2:

SPECIFYING THE FRAMEWORKS: GENERATION VERSUS EVALUATION

In 1.2, it was observed that both derivational and optimality frameworks spawn grammars that describe a function from underlying forms to surface forms. In order to develop a formal comparison of the two theoretical frameworks, we must specify how each constructs this function so we can look for structural correlates between the two.

As pointed out by Archangeli and Langendoen (1997:ix), there are two broad formal strategies that inform these frameworks, **generation** and **evaluation**. Generation involves the use of operations that modify (change or add to) given structures, evaluation involves measuring the extent to which given structures comply or fail to comply with constraints. The core of the derivational framework involves generation by a series of rules, the core of the optimality framework involves evaluation by constraints which are ranked to resolve any conflicts. We shall argue here that it will not do to compare Optimality-Theoretic constraint evaluation with the work of constraints in rule systems, since constraint evaluation is an additional development to the core devices of the derivational framework. Similarly, it will not do to compare rules with the Generator function of Optimality Theory, since, as we will show, this function is superfluous (despite its place in the popular conception of the structure of the theory). Rather, the essence of the two frameworks lies in one strategy or the other - generation or evaluation – so it is the very devices of generation and evaluation that a systematic formal comparison must compare.

This chapter's orientation is decidedly formal, since introductions to the theories have already been made. A formal approach is by nature highly powerful, forcing very basic properties to be stated explicitly. Often, our formulations may specify things which those who work with the theories know intuitively. In other instances, they may clarify the systems in ways which challenge popular views.

2.1 Generation in the Derivational Framework

The derivational analyst formulates a system in which a surface form is determined from the underlying form by a series of rule applications, each providing some mutation or augmentation of structure. This is the generation strategy. In this section we review the constructs of the derivational framework which derive the underlying-surface relation: derivational sequences, rules, and rule ordering.

2.1.1 Derivational Sequences

From a series of mutations or augmentations of structure by rules, a sequence of representations builds up - a linguistic derivation, then, is a **sequence**. The derivations of all the surface forms of an entire language form a class, and a generative grammar of the language defines this class. This can be given the following simple algebraic outline (adapted from Soames 1974:124 and Chomsky 1971:183-4)[1]:

(1) The class K of derivations in a grammar G is the class of finite sequences of representations $P_1,...,P_n$ such that:

(i) P_1 is an underlying representation,

(ii) Each pair $\langle P_i,P_{i+1}\rangle$ meets well-formedness requirements placed by G.

[1] The reference to work of such antiquity is a consequence of the peculiar history of generative grammar, since study of the nature of rule application had its heyday in the 1960s and 1970s. Soames (1974) specifically addresses the formalisation of derivational systems.

Since our concern is with how the relation between underliers and surface forms is mediated, we do not now pursue any further the question of *how they are set up* by the analyst, nor how they are to be *interpreted* in cognitive terms or otherwise. Since our inevitably limited focus excludes these from inquiry, we merely assume a set, call it ’Un’, whose members are precisely the underlying representations, and a set ’Su’ whose members are the surface representations. To say that a form is an underlying form is to say that it belongs to ’Un’, to say it is a surface form is to say that it belongs to ’Su’. The function specified by a grammar of the derivational framework is a serial composition of elementary functions, the rules, combined into a derivational sequence starting from the underlying form. The final member of that derivation is the corresponding surface form predicted by the grammar.

(2) Let the underlying-surface relation contain pairs

un_a	:	su_a
un_b	:	su_b
un_c	:	su_c
un_d	:	su_d
...		

Let $\mathrm{P}_1,...,\mathrm{P}_n$ be a derivation in K.

If $\mathrm{un}_x = \mathrm{P}_1$ for $x \in \{a,b,c,d,...\}$, then $\mathrm{P}_n = \mathrm{su}_x$.

The last member of a derivation is the surface counterpart of the underlying form at the start of the derivation.

What (1) does is to characterise the working of the grammar while (2) reveals its result, the determination of the surface from the underlier. The clear distinction of the working of the grammar from its output specifically characterises the classical generativist position: that the derivational grammar is blind to the surface it happens to traverse towards, and thereby offers an explanation of the facts at the surface.

We now turn our attention to rules, expressing the special relations between the successive structures of the derivation.

2.1.2 What Is A Rule?

Within a derivation $P_1,P_2,P_3,...,P_n$, each successive ordered pair of representations $\langle P_1,P_2 \rangle$, $\langle P_2,P_3 \rangle$, $\langle P_3,P_4 \rangle$, etc. – that is, each successive **step** of the derivation – constitutes the application of some rule. Thus, for the Sarcee forms in (3),

(3)	a.	dìní	'it makes a sound'
	b.	dìnítʼ-i	with relative suffix

the underlying form is / dìnítʼ /, as revealed by forms with vowel-initial suffix (3b) and corroborated by the fact that the speaker still *feels a ’tʼ* at the end of the word (3a) even though it is objectively absent from speech (Sapir 1933). Then, the derivation of (3a) contains the ordered pair, ⟨ dìnítʼ, dìní ⟩. Here, a rule of word-final consonant deletion has applied. A rule R is said to **apply** in a derivation $P_1,...,P_n$ if there exists some *i* such that $\langle P_i,P_{i+1} \rangle$ is a member of R. Thus, a rule defines a set of ordered pairs that can appear in the derivations of a language. A set of ordered pairs is a relation (Partee, ter Meulen and Wall 1990:29), so a rule is a **relation**.

As a relation, a rule has a **domain** - the set of structures from which the rule maps, and a **range** - the set of structures to which it maps. A rule of final consonant deletion has the domain 'structures with final consonants' and the range 'structures with no final consonants'. The rule only adds a new structure to the derivational sequence when the previous structure falls within its domain.[2] The domain and range, however, are not usually sufficient to define a relation,[3] and this is true of phonological rules. The range of final consonant deletion is ’structures with no final consonants', but this allows all sorts of possible outputs from / dìníť /, as in (4):

(4) a. / dìní /
b. / dìníťa /
c. / dìníťi /
d. / dì/
e. / pélí /
f. / víná /

Any form ending in a vowel falls within the range of the rule-relation, not just the desired form / dìní / (4a). In order to properly characterise a phonological rule, then alongside the specification of the domain, or **structural description** to which the rule applies, we replace the range condition with a specification of the **structural change** by which the second structure differs from the first. Thus word-final consonant deletion is formulated as follows:

(5) a. $C \rightarrow \varnothing / \_\#$
b. Structural Description: $C_i\#$
c. Structural Change: $C_i \rightarrow \varnothing$

The output of the rule must differ from the input by the absence of the particular final consonant that is identified by the structural description. We represent this by co-indexing the consonant

[2] A qualification needs to be made for rules which insert structure, whether a feature or an association relation or syllable structure. These rules have an ‘implicational’ format (Roca 1994:46), e.g., [+nasal]→[-continuant] rather than a ‘transformational’ format [+continuant]→[-continuant]. When a structure meets the domain condition of an implicational rule [+nasal]→[-continuant], the material to be inserted may already be present ([+nasal,-continuant]). In that case, the rule is said to apply **vacuously**, which would generate an identity mapping.

[3] To see this, consider a relation between the letters of the English alphabet (the domain) and the numbers 1 to 26 (the range). Now, what is 'a' mapped to? If we wish to adopt the conventional order and identify 'a' with '1', for example, we must additionally specify this convention, stating which letters are related to which numbers.

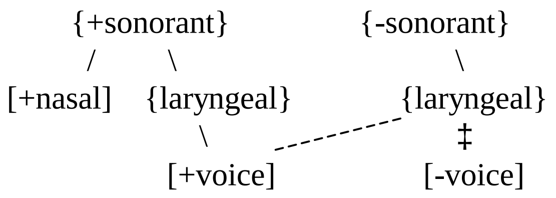
mentioned in the structural description (5b) with the consonant given in the structural change (5c): they are crucially one and the same. Such a definition now rules out augmenting the segmental string with a vowel in application of the rule as in (4b,c); rather, the rule states that the *t’* must be taken out. However, we also require that while the final consonant is deleted, *other* parts of the structure are not permitted to change at random, leading to forms like (4d) or (4e) or (4f). Pieces of structure must either exhibit the structural change (s.c.) of a rule, or else identity (id.), as in (6). This halts the absurdity of random variation.

```
(6)   d     ì     n     í     t’
      |     |     |     |     ¦
      d     ì     n     í     ∅

      id.   id.   id.   id.   s.c.
```

It remains possible for several structural changes to apply at once, by permitting two, or more, rule applications simultaneously within a single derivational step (and all other elements remaining identical). One question is what happens when the structural description of a single rule is met several times in a word. This can happen with rules applying to word-medial positions, such as consonant assimilations in words with several clusters, or vowel lengthening in words with multiple syllables. Chomsky and Halle (1968:344) considered that structural changes took place at all places where the structural description is met in a single step, but Johnson (1972) argued that it was necessary to separate them, applying structural changes singly at successive steps in the derivational sequence for positions from left to right or right to left in the word. The application of two *distinct* rules at the same step has also been countenanced in some proposals (Koutsoudas, Sanders and Noll 1974, Hyman 1993). In autosegmental phonology, it has remained ambiguous whether the spreading and delinking of features depicted within a single diagram apply one after the other or simultaneously (Kenstowicz 1994:103). We give the example of voicing assimilation of obstruents to nasals, where [+voice] spreads from the nasal to the obstruent and any [-voice] value of the obstruent delinks and deletes.

(7) *Spreading and delinking: together, or in some sequence?*



If the theory allows only one structural change at each step, then each structure in the sequence is uniquely determined by the rule applying at that step, and since uniquely determined output is the defining characteristic of a function (Partee, ter Meulen and Wall 1990:30), rules are **functions**. Otherwise, rules are a less stringent kind of relation, each failing to uniquely determine its outcome, and the theory requires an additional formal operation which takes all the rules that apply at one step and produces from them a single ordered pair containing all the structural changes.

2.1.3 Rule Ordering and Regular Sequencing Constraints

The next issue is the sequence in which rules apply. If one rule R_a always applies before another rule R_b in some language, because the mapping by R_a from one structure to the next occurs at an earlier point in the sequence than the mapping by R_b from one structure to the next, then the following statement holds over the class of well-formed derivations:

(8) For all derivations $P_1,P_2,P_3,...,P_n$: $\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in R_a$ & $\langle P_j,P_{j+1}\rangle \in R_b$) $\rightarrow$ i<j]

Whenever the two rules R_a and R_b both apply in a derivation, R_a always applies before R_b.

This captures the regular sequencing of two rules.[4] The regular sequencing of R_b with a third rule R_c would be captured by a similar well-formedness constraint:

(9) For all derivations $P_1,P_2,P_3,...,P_n$: $\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in R_b$ & $\langle P_j,P_{j+1}\rangle \in R_c$) $\rightarrow$ i<j]

Whenever the two rules R_b and R_c both apply in a derivation, R_b always applies before R_c.

Now, taking (8) and (9), it is not immediately possible to deduce (10), which regularises the sequential application of R_a before R_c.

(10) For all derivations $P_1,P_2,P_3,...,P_n$: $\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in R_a$ & $\langle P_j,P_{j+1}\rangle \in R_c$) $\rightarrow$ i<j]

Whenever the two rules R_a and R_c both apply in a derivation, R_a always applies before R_c.

The argument is as follows. In a derivation where all three rules apply, they must of course apply in the sequence R_a before R_b before R_c. But in a derivation where only R_a and R_c apply, but not R_b, neither (8) nor (9) says anything about the sequence in which they come (they are both vacuously true, by falsity of antecedent). So there is no reason why R_c may not apply before R_a, contrary to (10).

So regular sequencing constraints are not themselves **transitive**: R_a always precedes R_b and R_b always precedes R_c does not imply R_a always precedes R_c, though they are **irreflexive** (rules do not apply before themselves) and **asymmetric** (if R_a always applies before R_b, then R_b does not apply before R_a). Instead, (10) is achieved in a stronger theory in which rules are

[4] I have avoided the expression "order of application" and instead adopted the expression “regular sequencing”. This anticipates chapter five in which it is observed that derivational sequences are not necessarily orderable.

ordered, as in (11), because ordering relations are by definition irreflexive, asymmetric *and* transitive.

(11) Let < be an ordering on rules.[5]

If S < T, then for all derivations $P_1,P_2,P_3,...,P_n$,

$\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in S$ & $\langle P_j,P_{j+1}\rangle \in T$) → i<j]

Rules are ordered in a list, and each pair of rules always applies in the sequence given by their order.

Now in the rule ordering theory we have that if $R_a < R_b$, and $R_b < R_c$ then by transitivity $R_a < R_c$. *Then*, by (11), all three regular sequencing constraints (8), (9) and (10) *will* be imposed.[6]

In the only comparable study of the formal properties of derivations, Soames (1974) notes that transitivity is required, but overestimates the response that is needed. In his terminology, an ordering relation is not necessarily transitive; only a *linear* ordering is transitive. Thus, he presents a theory like (11): “if T1 and T2 are transformations, then the statement that T1 is ordered before T2 imposes the following constraint: $\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in T1$ & $\langle P_j,P_{j+1}\rangle \in T2$) → i<j]” (Soames 1974:130), but rejects it because “this characterisation does not require that the ordering [sic] relation holding between transformations be transitive” and “if it is the case that whenever grammars impose orderings, the orderings imposed are linear [and hence, transitive - RN], then we want a theory that is not just compatible with this result, but which predicts it.” (Soames 1974:131). Setting things straight, we do not want a theory which predicts linear order: linear order IS the theory. As Pullum (1979:25) points out, Soames need only add a statement of

[5] The notation S < T is standard in mathematics for "S precedes T", even though the opposite symbol ">" is more familiar in linguistics from historical derivations, e.g. **vin* > *vino* "*vin* is the antecedent of *vino*".

[6] (11) would need to be modified for theories of cyclic rule application, since the rules apply in sequence within one cycle, but the rules may apply again on the next cycle.

transitivity. Instead, Soames (1974:132) resorts to the more elaborate response of assigning numerical indices to transformations, ensuring linear order because the indices are “drawn from a linearly ordered system”. The natural numbers, to be sure, are linearly ordered, and provide a perspicuous notation (which I capitalise on), but they import a whole raft of other properties that have interested mathematicians for centuries but which have no use in derivational systems: for example, numbers are unbounded, so their incorporation implies that a grammar may contain an infinite number of rules, R1,R2,R3,R4,... , and that grammars with a finite set of rules (i.e. all real grammars) intrinsically stop short of the full capacity available. All we actually want is linear order.

In this section, we have recognised that rule-based grammars depend on an ordering relation, which is used to impose natural restrictions on the regular sequencing of rules in derivations. The overall structure of the framework is summarised in the next section.

2.1.4 Summary: Rule-Based Grammar

The three interrelated levels in (12) represent the derivational framework:

(12) Set of Ordered Rules
|
Class of Derivational Sequences
|
Underlying Representation - Surface Representation Pairs

The 'bottom' level has the list of underlying forms paired with the surface forms which realise them. These forms are the first and last members of the derivations, which themselves reside, as a class (as in 2.1.1), on the middle level. In the derivations, the determination of surface forms from underlying forms is decomposed into a series of ordered pairs whereby each successive form is mapped from its predecessor. What counts as a well-formed derivational sequence is determined from the rules and their ordering at the top level. Each ordered pair in the derivational sequence must constitute the application of some rule (as in 2.1.2) and the ordering of rules imposes constraints of regular sequencing on the application of rules (as in 2.1.3).[7] In table (13) below, we recall these formulations.

(13)

Rule System	**Effect of Rule System on Derivations**
A set of rules.	In a derivation $P_1,...,P_n$ in K, each pair of successive represention constitutes the application of some rule, i.e. $\langle P_i,P_{i+1}\rangle \in R$ for some rule R.
A (partial or total) ordering on rules: that is, a relation < between rules that complies with axioms of irreflexivity, asymmetry, transitivity.	Rule ordering statements S < T impose constraints on derivations $P_1,...,P_n$ of the form $\forall i \forall j$ [($\langle P_i,P_{i+1}\rangle \in S$ & $\langle P_j,P_{j+1}\rangle \in T$) $\rightarrow$ i<j].

Bromberger and Halle (1989) also cite other conventions used in the construction of derivations in phonology: the affiliation of rules to different strata, cyclic application of some rules over successively more inclusive morphological domains. These are substantial issues in their own right, and purely for simplicity's sake we delimit our formal enquiry to a single stratum of rules which all apply to the same morphosyntactic domain. This allows us to focus on comparing the essential formal system of rules and derivations with the optimality-theoretic alternative.

[7] The well-formedness of derivations also depends on the requirement that obligatory rules apply whenever a member of the sequence falls within their domain unless ruled out by the regular sequencing constraints and/or possibly other derivational constraints.

2.2 Evaluation in the Derivational Framework

Some work in generative phonology has proposed that the well-formedness of the steps in derivational sequences (see (1)) is decided not only by the basic system of ordered rules, but in part by constraints against which the structures produced by rules are measured. Here, we discuss the formalisation of constraining derivations this way, and show that the full complexity of this approach is greater than has been acknowledged, departing from the basic generation system.

2.2.1 Blocking Constraints

One example is from Modern Hebrew (McCarthy 1986). A rule of schwa deletion which fails to apply just in case the immediately adjacent consonants are identical. Form (14a) illustrates the deletion, which does not obtain in (14b) when the schwa is flanked by two identical consonants.

(14) a. *kaʃəru, kaʃru ’they tied’

b. titpaləli , *titpalli ’I will pray’

We would want a rule somewhat like (15a), whose structural description contains the condition that the flanking consonants must differ in some feature or other.

(15) a. $ə\rightarrow\varnothing/VC_1\_\_C_2V$ $(C_1\neq C_2)$

b. $ə\rightarrow\varnothing/VC\_\_CV$

c. $*C_1C_2$ where $C_1=C_2$

An alternative is to omit the condition that the neighbouring consonants differ, as in (15b), but concomitantly posit a constraint (15c) that prohibits identical adjacent consonants, which will *block* the application of (15b) where necessary. This constraint is the Obligatory Contour Principle (OCP). One can think of offending structures as being ruled out by the evaluation of a tentative rule application, accepted into the derivation depending strictly on satisfaction by the constraint (denoted in (16) by a tick √ or cross X).

(16)

kaʃəru ?↓	OCP
kaʃru	√

titpaləli ?↓	OCP
titpalli	X

A simple way of formalising constraints of this kind might be as in (17a): a constraint C defines a set of structures and no structure outside this set is allowed in a derivation. This would even require that underlying forms, as the first members of derivational sequences, must satisfy C. However, it has been proposed that derivational constraints are more selective, blocking only some rules (Kisseberth 1970a, Archangeli and Pulleyblank 1994, Idsardi 1997). The affected rule(s) may be named in the requirement on derivations as in (17b).

(17) Let $P_1,...,P_n$ be any derivation in K. Let i,j range over the subscripts 1 to n.

a. $\forall i[P_i \in C]$

All structures in the derivation must satisfy constraint C.

b. $\forall i[\ \langle P_i,P_{i+1}\rangle \in R \rightarrow P_{i+1}\in C]$

A derivation may contain an application of the rule R provided the resulting structure satisfies constraint C.

c. Applied to Modern Hebrew: a derivation may contain the rule of interconsonantal schwa deletion provided the outcome has no adjacent identical consonants.

In a similar example with a new twist, a vowel deletion rule in the Amerindian language Tonkawa applies in the environment VC__CV. This means that it fails just in case it would create clusters of three consonants. Kenstowicz (1994:527) offers two versions of the derivational constraint affecting this rule:

(18) a. $V\rightarrow\varnothing$ / X__Y

Condition: block if result violates constraint *CCC

b. $V\rightarrow\varnothing$ / σ__σ

Condition: block if result is not exhaustively syllabifiable

Version (18b) is intended to go beyond the segmental string and take account of contemporary syllable theory. Thus, the bias against clusters of three consonants is to be explained in turn by two constraints: (i) a general constraint against two consonants in either syllable onset or syllable coda - single onset and coda consonants lead to maximum clusters of two consonants word-medially: .CV**C.C**VC.; (ii) a requirement that all segments be licensed by (or have a legitimate place in) syllable structure - so no consonants between syllables, *.CV**C.<C>.C**VC. .

The complexity of this evaluation has not been made formally explicit, however. It cannot be done merely by testing the output of the rule. For, if we block the rule whenever it leaves an unsyllabified consonant, we would block every time - correctly (19, left) and incorrectly, indicated by 💣 (19, right):

(19)

CVC.CV.CV ?↓ Deletion	Licensing
CVC.<C>.CV	X

CV.CV.CV ?↓ Deletion	Licensing
CV.<C>.CV	💣X

In fact, the rule is blocked if it would create a string which is not merely unsyllabifi*ed*, but unsyllabifi*able*, a stronger condition that must be evaluated by taking into account further syllabification rules also, as in (20). Since the onset of the following syllable and the coda of the preceding syllable are both occupied, leaving no way to syllabify the consonant, *then* the original deletion rule is blocked:

(20)

VC.CV.CV ?↓ Deletion	Licensing
VC.<C>.CV ?↓ Onset Syllabification	
does not apply (*VC.CCV) ?↓ Coda Syllabification	
does not apply (*VCC.CV)	X

In contrast, CV.CV.CV will be reduced to CV.<C>.CV then CVC.CV by coda syllabification, and since the consonant is eventually licensed, no overzealous blocking will result.

Thus, while Kenstowicz's (1994:527) condition of unsyllabifiability is both accurate and true to contemporary syllable structure theory, a new degree - even dimension - of complexity has been added to the derivational system. We started out with the notion that rule application could be restricted by derivational constraints that prevent some outputs, keeping a sense of integrity to the particular derivational step as originally set out in (1ii) above and reiterated by Chomsky (1998) as an imperative of Minimalist theory. Now we have a scenario in (20) of evaluating not the outcome of the rule itself, but the outcome following a number of rules - this rule and the rules of syllabification. A new requirement on derivations, significantly more complex than the earlier formalisation of derivational constraints in (17b), is involved.[8] Yet this approach, underformalised and more complex than hitherto acknowledged, is not strictly essential. In principle, all restrictions on the application of rules can be put in the structural description which specifies the domain of the rule mapping. Indeed, Calabrese (1995) proposes that blocking reduces to precisely this, and applies it at least for a simple case. In the Tonkawa rule here, the rule needs to contain the condition that the syllable containing the vowel to be deleted (V) and the preceding syllable are both open syllables: ...C**V.**C**V.**CV... This condition is apt in its reference to syllable structure, without resorting to a complex evaluating mechanism.

[8] In order to maintain evaluation at the original step, it might be said to arise in a different way: all consequences of constraints are also constraints, so *CCC is a constraint because it follows from exhaustive syllabification and a ban on two consonants in onsets or codas. However, for structures where syllabification principles are in force, lack of CCC is simply an epiphenomenon, but *CCC as a constraint in its own right affects structures that syllabification principles do not (structures at derivational stages prior to their syllabification). By this very strength, *CCC is more than a logical consequence, it is an extension. Hence, the proposed grammar contains an enriched, self-extending system of derivational constraints, a development in complexity alternative to the evaluation in the text.

2.2.2 Constraints Triggering Repairs

In addition to the blocking facility, another function has been attributed to constraint statements - that of *triggering* repair-operations whose output satisfies the constraint violated by the input. For example, concatenations of morphemes can bring together material which violates a constraint statement. However, this appeal to constraints is essentially a re-conceptualisation of the structural description of a rule.

Yawelmani Yokuts employs the rule in (21) (Kisseberth 1970a):

(21) $\varnothing \rightarrow V$ / C__CC

Given this rule, one can identify the notions of constraint and repair strategy. One might say that the presence of CCC (or perhaps unsyllabifiable <C>) in some structure in a derivation is evaluated negatively and is subject to an operation to repair it. The structural change of the rule is the insertion of a V between first and second of the three consonants, a site denoted in (22) by $_1\varnothing_2$. Or, one might say, the vowel insertion is the repair operation to avert a *CCC violation.

(22)	Structural Description:	$C_1C_2C_3$	("Constraint: *CCC")
	Structural Change:	$_1\varnothing_2 \rightarrow V$	("Repair: $\varnothing \rightarrow V$")

Note, however, in (22) that the insertion of V cannot occur just anywhere in the word, it must be stated *where* in the configuration CCC it is employed - between the first and second consonants. The structural change of a rule crucially depends on the structural description for its intelligibility. Because of this, any additional independent constraint statement *CCC in the language is redundant. Myers (1991) makes the same argument with examples of rules of English. Constraints-as-triggers must be *none other than the structural descriptions* of rules, and repairs none other than structural changes, which are meaningless without being indexed to a structural description.

We briefly consider a couple of rejoinders to this. First, a putative advantage of appealing to constraints is that both effects of a constraint - triggering repairs and blocking other rules - may occur in a language. For example, Yawelmani Yokuts both repairs and blocks CCC sequences. Or the same constraint may block in one language and trigger repairs in another, such as the OCP (Yip 1988, Myers 1997a).[9] But in a derivational theory based on rules we must re-interpret the informal notion that constraints may both repair and block by saying that one configuration of phonological structure may be present both as the structural description of a rule and as a derivational constraint on other rules.

Second, Goldsmith (1990:318ff, 1993) attempts to generalise the blocking-and-triggering approach with the proposal that phonological rules apply if, and only if, their effect is to increase ‘harmony’ (i.e. increase satisfaction of constraints). In Yokuts, one might attempt to reduce the epenthesis rule (21) to a constraint *CCC (or *<C>) and the simple rule ∅→V, which can break up clusters in order to increase harmony with respect to *CCC. However, we still need to determine where the vowel goes: it could either go at C_CC or CC_C. If rules apply if and only if they would increase harmony, then we require the presence of a further constraint which deems that CV.**CVC.C**... is an improvement in harmony, but CV**C.CV.C**... is not. The problem is that both syllable patterns exist in the language: *su.doḳ.hun* ‘removes’; *wag.ci.wis* ‘act of dividing’. There being no constraint against either pattern in the language, harmony fails to distinguish between the two possible epenthesis sites. We must still index vowel insertion to the right position in the structure, which is what the rule (22) achieves.

[9] The supposed dual effect of constraints in blocking and repair is also a problematic ambiguity, as observed by Prince and Smolensky (1993:207) and Bird (1995:12-14): will a given constraint block the output of a rule, or will it admit the rule’s application but then trigger a repair of its output? Apparently this must be resolved on a case-by-case basis. This has led researchers either to abandon blocking-constraints (Myers 1991, Calabrese 1995), or to abandon rules (Scobbie 1991, Prince and Smolensky 1993).

In conclusion, constraints are an additional facility imposed on the essential system of generation by rules. Although the appeal to constraints is empirically motivated, we have found that the supposed blocking and triggering effects are underformalised, and that the technical difficulties encountered are overcome by reverting to rules.[10] This does not motivate a meaningful formal comparison between these constraints and the constraints in the optimality framework. Rather, rules and their effects must be compared with the optimality framework's constraints and their effects.

Having reviewed the appropriate specification of the derivational framework in 2.1 and 2.2, we move on to the optimality framework.

2.3 Generation in the Optimality Framework

Just as the use of constraints is a formally non-essential extension to the basic generation system of the derivational framework, we make the complementary but innovative claim that generation is eliminable from the evaluation system of the optimality framework.[11] Surface phonological representations are optimal among *all possible* representations defined by the theory of phonological representation, not some set of forms generated by mutations to the underlying representation.

[10] Constraint thinking can also be seen as antithetical to the explanatory intentions of the generation system set out in 2.1.1. For if the constraints which act on derivations are motivated by phonotactic patterns of the language (Sommerstein 1974, Singh 1987, Goldsmith 1993), then the derivation does not explain the surface facts but is itself driven by them (Scobbie 1991).

[11] This proposal was given to the 1997 Spring meeting of the Linguistic Association of Great Britain, and appears in Norton (1998). I am grateful to the LAGB audience and the editor of the volume for their comments.

2.3.1 Is Optimality Theory Derivational?

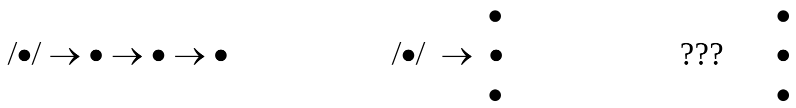
The structure of optimality-theoretic grammar given in the seminal texts (Prince and Smolensky 1993, McCarthy and Prince 1993a,1993b,1994) and maintained since (Kager 1999, McCarthy 2002) goes as in (23), where from each underlying form we generate a set of structures as candidates for the realisation of the form. This function is labelled 'GEN', short for 'generator'. This is followed up by an evaluation function 'EVAL' which assesses the relative adherence of candidates to a hierarchy of constraints, whereby one candidate is delivered up as optimal.[12]

(23) $\text{GEN (in)} \rightarrow \{ \text{out}_1, \text{out}_2, \ldots \}$

$\text{EVAL} (\{ \text{out}_1, \text{out}_2, \ldots \}) \rightarrow \{\text{out}_k\}$

This gives us a theory which *derives* an output from an input. The derivational perspective is suggestive of a computation of Gen and Eval (Ellison 1994), and suggests the possibility of extending the structure of the theory by re-applying the Gen-Eval combination in successive steps, either open-endedly (Prince and Smolensky 1993:4-5, McCarthy 2000), or a minimal number of times (Rubach 2000). In rule-based derivations, structures are generated one from the other in series, (24a). Prince and Smolensky (1993) developed the alternative in (24b) whereby several structures are generated together.

(24) a. Serial Generation b. Multiple Generation c. No Generation



[12] The seminal texts say that Eval "comparatively evaluates", "rates", "imposes an order on" the forms input to it, though their schematic representation, repeated here, shows a filter which outputs a single form rather than an ordering of forms.

The following sections argue that Optimality Theory has reached, without acknowledgement, a stage where generation of structure from structure plays no part at all (24c), and the grammar is purely an evaluative filter. While this requires abandonment of the popular conception of the theory, the result is a more explanatory system.

2.3.2 How Are Candidates Admitted?

To give concrete motivation to the discussion, a tableau is selected from the literature (Myers 1997a, 1997b) concerning some tone alternations in Shona. The brief data in (25) show that the high tone realised on the vowels of the word for ’knife’ is lost when the word follows the copula proclitic, an [i]-vowel itself with a high tone.

(25)	a.	bángá	’knife’
	b.	í banga	’it is a knife’

For the tableau analysis in (26), the concatenated input form comes with both high tones associated to their vowels. The constraints used are defined in (27). However, as candidate (26a), this form violates the OCP which prohibits adjacent identical elements. Candidate (b), for which the second high tone is absent, satisfies the OCP. It turns out that candidate (b) is better than a number of other candidates (c,d&e), and is passed as optimal. This predicts the surface form (25b) given above.

(26)

Input: H H | / \\ i banga	OCP	PARSE(T)	LEFT-ANCH	MAX-IO(T)	MAX-IO(A)
a. H H | / \\ i banga	*!				
☞b. H | i banga				*	**
c. H H / \\ i banga		*!			*
d. H H | | i banga			*!		*
e. H / \\ i banga			*!	*	**

(27) OCP Identical tones on adjacent tone-bearing units are prohibited.

PARSE(T) A tone must be associated with a tone bearer.

LEFT-ANCH If an output syllable α bears a tone, then α is the leftmost syllable in a tone span if and only if its input correspondent is the leftmost syllable in a tone span.

MAX-IO(T) Every tone in the input has a correspondent in the output.

MAX-IO(A) Every association relation in the input has a correspondent in the output.

Ranking: OCP, PARSE(T), LEFT-ANCH >> MAX-IO(T) >> MAX-IO(A)

Without pausing to examine the details of the rejection of the other candidates, we immediately raise a different point. What about the forms in (28) as candidates?

```
(28)   a.     H   H              b.     H  nga
              |                         |    / \
              i  banga                  i  ba  H
```

Candidates are typically admitted *onto linguists' tableaux* if they are plausible realisations and differ in crucial and informative ways from the selected one. In order to verify that a form is optimal, it is sufficient to show that alternatives which avoid its violations result only in worse ones (the Cancellation/Domination Lemma, Prince and Smolensky 1993). If, in order to avoid the MAX-IO(T) violation of candidate (b), we consider a candidate for which the second tone is retained as a floating tone, we will find that it does indeed fatally violate PARSE(T) as well as some other constraints. But this is the structure (28a). *In principle*, however, a large and potentially infinite quantity of candidates are produced by Gen. Myers (1997a) happened to omit candidate (28a,b), just as he also omitted the structure known as the Eiffel Tower, but are these admitted in principle?

Fortunately Myers is explicit about Gen. He assumes that Gen produces candidates from the input by freely employing optional, unordered operations that include insertion, deletion, linking and delinking of elements (Myers 1997a). (28a) is indeed arrived at by delinking and deleting the second tone, so is a candidate in principle. The formal monster in (28b), and the Eiffel Tower, are not produced by such operations. However, neither the Eiffel Tower not (28b) actually look like phonological forms. Is it possible to come up with other structures that are phonologically interpretable (unlike (28b) and the Eiffel Tower), but will still *not* be produced as candidates? Despite the received picture of OT grammar in (23), if Gen plays no decisive role and is merely in the background, then the statement that candidates are 'provided by Gen' lacks serious theoretical content.

2.3.3 The Theoretical Role of Gen

Gen is described by its creators as a function, with the qualities explained in (29). Let us then examine its nature as a function.

(29)

a."Gen consists of very broad principles of linguistic form, essentially limited to those that define the representational primitives and their most basic modes of combination." (McCarthy and Prince 1994:337)

b."Gen contains information about the representational primitives and their universally irrevocable relations." (Prince and Smolensky 1993:4)

c."Gen... generates for any given input a large space of candidate analyses by freely exercising the basic structural resources of the representational theory." (Prince and Smolensky 1993:5)

From (29a&b) particularly, it appears that principles of linguistic structure are intrinsic to what Gen is. Thus, Gen produces phonological structures, but in doing so might be understood as actually defining phonological structure, *generatively*, starting from some initial structures. If one asks the question of what constitutes a phonological structure, the answer will be:

- Any structure generatable from an input structure by Gen, and any input structure itself, is a phonological structure.

This faces the problem that, although the structure known as the Eiffel Tower cannot be generated from a phonologically plausible input, there seems nothing to stop the postulation of the Eiffel Tower, or (28b), as an input, and hence, by definition, as a phonological structure. And if we have (28b) as an input, we will have variations of this monster delivered by Gen as candidate outputs. However, there is another answer to what constitutes a phonological structure which is more principled, not dependent on some contingent input structures. This is an *axiomatic* definition:

- Any object consisting of phonological primitives (features, prosodic units) related to each other by some basic principles of permissible combination is a phonological structure.

This excludes the Eiffel Tower, and, with a little more work, (28b), but would be expected to admit all the input and output forms on tableau (26), and so on.[13]

To confirm the primacy of the latter definition, we refer to the basic mathematical theory of functions (Partee, ter Meulen and Wall 1990:30ff). A function is a set of ordered $\langle a,b \rangle$ whose left and right members are taken from two sets A and B, such that the right members are uniquely determined from the left members. Prince and Smolensky (1993:4) tell us that "each input is associated with a candidate set of possible analyses by the function Gen". So for Gen, the left members are the input structures and the right members are the candidate sets, and each input is uniquely associated with one particular collection of candidates. But these entities – inputs, candidate sets – rest on there being a pre-defined set of phonological structures. The situation is as in (30). First, the representational theory specifies what phonological structures may contain, which defines the set of possible phonological structures P (an explicit formulation is in Bird 1995). Gen is a function from structures (members of P) to sets of structures (members of the set of subsets of P, or **power set** of P). In each language a finite subset of these structures are the

[13] Quote (29c) seems closer to this approach, for it recognises a "representational theory" whose resources Gen must supposedly draw on, and which must therefore be distinct from Gen.

underlying forms, the set Un. The restriction of Gen to Un gives us the candidate sets from which the grammatical forms of that language are selected.

(30) Some Entities in the Theory

P the set of possible phonological structures

Gen a function from P to the power set of P

e.g. $p_k \rightarrow \{p_1,p_2,p_3,....\}$ where the p_i are phonological structures

Un a finite subset of P – *the set of underlying structures*

$\mathrm{Gen}|_{\mathrm{Un}}$ the restriction of Gen to Un – *which supplies a candidate set for each underlying form*

This brings out the theoretical claim that Gen embodies: the candidate set is a function of the underlying representation; each input determines which candidate outputs are in and which are out. One input form has one candidate set, another input form has another. We now show that this claim has effectively been abandoned.

The admission of candidates is guided by the principle of Inclusiveness:

(31) **Inclusiveness** (McCarthy and Prince 1994:336)

The constraint hierarchy evaluates a set of candidate analyses that are admitted by very general considerations of structural well-formedness.

The intention is that associated with an input is not just one possible derived form, but very many structures varying from the input in very general ways. McCarthy and Prince (1994,1995) build on the formative work of Prince and Smolensky (1993) by liberalising the generation of candidates. Epenthesis sites may be generated in Prince and Smolensky (1993), but McCarthy and Prince (1994) also propose that the segment structures of epenthesis also be generated. They justify this using Makassarese, as reproduced in (32), where epenthesis in coda position is constrained by the Makassarese restriction that admits only glottals stops and velar nasals in codas. Epenthetic segment structures are evaluated against the Coda Condition of Makassarese if they are generated. This condition and other interacting constraints are stated in (33).

(32) Makassarese (McCarthy and Prince 1994:336)

/jamal/	CODA-COND	ALIGNR	FINAL-C	MSEG	NO-NAS
☞jamal | aʔ		*		**	
jamal | aŋ		*		**	*!
jamal | at	*!	*		**	

(33) CODA-COND Codas may only contain ʔ or ŋ.

ALIGNR Align the right edge of the stem with the right edge of a syllable.

FINAL-C Words must end in a consonant.

MSEG Segments must be morphologically sponsored.

NO-NAS Segments must not be +nasal.

In the opposite case to epenthesis, phonetic absence is provided for in Prince and Smolensky (1993) by failure to parse input segments into syllable structure, but could be captured if input segmental structures are permitted to be absent from candidate outputs. This is advocated by McCarthy and Prince (1995:268), who point out that it avoids the problem of having to specify for an output constraint whether it refers to all elements or only the parsed elements, since they note that both kinds have been tried in the literature. For example, will the OCP prohibit any adjacent, identical elements in a language or only adjacent identical, parsed elements?[14]

Thus we have that candidates may have feature structure that is not present in the input, and they may lack features that are in the input. If these considerations are fully general, as the Inclusiveness principle suggests, then the logical result is that candidate structure varies from the input structure without limit, and Gen admits any, and hence every, possible phonological structure as a candidate, every time (34). This is Inclusiveness at its logical extreme. Whereas a rule generates a new form that differs from its predecessor in a specific and interesting way, Gen generates forms that differ from its input in literally *no* particular or crucial way.

(34) Gen: $\langle \text{in}_a, \text{P} \rangle, \langle \text{in}_b, \text{P} \rangle, \langle \text{in}_c, \text{P} \rangle$, ...

Gen maps every input back to the whole set P.

Gen is now a problematic item to retain in a theory, for it is now *unrestrictive*: nothing is ever crucially excluded. Moreover, Gen is *uninformative*: a different input never has a different candidate set; every input is always mapped to the same thing, ad nauseam. And Gen is *redundant*: what it produces (ad nauseam) is a set that is already known and independently specified. One can simply state – for all cases – that the possible candidate outputs are the members of P: this doesn’t need to be re-derived over and over from every input.

Summarising this section, we have that: (i) Gen is a function; (ii) Gen does not define phonological structure, but instead maps structures to structures and thus itself requires an independent definition of phonological structure; (iii) Gen would be of substance as a function if

[14] As pointed out by Idsardi (1998), however, the parsed/unparsed element distinction and the constraint PARSE is still necessary in order to require prosodic structure at all.

its outputs differed in useful ways depending on its input, but they do not; (iv) the optimal form is in all cases one of *the* set of all possible phonological structures, not one of a set that depends on the particular input at hand.

2.3.4 Maintaining Accountability To The Input

If we remove Gen from the general structure of optimality theory as in (35), however, it is not clear how the optimal output relates to the input. There must be some accountability to the input, without which all words would turn out the same (Chomsky 1995:224) – the one optimal member of P.[15]

(35) EVAL (P) → $\{\text{out}_k\}$

Accountability to the input is provided by the Correspondence Theory of relations between input and output structures developed in McCarthy and Prince (1995). The need for a theory of correspondence to maintain the accountability necessarily follows from the derestriction of Gen, and by incorporating it into the discussion we can demonstrate more comprehensively how the argument against Gen goes through.

As soon as McCarthy and Prince (1994) propose the provision of epenthetic segment structure, it becomes necessary that a distinction be made between segments originating in the input and the potential epenthetic segments. These are discriminated by a constraint MSEG in (33) or in the reformulation by McCarthy and Prince (1995), DEP, defined in (37) below. Augmenting the earlier tableau for Makassarese epenthesis, (36) shows that the candidate

[15] Heck *et al* (2002) argue that in syntax – unlike phonology – accountability to an input and deriving the candidate set from an input are *both* unnecessary.

structure *jamalal* may be arrived at by considering the stem *jamal* to be augmented in two possible ways, with different consequences against the additional constraint of CONTIGUITY.

(36) (Augmented version of (32))

/jamal/	CONTIGUITY	CODA-COND	ALIGNR	FINAL-C	DEP	NO-NAS
☞jamal | aʔ			*		**	
jamal | aŋ			*		**	*!
jamal | at		*!	*		**	
jamal | al		*!	*		**	
jama|la|l	*!	*			**	

(37) DEP A segment in the output must have a correspondent in the input.

CONTIGUITY The portion of the output string standing in correspondence forms a contiguous string.

Thus, the relation between a potential output and the input is not necessarily absolute, and various alternative relationships may be subjected to evaluation against Faithfulness constraints like MAX, DEP, and CONTIGUITY (McCarthy and Prince 1995). Furthermore, however, we may claim that the system of correspondence relations will leave generation redundant by taking over the job of relating the output back to the input which was originally an auxiliary benefit of a restrictive Gen.

Consider how the correspondence relations are assigned. Take the string /blurk/. /blurk/, like any other string, is guaranteed to appear on every tableau, as illustrated in (38). The /blurk/ example is taken by McCarthy and Prince (1995:14) to “emphasise the richness of Gen”, but of course that “richness” is a loss of theoretical content, because it means that Gen plays no role in selecting candidate sets.

(38)

b l u r k	

For the particular tableau associated with an input structure /blik/, say, the correspondence relations considered along with the candidate string /blurk/ are all and only those which relate the contents of /blurk/ to the contents of /blik/. The first six cases of /blurk/ in (39), where correspondence relations are illustrated by means of numerical co-indexing, are some of the many relations that will be evaluated for the input /blik/, but the correspondence relation given for the starred string, corresponding to, say, $/b_ia_{ii}n_{iii}/$, will not be evaluated.

(39)

$/b_1l_2i_3k_4/$	
$\mathbf{b_1l_2u_3rk_4}$	
$\mathbf{b\ {}_1l_2u_3r_4k}$	
$\mathbf{b_1l_3urk_4}$	
$\mathbf{b_1l_2urk}$	
$\mathbf{b_4l_2u_1r_3k}$	
b l u r k	
...	
$\mathbf{*b_ilu_{ii}r_{iii}k}$	

This means that although the output structures themselves that appear on tableaux are not a function of the input structure, the assignment of correspondence *is* a function of the input structure. A correspondence relation must correspond to the input, so the input is the decisive factor in deciding what is and is not an acceptable correspondence relation, even though it is not a factor in deciding what is and is not an acceptable output structure.[16]

Finally, we must crucially note that the argument thus far has been confined, by starting assumption, only to phonological structure. Although the phonological structure of candidate outputs can vary without limit, the morphological structure of the input is often assumed not to vary for the candidate output structures. Thus it could be objected that a Gen function freely generates phonological structure while holding other linguistic structure invariant. However, Gen is not necessary here either: a simple alternative is to assume that morphological and syntactic structure is constant across input/output correspondence relations. Thus, in tableau (36), candidate *jama|la|l* has a morphological Stem *jama...l* because those segments correspond to an input sequence identified as a Stem. Or we could assume that morphological (and syntactic) structure can be assigned freely in candidate structures, and evaluated against constraints. After all, optimality theory is offered as a theory of overall grammar, not merely phonology. Either way, the argument against Gen is not contingent on the simplificatory confinement to phonological structure, and goes through: if correspondence relations are assigned between possible outputs and the input, there is no motivation left for actually generating the outputs from the input.

[16] Of course, we could call the function that assigns correspondences between inputs and possible structures 'Gen' if we wish, as McCarthy and Prince (1995:263) do when they say "one can think of Gen as supplying correspondence relations between S1 [the first string of the correspondence, here the input - RN] and all possible structures over some alphabet", as suggested to them by others. But how is one to think about Gen? Gen is (or was) short for 'generator', and there is now no generation of structure from the underlying structure as there is in rule-based theory and as there is in the theory of Prince and Smolensky (1993), which came prior to the liberalisation which rendered generation vacuous.

2.3.5 Optimality Theory Without Gen

The origin of information on a tableau is now summed up in (40):

(40)

representational primitives and principles of combination

↓

set of possible structures P

↓ ↓ Constraint set *Con*

input in $\in$ P, outputs $\text{out}_i \in$ P Ranking <<

↓ ↓ ↓

/ in /	**C_1**	**C_2**	**C_3**	**…**
$\{\ldots\langle \text{in}, \text{out}_i, R_j \rangle\ldots\}$				

R_j is some correspondence

relation between in and out_i

The output structures that appear on the tableau are not derived from the input. Both come from the set of possible structures, which itself comes from the basic principles of what (phonological) structure looks like. The candidates themselves have a triple form, consisting of the input structure, a potential structure (from P), and one of the logically possible correspondence relations between that output and the input. The presence of the input structure and the correspondence relation within the candidate is necessary for evaluating the preservation of properties of the input in the output. For example, evaluating the preservation of linear order of elements requires reference to input order.

We have shown that, although it would not be viable merely to cancel the Gen function from the usual Gen-Eval scheme without losing all connection between the input and output, the use of correspondence relations between the input structure and the possible output structures enables an Eval function to stand without Gen:

(41) $\text{Eval}\,(\{\, \langle \text{in},\text{out}_i,R_j\rangle \mid \forall \text{out}_i \in \text{P}, \forall R_j \subseteq \text{in}{\times}\text{out}_i \,\}) = \{\, \langle \text{in},\text{out}_k,R_l\rangle \,\}$

For each input, evaluation of all possible 'input,output,correspondence' triple forms delivers some triple (or triples) as optimal.

In this way, Optimality Theory may abandon the generation of structure from structure and shift the explanatory burden totally, not merely primarily, over to evaluation. This brings the theory closer to the actual practice of optimality-theoretic analysis, since, following the liberalisation of Gen by McCarthy and Prince (1995), crucial recourses to Gen are not made anyway. Everything is accounted for, in an alternative formal system – an evaluation system.[17] It now remains to specify more fully how a candidate is evaluated as optimal in the evaluation system.

[17] This is not achieved by Russell (1997) who, while recognising that candidate sets are not a function of inputs, assumes that they are "primitive" (Russell 1997:115). However, it is important that the infinite candidate sets are *not* merely received as unanalysable, unending lists, but are entirely generated from an apt finite definition of phonological representations (Bird 1995). The difference between generating candidates from an input and generating them from the axioms of phonological structure is akin to generating the set of all positive integers from the positive integer '1' under the operation of addition, and generating them by constructing the number system in set theory.

2.4 Evaluation in the Optimality Framework

With generation being shown non-essential to the optimality framework, we now flesh out the form of optimality-theoretic evaluation in this section. We then have a pure evaluation system which can be compared to the generation system of 2.1.

2.4.1 Optimality

In Optimality Theory, the surface form is selected from a number of potential candidates by an evaluation which places them in an order of relative harmony, of which the most harmonic is said to be the ’optimal’ one. A tableau for representing this information takes the form outlined in (42) below, and we shall use its contents to explicate the form of optimality. The candidates, whose internal triple structure was discussed in 2.3.5, are for present purposes abbreviated to atomic alphabet symbols. All violations of constraints posted on the tableau are shown (by ’*’); just some of them are marked as crucial (by ’!’).

(42)

	C1	C2	C3	C4
a	**!	**		
b	*!	*		*
c		*!	*	
d			**!	*
e			*	*!
☞ f			*	
g		*!	*	
h	*!			

Each constraint C_i is a function associating a string of violation marks (*) to each candidate: $C_i(x)$ is the string of marks associated with candidate *x*. Candidates are then ordered as to how many violation marks they incur:

(43) Let *x* and *y* be candidates, and C_i a constraint.

$x \prec^{Ci} y$ iff $C_i(x)$ is longer than $C_i(y)$

x is less harmonic than y with respect to constraint C_i if and only if x violates C_i more times than y does.

Some pairs of candidates are not so ordered. For example, in (42) f and g are not discriminated by constraint C1. Such pairs are equally harmonic or 'iso-harmonic' (f≈g)[18]. The candidates fall into ordered equivalence classes of iso-harmony, {candidates with no marks} ≻ {candidates with one mark} ≻ {candidates with two marks} ≻ ...etc.

Moving from column to column, the evaluation is cumulative. The higher-ranked constraints have priority in discriminating between candidates, and the lower-ranked constraints discriminate more and more candidates. Some candidates fair better than others, and only the best survive, the others suffering crucial violations (*!) and they are shaded off for all subsequent columns. The overall harmonic ordering of the candidates in (42) is laid out in (44). Each line in (44) corresponds to evaluation against a portion of the hierarchy. The most harmonic candidates, which escape crucial constraint violations, are given in bold on the left.

[18] A reflexive, symmetric, transitive relation - an equivalence relation. The axiomatic properties are easily verifiable by consideration of the equivalence of having the same length of violation marks.

(44)

Portion of hierarchy considered	Overall relative harmonies
	abcdefgh
C1	**cdefg** $\succ$ bh $\succ$ a
C1>>C2	**def** $\succ$ cg $\succ$ bh $\succ$ a
C1>>C2>>C3	**ef** $\succ$ d $\succ$ cg $\succ$ bh $\succ$ a
C1>>C2>>C3>>C4	**f** $\succ$ e $\succ$ d $\succ$ cg $\succ$ h $\succ$ b $\succ$ a

Some harmony ratings are crucial to the non-optimality of the poorer-rated candidate: b $\prec^{C1}$ c, b is crucially violated by C1, c is not. Some are irrelevant to optimality: b $\prec^{C4}$ h: b does have more C4 marks than h, but both crucially violate C1 anyway), and some are overridden (b $\prec^{C4}$ a: b has more C4 marks – a has none, but a has more C1 marks and C1 is a higher constraint).

The ordering of relative harmony imposed by one constraint may be defined as in (45).

(45) Let ‘Cands’ be the set of candidates, and C$_i$ a constraint. Then

Eval$_{Ci}$(Cands) =$_{def}$ { ⟨*x*,*y*⟩ *x*,*y* e Cands such that C$_i$(*x*) is longer than C$_i$(*y*) }

Eval$_{Ci}$(Cands) is the ordering of candidates in relative harmony imposed by C$_i$.

Evaluation with respect to an entire constraint hierarchy Γ only admits ratings which do not contradict those imposed by higher-ranked constraints. This is defined in (46):

(46) Let Γ be a hierarchy of *n* ranked constraints C1, C2, …, C*n*

$\text{Eval}_{\Gamma} = \text{Eval}_{(n)}$, where

$$\text{Eval}_{(i)} =_{\text{def}} \begin{cases} \text{Eval}_{C1} & \text{if } i=1 \\ \text{Eval}_{(i\text{-}1)} \cup \text{Eval}_{Ci}/\text{Eval}_{(i\text{-}1)} & \text{if } i>1 \end{cases}$$

Eval$_{\Gamma}$ accumulates from each constraint Ci any discrimination in harmony between candidates not garnered from the higher constraints C1,…,C(i-1).

The most harmonic candidates at each stage, which escape crucial violations in (42) and are presented in bold in (43), may be picked out as in (47):[19]

(47) (i) For each *i*, $\max(\text{Eval}_{(i)}) = \{\, a : \neg\exists b \text{ such that } \langle b,a\rangle \in \text{Eval}_{(i)} \}$

max(Eval$_{(i)}$) is the subset of maximally harmonic candidates with respect to the hierarchy C1,…,Ci

(ii) $\max(\text{Eval}_{\Gamma}) = \{\, a : \neg\exists b \text{ such that } \langle b,a\rangle \in \text{Eval}_{\Gamma} \}$

max(Eval$_{\Gamma}$) is the subset of maximally harmonic candidates with respect to the entire hierarchy – i.e. the optimal candidate(s).

So while the derivational framework places structures in a sequence of which the end is the surface form, an OT grammar specifies harmony relationships between structures, of which the maximally harmonic candidate, or *optimal* candidate, contains the surface form for the underlying, input, form. How different harmony is from derivation remains to be examined.

[19] An alternative formulation is where filters for each of the constraints successively cream off the best of the best until all of the filters have been used and we are left with the optimal form (Eisner 1997a).

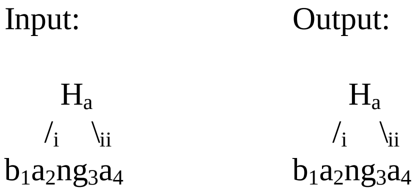
2.4.2 Optimality-Theoretic Constraints

Optimality-theoretic constraints are *violable*, but violation is *minimal* (McCarthy and Prince 1994:336). This appeals to a notion of *degree* of violation, spelt out by violation marks or the closely related harmony rating (no marks - rated top, 1 mark - rated next, etc.). An optimality-theoretic constraint C is a function from candidates (two structures in correspondence) to strings of *'s. A fundamental issue is how to define linguistic constraints which register violation marks against candidates for each point at which they fail to meet some linguistic requirement.

In Optimality Theory, phonology is seen in terms of interactions among two kinds of constraints, Faithfulness constraints and Markedness constraints (Prince 1997a, McCarthy 2002). Faithfulness constraints require the correspondence relation between the two structures to conform to some property – essentially keeping input and output alike in some particular respect. Markedness constraints place requirements particularly on output structures themselves. Although other kinds of constraints have sometimes been countenanced (e.g. Archangeli and Suzuki 1997, McCarthy 1999a), they fall outside the core proposals of Optimality Theory and we shall leave them aside here.

We adopt the autosegmental view of phonological representations as graphs (Goldsmith 1976, Coleman and Local 1991, Bird 1995) and assume that correspondence relations exist between these graphs. In particular, phonological representations consist of several **nodes** which occupy a number of **tiers** in which all the nodes are of the same sort, a particular feature, or segmental root node, etc. On each tier the nodes are **ordered**, and nodes on different tiers are related by **association** (associations are also ordered). In the correspondence relation, the elements (nodes) on a particular tier in the input representation take correspondents on the equivalent tier in the output representation. In (48) we give a simple example of coindexing between input and output for tonal and melodic tiers.

(48)



However, the “melodic tier” itself decomposes into a series of tiers for the segmental root and the various individual features. This view of correspondence then follows Lombardi (2001) and Zoll (1998) in assuming that Faithfulness constraints exist for all the feature tiers used in phonological representation.[20] Each feature tier τ will have the following Faithfulness constraints:

(49)

a. MAXIMALITY$_\tau$ (MAX$_\tau$),

Every element in the input has a correspondent in the output.

b. DEPENDENCE$_\tau$ (DEP$_\tau$)

Every element in the output has a correspondent in the input.

c. IDENTITY$_\tau$ (IDENT$_\tau$)

Correspondent elements have identical values.

d. LINEARITY$_\tau$

The order of input elements is preserved among their output correspondents.

e. INTEGRITY$_\tau$

No element in the input has more than one correspondent in the output.

f. UNIFORMITY$_\tau$

No element in the input has more than one correspondent in the input.

[20] However, we should not postulate Faithfulness constraints on prosodic constituents (syllable, foot, etc.), since no such effects are observed (McCarthy 2003).

These constraints specify all the recognisable natural properties in the mathematics of relations. If all these properties are met on each tier within a representation and the output is *fully faithful* to the input, then we have an identity isomorphism between the tier structure in the input and the tier structure in the output. If on the other hand some property is not met, violation marks will be awarded for each exception to the property. For example, MAX will award violation marks for *each* input element that has no correspondent in the output. Further Faithfulness constraints may have linguistic motivation. Thus, McCarthy and Prince (1995) propose CONTIGUITY constraints (requiring preservation of the word- or morpheme-internal sequence without insertion or deletion) and ANCHOR constraints (requiring retention of initial and final elements).

We may now turn to Markedness constraints. Some Markedness constraints are defined by a structural configuration. Examples are given in (50).

(50)	LO/TR	’No [+low] feature is accompanied by a [+ATR] feature’
	NONFINALITY	’No prosodic word ends with a foot edge’
	NOGEMINATES	‘No segmental root may be associated to two timing units’
	OCP_τ	‘No adjacent identical elements on tier τ’

Candidates incur violation marks each time the output structure (not the input structure) contains this configuration, so with LO/TR, every instance in an output structure where [+low] and [+ATR] coincide warrants a violation mark. Other Markedness constraints have an implicational form, requiring that if some structural node (or possibly sub-configuration of nodes) is present, it is accompanied by another. Examples are given in (51):

(51) NasVoi ’every nasal is accompanied by voicing’

Align(PrWd,R,σ,R) ’the right edge of every word coincides
with the right edge of a syllable’

Onset ’every syllable has an onset’

Every time the implication fails in the output structure, a violation mark is given. Thus, for Onset, each syllable in an output structure that does not have an onset warrants a violation mark. These two simple schemes, negation and implication of certain structural configurations, cover typical proposals for Markedness constraints.[21]

Some Markedness constraints have been given a powerful facility of awarding different kinds of violation mark, some more severe than others. This would complicate the formulation of harmony evaluation given here; however, we would argue that this facility is superfluous. There are two contexts in which it has arisen. The first is in connection with natural phonological scales. HNuc (Prince and Smolensky 1993) marks syllable nuclei with increasing severity the lower down the sonority scale they are (low vowels > mid vowels > high vowels > approximants > nasals > voiced fricatives > voiceless fricatives > voiced plosives > voiceless plosives). However, as Prince and Smolensky (1993:81) observe, this can be replaced with a finite series of constraints for each sonority level: *Peak/voiceless plosives >> *Peak/voiced plosives >> *Peak/voiceless fricatives >> … It is a general result that any constraint with a finite set of violation marks may be so reduced (Ellison 1994). This is shown in the text box below.

[21] Eisner (1997b) goes further and proposes that OT constraints be limited to specifying a negative or implicational relationship simply between a pair of structural elements or edges of constituents, such as ‘syllable’ and ‘onset’,or ‘low’ and ‘ATR’.

Reduction of a finite violation mark hierarchy:

Suppose a constraint C produces N different kinds of marks, $m_1,m_2,...,m_N$, in increasing order of severity ($m_1 \succ m_2 \succ ... \succ m_N$);

for each candidate *c*, C determines a list C(*c*) of marks, concatenations not of *'s but of $\{m_i\}(1 \leq i \leq N)$.

To separate out each kind of mark, let $f_1(C(c))$ be the string containing only the marks m_1 from C(*c*), and define f_i, i=2,..,N similarly;

C can be replaced by binary constraints $C_1,C_2,...,C_N$ such that $C_i(c)=f_i(C(c))$, so that each mark type is taken over by a separate constraint. Just as a mark m_2 is more costly to a candidate than a mark m_1, violation of C_2 is concomitantly more costly than C_1, and adoption of the ranking $C_2 >> C_1$ captures precisely this. In general, the mark hierarchy $m_1 \succ m_2 \succ ... \succ m_N$ is converted to the ranking $C_N >> ... >> C_1$, a hierarchy of constraints each assigning strings of a single mark.

The second place where severity of violation has been used does not give a finite hierarchy of marks. The Tagalog affix *-um-* appears as a prefix as in *um-aral* ‘teach’ provided that *m* is not parsed in a syllable coda position (V__.C). If this cannot be met, then it infixes as close to the left as possible: *gr-um-adwet *um-gradwet* ‘graduate’ (McCarthy and Prince 1993a:101). The pattern is analysed using the constraint NOCODA, 'every syllable has no coda' ranked above ALIGN([*um*]$_{\text{Af}}$,L,Stem,L), 'the left edge of every *um* affix coincides with the left edge of a stem' (the root and *um* affix together constitute a stem). A tableau for the infixed Tagalog form for 'graduate' is given in (52).

(52)

/um/,/gradwet/	NOCODA	ALIGN-*um*
a. [**um**.grad.wet.	***!	
b. [g-**um**.rad.wet.	***!	*(1)
c.☞[gr-**u.m**ad.wet.	**	*(2)
d. [grad.w-**u.m**et.	**	*(5)!

In candidates b.,c.,d. the affix is misaligned and so violates ALIGN-*um*. But it does so with increasing severity because of the increasing distance from the left edge of the stem. Once candidates a.,b. are eliminated by the other constraint NOCODA, the choice between c. and d. is settled purely on the fact that the misalignment in c. is less severe than that in d. This cannot be replaced by a hierarchy of constraints because the distance of misalignment depends on the number of segments in a stem, and this is not bounded – phonological structures are of arbitrarily length. There is however, another solution, namely that we reformulate the constraint as a generalisation about intervening material (Ellison 1994, Zoll 1998):

(53) NOINTERVENING:Segments([um],L,Stem,L) = 'Given an *um* affix, there is a stem such that no segments intervene between the left edges of the two'

This simply associates two violation marks * * to /gr-**u.m**ad.wet./ for the segments *gr* and five violation marks * * * * * to /grad.w-**u.m**et./ for the segments *gradw*, in the normal way. While Ellison (1994) and Zoll (1998) appear to implicitly assume such constraints generalise over segments, other NOINTERVENING constraints might generalise over another structural entity, such as the syllable, the feature [+nasal], etc.[22] This reformulation shows that constraints of alignment which incorporate marks of unbounded severity do reduce to constraints which use *'s, again preserving the formalisation of harmony evaluation already given.

In general, as we showed, a finite hierarchy of marks can be reconstructed with constraints employing a single string of marks; the difficulty comes when there is no bound on the possible severity of violation, since one could not then generate a set of constraints to replace the degrees of severity. The source of motivation for such a constraint is limited, however: any set of marks motivated by scales within substantive phonological theory will be finite, e.g. prosodic hierarchy, feature dependencies, sonority hierarchy, markedness hierarchies. Phonetic scales referring to articulatory or acoustic dimensions are infinite, but in practice it appears that only a certain list of threshold values are relevant to phonological analysis (e.g. Kirchner 1996). Unboundedness in phonology arises instead in the arbitrarily large size of phonological structures. This offers the possibility that constraints may have a severity of violation that increases with the size of a structure. Such constraints (such as constraints of constituent alignment) must have a unit of measurement of severity such as the segment or syllable. We know that alignment constraints of this kind reduce to generalisations on the structural element used as the unit of measurement, and other generalisations of similar complexity (unreported, but perhaps requirements of adjacency or licensing are abstract possibilities) are likely to reduce the same way. These considerations leave the window of plausible constraints with irreducible unbounded sets of violation marks vanishingly small.

[22] McCarthy (2002b) proposes that it is *prosodic constituents* such as the syllable or foot that are prohibited from intervening between two boundaries in a representation.

2.4.3 The Structure of Optimality Theoretic Grammar

The specification in (54) now outlines optimality-theoretic grammars.

(54) An Optimality-theoretic grammar is a quintuple $\langle$P, Un, Corr, Γ, Eval$_\Gamma$ $\rangle$ where:

P is the set of possible phonological structures

Un is a finite set of phonological structures

Corr: in$\mapsto\langle$in,P,in$\times$P$\rangle$ is a function which takes a phonological structure *in* as an input and associates with it triples $\langle in,p,in\times p\rangle$ for all phonological structures p and for each p, all correspondence relations between in and p. These are the candidates for a phonological input.

$\Gamma = \langle$ CON, << $\rangle$, a set of constraints CON with an ordering <<, where the constraints are functions which associate a string of *'s to triples $\langle in,p,in\times p\rangle$.

Eval$_\Gamma$ defines an ordering $\prec$ on triples $\langle in,p,in\times p\rangle$ from Γ.

This means that surface forms are determined as follows. For any underlying representation *un* in Un:

→The candidates are the set of triples given by Corr(*un*) – all structures in all correspondences to *un*

→The candidate triples are ordered by Eval$_\Gamma$(Corr(*un*)) – the harmony scale

→The optimal triple (or triples) is picked out by max(Eval$_\Gamma$(Corr(*un*)))

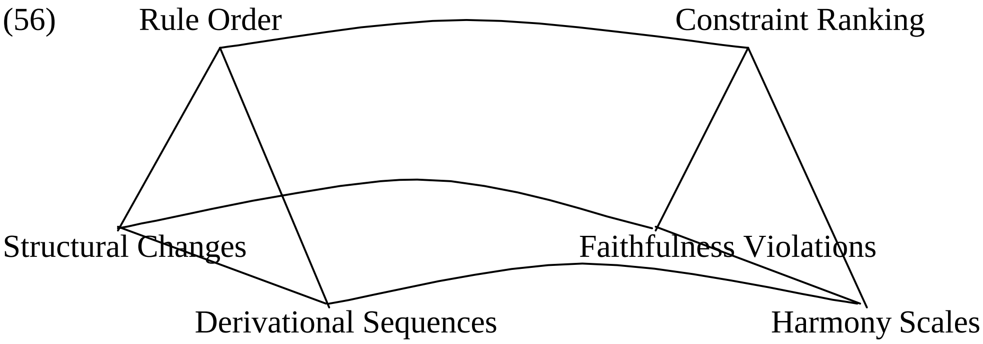
→The second member of this triple (or triples) is the corresponding surface representation.

2.5 Programme For Structural Comparison

We have a rule-based generation system in the derivational framework (constraints being underformalised and eliminable) and a constraint-based evaluation system in the optimality framework (generation being redundant).

A generation system and an evaluation system which generate the same surface forms from the same underlying forms describe the same function. A generation system and evaluation system which describe the same function are comparable in structure at three points:

(56)



Rules are comparable to Constraints in that Structural Descriptions and Structural Changes of rules are comparable with Markedness / Faithfulness interactions, and the fact that an ordering relation is defined on both rules and constraints. And the derivational sequences of structures, the last of which is the surface form, is comparable with the relative harmony of structures, of which the most harmonic is the surface form. These three structural analogies provide the basis for comparative studies of the frameworks, which we will pursue in the next three chapters.