

How Are Inputs Generated in Optimality Theory?*

Hideki Zamma

1 Introduction

In Optimality Theory (cf. Prince and Smolensky (1993) and McCarthy and Prince (1995) among others), it is assumed that phonological outputs are generated from their inputs by the function called Gen, which is the abbreviation for 'Generator.' Gen can produce any candidate outputs from an input, and a single output is selected as 'optimal' through the function H(armonic)-Eval(uation). This process is schematized in the following model:

- (1) a. Gen (In_k) \rightarrow {Out₁, Out₂, ...}
b. H-Eval (Out_i, $1 \leq i \leq \infty$) \rightarrow Out_{real}

(Prince and Smolensky (1993:4))

In this model, however, it is not obvious how inputs themselves are generated. Gen creates outputs for an input, but it cannot affect inputs in any way. Then, what 'generates' inputs at all?

This issue, in fact, has been rarely discussed in the OT literature so far. In this paper, we will attack this untrodden peak, and propose that a refinement of the theory is necessary to accommodate this issue, showing that the current OT framework is problematic. In particular, we will argue that there is an independent component which generates inputs. The new model, which is reminiscent of that of Harmonic Phonology (cf. Goldsmith (1990, 1993)), will serve to reduce the cost which OT inherently involves.

2 Lexicon Optimization

The only argument concerning the input issue in the OT literature is on the mechanism called Lexicon Optimization (cf. Prince and Smolensky (1993), Itô, Mester and Padgett (1995),

* This paper originated in a talk presented at the monthly meeting of the Tsukuba Circle of Phonologists, held on July 12th, 1997. I would like to thank the following people for comments and discussion: Shosuke Haraguchi, Toshiyuki Tabata, Takeru Honma, Masao Okazaki, Teruo Yokotani, Kan Sasaki, Ayako Hashimoto, June-ko Matsui (TCP members); Takeshi Shimada, and Joe Morita (TES reviewers). I am also grateful to Wayne Lawrence, who, not only gave me insightful comments, but also corrected stylistic errors in this article. This work was supported by JSPS Research Fellowships for Young Scientists and by Grant-in-Aid for JSPS Fellows.

Inkelas (1995)). Prince and Smolensky states this device in the following way:

(2) Lexicon Optimization (Prince and Smolensky (1993:192))

Suppose that several different inputs I_1, I_2, \dots, I_n when parsed by a grammar G lead to corresponding outputs O_1, O_2, \dots, 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 .

In other words, when several inputs are possible to attain the correct output, Lexicon Optimization says that the input which incurs the least serious violation must be chosen.

To embody this abstract idea into a concrete analytical method, Itô, Mester and Padgett (1995) propose a technique which they call "tableau des tableaux" (p.593). In (3), we will illustrate how this method works, with a Japanese word *tombo* 'dragonfly' as an example:

(3)

	Input	Output	License	NasVoi	Faith
a.	/tompɔ/ v	t o m b o v			*!*
b.	/tompɔ/ v	t o m b o v			*!
c.	/tombo/ v	t o m b o v			*!
d.	/tombo/ v v	t o m b o v			
e.	/tombo/ v v	t o m b o v v			*!*

The letter 'v' in the inputs and the outputs indicates [voice] feature. All the inputs in (3a-e) produce the correct output with the constraint ranking given. None of the outputs violate the constraints **License**, which requires that all features be licensed, and **NasVoi**, which requires

that nasals be voiced, but they differ in the number of **Faith** violations. By virtue of Lexicon Optimization, the input in (3d) is selected as the input for *tombo*, since it incurs no violation of **Faith**, which is the fewest among the inputs.¹

In essence, the role of Lexicon Optimization is the same as that of the Strict Cycle Condition of Lexical Phonology (cf. Kiparsky (1982, 1985)), which was first formulated by Mascaró (1976). We will give the formulation from Kiparsky (1982:41), omitting the strict definition which is irrelevant here:²

(4) Strict Cycle Condition

Cyclic rules apply only to derived representations.

As Kiparsky (1968, 1973) argues, given a phonological rule which neutralizes underlying distinctions, the underlying representation for non-alternating words becomes so ambiguous that two forms could be posited. For example, English Trisyllabic Shortening neutralizes both underlyingly long and short vowels to short vowels on the surface, when they lie in the antepenultimate syllable; e.g. *divine* → *divinity* (cf. Chomsky and Halle (1968)). So a word such as *camera* can be represented in two ways in underlying representation; one with an underlying short vowel, and the other with a long vowel which is shortened by Trisyllabic Shortening. The Strict Cycle Condition solves this problem: since *camera* is a monomorphemic word and thus does not constitute a derived environment, the word cannot undergo Trisyllabic Shortening, and hence it has an underlying short vowel. Lexicon Optimization plays the same role: when multiple inputs are possible to produce a certain output, the one which shows minimal disparity from the output is selected.

Although non-derived words are required to have the same representation for the underlying form and the surface form by virtue of the Strict Cycle Condition in Lexical Phonology, there are some phonological rules which apply to those words; that is, the rules

¹ See Itô, Mester and Padgett (1995) for details.

² Cole (1995) argues that the Strict Cycle Condition can be dispensed with, given Kiparsky's (1968, 1973) Revised Alternation Condition:

(i) Revised Alternation Condition

Obligatory neutralizing rules apply only in derived environments.

According to her, there are few arguments for the Reaching Back effect, which is another effect of the Strict Cycle Condition, and so the Revised Alternation Condition plays the role of restricting rule application more properly. We will not involve ourselves in this issue, because it is beyond the scope of this paper.

which produce allophones. For example, the /p/ of *pin* is aspirated to [p^h] although it is a monomorphemic word. Such rules are considered to apply at the post-lexical component, where the Strict Cycle Condition is dropped. This assumption of post-lexical rule application is based on another principle in Lexical Phonology, called Structure Preservation:

(5) Structure Preservation (from Steriade (1995:125))³

No lexical rule application will generate structures prohibited underlyingly.

In other words, one important role of this principle is to restrict lexical rules to produce only the phonemes of a language.⁴ Since [p^h] is not a phoneme in English, it is produced at the post-lexical level, where principles of lexicon do not apply.

In Optimality Theory, on the other hand, such principles are not assumed: Gen can alter inputs freely so long as the output satisfies important constraints, irrespective of whether the word is monomorphemic, or whether the segment produced is allophone. What is more, Lexicon Optimization predicts that some segments which are traditionally regarded as allophones are present in the input form. For example, Itô and Mester (1996) conclude that some Japanese words contain [ŋ], which is traditionally assumed to be a word-medial variant of /g/, as the input representation through Lexicon Optimization. In other words, Optimality Theory abolishes the traditional distinction between phonemes and allophones completely, and assumes that any segment be present in the input or produced by Gen. Does this abolition not raise problems in phonological theory? In fact, it does. In the following section we will see what kind of problems the present OT framework raises.

3 Problems Concerning Input Representations

3.1 Phonemes and Allophones

Allophones which appear in monomorphemic words do not raise problems in Optimality Theory, because, given Lexicon Optimization, such segments are considered to be present in input forms.⁵ Allophones in polymorphemic words, on the other hand, are produced by Gen to satisfy the constraints in the language (more precisely, the candidate with the allophone

³ Kiparsky (1985) does not give a specific formulation of this principle, although he introduces the idea. So, Goldsmith (1990) and Kenstowicz (1992), for example, also do not define Structure Preservation in a particular way.

⁴ Other roles concern underspecification, phonotactics, and so on.

⁵ Even these words raise the problem which we will see in the next section.

produced by Gen is selected as optimal through H-Eval). In this case, a phoneme in the input is replaced by an allophone in the output. The phoneme in the input is determined, again, through Lexicon Optimization. Take the Japanese [g]/[ŋ] alternation in Itô and Mester (1996) as an example. The morpheme-initial segment of [geta] is changed into [ŋ] in [niwa-ŋeta] for some speakers. The input is determined by means of correspondence with the monomorphemic form [geta], whose input is determined through Lexicon Optimization.

When polymorphemic words do not have an independent monomorphemic form, however, a problem arises: in such a case, the input form cannot be determined even with Lexicon Optimization. This is the case for Japanese verb inflection. Let us take the alternation of the stem-final segment in [kas-u] 'lend (indicative)' and [kaʃ-i] '(infinitive, nominal)' as an example. Since consonant-final verb stems do not have independent forms with no suffix, the input forms for the stems cannot be determined by means of correspondence. Thus, we must consider at least two cases for each of [kas-u] and [kaʃ-i]: the one with [s] in the input, and the other with [ʃ] in the input.

Before we see this in the familiar OT tableau, we should first define the constraints we use in our discussion. The following constraints will do for the time being:

- (6) a. ***si**: the sequence [si] should not be produced.⁶
 b. ***ʃ**: the segment [ʃ] should not be produced.
 c. **Faith**: the output form must be faithful to the input form.

Now we will see how the alternation is treated with these constraints. First, we will consider the case in which the input contains [s].

- (7) a.

kas-u	*si	*ʃ	Faith
kasu			
kaʃu		*!	*!

⁶ Itô and Mester (1995) propose a constraint called **CV Linkage**, which prohibits coronal onsets to be followed by front vowels. We will use ***si** instead just because this is easier to understand. So, our usage of ***si** does not mean that we deny **CV Linkage**, of course.

b.

kas-i	*si	*f	Faith
kasi	*!		
kas̩ kafi		*	*

In the indicative form (7a), the first candidate wins because the candidate violates none of the constraints. On the other hand, the candidate with [s] is not selected in the infinitive form (7b), because the candidate violates the topmost constraint *si, which is fatal. In this way, the distributional fact is accounted for that [f] appears only when [i] follows it and otherwise [s] appears.

Next, consider the case of the input containing [f] as the stem-final segment.

(8) a.

kaf-u	*si	*f	Faith
kasu			*
kafu		*!	

b.

kaf-i	*si	*f	Faith
kasi	*!		*
kas̩ kafi		*	

In this case, the indicative form [kas-u] is selected at the cost of a **Faith** violation. (Note that the ranking between *f and **Faith** is determined in this case; otherwise the correct output [kas-u] cannot be selected. We will discuss this issue later.)

In addition to the two cases above, we should consider another possibility; i.e. the case of the input with underspecified /S/. In this case, the stem-final segment in the input is underspecified for the feature [anterior].

(9) a.

kaS-u	*si	*f	Faith
kasu			*
kafu		*!	*

b.

kaS-i	*si	*f	Faith
kasi	*!		*
☞ kaf-i		*	*

All the candidates in (9a) and (9b) violate **Faith**, because they are specified with the feature [anterior], which is absent in the input form. Although the tableau above does not contain the output candidates with underspecified [S] (i.e. [kaS-u] and [kaS-i]), such candidates would lose because of the undominated constraint which require full-specification in the output, as Itô and Mester (1996) implicitly assume in their analysis. In this case, again, the actual forms [kas-u] and [kaf-i] are selected in the constraint hierarchy. (The ranking between *f and **Faith** is not determined: note that the correct form is predicted even with the ranking **Faith** » *f.)

Now, with the "tableau des tableaux" procedure, that is, with Lexicon Optimization, we will determine the input form for the cases at hand.

(10) a.

input	output	*si	*f	Faith
☞ kas-u	kasu			
kaf-u	kasu			*!
kaS-u	kasu			*!

b.

input	output	*si	*f	Faith
kas-i	kaf-i		*	*!
☞ kaf-i	kaf-i		*	
kaS-i	kaf-i		*	*!

As the tableau in (10) shows, Lexicon Optimization selects different input forms for each of [kas-u] and [kaf-i], since they are superior to their competitors in that they do not incur **Faith** violations. This indicates that every inflectional form of the verb has a distinct input form, that is, with /s/ or /f/ stem-finally. If this is the case, then it is no longer possible to assume that they are all produced from the same verb-stem and inflectional suffixes: they are independent words.

Correspondence constraints (cf. McCarthy and Prince (1995), etc.) do not solve this

problem. As we have argued, the stem does not appear as an independent word (that is, without being affixed), with which suffixed forms are compared with respect to correspondence. Moreover, even if we assume such a constraint that requires inflectional forms to be identical to the indicative form, the situation does not change. Let us call this putative constraint **Ident** (ind.). This constraint is dominated by ***si**, otherwise *kafi* would never occur. Below are the tableaux including **Ident** (ind.):

(11) a.

kas-u	*si	*ʃ	Ident (ind.)	Faith
kasu				
kafu		*!		*!

b.

kas-i	*si	*ʃ	Ident (ind.)	Faith
kasi	*!			
kafi		*	*	*

(12) a.

kaf-u	*si	*ʃ	Ident (ind.)	Faith
kasu				*
kafu		*!		

b.

kaf-i	*si	*ʃ	Ident (ind.)	Faith
kasi	*!			
kafi		*	*	*

(13) a.

kaS-u	*si	*ʃ	Ident (ind.)	Faith
kasu				*
kafu		*!		*

b.

kaS-i	*si	*f	Ident (ind.)	Faith
kasi	*!			
kas-i		*	*	*

Ident (ind.) does not affect the indicative form, of course: no matter what shape the form takes, the form always satisfies the constraint. On the other hand, the lower candidates of the infinitive form all violate **Ident** (ind.). As for the rest of the constraints, the violation marks appear in the same column as (7), (8) and (9).⁷ The "tableau des tableaux" below selects the optimal inputs:

(14) a.

input	output	*si	*f	Ident (ind.)	Faith
kas-u	kasu				
kaf-u	kasu				*!
kaS-u	kasu				*!

b.

input	output	*si	*f	Ident (ind.)	Faith
kas-i	kafi		*	*	*!
kaf-i	kafi		*	*	
kaS-i	kafi		*	*	*!

Again, Lexicon Optimization selects the different stems for each of the forms. This is because **Ident** (ind.) is irrelevant for the indicative form, and the optimal output for the infinitive form must always violate **Ident** (ind.) and thus all the candidate inputs are on a par with respect to this constraint. Hence, the result is the same as the case in which **Ident** (ind.) is not assumed.

The same is true if we define the relevant constraint in the other way, so that outputs are required to be identical to the infinitive form; i.e. **Ident** (inf.). Below we give the "tableau des

⁷ The place of the constraint **Ident** (ind.) in the ranking cannot be determined in (12), because there is no evidence to show whether this constraint is more important than *f or than **Faith**. Although *f and **Faith** are ranked with respect to each other as *f » **Faith** (from the fact that [kasu] is selected in (12a)), **Ident** (ind.) can be ranked in the same place either as *f or **Faith**. We employ double vertical lines to indicate this situation.

tableaux" of such a case:⁸

(15) a.

input	output	*si	*f	Faith	Ident (inf.)
kas-u	kasu				*
kaf-u	kasu			*!	*
kaS-u	kasu			*!	*

b.

input	output	*si	*f	Faith	Ident (inf.)
kas-i	kafi		*	*!	
kaf-i	kafi		*		
kaS-i	kafi		*	*!	

The optimal outputs cannot help violating **Ident** (inf.) in the indicative form, hence this

⁸ For those who wish to ascertain the correctness of (15), below are the tableaux in which the optimal outputs are selected:

- (i) a. kas-u *si | *f : Faith : Ident (inf.)
 kasu | : : *
 kafu | * : *!
 b. kas-i *si | *f : Faith : Ident (inf.)
 kasi *! | : :
 kafi | * : *
 (ii) a. kaf-u *si | *f | Faith : Ident (inf.)
 kasu | | * : *
 kafu | *! | :
 b. kaf-i *si | *f | Faith : Ident (inf.)
 kasi *! | | * :
 kafi | * | :
 (iii) a. kaS-u *si | *f || Faith || Ident (inf.)
 kasu | || * || *
 kafu | *! || * ||
 b. kaS-i *si | *f || Faith || Ident (inf.)
 kasi *! | || * ||
 kafi | * || * ||

In (i), *f, Faith, and Ident (inf.) do not have to be ranked: even if they are ranked in the same place, the lower candidate in (ia) loses because this candidate has more violation marks than the upper candidate. In (iii), Faith can be ranked in the same place either as *f or Ident (inf.).

constraint has no effect in the selection of the optimal input in (15a). For the infinitive form, this constraint is irrelevant because any candidate vacuously satisfies it. Accordingly, the result is the same: distinct inputs are selected through *Lexicon Optimization* for the indicative and the infinitive forms.

To summarize the discussion so far, *Lexicon Optimization* is problematic in that it selects different inputs for some inflectional forms of a stem. This is an undesirable result, since, if such a divergence were admitted, it would become impossible to analyze inflectional forms as being produced from the same stem. Introducing various correspondence constraints does not solve this problem.

There is another crucial problem with *Lexicon Optimization*. As the reader might have noticed already, ranking of the constraints which is to be compared in the "tableau des tableaux" is often inconsistent among the tableaux of each input. Compare, for example, the tableaux in (7), (8), and (9). For the input forms *kas-* and *kaS-*, the ranking between *f and **Faith** cannot be determined since there is no evidence that suggests it. For the input form *kaf-*, on the other hand, *f is ranked above **Faith**, since, as (8a) shows, violation of the former is fatal in the selection. In (10) we have compared these non-identical rankings, but strictly speaking, such comparison is problematic. In *Optimality Theory*, the degree of improperness is not scaled by the number of violation marks, but by the seriousness of the constraint violation. Then, when the ranking is distinct and thus the importance of the constraints is not the same, is it really possible to compare the degree of constraint violations among such different rankings?

In fact, such a situation frequently occurs in *Optimality Theory*. Previous studies seem to solve this problem by assuming the *Least Common Multiple*-like constraint ranking.⁹ But the cost of doing this is tremendously high and thus should not be supported, as we will see in the next section.

The difficulty of inconsistent constraint ranking derives directly from the premise of the present *Optimality Theory*; that is, the input is unrestricted. Note that the constraint ranking is

⁹ Itô and Mester (1996), for example, assume the following constraint ranking to account for the distribution of [g] and [ŋ] observed among (old) speakers of Tokyo Japanese:

(i) *[ŋ] » *g » **Ident LS** [i.e. **Faith**; HZ]

There is, however, only one input form (i.e. *kagi*) which displays this ranking among the six (three candidates each for two words) forms they consider. For the rest of the candidates, *g and **Ident LS** do not have to be ranked with respect to each other. The ranking in (i) is thus the least common multiple.

not determined until both the input form and the output form are available. The constraint is ranked so that the correct output is selected for a given input. Therefore, if the input form differs, the constraint ranking can be different. If there is only one input for a given form, on the other hand, the ranking problem we have discussed never occurs.¹⁰

3.2 Phonemes and Non-phonemes

The number of segments which a language utilizes is limited, of course. For example, of 78 consonants which the International Phonetic Alphabet lists,¹¹ Japanese uses only 14 as 'phonemes' (i.e. /p/, /b/, /t/, /d/, /k/, /g/, /m/, /n/, /r/, /s/, /z/, /h/, /j/, and /w/).¹² In Optimality Theory, the segments which are not admitted in a given language are prevented from appearing in the optimal candidate by means of constraints which militate against them. Given these segmental constraints S_1 , a set of faithfulness constraints f , and constraints S_2 which require the admitted segments not to appear, the phonemic inventory of a language is defined by the following ranking, as Itô and Mester (1995b) demonstrate:

$$(16) S_1 \gg f \gg S_2$$

S_1 being ranked above f , segments prohibited by S_1 are altered to other segments which are not referred to by S_1 , or completely deleted (depending on the nature of f). Segments referred to by S_2 , on the other hand, appear as they are since f is ranked above S_2 .

Let us consider a hypothetical example in Japanese. When a speaker generates a new input such as /θyk/, perhaps in a new word formation, this word will appear as a different form, say [suki], because /θ/ and /y/ is prohibited by S_1 (and coda /k/ by Coda Condition (cf. Itô (1986))) in Japanese.

¹⁰ This is the case for the rule ordering of the traditional rule-based theory, in which a single input (i.e. underlying form) is available. Rules are ordered so that the correct output (i.e. surface form) is produced.

¹¹ The IPA referred to here is revised to 1993. Phones with diacritics are not included in counting of the consonants.

¹² Wayne Lawrence (p.c.) has pointed out to me that it is possible to establish a different phonemic inventory for Japanese, because some phones are in complementary distribution in a certain environment. We assume the traditional inventory for this language in this paper.

(17)

θyk	S ₁	CodaCon	Faith	S ₂
θyk	*!*	*		*
すき suki			***	***

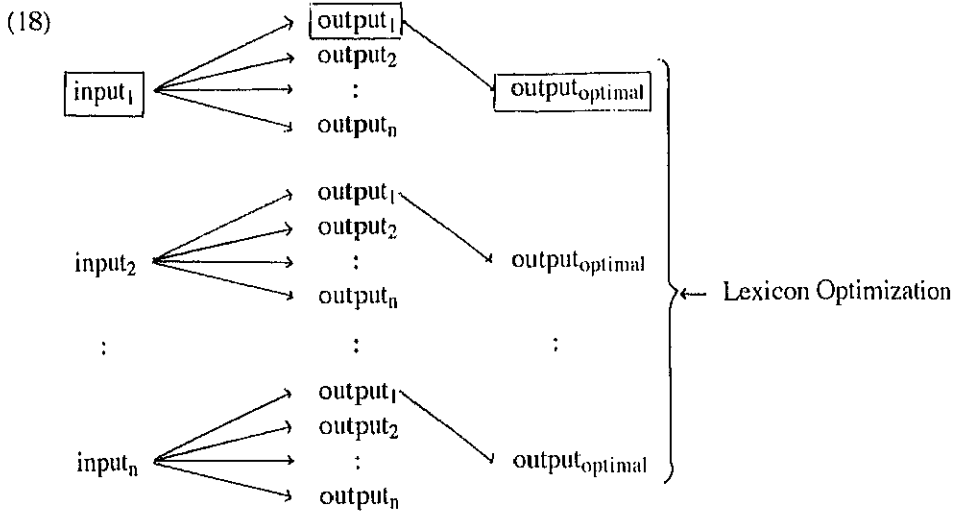
The upper candidate has two violation marks of *S*₁, containing [θ] and [y], but no mark of **Faith**. On the other hand, the lower candidate involves three violation marks of **Faith**, changing /θ/ and /y/ into [s] and [u] respectively and inserting [i] to the right of /k/, but no *S*₁ violation. With the ranking given, the lower candidate wins.

Other output candidates, however, are possible for the input /θyk/; e.g. [tʃiku], [tsuki], [ʃiki], etc. It depends on the ranking of faithfulness constraints concerning relevant features; e.g. **Faith** [anterior], **Faith** [strident], etc. But the ranking cannot be determined, because Japanese does not have such an alternation that the input /θ/ and /y/ are changed to another segment, unlike the case of the allophonic alternation such as /s/ and /ʃ/.¹³

Moreover, the problem with Lexicon Optimization we have seen in the previous section arises also in this case. Note that, in addition to /θyk/, the input form /suki/ is possible for the output [suki], because input forms are not restricted in Optimality Theory. As for the input /suki/, a set of constraints *f* is irrelevant and thus the entire ranking of the constraints is not determined. As we have discussed, Lexicon Optimization cannot work until the consistent ranking of constraints become obvious, but how can we establish the ranking among *f* when there is no evidence?

In addition to these problems, Lexicon Optimization involves a general problem of the cost in its operation. As we have seen, the input /θyk/ may realize as [suki] in Japanese. This means that for the output [suki] 'spade (n.)', /θyk/ is a possible input. Now, the range in which Lexicon Optimization seeks the best input becomes tremendously wide: innumerable inputs such as /ʃix/ and /ðyχθ/ must be taken into consideration in the procedure. Let us illustrate the cost which Lexicon Optimization holds in (18):

¹³ As for the epenthetic vowel after the coda consonant, there is much evidence in Sino-Japanese words that it is /i/ or /u/ (cf. Tateishi (1990), etc.).



As illustrated above, the optimal output can be obtained from every possible inputs after evaluating all the possible outputs Gen can produce for each of the inputs. Lexicon Optimization selects the best input by examining all these optimal outputs. This means that the pair of real input and output is not determined until innumerable possibilities are all examined. What is more, it is after this examination that the Least Common Multiple-like ranking is determined. It is needless to say that this procedure is tremendously costly.

Another potential problem with the cost which Optimality Theory contains is that it is necessary to posit quite a few constraints for prohibiting numerous segments which never appear in a particular language. As I have mentioned, Japanese use only 14 consonants as phonemes, of 78 which the International Phonetic Alphabet lists. In addition, languages often contain consonants with secondary articulation, such as palatalization, nasalization, and aspiration. In order to eliminate most of these sounds from a language, an innumerable number of constraints are necessary.

The number can be reduced when the constraints refer to features (e.g. *[F]), not to individual segments, because prohibiting a feature implies prohibiting a class of segments which include it as a whole. However, it is often the case that some, not all, of the members in a class are prohibited; for example, Japanese has velar plosives but not fricatives (and even nasal as a 'phoneme') while both plosives and fricatives (and also a nasal) are available for alveolar articulation. Thus, if constraints are stated in terms of features, then the combination of the features becomes crucial. Although many of the logically possible combinations can be

excluded by some reasons (for example, nasals cannot be lateral simultaneously), there still remain a huge number of possible featural combinations.

Speaking of combination, the issue of phonotactics may also be a part of the 'cost' problem. For example, English has a limited set of consonant clusters although it allows at most three consonants in the onset and the coda. Although the Sonority Sequencing Principle (cf. Clements (1990), etc.) eliminates some of the clusters, there still remain many illegal clusters which are predicted by the principle to be legal (e.g. /ts/), and legal clusters which should be illegal (e.g. /st/). To posit constraints which eliminate or include these exceptional clusters seems highly costly, because there are too many possibilities in the combination of segments.¹⁴

Finally, another problem concerning the cost arises if we assume underspecification in Optimality Theory (cf. Inkelas (1995)). Recall that we have examined such a possibility in the previous section. In (9) and (13) we have posited for the input an underspecified segment /S/ which lacks specification for [anterior]. This is, however, just one way of underspecification for the segment [s]. A segment, as we have mentioned, consists of a bundle of features, and thus there are possibilities of underspecification for each of the features. So, each of the possible underspecified segments must be examined through Lexicon Optimization, and the cost becomes enormous.

All of these problems might not arise in Golston's (1997) Direct Optimality Theory, in which all lexical items are represented with violation marks of constraints. It is true that it is unnecessary in this theory to examine candidates with impossible segments (because the evaluation is achieved in terms of faithfulness of the lexical violation marks; that is, how well the violation marks are parsed), but there arises a new problem: What kind of violation marks can be employed in a given language?

* * * * * * *

The discussion above does not entail that Optimality Theory itself is problematic.¹⁵ What is problematic is the procedure which determines the input in this theory: the procedure is too costly. This is because Lexicon Optimization is a mechanism which *selects* an input for a given

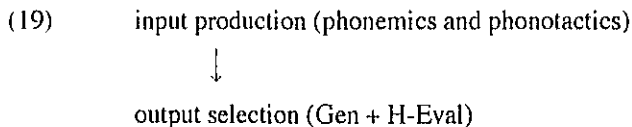
¹⁴ One way to resolve this problem might be to posit templatic restrictions, as Cairns (1988) does. However, it is not obvious how to translate these restrictions into OT constraints.

¹⁵ Actually, McCarthy (1997) does not deny the possibility of multi-stratal OT, which we explore in the next section.

output, not the one which *produces* an input. Equipped with a mechanism of production, none of the problems we have seen in the discussion above will remain in the theory. In the next section, we will explore such a revision of the theory.

4 A Proposal

We have argued that it is necessary to postulate in Optimality Theory a mechanism of input production, which involves restrictions on phonemes and phonotactics. With these restrictions, inputs are generated by utilizing a limited number of phonemes (cf. Structure Preservation) and provided with only one shape. Because the established OT procedure of output selection (i.e. Gen followed by H-Eval) itself is not problematic and inputs must be present before these mechanisms come into play, it is reasonable to assume that there is a component of input production before Gen and H-Eval. This assumption is schematized as below:



Within the framework in (19), none of the problems we have seen above arise. Since phonemes of a language are restricted in number, it is unnecessary to examine the candidates with non-phonemes of that language. In other words, the existence of the component of input production serves to reduce the cost of Lexicon Optimization.¹⁶ Moreover, because each input is provided with only one form, the problem of multiple inputs for a stem never arises.

We will illustrate how our proposal works, taking the alternation found in Japanese inflection as an example. In the component of input production, only those segments which are traditionally called 'phonemes' are present; i.e. /p/, /b/, /t/, /d/, /k/, /g/, /m/, /n/, /r/, /s/, /z/, /h/, /j/, and /w/.¹⁷ An input for a morpheme is created by arranging these phonemes into a legal sequence in the language. For the verb stem expressing the action of lending, phonemes /k/, /a/, and /s/ are arranged to form /kas-/. On the other hand, the form *kaʃ-* cannot be produced as

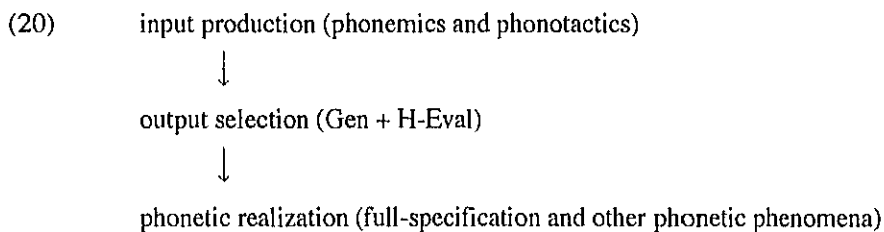
¹⁶ It is not clear whether Lexicon Optimization can be dispensed with completely. We should await further research on this issue.

¹⁷ The phonemes stored in the component of input production can be different among the sublexicons (cf. Itô and Mester (1995a)). Note that loanwords in Japanese can contain segments which are not allowed in native words.

an input, since [ʃ] is regarded as an allophone of /s/ before /i/ in Japanese. Now that the verb stem in question is provided with the form /kas-/ , the indicative and the infinitive forms are created by attaching the suffixes /u/ and /i/ to the stem, resulting in /kas-u/ and /kas-i/ respectively. Gen creates several possible outputs for these inputs, and the forms [kas-u] and [kaʃ-i] are selected as the optimal outputs through H-Eval, which contains the constraints in (6). The output selection in this case might be the same as the tableaux in (7) (or (9) if the phoneme /s/ is underspecified with some feature).

Note that it is unnecessary in this framework to examine the cases of other input forms such as /kaʃ-u/ and /kaʃ-i/. Since only one input is possible for an input, a single tableau for a combination of morphemes is sufficient, in this case (7) (or (9)). It goes without saying that this leads to a reduction in the cost which the current Optimality Theory involves. Moreover, the cost of positing segmental constraints is also reduced. Since phonemes are restricted, it is unnecessary to posit constraints against non-phonemes in output evaluation. In this way, our framework in (19) succeeds in simplifying the present Optimality Theory.

Finally, we consider further refinement of the framework we have proposed. In addition to the components in (19), another component can be posited after the component of output selection; that is, a component of phonetic realization. Although the current OT deals with many phonetic phenomena in its framework, it is not obvious whether ALL the phonetic phenomena can be treated in the component of output selection. Moreover, Honma (1996) proposes that underspecification is necessary even in Optimality Theory, showing that some outputs in Yoruba contain underspecified segments. This suggests the existence of another component which specifies the absent features after output selection. If this is true, the overall picture will be:

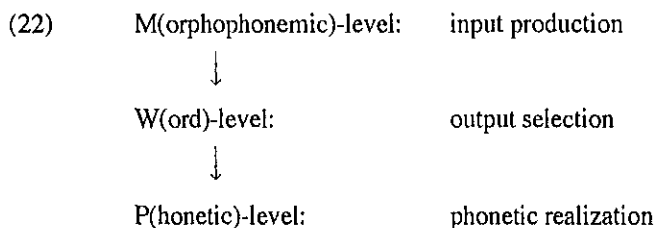


This model reminds us of Goldsmith's (1990, 1993) Harmonic Phonology, in which three levels are posited in phonology. Below are the levels he assumes:

- (21) M-level: a morphophonemic level; the level at which morphemes are phonologically specified.
- W-level: the level at which expressions are structured into well-formed syllables and well-formed words, but with a minimum of redundant phonological information.
- P-level: a level of broad phonetic description that is the interface with the peripheral articulatory and acoustic devices.

(Goldsmith (1993:32))

Although Harmonic Phonology is a theory which utilizes traditional phonological rules, the model in (21) itself, we assume, is not incompatible with Optimality Theory. Recall that three levels seem to be necessary also in Optimality Theory to produce inputs and to specify underspecified features, as we have seen. Because the role of each levels in (21) seem to be similar to that of the levels in (20), it might be possible to correlate them with each other:



Of course, it is after the nature of each component in (20) becomes clear that we can decide whether this correlation is appropriate or not.

In this paper, we have argued that the component of input production is necessary in Optimality Theory, providing empirical and theoretical discussions. On the other hand, little is known about the component of phonetic realization. It is necessary, for example, to limit the range in which Gen and H-Eval (i.e. the mechanisms of output selection) can deal with phonetic phenomena, and to decide what kind of phonetic phenomenon is treated in the component of phonetic realization. Because details of the newly-proposed components (i.e. M-level and P-level, in Goldsmith's terms) are yet not clear, we should await further studies.

5 Conclusion

The present Optimality Theory does not involve a mechanism which produces inputs. As a result, the cost of input selection and constraint ranking is tremendously high. Moreover, Lexicon Optimization includes a problem in its operation, as we have seen in Japanese

inflection. In order to solve these problems, we propose a new model of Optimality Theory, in which a level of input generation precedes the established OT procedures. This model not only raises none of the problems which arise in the current OT, but also reduces the cost that OT essentially includes.

References

- Beckman, Jill, Laura Walsh Dickey, and Suzanne Urbanczyk, eds. 1995. *University of Massachusetts occasional papers in linguistics 18: Papers in Optimality Theory*. GLSA, University of Massachusetts, Amherst.
- Cairns, Charles E. 1988. Phonotactics, markedness, and lexical representation. In *Phonology* 5, 209-236. Cambridge: Cambridge University Press.
- Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Harper & Row.
- Clements, George N. 1990. The role of the sonority cycle in core syllabification. In *Papers in laboratory phonology 1: Between the grammar and physics of speech*, ed. J. Kingston and M. Beckman, 283-333. Cambridge: Cambridge University Press.
- Cole, Jennifer. 1995. The cycle in phonology. In Goldsmith 1995, 70-113.
- Goldsmith, John A. 1990. *Autosegmental & metrical phonology*. Cambridge, MA: Basil Blackwell.
- Goldsmith, John A. 1993. Harmonic phonology. In *The last phonological rule*, ed. J. Goldsmith, 21-60. Chicago: The University of Chicago Press.
- Goldsmith, John A. ed. 1995. *The handbook of phonological theory*. Cambridge, MA: Basil Blackwell.
- Golston, Chris. 1996. Direct Optimality Theory: Representation as pure markedness. *Language* 72: 713-748.
- Honma, Takeru. 1996. Saizensei-Riron to hukanzenshitei (Optimality Theory and underspecification). *Onseigakkai Kaihou (The Bulletin of the Phonetic Society of Japan)* 213: 16-26.
- Inkelas, Sharon. 1995. The consequences of optimization for underspecification. In *Proceedings of NELS 25*, 287-302. GLSA, University of Massachusetts, Amherst.
- Itô, Junko. 1986. Syllable theory in prosodic phonology. Doctoral dissertation, University of Massachusetts, Amherst. [New York: Garland Press, 1988.]
- Itô, Junko, and Armin Mester. 1995a. Japanese Phonology. In Goldsmith 1995, 817-847.

- Itô, Junko, and Armin Mester. 1995b. The core-periphery structure of the lexicon and constraints on reranking. In Beckman, Walsh Dickey, and Urbanczyk 1995, 181-209.
- Itô, Junko, and Armin Mester. 1996. Correspondence and compositionality: The ga-gyô variation in Japanese phonology. To appear in *Derivations and constraints in phonology*, ed. I. Roca. London: Oxford University Press.
- Itô, Junko, Armin Mester, and Jaye Padgett. 1995. Licensing and underspecification in Optimality Theory. *Linguistic Inquiry* 26: 571-613.
- Kenstowicz, Michael. 1994. *Phonology in generative grammar*. Cambridge, MA: Basil Blackwell.
- Kiparsky, Paul. 1968. How abstract is phonology? Bloomington: Indiana University Linguistics Club. [Reprinted in Kiparsky 1973b.]
- Kiparsky, Paul. 1973a. Abstractness, opacity, and global rules. Bloomington: Indiana University Linguistics Club. [Reprinted in Kiparsky 1973b.]
- Kiparsky, Paul. 1973b. Phonological representations. In *Three dimensions of linguistic theory*, ed. O. Fujimura, 1-136. Tokyo: TEC.
- Kiparsky, Paul. 1982. Lexical phonology and morphology. In *Linguistics in the morning calm*, ed. I. S. Yang, 3-91. Seoul: Hanshin.
- Kiparsky, Paul. 1985. Some consequences of lexical phonology. In *Phonology Yearbook 2*, 83-138.
- Mascaró, Joan. 1976. Catalan phonology and the phonological cycle. Doctoral dissertation, MIT.
- McCarthy, John J. 1997. Process-specific constraints in Optimality Theory. *Linguistic Inquiry* 28: 231-251.
- McCarthy, John J., and Alan Prince. 1995. Faithfulness and reduplicative identity. In Beckman, Walsh Dickey, and Urbanczyk 1995, 249-384.
- Prince, Alan, and Paul Smolensky. 1993. Optimality Theory: Constraints interaction in generative grammar. Ms., Rutgers University and University of Colorado.
- Steriade, Donca. 1995. Underspecification and markedness. In Goldsmith 1995, 114-174.
- Tateishi, Koichi. 1990. Phonology of Sino-Japanese morphemes. In *University of Massachusetts occasional papers in linguistics 13*, ed. G. Lamontagne and A. Taub, 207-235. GLSA, University of Massachusetts, Amherst.