded above by, one that lacks φ.

The strategy of Harmonic Bounding was implicitly involved, for example, in the analysis of the minimal word phenomenon (§4.3.1). In this case, the impossible structure is $\varphi = [\mu]_{\text{PrWd}}$. We examined the most important type of input, a monomoraic one like /re/, and showed that the analysis containing φ, $A = [[\text{ré}]_F]_{\text{PrWd}}$, is less harmonic than a competitor B = ⟨re⟩, the Null Parse, which lacks φ. The method of constructing B from A is simply to replace structure with no structure.

To complete the demonstration that the Latin constraint hierarchy allows no monomoraic words in the output, we must consider every input that could give rise to a monomoraic word. We need to examine inputs with less than one mora, showing that they do not get overparsed as a single empty mora: $[[\acute{\square}_\mu]_F]_{\text{PrWd}}$. We also must consider inputs of more than one mora, showing that these do not get underparsed, with only one mora being parsed into the PrWd: $[[\mu]_F]_{\text{PrWd}}\langle\mu\ldots\rangle$. Both of these are also harmonically bounded by the Null Parse of the relevant inputs. On top of whatever violation marks are earned by complete structuring of monomoraic input – marks that are already sufficient to establish the superiority of the Null Parse – these moraic over- and underparses incur *Fill and *Parse marks as well, and it is even clearer that a monomoraic parse cannot be optimal.

Similarly, in the analysis of Lardil in §7 [omitted here – Ed.], we provided the core of the explanation for why no words in its inventory can be monomoraic. The result is the same as in Latin, but enforcement of Lx≈Pr and FtBin for monomoraic inputs is now by overparsing rather than by underparsing, due to differences in the constraint ranking. The structure we wish to exclude is again $\varphi = [\mu]_{PrWd}$, and, as in Latin, we examined monomoraic inputs such as /maɽ/ to see if their parses contained φ. In all such cases, the optimal parses are bisyllabic competitors B, the second mora of which is unfilled. We also examined vowel-final bimoraic inputs like /wiʈe/ because, for longer inputs, a final vowel is optimally unparsed, a pattern which would lead to monomoraicity if universally applied. However, both moras in bimoraic inputs must be parsed, so again we fail to produce a monomoraic output. Inputs with three or more moras leave a final vowel unparsed, but parse all the others (184). Thus, there are no inputs, long or short, which produce monomoraic outputs.

It is worth emphasizing that, even though the lack of monomoraic words in the Latin and Lardil inventories is a result of the high ranking of Lx≈Pr and FtBin in the domination hierarchy, it would be distinctly incorrect to summarize the Optimality Theory explanation as follows: "Lx≈Pr and FtBin are superordinate therefore unviolated, so any monomoraic input is thereby rendered absolutely ill-formed." An accurate summary is: "Lx≈Pr and FtBin dominate a Faithfulness constraint (Parse in Latin; Fill in Lardil), so for any input at all – including segmentally monomoraic strings as a special case – monomoraic parses are always less harmonic than available alternative analyses (Null Parse for Latin, bisyllable for Lardil); therefore outputs are never monomoraic."

Successful use of the Harmonic Bounding argument does not require having the optimal candidate in hand; to establish *φ in the absolute sense, it is sufficient to show that there is always a B-without-φ that is better than any A-with-φ. Whether any such B is optimal is another question entirely. This can be seen clearly in the kind of argument pursued repeatedly above in the development of the Basic Segmental Syllable Theory in §8. For example, as part of the process of deriving the typology of segmental inventories licensed by various syllable positions, we showed that the inventory of possible nuclei could not include a segment α in any language in which $*P/\alpha \gg \{\text{Fill}^{Nuc}, *M/\alpha\}$).[18] These are languages in which it is

(i) more important to keep α out of the Nucleus (P = 'peak') than to fill the Nucleus, and
(ii) more important to keep α out of the Nucleus than to keep it out of the syllable margins.

The φ we want to see eliminated is the substructure Nuc/α, in which the segment α is dominated by the node Nucleus. Let A denote an arbitrary parse containing Nuc/α = ά, so that a segment α appearing in the input string is parsed as a nucleus: A = ~ά~. The bounding competitor B is identical to A except that the structure in question, Nuc/α, has been replaced by the string in which α is an onset sandwiched between two empty nuclei: B = ~□́.α□́~. In terms of the slash-for-domination notation, the crucial replacement pattern relating A to B can be shown as

A = . . . Nuc/**α** . . . B = . . . Nuc/□. Ons/**α** Nuc/□. . . .

We have then the following argument:

(278) Harmonic Bounding Argument, showing α is an impossible nucleus

a. Assumed constraint ranking
$*P/\alpha \gg \{\textsc{Fill}^{Nuc}, *M/\alpha\}$

b. Structures

i. $\varphi = \acute{\alpha}$ (segment α qua nucleus)

ii. $A = \sim \acute{\alpha} \sim$ (any parse taking α to be a nucleus)

iii. $B = \sim \acute{\square}.\alpha\acute{\square} \sim$ (analysis A modified in a specific way to make α nonnuclear)

c. Argument: show that B bests A.

It should be clear that B is always more harmonic than A in the given languages. The mark *P/α incurred by nucleizing α in A is worse than both the marks *M/α (for marginalizing α) and $*\textsc{Fill}^{Nuc}$ (for positing empty nuclei) that are incurred by B. Hence, in such a grammar the optimal parse can never include φ = Nuc/α, no matter what the input. The conclusion is that α is not in the inventory of possible nuclei for these languages. However, we cannot conclude that every occurrence of α is in onset position, as in the bounding analysis B, or indeed, without further argument, that *any* occurrence of α is in onset position. There may be other analyses that are even more harmonic than B in specific cases; but we are assured that α will never be a nucleus in any of these.

The Harmonic Bounding strategy is implicitly involved in a number of results derived above. Samek-Lodovici (1992) makes independent use of the same method of proof (taking B to be a kind of Null Parse) to establish the validity of his Optimality theoretic analysis of morphological gemination processes.

9.1.2 Segmental inventories

Having illustrated the way prosodic inventories are delimited, from the structural level of the syllable position (e.g., Nuc) up through the syllable itself to the word, we can readily show how the technique extends downward to the level of the segment. Now we take as inputs not strings of already formed segments, but rather strings of feature sets. These must be optimally parsed into segments by the grammar, just as (and at the same time as) these segments must be parsed into higher levels of phonological structure. The segmental inventory of a language is the set of segments found among the optimal *output parses* for all possible inputs.

We now illustrate this idea by analyzing one particular facet of the segmental inventory of Yidin^y (Kirchner 1992b). Our scope will be limited: the interested reader should examine the more comprehensive analysis of the Yidin^y inventory developed in Kirchner's work, which adopts the general Optimality Theory approach to inventories, but pursues different analytic strategies from the ones explored here.

The consonant inventory of Yidin^y looks like this:

Labial	*Coronal*	*Retroflex coronal*	*Palatalized coronal*	*Velar*
b	d		d^y	g
m	n		n^y	ŋ
	l			
	r	ɽ		

Here [r] is a "trilled apical rhotic" and [ɽ] an "apical postalveolar (retroflex) rhotic continuant," according to Dixon (1977: 32).

Complex articulations are found only at coronal place of articulation; this is the generalization we wish to derive. The complexities include palatalization in [d^y, n^y] and the retroflexion in [ɽ]. We propose to analyze the normal and palatalized coronals as follows, along lines developed in Clements 1976, 1991 and Hume 1992:

(279) Representation of Coronals

a. Normal

```
PLACE
  |
 C-Pl
  |
 Cor
```

b. Palatalized

```
      PLACE
     /     \
  C-Pl     V-Pl
   |         |
  Cor       Cor
```

In line with the findings of Gnanadesikan 1992 and Goodman 1993, we hold that retroflexion is dorsalization rather than coronalization (as it is in Kirchner 1992b). To focus the discussion, we will deal only with the coronalized coronals. As a compact representation of these structures, we will use bracketing to denote the structure of the Place node, according to the following scheme:

(280) Bracketing Notation for Place Geometry

a. [α] 'feature α occupies C-Place, there is no V-Place' node

b. [α β] 'feature α occupies C-Place and feature β occupies V-Place'

With this notation, structure (279a) is denoted by [Cor] and structure (279b) is denoted by [Cor Cor].

In this representational system, the palatalized coronals are literally *complex*, with two places of articulation, while the other, unmarked coronals are literally simple. The generalization is now clear: of all the possible structurally complex places, only one is admitted into the Yidin^y lexicon: the one in which the primary and secondary places are both Cor – generally held to be the unmarked place of articulation (Avery & Rice 1989, and see especially the papers in Paradis & Prunet 1991, reviewed in McCarthy & Taub 1992).

Informally speaking, two generalizations are involved:

(281) Coronal Unmarkedness (Observation)
"Don't have a place of articulation other than Coronal."

(282) Noncomplexity (Observation)
"Don't have structurally complex places of articulation."

Our goal is to analyze the interaction between coronal unmarkedness and complexity markedness. This is of particular interest because it exemplifies a common pattern of interaction: each constraint is individually violable, but no form is admitted which violates both of them at once. There are consonants with single Lab or Vel specifications, violating coronal unmarkedness, and there are consonants with two place specifications, violating noncomplexity. But no consonant with any noncoronal place feature has a complex specification. We dub this generalization pattern *banning the worst of the worst*.

The worst-of-the-worst interaction is absent in the Basic CV Syllable Structure Theory. The two dimensions of well-formedness there – Onset well-formedness (more harmonic when present) and Coda well-formedness (more harmonic when absent) – operate independently. Requiring Onset, prohibiting Coda will generate the entire Jakobson Typology; the *worst-of-the-worst languages do not appear. Such a language would allow onsets to be absent, and codas to be present, but not in the same syllable; its inventory would include CV, V, CVC but exclude VC. This inventory is not possible according to the Basic CV Syllable Structure Theory, and we know of no reason to believe that this is anything but a desirable result.

The techniques already developed enable a direct account of the interaction between coronality and structural complexity. We assume that the input to the grammar is a string of root nodes each with a set of (unassociated) features. The output is an optimal parse in which these features are associated to root nodes (with the root nodes associated to syllable-position nodes, and so on up the prosodic hierarchy). To minimize distractions, let's assume a universally superordinate constraint requiring root nodes to have a child PL (Place) node. (This parallels the assumption made in §6 that the syllable node always has a child Nuc, due to universal superordinance of the relevant constraint NUC.) For the present analysis of consonant inventories, we similarly assume a universally superordinate constraint, or restriction on Gen, to the effect that in consonants the presence of V-Place entails the presence of C-Place. (This head/dependent type of relationship is conveniently encoded in the bracketing notation of (280), because the configuration [α] is always interpreted as 'α is C-Pl'.)

Our focus will be on which of the place features in an input feature set gets associated to the PL node. As always, unparsed input material is phonetically unrealized; underparsing is therefore a principal means of barring certain feature combinations from the inventory. If certain infelicitous combinations of features should appear in an input feature set, the grammar may simply leave some of them unparsed; the feature combinations which surface phonetically define a segmental inventory from which certain ill-formed feature combinations have been absolutely banned.

In Yidin^y, the feature set {Cor, Cor} gets completely parsed. Both Cor features are associated to the PL node in the optimal parse, and the segment surfaces as *d*^{y} or *n*^{y}, depending on which other features are in the set. On the other hand, the set {Lab, Lab} does *not* get completely parsed: the inventory does not include complex labials. In contrast, the unit set {Lab} does get completely parsed; the language has simple labials.

To minimize notation we will deal only with Cor and Lab; any other noncoronal place features receive the same analysis for present purposes as Lab.

Coronal unmarkedness can be formally stated as the following universal Harmony scale:

(283) Coronal Unmarkedness: Harmony Scale
PL/Cor ≻ PL/Lab

The notation 'PL/Cor' refers to a structural configuration in which PL dominates Cor, understood to be through some intermediate node – either C-Pl or V-Pl. The simplest theory, which we develop here, treats the two intermediate nodes alike for purposes of Harmony evaluation.

Following the same analytic strategy as for Universal Syllable Position/Segmental Sonority Prominence Alignment of §8, we convert this Harmony scale to a domination ranking of constraints on associations:

(284) Coronal Unmarkedness: Domination Hierarchy
*PL/Lab ≫ *PL/Cor

Following the general 'Push/Pull' approach to grammatical parsing summarized in §8, the idea here is that all associations are banned, some more than others. The constraint hierarchy (284) literally says that it is a more serious violation to parse labial than to parse coronal. Coronal unmarkedness in general means that to specify PL as coronal is the least offensive violation. The constraint *PL/Lab is violated whenever Lab is associated to a PL node; this constraint universally dominates the corresponding constraint *PL/Cor because Lab is a less well-formed place than Cor. In addition to these two associational constraints we have the usual Faithfulness constraints Parse and Fill. They are parametrized by the structural elements they pertain to; in the present context, they take the form:

(285) $\text{Parse}^{\text{Feat}}$
An input feature must be parsed into a root node.

(286) Fill^{PL}
A PL node must not be empty (unassociated to any features).

Just as with the segmental syllable theory, we have a set of deeply conflicting universal constraints: association constraints (*PL/Lab, *PL/Cor), which favor no associations, and Faithfulness constraints which favor associations ($\text{Parse}^{\text{Feat}}$ from the bottom up, Fill^{PL} from the top down). This conflict is resolved differently in different languages by virtue of different domination hierarchies. The four constraints can be ranked in 4! = 24 ways overall; Universal Grammar, in the guise of Coronal Unmarkedness (283), rules out the half of these in which *PL/Lab is ranked below *PL/Cor, leaving 12 possible orderings, of which 8 are distinct. These induce a typology of segment inventories which includes, as we will shortly see, the Yidin^y case.

In languages with a wider variety of complex segments than Yidin^y, we need to distinguish an input which will be parsed as [Cor Vel] – a velarized coronal like [t^ɣ] – from an input which will be parsed as [Vel Cor] – a palatalized velar like [k^y]. (Both these segments occur, for example, in Irish and Russian.) For this purpose we assume that the feature set in the first input is {Cor, Vel′} and in the second, {Cor′, Vel}; the notation f′ means that the feature f is designated in the feature set as secondary, one which is most harmonically parsed in the secondary place position. That is, we have the constraint:

(287) * [f′
f′ is not parsed as the primary place of articulation (not associated to C-Pl).

Since f and f′ designate the same place of articulation, parsing either of them incurs the same mark *PL/f; there are no separate marks *PL/f′ because *PL/f refers only to the place of articulation f.

Now we are ready to analyze the interaction between coronal unmarkedness and complexity in Yidin^y. The analysis is laid out for inspection in table (288).

(288) Segmental Inventory

Input POA's	*Candidates*	FILL^{PL}	*PL/Lab	PARSE^{Feat}	*PL/Cor	*[f′
Coronalized coronal {PL, Cor, Cor′}	*a.* ☞ [Cor Cor′]				**	
	b. [Cor′ Cor]				**	* !
	c. [Cor] ⟨Cor′⟩			* !	*	
	d. [Cor′] ⟨Cor⟩			* !	*	*
	e. [ ] ⟨Cor, Cor′⟩	* !		**		
Labialized labial {PL, Lab, Lab′}	*f.* [Lab Lab′]		** !			
	g. ☞ [Lab] ⟨Lab′⟩		*	*		
	h. [ ] ⟨Lab, Lab′⟩	* !		**		
Coronalized labial {PL, Lab, Cor′}	*i.* [Lab Cor′]		* !		*	
	j. [Lab] ⟨Cor′⟩		* !	*		
	k. ☞ [Cor′] ⟨Lab⟩			*	*	*
	l. [ ] ⟨Lab, Cor′⟩	* !		**		
Labialized coronal {PL, Cor, Lab′}	*m.* [Cor Lab′]		* !		*	
	n. ☞ [Cor] ⟨Lab′⟩			*	*	
	o. [Lab′] ⟨Cor⟩		* !	*		*
	p. [ ] ⟨Cor, Lab′⟩	* !		**		
Simple coronal {PL, Cor}	*q.* ☞ [Cor]				*	
	r. [ ] ⟨Cor⟩	* !		*		
Simple labial {PL, Lab}	*s.* ☞ [Lab]		*			
	t. [ ] ⟨Lab⟩	* !		*		

The size of the table gives a misleading impression of intricacy. The idea behind this analysis is quite simple. Association must be forced, since the anti-association constraints *PL/α militate against it. The location of Parse amid the anti-association constraints marks a kind of cut-off point: those *PL/α below Parse are overruled and association of their α is compelled; those above Parse, by contrast, are under no bottom-up pressure to associate. Only the top-down pressure of Fill will compel association – but since violations must be minimal, only minimal association can be forced. Glancing across the top of the tableau, one can see that all Cor's will be forced into association by Parse, but Lab-association, driven only by Fill, will be minimal.

Here we give the details of the argument just outlined. Since *PL/Lab $\gg$ Parse$^{\text{Feat}}$, it is more harmonic to leave Lab features unparsed (incurring *Parse$^{\text{Feat}}$) than to associate them to PL (incurring *PL/Lab). Thus, *ceteris paribus*, Lab features remain unparsed.

The only reason that Lab nodes are ever parsed at all is to satisfy Fill$^{\text{PL}}$, which dominates *PL/Lab. Fill is exactly the *ceteris* that is not *paribus*. If the only features available in the set are Lab features, then failing to parse all of them would leave PL unfilled, earning a worse mark *Fill$^{\text{PL}}$ than is incurred by parsing one of the Lab nodes.

On the other hand, only *one* Lab feature need be parsed to satisfy Fill$^{\text{PL}}$. When two are available, as in (f–h), parsing both would only increase the degree of violation of *PL/Lab. Since violations are minimal, the least necessary concession is made to Fill$^{\text{PL}}$. If two Labs are available in the set, one of them satisfies its intrinsic tendency to remain unparsed, while the other sacrifices this for the higher goal of ensuring that PL is not completely empty.

The situation is reversed for Cor, however; it is more harmonic to parse these features than to leave them unparsed, because Parse$^{\text{Feat}}$ $\gg$ *PL/Cor.

As we see from the tableau, the Yidin^{y} inventory includes simple labials, as in rows (g,s), simple coronals, as in rows (k,n,q), and complex coronals as in row (a) but no other complex Places.[19] The grammar foils the attempt to create a complex labial from the input {PL,Lab,Lab′} in rows (f–h) by underparsing this set: a simple labial is output, as in (g), with one of the Lab features unparsed. The input {PL,Lab,Cor′} in rows (i–l) also fails to generate a complex segment, because the grammar parses only the Cor feature, outputting a simple coronal, row (k). The same output results from the input {PL,Cor,Lab′} of rows (m–p). This then is an instance of what we called 'Stampean Occultation' in §4.3.1 [not included here – Ed.]; potential complex places involving Lab cannot surface, because the grammar always interprets them as something else, behind which they are effectively hidden. In the simplest case, the learner would never bother to posit them (see §9.3 for discussion).

9.2 Universal inventories

In addition to language-particular inventories, any theory must make possible an account of universal inventories. We have already seen a number of examples of universal inventory construction, and the preceding analysis of segmental inventories provides yet another, which we will now explore. The general issue of universal inventories has two aspects which we will exemplify; the following statements are intended to fix the terms of the discourse.

(289) Absolute Universal Inventory Characterizations
a. *Absence*. A structure φ is absent from the universal inventory if, for every possible grammar and every possible input, the optimal output parse of that input for that grammar lacks φ.
b. *Presence*. A structure φ is universally present in language inventories if, for any possible grammar, there is some input whose optimal parse in that grammar contains φ.

(290) Relative Universal Inventory Characterizations
An *implicational universal* of the form 'ψ in an inventory implies φ in the inventory' holds if, for every possible grammar in which there is some input whose optimal parse includes ψ, there is an input whose optimal parse in that same grammar includes φ.

The phrase 'possible grammar' refers to the well-formedness constraints provided by Universal Phonology, interacting via a particular domination hierarchy consistent with the domination conditions imposed by Universal Phonology.

9.2.1 Segmental inventories

The segmental inventory of Yidin^y, barring only the worst-of-the-worst (complex, with at least one noncoronal Place), is but one of the inventories in the universal typology generated by the 12 possible domination hierarchies which can be constructed from the four constraints *PL/Cor, *PL/Lab, FILL^{PL}, PARSE^{Feat}, consistent with the universal domination condition (283) that yields Coronal Unmarkedness. This typology includes, for example, inventories which exclude all segments with complex places, and inventories which exclude all labials. The basic sense of the typology emerges from a couple of fundamental results, which correspond directly to the informal observations of Noncomplexity (282) and Coronal Unmarkedness (281), taken as implicational universals:

(291) Complex ⇒ Simple
[π ψ] ⇒ [π], [ψ]
If the segment inventory of a language includes a complex segment with primary place π and secondary place ψ, it has a simple segment with place π and a simple segment with place ψ.

(292) Lab ⇒ Cor
[. . . Lab . . .] ⇒ [. . . Cor . . .]
If the segment inventory of a language admits labials, it admits coronals.

a. Harmonic Completeness w.r.t. Simple Segments [Lab] ⇒ [Cor]
If a language has simple labials, then it has simple coronals.

b. Harmonic Completeness w.r.t. Primary Place [Lab ψ′] ⇒ [Cor ψ′]
If a language has a complex segment with primary place Lab and secondary place ψ, then it has a complex segment with primary place Cor and secondary place ψ.

c. Harmonic Completeness w.r.t. Secondary Place [π Lab′] ⇒ [π Cor′]
If a language has a complex segment with secondary place Lab and primary place π, then it has a complex segment with secondary place Cor and primary place π.

Recall that we are using 'Lab' to denote any noncoronal place of articulation. All noncoronals satisfy these implicational universals, because like Lab they all satisfy the Coronal Unmarkedness constraint domination condition (284). Both 'Lab' and 'Cor' should be taken here as no more than concrete place-holders for 'more marked entity' and 'less marked entity'.

Harmonic completeness means that when a language admits forms that are marked along some dimension, it will also admit all the forms that are less marked along that dimension. More specifically, if some structure is admitted into a language's inventory, and if a subpart of that structure is swapped for something more harmonic, then the result is also admitted into that language's inventory. The implications Complex ⇒ Simple and Lab ⇒ Cor ensure harmonic completeness in exactly this sense.

These results entail that *only* harmonically complete languages are admitted by the constraint system, no matter what rankings are imposed. In other words, harmonic completeness in POA is a necessary condition for the admissibility of a language under the constraint system at hand. This result is not as strong as we would like: it leaves open the possibility that there are nevertheless some harmonically complete languages that the system does not admit. For example, if the factorial typology turned out to generate only those languages where the distinctions among the coronals were *exactly the same* as those among the labials, the theorems Complex ⇒ Simple and Lab ⇒ Cor would still hold true, for such languages are harmonically complete. (In fact, we know by construction that this is not the case: the Yidin^y hierarchy allows secondary articulations among the coronals but nowhere else.) What we want, then, is that harmonic completeness be also a sufficient condition for admissibility, so that *all* harmonically complete languages are admitted. Let us single out and name this important property:

(293) Strong Harmonic Completeness (SHARC) Property
If a typology admits *all and only* the harmonically complete languages, then we say that it has Strong Harmonic Completeness (SHARC).

If a typology has the SHARC, then it manifests what has been referred to in the literature as 'licensing asymmetry'. For place of articulation, in the circumscribed realm we have been examining, this comes out as follows:

(294) POA Licensing Asymmetry
In any language, if the primary place Lab licenses a given secondary place, then so does Cor; but there are languages in which the secondary places licensed by Cor are a strict superset of those licensed by Lab.

In the common metaphor, Cor is a 'stronger' licenser of secondary places than Lab. With the SHARC, there is the broader guarantee that every asymmetric system is possible. We know that the system of constraints examined here has the POA licensing asymmetry property, because harmonic completeness is a necessary property of admitted languages, and because we have produced at least one (Yidin^{y}) where the secondary articulations among the coronals are a strict superset of those permitted with labials. The factorial typology of the constraint system presented here does not in fact have the SHARC, as the reader may determine, but is a step in that direction.

It is worth noting that the SHARC is undoubtedly not true of POA systems in languages, and therefore not true of the entire UG set of constraints pertaining to POA. Indeed, it is unlikely that harmonic completeness is even a necessary condition on POA systems, as John McCarthy has reminded us. With respect to labialization, for instance, many systems have k^w or g^w with no sign of t^w or d^w. With respect to Simple ⇒ Complex, one recalls that Irish has velarized labials and palatalized labials, but no plain labials. McCarthy points to the parallel case of Abaza, which has pharyngealized voiceless uvulars but not plain ones. We do not see this as cause for dismay, however. Virtually any theory which aims to derive implicational universals must include subcomponents which, in isolation, predict the necessity of harmonic completeness and even its sufficiency as well. The constraints discussed here are a very proper subset of those relevant to POA. In particular, the key domination hierarchy is concerned only with context-free comparison of single features, and contains no information about effects of combination (labial+velar, round+back, ATR+high, etc.), which greatly alter the ultimate predictions of the system (Chomsky & Halle 1968: ch. 9, Cairns 1969, Kean 1974, Stevens & Keyser 1989, Archangeli & Pulleyblank 1992). Optimality Theory, by its very nature, does not demand that individual constraints or constraint groups must be *true* in any simple a-systematic sense. What this means is that an established subsystem or module can be enriched by the introduction of new constraints, without necessarily revising the original impoverished module at all. (We have already seen this in the transition from the basic syllable structure theory to the analysis of Lardil.) This fact should increase one's Galilean confidence that finding a subtheory with the right properties is a significant advance.

The POA subtheory examined here derives the relative diversity of coronals in inventory from the single fact of their unmarkedness. These two characteristics are so commonly cited together that it can easily be forgotten that underspecification theory cannot relate them. This important point comes from McCarthy & Taub 1992:

> Equally important as evidence for the unmarked nature of coronals is the fact that they are extremely common in phonemic inventories, where they occur with great richness of contrast. . . . [The] phonetic diversity of coronals is represented phonologically by setting up a variety of distinctive features that are dependent on the feature coronal. . . .
>
> As explanations for different aspects of coronal unmarkedness, underspecification and dependent features are distinct or even mutually incompatible. By the logic of dependency, a segment that is specified for a dependent feature . . . must also be specified for the corresponding head feature . . . For example, even if the English plain alveolars

> *t*, *d*, *l*, *r* and *n* are underspecified for [coronal] the dentals *θ*/*ð* and palato-alveolars *č*/*ǰ*/*š*/*ž* must be fully specified to support the dependent features [distributed] and [anterior]. As a consequence, the dentals and palato-alveolars should not participate in the syndrome of properties attributed to coronal underspecification, and conversely, the plain alveolars should not function as a natural class with the other coronals until application of the [coronal] default rule.

It seems clear that the only way out is to abandon underspecification in favor of markedness theory (cf. Mohanan 1991). This is an ill-advised maneuver if it means embracing nothing more substantial than an elusive hope. The present theory shows that solid formal sense can be made of the notion of markedness, and, more significantly, that results about subtleties of inventory structure – permitted featural combinations – can be deduced from hypotheses about the relative markedness of individual atomic features. The coronal diversity result parallels the result in §8 that onsets are stronger licensers of segments than codas. In the syllable structure case, it is the structural markedness of the Cod node relative to the Ons node which impairs its ability to license segments. Here, licensing is diminished by the markedness of Lab as a place relative to Cor. Formally, the relationship of licenser to licensed is quite different in the two cases, but in both cases the markedness of the licenser governs its ability to license. We have, then, a very general mode of subtheory construction within Optimality Theory which allows us to argue from the markedness of atomic components to limitations on the structure of systems.

We now turn to the demonstrations of (291) and (292), with the goal of identifying a general technique for establishing such implicational universals.[20]

The argument establishing (291) runs as follows:

(295) Proof of Complex ⇒ Simple

For the case of the secondary place, i.e., proof that if a language has [π ψ′] it has [ψ]:

a. By definition of admission into the inventory, the output [π ψ′] must appear in an optimal parse of some input; the only possible such input is {PL,π,ψ′}.
b. This means that [π ψ′] (incurring two marks *PL/π, *PL/ψ) must be more harmonic than all competing parses of the input {PL,π,ψ′}, including [π]⟨ψ′⟩ (incurring the marks *PL/π, *PARSE$^{\text{Feat}}$).
c. This entails that PARSE$^{\text{Feat}}$ must dominate *PL/ψ.
d. This in turn implies that with the input {PL,ψ}, the parse [ψ] (incurring *PL/ψ) is more harmonic than its only competitor, [ ]⟨ψ⟩ (incurring *PARSE$^{\text{Feat}}$ [as well as *FILL$^{\text{PL}}$]), hence [ψ] is the optimal parse.
e. Which means that the simple segment [ψ] is admitted into the segmental inventory.

Broadly put, the argument runs like this. Association must be compelled, over the resistance of the anti-association constraints. Either PARSE or FILL can be responsible. The existence of [π ψ′] in an optimal output guarantees that association of ψ is in fact compelled by the grammar and indeed compelled by PARSE, since FILL would be satisfied by merely parsing π. Therefore, the association [ψ] must also occur, driven by PARSE. A similar but slightly more complex argument also establishes that [π] must be admitted.

The parallel argument establishing (292) is just a little more complicated:

(296) Proof of Lab ⇒ Cor
For the case of simple segments, (292a):

a. If a grammar admits simple labials, then the feature Lab in some input feature set must get associated to PL: [Lab] must appear in the optimal parse of this input.
b. In order for this to happen, the association [Lab incurring *PL/Lab, must be more harmonic than leaving Lab unparsed (incurring $*\textsc{Parse}^{\text{Feat}}$, and also possibly $*\textsc{Fill}^{\text{PL}}$ if there are no other features in the set to fill PL).
c. This means the language's domination hierarchy must meet certain conditions: either
   (i) $\textsc{Parse}^{\text{Feat}} \gg$ *PL/Lab
   or
   (ii) $\textsc{Fill}^{\text{PL}} \gg$ *PL/Lab.
d. These conditions (i–ii) on the ranking of *PL/Lab entail that the same conditions must hold when *PL/Lab is replaced by the universally lower-ranked constraint *PL/Cor: since *PL/Lab ≫ *PL/Cor, by Coronal Unmarkedness (283), if (i), then:
   (i′) $\textsc{Parse}^{\text{Feat}} \gg$ *PL/Lab ≫ *PL/Cor;
   if (ii), then:
   (ii′) $\textsc{Fill}^{\text{PL}} \gg$ *PL/Lab ≫ *PL/Cor.
e. This in turn entails that parsing Cor must be better than leaving it unparsed: the input {PL,Cor} must be parsed as [Cor] (incurring *PL/Cor), since the alternative [ ] ⟨Cor⟩ would incur both $*\textsc{Fill}^{\text{PL}}$ and $*\textsc{Parse}^{\text{Feat}}$, at least one of which must be a worse mark than *PL/Cor by d.
f. This means that coronals are admitted into the inventory.

Again, the argument can be put in rough-and-ready form. Association must be compelled, either bottom-up (by PARSE) or top-down (by FILL). The appearance of [Lab – primary labial place – in an optimal output of the grammar guarantees that labial association has in fact been compelled one way or the other. Either a dominant PARSE or a dominant FILL forces violation of *PL/Lab 'don't have a labial place'. The universal condition that labial association is worse than coronal association immediately entails that the less drastic, lower-ranked offense of coronal association is also compelled, by transitivity of domination.

These two proofs (295, 296) illustrate a general strategy:

(297) General Strategy for Establishing Implicational Universals ψ ⇒ φ

a. If a configuration ψ is in the inventory of a grammar G, then there must be some input I_ψ such that ψ appears in the corresponding output, which, being the optimal parse, must be more harmonic than all competitors.

b. Consideration of some competitors shows that this can only happen if the constraint hierarchy defining the grammar G meets certain domination conditions.
c. These conditions entail – typically by dint of universal domination conditions – that an output parse containing φ (for some input I_φ) is also optimal.

9.2.2 Syllabic inventories

The general strategy (297) was deployed in §8 for deriving a number of implicational universals as part of developing the Basic Segmental Syllable Theory. One example is the Harmonic Completeness of the inventories of Possible Onsets and Nuclei (216), which states that if τ is in the onset inventory, then so is any segment less sonorous than τ, and if α is in the nucleus inventory, then so is any segment more sonorous than α. A second example is (255), which asserts that if τ is in the inventory of possible codas, then τ is also in the inventory of possible onsets. The fact that the converse is *not* an implicational universal is the content of the Onset/Coda Licensing Asymmetry (259).

So far, our illustrations of universal inventory characterizations have been of the implicational or relative type (290). Examples of the absolute type (289) may be found in the Basic CV Syllable Structure Theory. A positive example is the result (128) that every syllable inventory contains CV, the universally optimal syllable. A negative example is the result (145) which states that, in syllabic theory (which does not include constraints like Lx≈Pr), two adjacent empty syllable positions (phonetically realized as two adjacent epenthetic segments) are universally impossible: the universal word inventory, under the Basic Theory, includes no words with two adjacent epenthetic segments.

9.3 Optimality in the lexicon

The preceding discussions have been independent of the issue of what inputs are made available for parsing in the actual lexicon of a language. Under the thesis that might be dubbed *Richness of the Base*, which holds that *all* inputs are possible in all languages, distributional and inventory regularities follow from the way the universal input set is mapped onto an output set by the grammar, a language-particular ranking of the constraints. This stance makes maximal use of theoretical resources already required, avoiding the loss of generalization entailed by adding further language-particular apparatus devoted to input selection. (In this we pursue ideas implicit in Stampe 1969, 1973/79, and deal with Kisseberth's grammar/lexicon 'duplication problem' by having no duplication.) We now venture beyond the Richness of the Base to take up, briefly, the issue of the lexicon, showing how the specific principles of Optimality Theory naturally project the structure of a language's grammar into its lexicon.

Consider first the task of the abstract learner of grammars. Under exposure to phonetically interpreted grammatical outputs, the underlying inputs must be inferred. Among the difficulties is one of particular interest to us: the many-to-one nature of the grammatical input-to-output mapping, arising from the violability of Faithfulness. To take the example of the Yidin^y segmental inventory illustrated above in the tableau (288), two different inputs surface as a simple labial: the input {PL,Lab} which earns the faithful parse [Lab], and the input {PL,Lab,Lab′} which is parsed [Lab]⟨Lab′⟩. These outputs are phonetically identical: which underlying form is the learner to infer is part of the underlying segmental inventory? Assuming that there is no morphophonemic evidence bearing on the choice, the obvious answer – posit the first of these, the faithfully parsable contender – is a consequence of the obvious principle:

(298) Lexicon Optimization[21]
Suppose that several different inputs $I_1, I_2, \ldots, I_n$ when parsed by a grammar G lead to corresponding outputs $O_1, O_2, \ldots, O_n$, all of which are realized as the same phonetic form Φ – these inputs are all *phonetically equivalent* with respect to G. Now one of these outputs must be the most harmonic, by virtue of incurring the least significant violation marks: suppose this optimal one is labelled O_k. Then the learner should choose, as the underlying form for Φ, the input I_k.

This is the first time that parses of *different inputs* have been compared as to their relative Harmony. In all previous discussions, we have been concerned with determining the output that a given input gives rise to; to this task, only the relative Harmony of competing parses of the same input is relevant. Now it is crucial that the theory is equally capable of determining which of a set of parses is most harmonic, even when the inputs parsed are all different.

Morphophonemic relations can support the positing of input–output disparities, overriding the Lexicon Optimization principle and thereby introducing further complexities into lexical analysis. But for now let us bring out some of its attractive consequences. First, it clearly works as desired for the Yidin^y consonant inventory. Lexicon Optimization entails that the analysis of the Yidin^y constraint hierarchy (288) simultaneously accomplishes two goals: it produces the right *outputs* to provide the Yidin^y inventory, and it leads the learner to choose (what we hypothesize to be) the right *inputs* for the underlying forms. The items in the Yidin^y lexicon will not be filled with detritus like feature sets {PL,Cor,Lab′} or {PL,Lab,Lab′}. Since the former surfaces just like {PL,Cor} and the latter just like {PL,Lab}, and since the parses associated with these simpler inputs avoid the marks *PARSE$^{\mathrm{Feat}}$ incurred by their more complex counterparts, the needlessly complex inputs will never be chosen for underlying forms by the Yidin^y learner.[22]

Lexicon Optimization also has the same kind of result – presumed correct under usual views of lexical contents – for many of the other examples we have discussed. In the Basic CV Syllable Structure Theory, for example, Lexicon Optimization entails that the constraints on surface syllable structure will be echoed in the lexicon as well. In the typological language family $\Sigma^{\mathrm{CV}}{}_{\mathrm{del,del}}$, for example, the syllable inventory consists solely of CV. For any input string of Cs and Vs, the output will consist entirely of CV syllables; mandatory onsets and forbidden codas are enforced by underparsing (phonetic nonrealization). Some inputs that surface as [CV] are given here:

(299) Sources of CV in $\Sigma^{CV}{}_{del,del}$

a. /CVV/ → .CV.⟨V⟩

b. /CCVV/ → ⟨C⟩.CV.⟨V⟩

c. /CCCVVV/ → ⟨C⟩⟨C⟩.CV.⟨V⟩⟨V⟩

The list can be extended indefinitely. Clearly, of this infinite set of phonetically equivalent inputs, /CV/ is the one whose parse is most harmonic (having no marks at all); so *ceteris paribus* the $\Sigma^{CV}{}_{del,del}$ learner will not fill the lexicon with supererogatory garbage like /CCCVVV/ but will rather choose /CV/. Ignoring morphological combination (which functions forcefully as *ceteris imparibus*) for the moment, we see that CV-language learners will never insert into the lexicon any underlying forms that violate the (surface) syllable structure constraints of their language; that is, they will always choose lexical forms that can receive faithful parses given their language's syllable inventory.

Morphological analysis obviously enlivens what would otherwise be a most boringly optimal language, with no deep/surface disparities at all. [See ch. 5, §2, of Tesar & Smolensky 2000 for some recent discussion. – Ed.]

[. . .]

While properly reformulating Lexicon Optimization from a form-by-form optimization to a global lexicon optimization is a difficult problem, one that has remained open throughout the history of generative phonology, a significant step towards bringing the Minimal Lexical Information principle under the scope of Lexicon Optimization as formulated in (298) is suggested by a slight reformulation, the Minimal Redundancy principle: to the maximal extent possible, information should be excluded from the lexicon which is predictable from grammatical constraints. Such considerations figure prominently, e.g., in discussions of underspecification (e.g., Kiparsky's Free Ride). An example of the consequences of this principle, if taken to the limit, is this: in a language in which *t* is the epenthetic consonant, a *t* interior to a stem which happens to fall in an environment where it would be inserted by epenthesis if absent in underlying form should for this very reason *be* absent in the underlying form of that stem. A rather striking example of this can be provided by the CV Theory. Consider a $\Sigma^{(C)V}{}_{ep}$ language (onsets are optional and codas are forbidden, enforced by overparsing – 'epenthesis'). The Minimal Lexical Redundancy principle would entail that a stem that surfaces as .CV.CV.CV. must be represented underlyingly as /CCC/, since this is overparsed as .C□́.C□́.C□́., which is phonetically identical to .CV.CV.CV.: it is redundant to put the V's in the lexicon of such a language. Given the constraints considered thus far, Lexicon Optimization as stated in (298) selects /CVCVCV/ and not /CCC/ in this case; again, avoiding deep/surface disparities whenever possible. But this is at odds with the principle that the lexicon should not contain information which can be predicted from the grammar.

The approach to parsing we have developed suggests an interesting direction for pursuing this issue. As stated in (186), the Push/Pull Parsing approach views parsing as a struggle between constraints which *prohibit* structure and constraints which *require* structure. As noted in §3.1 [omitted here – Ed.], the most general form of the structure-prohibiting constraint is *Struc which penalizes any and all structure. There is a specialization of it which would be invisible during parsing but which can play an important role in learning:

(303) *Spec
Underlying material must be absent.

Each underlying feature in an input constitutes a violation of this constraint.[23] But these violations cannot influence parsing since the underlying form is fixed by the input, and no choice of alternative output parses can affect these violations of *Spec. But Lexicon Optimization is an inverse of parsing: it involves a fixed phonetic *output*, and varying underlying *inputs*; thus, among phonetically equivalent inputs, *Spec favors those with fewest featural and segmental specifications.

Now an interesting change occurs if *Spec outranks Faithfulness: Lexicon Optimization (298) selects /CCC/ over /CVCVCV/ in the CV theory example – since minimizing Faithfulness violations (and thereby deep/surface disparities) is now less important than minimizing underlying material. If, on the other hand, Faithfulness dominates *Spec, we are back to /CVCVCV/ as the optimal underlying form.

Clearly a great deal of work needs to be done in seriously pursuing this idea. Still, it is remarkable how the addition of *Spec to the constraint hierarchy can allow Lexicon Optimization – in its original straightforward formulation (298) – to capture an important aspect of the Minimal Lexical Information and Minimal Redundancy principles. It remains to be seen whether a constraint like *Spec can supplant other possible constraints aimed specifically at limiting allomorphy, demanding (for example) a 1:1 relation between a grammatical category and its morphemic exponent. It is important to note that the addition of *Spec makes no change whatever to any of the analyses we have considered previously. This raises the intriguing question of whether there are other constraints which are invisible to parsing – the operation of the grammar – but which play indispensable roles in grammar acquisition.
[. . .]

Notes

1 This kind of reasoning is familiar at the level of grammar selection in the form of the Evaluation Metric (Chomsky 1951, 1965). On this view, the resources of Universal Grammar (UG) define many grammars that generate the same language; the members of that set are evaluated, and the optimal grammar is the real one.

2 An interesting variant is what we might call 'anharmonic serialism', in which Gen produces the candidate set by a nondeterministic sequence of constrained procedures ('do one thing; do another one') which are themselves not subject to harmonic evaluation. The candidate set is derived by running through every possible sequence of such actions; harmonic evaluation looks at this candidate set. To a large extent, classical Move-α theories (Chomsky 1981) work like this.

3 Glosses are *ratkti* 'she will remember'; *bddl* 'exchange!'; *maratgt* 'what will happen to you?'; *tftkt* 'you suffered a strain'; *txznt* 'you stored'; *txznakk^w* 'she even stockpiled'; *tzmt* 'it (f.) is stifling'; *tmzħ* 'she jested'; *trglt* 'you locked'; *ildi* 'he pulled'; *ratlult* 'you will be born'; *trba* 'she carried-on-her-back'; where 'you' = second person singular and the English past translates the perfect.

4 Not the least of these is that syllables can have codas; the DEA serves essentially to locate syllable nuclei, which requires that onsets be taken into consideration. But it is not difficult to imagine plausible extensions which lead to adjunction of

codas. More subtle, perhaps, are these phenomena:

a. obstruents are always nonsyllabic in the envs. #— and —#.
b. sonorant C's are optionally nonsyllabic —# under certain conditions.
c. the 1st element of a tautomorphemic geminate is never an onset.

In addition, the DEA does not completely resolve sequences /~aa~/, which according to other sources, surface as *~aya~* (Guerssel 1986). The appropriate approach to epenthetic structure within OT involves the constraint Fill, which makes its appearance below in §3.1 [omitted here – Ed.] and receives full discussion in §6.

5 We deal with the fact that [a] cannot occupy syllable margins in §8. The commonly encountered relaxation of the onset requirement in initial position is resolved in McCarthy & Prince 1993 in terms of constraint interaction, preserving the generality of Ons. Dell & Elmedlaoui are themselves somewhat ambivalent about the need for directionality (Dell & Elmedlaoui 1985: 108); they suggest that "the requirement [of directionality] is not concerned with left to right ordering *per se*, but rather with favoring applications of [the DEA] that maximize the sonority differences between [onset and nucleus]" (Dell & Elmedlaoui 1985: 127 fn. 22). In addition, they note that directionality falsely predicts **.i.tBd.rin.* from /i=t-!bdri-n/ 'for the cockroaches', whereas the only licit syllabification is *.it.bD.rin.* The reason for this syllabification is not understood. A directionless theory leaves such cases open for further principles to decide.

6 We show the form predicted by the DEA. The form is actually pronounced rat.lult. because obstruents cannot be nuclear next to phrase boundaries, as mentioned in n.4.

7 These are exactly the sort of questions that were fruitfully asked, for example, of the classic Transformational Grammar (TG) rule of Passive that moved subject and object, inserted auxiliaries, and formed a PP: why does the post-verbal NP move up not down? why does the subject NP move at all? why is by+NP a PP located in a PP position? and so on.

8 Further development of this idea could eliminate complications at the level of the general theory; in particular, the appearance of obeying the Free Element Condition during serial building of structure could be seen to follow from the fact that disobeying it inevitably decrements the Harmony of the representation.

9 It is also possible to conceive of the operative constraint in a kind of 'contrapositive' manner. Because all underlying segments of ITB are parsed, a segment is a nucleus iff it is not a member of the syllable margin. Consequently, negative constraints identifying the badness of syllable margins can have the same effect as positive constraints identifying the goodness of nuclei. We investigate this approach in §8.

10 Properly speaking, if we limit our attention to the core syllable stage of the procedure, we should be comparing core *.u.* with core *.wL.* But the comparison remains valid even after coda consonants are adjoined and we wish to emphasize that the two cited analyses of /ħaul-tn/ differ only in treatment of the sequence /ul/.

11 In §5.1 we define several formally distinct orders in terms of one another. At the risk of overburdening the notation, in this section only, we use superscripts like $^{\text{parse}}$ and $^{(*)}$ to keep all these orders distinct. We prefer to resist the temptation to sweep conceptual subtleties under the rug by using extremely concise notation in which many formally distinct relations are denoted by the same symbol. It is important to remember, however, that the symbols '$\succ$' and '$\approx$' – no matter what their subscripts and superscripts – always mean 'more harmonic' and 'equally harmonic'. We need to compare the Harmonies of many different kinds of elements, and for clarity while setting up the fundamental definitions of the theory, we distinguish these different Harmony comparison operators. Once the definitions are grasped, however, there is no risk of confusion in dropping superscripts and subscripts; this we will do elsewhere. The superscripts and subscripts can always be inferred from context – once the whole system is understood.

12 A simple example of how this definition (97) works is the following demonstration that

$(*\mathbb{C}) \succ^{(*)} (*\mathbb{C}, *\mathbb{C})$.

Define α and β as follows (we use '≡' for 'is defined to be'):

$\alpha \equiv (*\mathbb{C}) \quad \beta \equiv (*\mathbb{C}, *\mathbb{C})$.

Then

$\alpha \succ^{(*)} \beta$ because
(97.ii) $\mathrm{FM}(\alpha) = *\mathbb{C} \approx^{*} *\mathbb{C} = \mathrm{FM}(\beta)$ and
$\mathrm{Rest}(\alpha) = (\ ) \succ^{(*)} (*\mathbb{C}) = \mathrm{Rest}(\beta)$;

where the last line, $(\ ) \succ (*\mathbb{C})$, is in turn demonstrated by letting

$\alpha' \equiv (\ ) \ \beta' \equiv (*\mathbb{C})$

and noting that

$\alpha' \succ^{(*)} \beta'$ because
(97.i) $\mathrm{FM}(\alpha') = \varnothing \succ^{*} *\mathbb{C} = \mathrm{FM}(\beta')$

by (96).

13 Both Fill and Parse are representative of families of constraints that govern the proper treatment of child nodes and mother nodes, given the representational assumptions made here. As the basic syllable theory develops, Fill will be articulated into a pair of constraints (see §6.2.2.2):

Fill^{Nuc}: Nucleus positions must be filled with underlying segments.
Fill^{Mar}: Margin positions (Ons and Cod) must be filled with underlying segments.

Since unfilled codas are never optimal under syllable theory alone, shown below in §6.2.3 (141), Fill^{Mar} will often be replaced by Fill^{Ons} for perspicuity.

14 For versions of the structural constraints within the perhaps more plausible moraic theory of syllable structure see Kirchner 1992b,c, Hung 1992, Samek-Lodovici 1992, 1993, Zoll 1992, 1993, McCarthy & Prince 1993.

15 On *complex* margins, see Bell 1971, a valuable typological study. Clements 1990 develops a promising quantitative theory of cross-linguistic margin-cluster generalizations in what can be seen as harmonic terms. The constraint *Complex is intended as no more than a cover term for the interacting factors that determine the structure of syllable margins. For a demonstration of how a conceptually similar *complex* vs. *simple* distinction derives from constraint interaction, see §9.1–2 below.

16 It would also be possible to break this yoke by having two separate Parse constraints, one that applies to C and another to V. Basic syllable structure constraints that presuppose a C/V distinction, however, would not support the further development of the theory in §8, where the segment classes are derived from constraint interactions.

17 The demonstration will require some work, however; perhaps this is not surprising, given the simplicity of the assumptions.

18 Here, *M/α and *P/α are the constraints against parsing α as a Margin (Onset, Coda) and as a Peak (Nucleus), respectively; this is the contrapositive of the Possible Peak Condition (231).

19 In the tableau, a label like 'Labialized Labial' for the input {PL,Lab,Lab′} is keyed to what *would result* from a faithful parse. The actual grammar underparses this input, and the output is a simple labial. Such labels are intended to aid the reader in identifying the input collocation and do not describe the output.

20 Another related technique, used in §8 and to an extended degree in Legendre, Raymond, & Smolensky 1993, can be effectively used here as well; the results are more general but the technique is a bit more abstract. This other technique, which might be called the Technique of Necessary and Sufficient Conditions, goes as follows:

Step 1: Determine necessary and sufficient conditions on the ranking of constraints in a hierarchy in order that each of the relevant structures be admitted into the inventory by that constraint ranking.
Step 2: Examine the logical entailments that hold between these conditions. These are arguments of the form: in order to admit structure φ it is necessary that the constraints be ranked in such-and-such a way, and this entails that the constraint ranking meets the sufficient conditions to admit structure ψ.

To carry out Step 1, to determine the necessary and sufficient conditions for a structure φ to be admitted, one takes a general parse containing φ and compares it to all alternative parses of the same input, and asks, how do the constraints have to be ranked to ensure that φ is more harmonic than all the competitors? And this in turn is done by applying the Cancellation-Domination Lemma (§8.2.6 [omitted here, but see (192) – Ed.]): for each mark m incurred by φ, and for each competitor C, if m is not cancelled by an identical mark incurred by C then it must be dominated by at least one mark of C.

In the present context, this technique gives the following results (Step 1):

(α) In order that [χ] be admitted into a inventory it is necessary and sufficient that:
either Parse^{feat} or $\text{Fill}^{PL} \gg$ *PL/χ

(β) In order that [π ψ] be admitted into an inventory it is necessary and sufficient that:
a. $\text{Parse}^{feat} \gg$ *PL/ψ, and
b. either Parse^{feat} or *[f′ ≫ *PL/π, and
c. either Parse^{feat} or $\text{*Fill}^{PL} \gg$ *PL/π

From here, Step 2 is fairly straightforward. The result Complex ⇒ Simple (291) for the secondary place ψ follows immediately, since (β.a) ⇒ (α) for χ = ψ. The result Complex ⇒ Simple for the primary place π follows similarly since (β.c) ⇒ (α) for χ = π.

For the Harmonic Completeness results (292), we use the Coronal Unmarkedness domination condition (284): *PL/Lab ≫ *PL/Cor. This means that whenever any of the domination conditions in (α) or (β) hold of the feature Lab, it must also hold of the feature Cor; for in that case, each asserts that some constraint must dominate *PL/Lab, which means the same constraint must also dominate *PL/Cor since *PL/Lab ≫ *PL/Cor. Spelling this observation out in all the cases a–c of (292) proves the result Lab ⇒ Cor.

21 The term ‘lexicon’ here is really overly restrictive, since this is actually a principle for inducing underlying forms in general, not just those of lexical entries. For example, it can apply in syntax as well. The rules of the syntactic base might well generate structures such as $[[[[[\text{he}]_{NP}]_{NP}]_{NP}]_{NP}]_{NP}$ as well as simple $[\text{he}]_{NP}$. But, as we shall see, the principle (298) will imply that the simpler alternative will be selected as the underlying form.

22 The Yidin[y] system follows the pattern called ‘Stampean Occultation’ in §4.3.1 [omitted here – Ed.]. The principle of Lexicon Optimization thus makes explicit the content of the Occultation idea.

23 The constraint is thus identical to the featural measure of lexical complexity in Chomsky & Halle 1968: 381.

References

Anderson, Stephen R. 1969. *West Scandinavian Vowel Systems and the Ordering of Phonological Rules*. Doctoral dissertation. MIT, Cambridge, MA.

Aoun, Joseph. 1986. *Generalized Binding: The Syntax and Logical Form of Wh-interrogatives*. *Studies in Generative Grammar 26*. Dordrecht: Foris.

Archangeli, Diana. 1984. *Underspecification in Yawelmani Phonology and Morphology*. Doctoral dissertation. MIT, Cambridge, MA. [New York: Garland Press, 1988.]

Archangeli, Diana and Douglas Pulleyblank. 1992. Grounded phonology. MS. University of Arizona and University of British Columbia. ▶

Avery, P. and Keren Rice. 1989. Segment structure and coronal underspecification. *Phonology* 6, 179–200.

Bell, Alan. 1971. Some patterns of occurrence and formation of syllable structures. In *Working Papers on Language Universals*, no. 6, 23–137. Language Universals Project, Stanford University.

Black, H. Andrew. 1991. The optimal iambic foot and reduplication in Axininca Campa. *Phonology at Santa Cruz* 2, 1–18.

Broselow, Ellen. 1982. On the interaction of stress and epenthesis. *Glossa* 16, 115–32.

Browning, M. A. 1991. Bounding conditions on representation. *Linguistic Inquiry* 22, 541–62.

Cairns, Charles. 1969. Markedness, neutralization and universal redundancy rules. *Language* 45, 863–85.

Cairns, Charles. 1988. Phonotactics, markedness, and lexical representation. *Phonology* 5, 209–36.

Cairns, Charles and Mark Feinstein. 1982. Markedness and the theory of syllable structure. *Linguistic Inquiry* 13, 193–226.

Chomsky, Noam. 1951. *Morphophonemics of Modern Hebrew*. Master's thesis. University of Pennsylvania, Philadelphia.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper and Row.

Clements, G. N. 1976. Palatalization: linking or assimilation? In *Proceedings of the Twelfth Regional Meeting of the Chicago Linguistic Society*, ed. S. S. Mufwene, C. A. Walker, and S. B. Steever, 96–109. Chicago: Chicago Linguistic Society.

Clements, G. N. 1990. The role of the sonority cycle in core syllabification. In *Papers in Laboratory Phonology I*, ed. John Kingston and Mary Beckman, 283–333. Cambridge: Cambridge University Press.

Clements, G. N. 1991. Place of articulation in vowels and consonants: a unified theory. In *L'Architecture et la géométrie des représentations phonologiques*, ed. B. Laks and A. Rialland. Paris: CNRS-Editions.

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

Dell, François and Mohamed Elmedlaoui. 1985. Syllabic consonants and syllabification in Imdlawn Tashlhiyt Berber. *Journal of African Languages and Linguistics* 7, 105–30.

Dell, François and Mohamed Elmedlaoui. 1988. Syllabic consonants in Berber: some new evidence. *Journal of African Languages and Linguistics* 10, 1–17.

Dell, François and Mohamed Elmedlaoui. 1989. Quantitative transfer in the nonconcatenative morphology of Imdlawn Tashlhiyt Berber. *Journal of Afroasiatic Languages* 3, 89–125.

Dixon, R. M. W. 1977. *A Grammar of Yidin*^{y}. Cambridge: Cambridge University Press.

Gnanadesikan, Amalia. 1992. The feature geometry of coronal subplaces. MS. University of Massachusetts, Amherst. ▶

Goldsmith, John. 1991. Phonology as an intelligent system. In *Bridges between Psychology and Linguistics: A Swarthmore Festschrift for Lila Gleitman*, ed. Donna Jo Napoli and Judy Kegl, 247–67. Hillsdale, NJ: Lawrence Erlbaum Associates.

Goldsmith, John. 1993. Harmonic phonology. In *The Last Phonological Rule*, ed. John Goldsmith, 21–60. Chicago: University of Chicago Press.

Goodman, Beverley. 1993. *The Integration of Hierarchical Features into a Phonological System*. Doctoral dissertation. Cornell University, Ithaca.

Guerssel, M. 1986. Glides in Berber and syllabicity. *Linguistic Inquiry* 17, 1–12.

Hale, Kenneth. 1973. Deep-surface canonical disparities in relation to analysis and change: an Australian example. *Current Trends in Linguistics* 11, 401–58.

Hayes, Bruce. 1991. *Metrical Stress Theory: Principles and Case Studies*. Samizdat. Los Angeles. ▶

Hume, Elizabeth V. 1992. *Front Vowels, Coronal Consonants, and Their Interaction in Nonlinear Phonology*. Doctoral dissertation. Cornell University, Ithaca.

Hung, Henrietta. 1992. Relativized suffixation in Choctaw: a constraint-based analysis of the verb grade system. MS. Brandeis University, Waltham.

Itô, Junko. 1986. *Syllable Theory in Prosodic Phonology*. Doctoral dissertation. University of Massachusetts, Amherst.

Itô, Junko. 1989. A prosodic theory of epenthesis. *Natural Language and Linguistic Theory* 7, 217–60.

Jakobson, Roman. 1962. *Selected Writings 1: Phonological Studies*. The Hague: Mouton.

Kaye, Jonathan and Jean Lowenstamm. 1984. De la syllabicité. In *Forme sonore du langage*, ed. F. Dell, D. Hirst, and J.-R. Vergnaud. Paris: Hermann.

Kean, Mary-Louise. 1974. *The Theory of Markedness in Generative Grammar*. Doctoral dissertation. MIT, Cambridge, MA.

Kiparsky, Paul. 1973. 'Elsewhere' in phonology. In *A Festschrift for Morris Halle*, ed. S. Anderson and P. Kiparsky, 93–106. New York: Holt, Rinehart, and Winston.

Kirchner, Robert. 1992a. Lardil truncation and augmentation: a morphological account. MS. University of Maryland, College Park.

Kirchner, Robert. 1992b. *Harmonic Phonology within One Language: An Analysis of Yidin*^{y}. MA thesis. University of Maryland, College Park.

Kirchner, Robert. 1992c. Yidin^{y} prosody in Harmony Theoretic Phonology. MS. University of California, Los Angeles.

Klokeid, Terry Jack. 1976. *Topics in Lardil Grammar*. Doctoral dissertation. MIT, Cambridge, MA.

LaPointe, Steven and Mark Feinstein. 1982. The role of vowel deletion and epenthesis in the assignment of syllable structure. In *The Structure of Phonological Representations. Part II*, ed. H. van der Hulst and N. Smith, 69–120. Dordrecht: Foris.

Legendre, Géraldine, William Raymond, and Paul Smolensky. 1993. Analytic typology of case marking and grammatical voice. In *Proceedings of the Berkeley Linguistics Society*, 19. ▶

Levin [Blevins], Juliette. 1985. *A Metrical Theory of Syllabicity*. Doctoral dissertation. MIT, Cambridge, MA.

McCarthy, John. 1979. *Formal Problems in Semitic Phonology and Morphology*. Doctoral dissertation. MIT, Cambridge, MA.

McCarthy, John and Alan Prince. 1990. Prosodic Morphology and Templatic Morphology. In *Perspectives on Arabic Linguistics: Papers from the Second Symposium*, ed. M. Eid and J. McCarthy, 1–54. Amsterdam: John Benjamins.

McCarthy, John and Alan Prince. 1993. *Prosodic Morphology I: Constraint Interaction and Satisfaction*. MS. University of Massachusetts, Amherst, and Rutgers University, New Brunswick, NJ. ROA-482.

McCarthy, John and Alison Taub. 1992. Review of Carole Paradis and Jean-François Prunet, *The Special Status of Coronals*. *Phonology* 9, 363–70.

Mester, Ralf-Armin. 1992. The quantitative trochee in Latin. SRC-92-06, Syntax Research Center, University of California, Santa Cruz. To appear in *Natural Language and Linguistic Theory*. ▶

Mohanan, K. P. 1991. On the bases of radical underspecification. *Natural Language and Linguistic Theory* 9(2), 285–325.

Myers, Scott. 1991. Persistent rules. *Linguistic Inquiry* 22, 315–44.

Paradis, Carole. 1988a. On constraints and repair strategies. *Linguistic Review* 6, 71–97.

Paradis, Carole. 1988b. Towards a theory of constraint violations. *McGill Working Papers in Linguistics* 5, 1–44.

Paradis, Carole and Jean-François Prunet, eds. 1991. *The Special Status of Coronals: Internal and External Evidence. Phonetics and Phonology*, vol. 2. New York: Academic Press.

Payne, David. 1981. *The Phonology and Morphology of Axininca Campa*. Arlington, TX: Summer Institute of Linguistics.

Payne, David, Judith Payne, and Jorge Sánchez S. 1982. *Morfología, fonología y fonética del Asheninca del Apurucayali* (*Campa–Arawak preandino*). Yarinacocha, Peru: Ministry of Education. Summer Institute of Linguistics.

Piggott, Glyne and Raj Singh. 1985. The phonology of epenthetic segments. *Canadian Journal of Linguistics* 30, 415–53.

Prince, Alan. 1983. Relating to the grid. *Linguistic Inquiry* 14, 19–100.

Prince, Alan. 1985. Improving tree theory. In *Proceedings of the Eleventh Annual Meeting of the Berkeley Linguistics Society*, ed. Mary Niepokuj, Mary VanClay, Vassiliki Nikiforidou, and Deborah Jeder, 471–90. Berkeley: University of California, Berkeley.

Samek-Lodovici, Vieri. 1992. Universal constraints and morphological gemination: a crosslinguistic study. MS. Brandeis University. ROA-149.

Samek-Lodovici, Vieri. 1993. A unified analysis of crosslinguistic morphological gemination. In *Proceedings of CONSOLE-1*, ed. P. Ackema and M. Schoorlemmer. The Hague: Holland Academic Graphics.

Selkirk, Elisabeth. 1981. Epenthesis and degenerate syllables in Cairene Arabic. In *Theoretical Issues in the Grammar of the Semitic Languages*, ed. H. Borer and J. Aoun. Cambridge, MA: MIT.

Smolensky, Paul. 1983. Schema selection and stochastic inference in modular environments. In *Proceedings of the National Conference on Artificial Intelligence*, 378–82. Menlo Park, CA: American Association for Artificial Intelligence.

Smolensky, Paul. 1986. Information processing in dynamical systems: foundations of

Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: *Foundations*, ed. D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, ch. 6, 194–281. Cambridge, MA: MIT Press/Bradford Books.

Spring, Cari. 1990. *Implications of Axininca Campa for Prosodic Morphology and Reduplication*. Doctoral dissertation. University of Arizona, Tucson.

Stampe, David. 1969. The acquisition of phonetic representation. In *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society*, eds. Robert I. Binnick et al., 433–44. Chicago: Chicago Linguistic Society.

Stampe, David. 1973/79. *A Dissertation in Natural Phonology*. Doctoral dissertation. University of Chicago. [New York: Garland.]

Steriade, Donca. 1982. *Greek Prosodies and the Nature of Syllabification*. Doctoral dissertation. MIT, Cambridge, MA.

Stevens, Kenneth and S. J. Keyser. 1989. Primary features and their enhancement in consonants. *Language* 65, 81–106.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

Wilkinson, Karina. 1988. Prosodic structure and Lardil phonology. *Linguistic Inquiry* 19, 325–34.

Zoll, Cheryl. 1992. When syllables collide: a theory of alternating quantity. MS. Brandeis University.

Zoll, Cheryl. 1993. Ghost segments and optimality. In *Proceedings of the West Coast Conference on Formal Linguistics 12*, ed. Erin Duncan et al., 183–99. Stanford, CA: Stanford Linguistics Association.

Study and Research Questions

1 Redefine HNUC ((12) in §2) so that it assigns *'s (cf. §5.2.1.2). Show that your redefined constraint works by substituting it in tableaux (16) and (17) of the same section. (More ambitious question: Prove that, in *all* cases, your definition gives exactly the same results as the one in the text.)

2 If the ranking of ONS and HNUC were reversed in Berber, how would syllabification change? Provide tableaux for some relevant examples.

3 Pick a tableau from anywhere in this book. The tableau should have at least three ranked constraints and at least three candidates. Show how the rule (105) in section 5 applies recursively to find the optimal candidate.

4 Show how HNUC can be replaced in the analysis of Berber by the universal constraint subhierarchies in (187) of §8. (You may find it helpful to look at chapter 9 first.)

5 Redo the Basic CV Syllable Theory (§6.2) using the MAX and DEP constraints of chapter 3. Show that the same results are obtained. (Some issues to contend with: overparsing and underparsing of segments in the redone theory; dealing with the motivation for the FILL-ONS/FILL-NUC distinction.)

6 Within the Basic CV Syllable Theory, a /CVCV/ sequence must be parsed as [CV.CV]. Other constraints that go beyond the Basic CV Theory could, however, force the parsing [CVC.V] under certain conditions. Think of some linguistically plausible constraints that might do this. (Suggestion: stress is a good place to look.)

7 Any language that allows CC coda clusters also allows simple C codas. Show how this implicational universal can be explained using the logic of Harmonic Bounding (§9.1.1).

8 Section 5.2.3.1 mentions a "nasty conceptual problem" that interferes with efforts to construct a local, constituent-by-constituent version of Eval. Describe that problem and some possible solutions to it. (Suggestion: think about foot theory as an example – see Part III.)