UDK 811.531342.622
Izvorni znanstveni članak
Prihvaćeno za tisak 13. travnja 2011.

Charles J. Pindziak II
University of Florida
cpindziak@ufl.edu

Sean M. Witty
University of Florida
wittysan@ufl.edu

Framing Korean Complex–Coda Resolution with Optimality Theory

Korean phonotactics prohibits complex codas; therefore, the language employs two strategies that allow for the breakup of consonant clusters to conform to this prohibition. These strategies include "relinking" (as used by Choo & O'Grady 2003:58–9), that is, the realization of consonant clusters across syllable boundaries, and deletion. The preference of these strategies fits into an Optimality–Theoretic framework, which this paper explicates. By using well established constraints, this analysis complements OT research on similar phonological processes in Korean so that these processes can be unified under a single constraint ranking.

1. Introduction

There are three phonological processes that are important to the phonotactics of Korean: the Head Consonant Rule, Syllable Contact, and Complex–Coda Resolution. The Head Consonant Rule is no longer active in Korean but is important in loanwords that were added to the language during the period in which it was productive. The Head Consonant Rule enforced a prohibition on word–initial liquids. Syllable Contact resolves forbidden intervocalic consonant clusters. Complex–Coda Resolution separates consonant clusters into separate syllables, or deletes one of the consonants of the cluster to produce a simple coda. Since these processes all seem to perform similar functions but on different areas of the word, it would be logical to analyze them together; however, since each process is quite complex, it is easier to handle them separately but in a way such that the analysis of one is compatible with the analyses of the others. For this reason, this paper utilizes modularization of constraints, where each process has a set of constraints, or *module*, that pertains to that process. The ranking of the set of constraints should be compatible with modules for the other processes, so that all the constraints can operate on the same stratum.

Previous Optimality–Theoretic analyses have created modules for the Head Consonant Rule and Syllable Contact. Witty & Pindziak (2010) provide a set of constraints for the Head Consonant Rule. Davis & Shin (1999) and Um (2002), for example, describe Syllable Contact of sonorants with sonority–based segmental constraints. This paper will contribute a module for Complex–Coda Resolution; the Optimality–Theoretic framework to be presented is a complete representation of how Korean avoids the realization of complex codas. With modules for all three of these processes, future research will focus on the interaction of the modules in order to unify these similar phonological processes under a single constraint ranking.

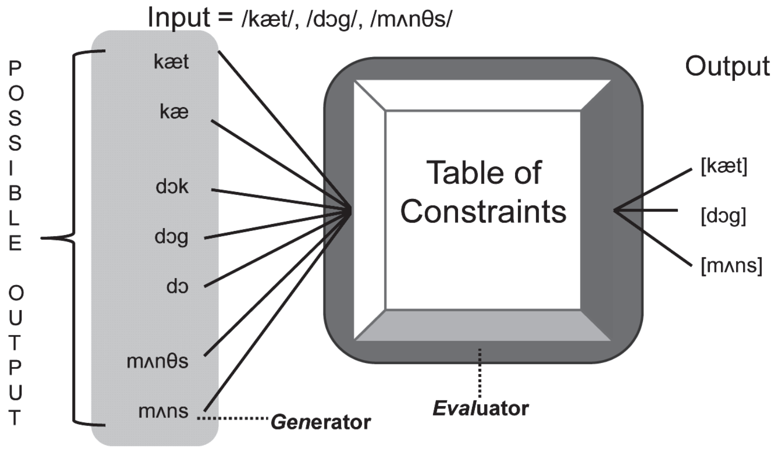
2. Background

2.1 Overview of Optimality Theory

Optimality Theory (OT) is a framework that maps phonemic underlying representations to phonetic realizations using a set universal constraints ranked in a language–specific hierarchy. It differs from traditional rule–based approaches in that it requires no intermediate forms or repair strategies; that is, OT has been successful[1] in determining surface forms (output) from underlying representations (input) in a single step. The OT architecture was first proposed by Prince & Smolensky (1993) and consists of two theoretical constructs: the Generator (*Gen*) and the Evaluator (*Eval*). Given an underlying representation as input, the *Gen* offers possible output candidates from a rich base and passes them to the *Eval*, which in turn scrutinizes the candidates under a table of constraints. The table of constraints determines the optimal candidate of the lot, which in a well–constructed table corresponds to the surface form. This is summarized schematically in (1).

(1) A schematic sketch of OT architecture, with the English words *cat, dog,* and *months* as examples.

1 An exception to OT's success is apparent in cases of opacity. For more information on how OT handles opaque surface forms, see Kager (1999:§9); however, this information is not necessary to understand the OT framing of the phonological processes described herein, which do not produce opaque forms.



There are two types of constraints in OT: markedness constraints and faithfulness constraints. Markedness constraints demand that the output be as least marked as possible, whereas faithfulness constraints insist that the output retain as many characteristics of the input as possible. As such, these types of constraints always have conflicting interests. Constraints are language–universal, but different languages apply more weight to some constraints over others. For example, voiced obstruents in a coda position are marked[2], which has led to the postulation of a markedness constraint that militates against voiced obstruents in codas, which can be written[3] as *VcdObsCoda (2a). German, among other languages, gives much weight to this constraint, since voiced obstruents can never occur in German codas. English, on the other hand, gives precedence to preserving the voicing contrast over neutralizing it for the sake of markedness. For English, a faithfulness constraint calling for segments in the surface form to have the same specification for voice as in the underlying representation is more important than *VcdObsCoda. This faithfulness constraint that preserves the voicing of the underlying representation is written ID–IO[voice] (2b). The respective importance of these two constraints in German and English demonstrates one of the fundamental concepts of OT: languages *rank* universal constraints in a language–specific order, and this hierarchy determines the surface forms by processing the underlying representations. Thus, German ranks *VcdObsCoda over ID–IO[voice] (2c) whereas English ranks them conversely (2d). The candidate provided by the *Gen* that incurs the smallest number of violations of higher–ranked constraints at the *Eval* is said to be optimal.

2 Traditionally, it is meaningful to describe phonological phenomena on a gradient (“more marked” or “less marked”), but OT gets away from relative terminology and handles this issue intrinsically.

3 An asterisk (*) is often used in constraints as a shorthand to signify that the following construction is illegal.

(2) Constraint interaction

a	*VcdObsCoda	No voiced obstruents in a coda position		
b	ID–IO[voice]	Segments specified as [αvoice] in the input are also specified as [αvoice] in the output; no change in voice.		
c	German ranking	*VcdObsCoda	>>	ID–IO[voice]
d	English ranking	ID–IO[voice]	>>	*VcdObsCoda

In a correctly ordered tableau, the optimal form is the same as the surface form in the language. The rankings of the constraints *VcdObsCoda and ID–IO[voice] for German and English are displayed within tableaux in (3) and (4). These tableaux are representative of the markup used for all tableaux in printed OT research. Note that an asterisk (*) indicates a violation, which when combined with an exclamation mark (*!) indicates the violation is *fatal*; that is, the suggested candidate is not optimal. For a more detailed account of OT, see Kager (1999).

(3) Example constraint ranking: German
[ʃpraːxbʊnt] is optimal for input /ʃpraːxbʊnd/.

Input: /ʃpraːxbʊnt/	*VoicedObsCoda	ID–IO [voice]
a. ʃpraːxbʊnt	*!	
b. ☞ ʃpraːxbʊnt		*b

(4) Example constraint ranking: English
[gʊd] is optimal for input /gʊd/.

Input: /gʊd/	*VoicedObsCoda	ID–IO [voice]
a. ☞ gʊd		*
b. gʊt	*!	

2.2 Head Consonant Rule

The Head Consonant Rule is a process that affects word–initial liquid consonants, and word–initial nasal consonants before a high front vowel or glide. Although this process is no longer active in Korean, it is important to analyze because it was productive during a period of heavy borrowing from Chinese. In a rule–based approach, a word–initial liquid become a nasal (5a), and then a word–initial nasal is deleted when followed by a high front vowel or glide (5b). In non–word–initial contexts, the Head Consonant Rule does not apply, and therefore, the underlying liquids and nasals surface when they occur word–medially in compounds. Since word–initial liquids are very rare in native Korean words, the Head Consonant Rule is primarily associated with Chinese borrowings (specifically those brought into Korean during the Head Consonant Rule's period of productivity). Examples of where the Head Consonant Rule applies (and does not apply) are shown in (6). An underlying liquid becomes realized as a nasal in (6a); an underlying nasal is deleted in front of a [+high,

+front] segment in (6c); and an underlying liquid is ultimately deleted after both steps of the rule apply (/l/ → n → ∅) in (6e). Examples (6b, d, f) show the same lexical items word–medially, and therefore, the Head Consonant Rule does not apply and the underlying liquids and nasals are preserved on the surface.

(5) Head Consonant Rule
a. 1. [liquid] → [nasal] / #___
b. 2. [nasal] → ∅ / ___[+high, +front]

(6) Examples of the Head Consonant Rule[4]

a	<락> "fall"	+	<심> "mind"	=	<낙심> "discouraged"
	/lak/	+	/ʃim/	=	[nakʃim]
			but		
b	<부> "division"	+	<락> "fall"	=	<부락> "village"
	/bu/	+	/lak/	=	[bulak]
c	<녀> "woman"	+	<성> "gender"	=	<여성> "womankind"
	/njɔ/	+	/sʌŋ/	=	[jɔsʌŋ]
			but		
d	<남> "man"	+	<녀> "woman"	=	<남녀> "men and women"
	/nam/	+	/njɔ/	=	[namnjɔ]
e	<룡> "dragon"	+	<산> "mountain"	=	<용산> "Dragon Hill"
	/ljoŋ/	+	/san/	=	[joŋsan]
			but		
f	<계> "chicken"	+	<룡> "dragon"	=	<계룡> "cockatrice"
	/gje/	+	/ljoŋ/	=	[gjeljoŋ]

Witty & Pindziak (2010) show that the Head Consonant Rule is driven by two markedness constraints: IdealOnset, which is a gradient constraint that favors syllable onsets with low sonority values, and CoronalPalatalization, which militates against certain dissimilar consonant clusters. More recent borrowings into Korean do not exhibit effects of the Head Consonant Rule (7). Witty & Pindziak (2010) account for the obsolescence of the Head Consonant Rule by demonstrating that diachronically, faithfulness constraints have been

4 These examples, and many more, are documented in Grant (1979), Witty & Pindziak (2010), and 이기문 (1999).

re–ranked above IdealOnset and CoronalPalatalization, and that the Head Consonant Rule–affected forms of older borrowings have become fossilized.

(7) Examples of more recent borrowings; Head Consonant Rule does not apply.

a.	<라면> "ramen (noodles)"	/lamjʌn/ → [lamjʌn], *[namjʌn]
b.	<라디오> "radio"	/ladio/ → [ladio], *[nadio]
c.	<뉴스> "news"	/njusʉ/ → [njusʉ], *[ɲusʉ]

2.3 Syllable Contact

When syllables are concatenated, Korean exhibits an assimilation process in adjacent consonants. The assimilation process is triggered when two consonants are next to each other and at least one of them is a sonorant. Examples of Korean assimilation are shown in (8).

a.	<심> "mind"	+	<리> "reason"	+	<학> "study"	=	<심리학> "psychology"
	/ʃim/	+	/li/	+	/hak/	=	[ʃimnihak]
b.	<신> "new"	+	<라> "arrange"	+	<방> "room"	=	<신라방> "shared room"
	/ʃin/	+	/la/	+	/baŋ/	=	[ʃilːabaŋ]
c.	<입> "be"	+	<니> "FORMAL"	+	<다> "DECL"	=	<입니다> "be (formal)"
	/ib/	+	/ni/	+	/da/	=	[imnida]
d.	<대학> "university"			+	<로> "road"	=	<대학로> "University Street"
	/dæhak/			+	/lo/	=	[dæhaŋno]

The assimilation process is quite complicated, especially in (8d). It is difficult to capture using a rule–based approach; however, it has been modeled with OT. Davis & Shin (1999:285–6), citing Vennemann (1988:40), propose a Syllable Contact constraint that requires that for adjacent consonants separated by a syllable boundary, the sonority value of the first consonant must greater than or equal to the sonority value of the second consonant. This constraint is chiefly responsible for the assimilation process in (8), and explains why nasal–liquid and obstruent–sonorant clusters are changed to clusters of consonants with level (i.e. not rising) sonority. The preservation of place of articulation, represented in OT by the faithfulness constraint ID–IO[PLACE], is also important to this process, and the ranking of the Syllable Contact constraint over ID–IO[PLACE] accounts for two nasal consonants that correspond in [place] with the underlying form in (8d). By ranking the Syllable Contact constraint over a collection of other faithfulness constraints from the Identity and MAXIMALITY constraint families, Davis & Shin (1999) create a model that predicts the correct surface forms for this assimilation process given the underlying forms.

Um (2002) provides another analysis of Korean assimilation. Um gives special attention to the specific prohibition of /ln/ and /nl/ clusters in Korean, which must be assimilated to either [lː] or [nː]. Um also cites Vennemann (1988) to propose a Syllable Contact constraint, which appears to be necessary for any OT analysis to accurately capture this process. Neither Davis & Shin (1999) nor Um (2002), however, address obstruent–obstruent clusters, which is the subject of a different phonological process: Complex–Coda Resolution.

2.4 Complex–Coda Resolution[5]

Since Korean prohibits complex syllable margins to surface from underlying representations with consonant clusters, the language has two strategies for assuring that consonant clusters are separated across syllables: "relinking" and deletion. Relinking (as used by Choo & O'Grady 2003:58–9) is the process of realizing a two–consonant cluster with the first consonant as the coda of the previous syllable and with the second consonant as the onset of the following syllable[6] (9). All consonants can undergo this relinking process with the exception of /ŋ/ (Chung 2001).

(9) Relinking

a	<값> "price"	+	<을> accusative marker	=	<값을> "price.ACC"
	/gabs/	+	/ɨl/	=	[gap.sɨl]
b	<몫> "share"	+	<을> accusative marker	=	<몫을> "share.ACC"
	/mogs/	+	/ɨl/	=	[mog.sɨl]
c	<꽃> "flower"	+	<이> accusative marker	=	<꽃이> "flower.NOM"
	/kotʃʰ/	+	/i/	=	[ko.tʃʰi]

Relinking is employable with consonant clusters only when a connectable syllable (i.e. a syllable with an empty onset) follows. In all other situations, it is necessary to reduce the cluster by deletion (10). In all situations, Korean prefers relinking and tolerates deletion only where necessary to avoid complex syllable margins. The consonant to be deleted in the cluster is the one with the highest sonority value.

5 Similar examples to the ones presented in this section are documented in Choo & O'Grady (2003) and 이기문 (1999). In addition, an explanation of Korean complex–coda resolution with feature geometry can be found in Kim (1995).

6 This definition for *relinking* is more elegant than the one used by Choo & O'Grady, who coined the term and defined it as follows: "A consonant that occurs at the end of one syllable is pushed into the next syllable when the second syllable starts with a vowel sound" (2003:58). Their definition implies that syllable boundaries are underlying (or that syllabification occurs more than once between the underlying and surface forms), which is inconsistent with mainstream phonology. For lack of a better term, we have used *relinking* as the label for this process but with the improved definition above

(10) Cluster scenarios and how Korean handles them

a. /VCCV/ → [VC.CV]

For two–consonant clusters followed by a vowel, the first consonant is realized as the coda of the previous syllable, and the second consonant forms the onset of the following syllable.

b. /VCCCV/ → [VC.CV]

For three–consonant clusters followed by a vowel, one of the consonants must be deleted, and the other two are distributed across syllable boundaries as in (a).

c. /VCC/ → [VC]

For consonant clusters that are not followed by a vowel, one of the consonants must be deleted.

Another effect that relinking has is to preserve certain features of the underlying representation that would be neutralized in the coda. Korean, like German, does not permit voiced obstruents in coda positions in surface forms. Thus, words like /bab/ "rice" in (11) are realized with a voiceless stop (11a) in the coda unless that consonant is able to relink to an empty onset in the next syllable (11b).

a.	In isolation:	<밥>	/bab/	→	[bap]	"rice"
b.	With nominalizer |i|:	<밥이>	/bab+i/	→	[babi]	"rice.NOM"
c.	In isolation:	<각>	/gag/	→	[gak]	"angle"
d.	With nominalizer |i|:	<각이>	/gag+i/	→	[gagi]	"angle.NOM"
e.	In isolation:	<닫>	/dad/	→	[dat]	"close"
f.	With nominalizer |ʉb|:	<닫음>	/dad+ʉb/	→	[dadʉp]	"closing"

The Korean coda is a very limited position; in fact, the only obstruent able to occupy it is a voiceless stop. Non–sonorant continuants (sibilants, for example) must either be relinked to the next syllable or changed to a stop consonant with the same place of articulation (12).

a.	In isolation:	<것>	/gʌs/	→	[gʌt]	"thing"
b.	With accusative |ʉl|:	<것을>	/gʌs+ʉl/	→	[gʌsʉl]	"thing.ACC"
c.	In isolation:	<맛>	/mas/	→	[mat]	"flavor"
d.	With accusative |ʉl|:	<맛을>	/mas+ʉl/	→	[masʉl]	"flavor.ACC"
e.	In isolation:	<빗>	/bis/	→	[bit]	"comb"
f.	With accusative |ʉl|:	<빗을>	/bis+ʉl/	→	[bisʉl]	"comb.ACC"

As generalized earlier in this section, a deletion process takes place in consonant clusters when relinking is not an option. The relinking–versus–deletion determination is visible with two different conjugations of the verb /ilg/ "read":

/ilg+ʌt/ “read+PAST (BASE)” and /ilg+da/ “read+DECLARATIVE (PRESENT)” (13ab). Given the rules in (11) and (12), it makes sense that the consonant that remains is the stop (and made voiceless per (11)), while the other consonant of the cluster is deleted.

(13) When relinking is not possible, deletion of the non–stop consonant occurs.

a.	Relinkable cluster	<읽었>	/ilg+ʌt/ read+PAST	→	[il.gʌt]
b.	Non–relinkable cluster	<읽다>	/ilg+da/ read+DECL	→	[ikda]
c.	Relinkable cluster	<밟었>	/balb+ʌt/ step+ PAST	→	[bal.bʌt]
d.	Non–relinkable cluster	<밟다>	/balb+da/ step+DECL	→	[bapda]

To review and clarity, Complex–Coda Resolution differs from the assimilation process of Syllable Contact (8) in that the Syllable Contact assimilation process is triggered only when at least one of the two consonants at the morpheme edges is a sonorant. When two obstruents come in contact, however, no assimilation takes place, and depending on the complexity of the cluster, the Complex–Coda Resolution process may need to delete a consonant (as in 10b). In addition to the contact of obstruents, Complex–Coda Resolution applies in order to distribute consonant cluster across syllable when possible (10a) or reduces the cluster otherwise (10c).

3. Optimality–Theoretic Analysis of Korean

3.1 Deletion

As seen in §2.4, Korean avoids complex codas by a relinking process and a deletion process. Let us first establish the set of constraints responsible for the Korean deletion process. One of the primary reasons that deletion takes place is because Korean does not permit complex syllable margins. Therefore, the markedness constraint *COMPLEX (14a), which militates against complex margins, is involved in this hierarchy. Since the output of the Complex–Coda Resolution process never contains a complex margin, the constraint *COMPLEX is undominated. That is, since optimal output should never violate this constraint, it is ranked as one of the most important, must–satisfy constraints (14b).

(14) Motivating deletion

a. *COMPLEX — Syllables do not have complex margins (Kager 1999:288).

b. *COMPLEX is undominated.

When a word with a potential complex coda, like <값> /gabs/, occurs in isolation, the cluster is reduced to a single consonant by deletion ([gap]). In this process, part of the input is lost for the sake of markedness, which in OT is represented with the dominance of *COMPLEX over some violated faithfulness constraint. The constraint in this case is MAX–IO, which disallows deletion (15a). Since Korean prefers deletion to a surfacing complex coda, *COMPLEX ranks over MAX–IO (15b).

(15) Deletion is acceptable in Korean.

a MAX–IO — Input segments have output correspondents; no deletion (Kager 1999:205).

b *COMPLEX >> MAX–IO

Another strategy that some languages use to avoid complex syllable margins is epenthesis, e.g., the insertion of the segment not present in the underlying representation to break up a cluster in the surface form; however, no epenthesis process occurs in Korean. Therefore, it is necessary to add a no–epenthesis constraint to the OT hierarchy: DEP–IO (16a). It ranks co–dominant with *COMPLEX as part of the undominated set of constraints, since no proper output violates it (16b). The tableau in (17) shows how this partial hierarchy marks as optimal the correct output [gap] for input <값> /gabs/.

(16) Korean does not allow epenthesis.

a DEP–IO — Output segments have input correspondents; no epenthesis (Kager 1999:205).

b *COMPLEX, DEP–IO >> MAX–IO

(17) <값> /gabs/ → [gap]

Input: /gabs/	*COMPLEX	DEP–IO	MAX–IO
a. gabs			
b. ☞ gap			*
c. gabsɨ		*!	

This tableau, however, contains a limited set of output candidates to serve as only an example of the evaluation of the constraints presented thus far, and is incomplete. It takes for granted the rule that obstruents are voiceless in a coda position (11), and would incorrectly mark the candidate [gab] as optimal, as well. Hence, it is necessary to add constraints that account for the featural limitations on consonants that surface in coda positions.

3.2 Coda conditions

Restrictions on the consonants in coda positions are common cross–linguistically (see, for example, Blevins 1995:§6.1, Flack 2009, Itô 1986), and this evidence gives ample motivation for markedness constraints that put prohibitions on coda consonants. As seen in <밥> /bab/ → [bap] (11a), Korean is like German in reducing voiced obstruents to voiceless in the coda. The markedness constraint behind this change, *VcdObsCoda, was used as an example in §2.1 and is reproduced in (18a). The discussion in §2.1 also mentions that *VcdObsCoda clashes theoretically with ID–IO[voice]. Since there is no justification to single out the feature [voice] in this more complicated hierarchy, the faithfulness constraint is generalized to ID–IO, which calls for the preservation of all features of the input in the output (18b). Since voiced obstruents never occur in Korean codas, *VcdObsCoda is ranked co–dominant with the other two undominated constraints of the hierarchy so far, and ID–IO must rank below it (18c).

(18) Accounting for coda conditions: no voiced obstruents

a. *VcdObsCoda — No voiced obstruents in a coda position

b. ID–IO — For all segments in the input that have a corresponding segment in the output, if S_{input} is specified as [αF_1, βF_2, γF_3, ...], then S_{output} is also specified as [αF_1, βF_2, γF_3, ...]; no change in features.

c. *Complex, Dep–IO, *VcdObsCoda >> (ID–IO, Max–IO)

It is still necessary to establish a ranking between ID–IO and Max–IO, if possible. A ranking of Max–IO over ID–IO would mean that Korean prefers changing the feature of a segment over deleting the segment altogether, whereas a ranking of ID–IO over Max–IO would indicate the converse. As manifested by <밥> /bab/ → [bap], *[ba], it is evident the Max–IO dominates ID–IO (19a), yielding the hierarchy so far in (19b). The success of this hierarchy for handling coda devoicing is demonstrated by the correct evaluation of input <밥> /bab/ (20).

(19) Ranking the dominated faithfulness constraints

a. Max–IO >> ID–IO

b. *Complex, Dep–IO, *VcdObsCoda >> Max–IO >> ID–IO

(20) <밥> /bab/ → [bap]

Input /bab/	*Complex	Dep–IO	*VcdObsCoda	Max–IO	ID–IO
a. bab			*!		
b. ba				*!	
c. babʉ		*!			
d. ☞ bap					*
e. pap					**!

Still, this hierarchy is incomplete and must account for the reduction of obstruent continuants in a coda position (12), like in <것> /gʌs/ → [gʌt]. This can be accomplished with a markedness constraint that prohibits obstruent continuants in codas: *ObsConCoda (21a). This constraint is an adaptation of a constraint proposed by Kim (2000). Since [+obstruent, +continuant] segments can never surface in a coda, *ObsConCoda is undominated (21b). The addition of the constraint *ObsConCoda may seem to be sufficient to correctly evaluate <것> /gʌs/ as input, but consider what happens in (22).

(21) Accounting for coda conditions: no obstruent continuants
  a. *ObsConCoda No obstruent continuants in a coda position
  b. *Complex, Dep–IO, *ObsConCoda, *VcdObsCoda >> Max–IO >> ID–IO

(22) <것> /gʌs/ → [gʌt], naïve

Input /gʌs/	*Complex	Dep–IO	*ObsCon Coda	*VcdObs Coda	Max–IO	ID–IO
a. gʌs			*!			
b. gʌ					*!	
c. ☞ gʌt						*
d. ☞ gʌn						*
e. gʌsɨ		*!				

The present constraints require that the obstruent continuant (in this case, /s/) not surface in the output, but they place no requirement on what segment can be a substitute for it, except that it not be deleted. As this hierarchy stands now, [gʌt], [gʌn], and any other [+alveolar, ±obstruent, ∓continuant] would be marked as co–optimal, which is a failure in evaluation. Thus, it is necessary to include a faithfulness constraint that retains the feature [obstruent] in the output: ID–IO[obs] (23a). This constraint is undominated because no segments are changed for the feature [obstruent] in the Korean coda–reduction process[7] (23b). Since the more general identity constraint ID–IO must be dominated by Max–IO, the more specific ID–IO[obs] is indispensable. With this new constraint, the hierarchy properly evaluates <것> /gʌs/ (24) and <값> /gabs/ (25) without taking any rules for granted.

(23) Adjusting the hierarchy to preserve [obstruent]
  a. ID–IO[obs] Segments specified as [αobstruent] in the input are also specified as [αobstruent] in the output; no change in the feature [obstruent].

7 In the Syllable Contact assimilation process outlined in §2.3, underlying obstruents often surface as sonorants, so in a wider analysis of the language, ID–IO[obs] is not undominated; however, a lower ranking of ID–IO[obs] is not incompatible with the present analysis. ID–IO[obs] is ranked as undominated herein because there is no justification for a lower ranking by the process under discussion (i.e. coda resolution).

b. *COMPLEX, DEP–IO, ID–IO[OBS], *OBSCONCODA, *VCDOBSCODA >> MAX–IO >> ID–IO

(24) <것> /gʌs/ → [gʌt]

Input /gʌs/	*COMPLEX	DEP–IO	ID–IO [OBS]	*OBSCON CODA	*VCDOBS CODA	MAX–IO	ID–IO
a. gʌs				*!			
b. gʌ						*!	
c. ☞ gʌt							*
d. gʌn			*!				*
e. gʌsʉ		*!					

(25) <값> /gabs/ → [gap]

Input /gabs/	*COMPLEX	DEP–IO	ID–IO [OBS]	*OBSCON CODA	*VCDOBS CODA	MAX–IO	ID–IO
a. gabs	*!			*!	*!		
b. gaps	*!			*!			*
c. gab					*!	*	
d. gas				*!		*	
e. ☞ gap						*	*
f. ga						**!	

There is one more input type that must be considered before moving on to the Korean relinking process. This input type is consonant clusters formed by a liquid and obstruent stop like <읽다> /ilg–da/ → [ikda] (13b), where the liquid undergoes deletion. The preservation of the stop consonant suggests that the qualified faithfulness constraint MAX–IO[STOP] (also used by Lombardi 2001:233, for example) is active in this process (26a). MAX–IO[STOP] must be ranked above MAX–IO to ensure that the non–stop consonant in the cluster is deleted (26b). Since Korean never deletes a stop in the complex–coda–resolution process, the constraint MAX–IO is undominated in the hierarchy[8] (26c). This set of constraints is now successful in the evaluation of <읽다> /ilg–da/ (27) and is complete for the Korean deletion process and coda conditions. This hierarchy facilitates the analysis of the relinking process.

8 If another Korean phonological process causes the deletion of stops, it is still compatible with this hierarchy, which only requires that MAX–IO[STOP] dominate MAX–IO.

(26) Korean deletes non–stops in non–intervocalic clusters.

a. Max–IO[stop] For each stop in the input, there is a corresponding segment in the output; no deletion of stops.[9]

b. Max–IO[stop] >> Max–IO

c. *Complex, Dep–IO, ID–IO[obs], Max–IO[stop], *ObsConCoda, *VcdObsCoda >> Max–IO >> ID–IO

(27) <읽다> /ilg–da/ → [ikda]

Input /ilg–da/	*Cx	Dep	ID[obs]	Max [stop]	*ObConC	*VcdObC	Max	ID
a. ilgda	*!					*!		
b. ilgʉda		*!						
c. ilda				*!			*	
d. igda						*!	*	
e. ☞ ikda							*	*
f. ida							**	

3.3 Relinking

The constraints in the hierarchy so far accurately handle any input to prevent a complex coda, but the relinking process applies to simple codas, as well, as in <밥이> /bab+i/ → [ba.bi] "rice (nominative)" (11b). This evidence suggests the presence of the No–Coda markedness constraint (28a), but since simple codas legally occur in Korean, No–Coda is not undominated. Simple codas are never resolved by deletion, so Max–IO dominates No–Coda. For the moment, let us leave No–Coda co–dominant with ID–IO at the bottom of the hierarchy (28b); this will be readdressed in (31). The addition of the No–Coda constraint, albeit to a low position in the hierarchy, guarantees that single intervocalic consonants surface in the onset of the following syllable rather than in the coda of the previous syllable, as seen with <바비> /bab+i/ in (29) and <것을> /gʌs+ʉl/ in (30).

(28) Account for relinking

a. No–Coda Syllable do not have codas (Kager 1999).

b. *Complex, Dep–IO, ID–IO[obs], Max–IO[stop], *ObsConCoda, *VcdObsCoda >> Max–IO >> (ID–IO, No–Coda)

9 Specifically, Max–IO[stop] requires only that a stop consonant have a correspondent in the output, but it does not have to be a stop necessarily. That is, a change in feature (even [manner]) does not violate any Maximality constraint. See Farris–Trimble (2008:62–82) for more examples of this Maxmality–Identity interaction.

(29) <밥이> /bab+i/ → [ba.bi]

Input /bab+i/	*Cx	Dep	ID[obs]	Max [stop]	*ObConC	*VcdObC	Max	ID	NC
a. bab.i						*!			*
b. ☞ ba.bi									
c. bap.i								*!	*!
d. ba.pi								*!	

(30) <것을> /gʌs+ʉl/ → [gʌ.sʉl]

Input /gʌs+ʉl/	*Cx	Dep	ID[obs]	Max [stop]	*ObConC	*VcdObC	Max	ID	NC
a. gʌs.ʉl					*!				**
b. ☞ gʌ.sʉl									*
c. gʌt.ʉl								*	**!
d. gʌ.ʉl							*!		*

Chung (2001) postulates that an exception must be made for the velar nasal /ŋ/, stating that it cannot relink and must always surface in a coda position. Chung proposes the constraint OnsetCond to represent this prohibition in OT (31a), which must be then undominated. The revelation that /ŋ/ is realized in the coda position permits a ranking between ID–IO and No–Coda. Since [ŋ] remains in a coda and does not undergo a featural change (e.g. to [n]), ID–IO, which preserves the features of the input, dominates No–Coda. The addition of this constraint and the ranking of ID–IO over No–Coda yields the hierarchy in (31b), which successfully evaluates input with an intervocalic /ŋ/, *like* <방이> /baŋ+i/ "room (nominative)" (32).

(31) Korean requires that /ŋ/ be realized in a coda.
a. OnsetCond "The velar nasal /ŋ/ is not allowed as an onset element in Korean" (Chung 2001:182).
b. *Complex, Dep–IO, ID–IO[obs], Max–IO[stop], *ObsConCoda, OnsetCond, *VcdObsCoda >> Max–IO >> ID–IO >> No–Coda

(32) <방이> /baŋ+i/ → [baŋ.i]

Input: /baŋ+i/	*Cx	Dep	ID[obs]	Max [stop]	*ObConC	$*_{\sigma}$[ŋ	*VcdObC	Max	ID	NC
a. ☞ baŋ.i										*
b. ba.ŋi						*!				
c. ba.ni									*!	
d. ba.i				*!				*		

A reexamination of the evaluations of <밥이> /bab+i/ (29) and <것을> / gʌs+ɨl/ (30) shows that the ranking of ID–IO over NO–CODA does not produce any unexpected results with the relinking process. Hence, the hierarchy of constraints is complete, as reproduced in (33). This hierarchy successfully produces output in a single step for the Korean Complex–Coda Resolution process, which takes multiple steps in a traditional rule–based analysis. This combined hierarchy is showcased in the evaluation of <값을> /gabs+ɨl/ "price (accusative)" (34), an example of input that undergoes both a coda condition (i.e. /b/ → [p]) and relinking.

(33) Complete hierarchy for Korean complex–coda resolution
*COMPLEX, DEP–IO, ID–IO[OBS], MAX–IO[STOP], *OBSCONCODA, ONSETCOND, *VCDOBSCODA >> MAX–IO >> ID–IO >> NO–CODA

(34) <값을> /gabs+ɨl/ → [gap.sɨl]

Input: /gabs+ɨl/	*CX	DEP	ID[OBS]	MAX [STOP]	*OBCONC	*$_{\sigma}$[ŋ	*VCDOBC	MAX	ID	NC
a. gabs.ɨl	*!				*!		*!			**
b. gab.sɨl							*!			**
c. ga.bɨl								*!		*
d. ga.sɨl				*!				*		*
e. gaps.ɨl	*!								*	**
f. ☞ gap.sɨl									*	**
g. ga.bɨ.sɨl		*!								*

4. Conclusion

The Korean Complex–Coda Resolution process can be managed succinctly and completely within the OT framework. The constraint hierarchy presented herein, as it appears in (33), encapsulates the relinking and deletion processes of Complex–Coda Resolution, and accounts for the conditions that Korean imposes on consonants that surface in simple codas. By outlining the ranking of constraints for Korean Complex–Coda Resolution, this analysis complements the previous research of the Syllable Contact assimilation process and the Head Consonant Rule. Future research should focus on the compatibility of these OT modules with each other and other Korean phonological processes, as the theory predicts that the ranking of constraints is consistent for all processes in a language.

References

Blevins, Juliette. 1995. The syllable in phonological theory. In the handbook of phonological theory, ed. John Goldsmith. Cambridge: Blackwell.

Choo, Miho, and William O'Grady. 2003. The sounds of Korean – A pronunciation guide. Honolulu: University of Hawai'i Press.

Chung, Chin Wan. 2001. The behavior of velar nasal and syllabification. Studies in Phonetics, Phonology and Morphology 7.1, 177–89.

Davis, Stuart, and Seung–Hoon Shin. 1999. The syllable contact constraint in Korean: an Optimality–Theoretic analysis. Journal of East Asian Linguistics 8, 285–312.

Farris–Trimble, Ashley. 2008. Cumulative faithfulness effects in phonology. Indiana University: Ph.D. dissertation.

Flack, Kathryn. 2009. Constraints on onsets and codas of words and phrases. Phonology 26.2, 269–302.

Grant, Bruce K. 1979. A guide to Korean characters: reading and writing Hangŭl and Hanja. Elizabeth, N.J.: Hollym International Corp.

Itô, Junko. 1986. Syllable Theory in prosodic phonology. University of Massachusetts Amherst: Ph.D. dissertation.

Kager, René. 1999. Optimality Theory. Cambridge: Cambridge University Press.

Kim, Gyung–Ran. 2000. Revisit to Korean irregular verbs. Studies in Phonetics, Phonology and Morphology 6, 21–39.

Kim, Sun–Hee. 1995. Stray erasure and the coda condition in Korean. Cahiers de linguistique – Asie orientale 24.1, 139–54. Available at <http://www.persee.fr/web/revues/home/prescript/article/clao_0153–3320_1995_num_24_1_1469>.

Lombardi, Linda. 2001. Segmental phonology in Optimality Theory: constraints and representations. New York: Cambridge University Press.

McCarthy, John. 2007. Slouching toward optimality: Coda reduction in OT–CC. University of Massachusetts Amherst: Linguistics Department Faculty Publication Series, paper 74.

Prince, Alan, and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. New Brunswick, NJ: Rutgers University.

Um, Yongnam. 2002. An ordering paradox as constraint interaction: alternation of n and l in Korean. Studies in Phonetics, Phonology and Morphology 9, 111–33.

Vennemann, Theo. 1988. Preference Laws for Syllable Structure. Berlin: Mouton de Gruyter.

Witty, Sean, and Charles Pindziak. 2010. Stratification and Optimality: The Korean Head Consonant Rule. Chomskyan Studies 5.1: 63–77.

Složena koda u korejskome u okviru teorije optimalnosti

Fonotaktika korejskog jezika ne dozvoljava složenu kodu, dakle korejski jezik rabi dvije strategije kako bi razbio suglasničke skupine i prilagodio se tom stanju. Prva je strategija ponovno povezivanje – "relinking" (termin koji rabe Choo&O'Grady 2003:58–9), tj. realizacija suglasničke skupine preko granice sloga, a druga je strategija brisanje. Teorija optimalnosti pogodna je da se njome objasni način na koji se rabe ove dvije strategije, što je ujedno i tema ovoga rada. Koristeći se već ustanovljenim ograničenjima, ovaj rad nadopunjuje istraživanja unutar teorije optimalnosti o sličnim fonološkim procesima u korejskome tako da se navedeni procesi mogu ujediniti pod jednim rangiranjem ograničenja.

Key words: Korean, coda reduction, Optimality Theory, phonology, phonotactics, assimilation of sound

Ključne riječi: korejski jezik, redukcija kode, teorija optimalnosti, fonologija, fonotaktika, asimilacija glasova