Chapter 3 Actual Analyses: Implementation of the Tools

Building on the discussion of the constraint interaction in chapter 2, I will analyze actual languages in order to confirm whether the predicted constraint rankings are valid. In section 3.1, a morpheme structure constraint (MSC) in Ponapean will be investigated as an example of Type 2. Dakota coronal dissimilation will be analyzed as a representative of Type 3 in section 3.2, and section 3.3 will examine Basque deletion and spirantization in consonant clusters as a Type 4 language.

3.1 Type 2: Morpheme Structure Constraint (MSC) in Ponapean

As I pointed out in section 2.2.1.2, the grammars of many languages exhibit feature fusion so as not to violate the OCP. I classify such a language as Type 2. The ranking for Type 2 proposed in section 2.3.3.2 is:

(1) Constraint Ranking for Type 2:

(a)

```
DEP[F],    OCP[F],      MAX-IO
  |__________|__________|
             |
      UNIFORMITY[F]
```

(b)

```
MAX[F],    OCP[F]
   |__________|
        |
 UNIFORMITY[F]
```

In a Type 2 language, either {DEP[F], OCP[F], MAX-IO} or {MAX[F], OCP[F]} must outrank UNIFORMITY[F]. For example, the rankings in (1) entails a language in which MAX-[F], DEP[F], OCP[F], and MAX-IO are higher ranked than UNIFORMITY[F] as follows:

(2) a possible ranking for Type 2:

```
MAX[F], DEP[F],    OCP[F],      MAX-IO
  |________|__________|____________|
                 |
         UNIFORMITY[F]
```

As discussed in section 2.3.3.2, I claim that feature fusion violates neither MAX[F] nor faithfulness constraints for association lines. On the other hand, it does violate UNIFORMITY[F] which is lower ranked than the other proposed constraints. I stated that featural fusion is exactly parallel to segmental coalescence in this sense.

In this section, I will analyze the case of Ponapean to confirm that the ranking for Type 2 is found in an actual grammar.

3.1.1 Ponapean Labials

In Ponapean, a word cannot contain two [labial] features which do not share the same value for backness, while it can contain those which share the same backness as the following data show.

(3) Ponapean Labials (Mester 1986):

p + p	paip	'boulder'
	pap	'swim'
m + m	mem	'sweet'
	kamam	'to enjoy kava'
p + m	parem	'nipa palm'
	madep	'species of sea cucumber'
p^w + p^w	p^wup^w 'to fall'	
	p^wop^we	'shoulder'
m^w + m^w	sum^wum^w	'trouchus'
	kam^wam^w	'to exhaust'
	m^waam^w	'fish'
m^w + p^w	m^wop^w	'out of breath'
*p^w ap	*p^w ap	DOES NOT EXIST

[C^w] in the data indicates velarization which is represented by an addition of [back]. Mester (1986) accounts for these Ponapean data by evoking the OCP. When the two [labial] features share the same value for backness, they can be fused into one [labial]. Hence, an OCP violation on labial does not take place as in (4a) or (4b). On the other hand, when the two [labial] features do not share the same value for the [back] feature, they cannot be fused, resulting in an OCP violation on [labial] as in (4c). Consequently, two labial features with different values for backness will never surface in the language.

(4) Possible and Impossible sequences in Ponapean:

```
(a).              (b).             *(c).

  p^wup^w          p  a p            p^w  a    p
   \   /            \   /             \        /
   [lab]            [lab]            [lab]   [lab]
     |                                 |
   [back]                            [back]
  (possible)       (possible)        (impossible)
```

We can conclude that the whole word is the domain of the effect of the OCP on [lab] in this language.

3.1.2 An Analysis within the OT Framework

3.1.2.1 OCP[lab] >> UNIFORMITY[lab]

In Ponapean, from the surface forms of [p^wap^w] or [pap], two kinds of structures for each are possible.

(5) two possible structures for [p^wup^w] :

```
(a)  p^wap^w                    (b)  p^wap^w
      |   |                            \  /
   [lab] [lab]         OR             [lab]
     |     |                            |
  [back][back]                        [back]
```

(6) two possible structures for [pap]:

```
(a)  p a  p                     (b)  p a  p
     |    |                           \  /
   [lab] [lab]         OR             [lab]
```

In both sequences, I assume that (5b) and (6b) are the actual structures, because of the impossibility of *[p^wap].

(7) only one possible structure for *[p^wap]:

```
(a)  *p^wap
       |   |
    [lab][lab]
       |
    [back]
```

For the sequence of *[p^wap], a fused structure is not possible because the two labials do not share the backness feature. Thus, I claim that the structure illustrated in (5a), (6a), and (7a) is ruled out by the Ponapean constraint ranking.

First, let us observe how the constraint ranking can account for the difference between (a) and (b) in (5) or (6) whose surface forms are exactly the same. I propose that OCP[lab] outranks UNIFORMITY[lab].

(8)

OCP[lab]: *[lab][lab]

UNIFORMITY[lab]: No output labial feature may have multiple correspondents in the input.

(9)

/ p^wa p^w/ | | $[lab]_1$ $[lab]_2$ | | [back] [back]	OCP[lab]	UNIFORMITY[lab]
a. p^wa p^w | | $[lab]_1$ $[lab]_2$ | | [back][back] Faithful to input	*!	
☞b. p^wa p^w \ / $[lab]_{1,2}$ | [back] Feature Fusion		*

The ranking in (9) explains why the structure in (a) is ruled out, while the fused one in (b) is optimal. I will discuss the validity of this ranking in more detail in section 3.1.2.5 by examining the impossible sequence of [p^{w}ap].

3.1.2.2 MAX[lab] >> UNIFORMITY[lab]

The next ranking which I would like to discuss is between the two constraints MAX[F] and UNIFORMITY[F].

(10)

MAX[lab]: An input labial feature must have an output correspondent

I claim that featural fusion does not violate MAX[F] since the input labial feature is realized in the output.

(11)

/ p^w a p^w/ | | [lab]$_1$ [lab]$_2$ | | [back] [back]	OCP[lab]	MAX[lab]	UNIFORMITY[lab]
☞a. p^wa p^w \\ / [lab]$_{1.2}$ | [back] Feature Fusion			*
b. p^wa p^w | | [lab]$_1$ [lab]$_2$ | | [back][back] Faithful to input	*!		
c. p^wa p^w | [lab]$_1$ | [back] one of the [lab] feature deletes		*!	

In both candidates (a) and (c), only one [labial] surfaces. Nevertheless, (c) loses due to its violation of MAX[lab]: one of the input labial features does not have an output correspondent in (c). On the other hand, both of the labial features of the input do have the correspondents in the output in (a). Although (a) violates UNIFORMITY[lab], this violation is not fatal since it is lower ranked than OCP[lab] and MAX[lab].

Thus, I propose that OCP[lab] and MAX[lab] outrank UNIFORMITY[lab] in Ponapean.

(12)

```
OCP[lab]          MAX[lab]
   |________________|
            |
    UNIFORMITY[lab]
```

The ranking in (12) follows what I have discussed for Type 2.

3.1.2.3 MAX-IO

The next constraint I will discuss is MAX-IO. Since featural fusion is preferred to segmental deletion in this language, one might assume that MAX-IO also outranks UNIFORMITY[lab].

(13)

/ $p^w a$ p^w/ | | $[lab]_1$ $[lab]_2$ | | [back] [back]	OCP[lab]	MAX-[lab]	MAX-IO	UNIFORMITY[lab]
☞a. $p^w a$ p^w \\ / $[lab]_{1,2}$ | [back] Feature Fusion				*
b. / $p^w a$ / | $[lab]_1$ | [back] segmental deletion		*!	*!	

However, if MAX[lab] is higher ranked than UNIFORMITY[lab] as indicated in section 3.1.2.2, then the ranking of MAX-IO does not matter. This is because wherever MAX-IO may rank, the entire ranking correctly rules out candidate (b).

(14)

/ p^wa p^w/ | | $[lab]_1$ $[lab]_2$ | | [back] [back]	OCP[lab]	MAX-[lab]	MAX-IO	UNIFORMITY[lab]
☞a. p^wa p^w \\ / $[lab]_{1,2}$ | [back] Feature Fusion				*
b. p^w a | $[lab]_1$ | [back] segmental deletion		*!	*	

3.1.2.4 DEP[F]

In Ponapean, another possible repair strategy, changing the feature, also does not take place. Consequently, it seems that DEP[F] also outranks UNIFORMITY[lab]. However, the ranking of DEP[F] also does not matter because the entire ranking correctly rules out the candidate in which insertion of the feature takes place.

(15)

/ p^wa p^w/ | | [lab]$_1$ [lab]$_2$ | | [back] [back]	OCP[lab]	Max-[lab]	Dep[cor]	Uniformity[lab]
☞a. p^wa p^w \\ / [lab]$_{1,2}$ | [back] Feature Fusion				*
b. [cor] | **t**a p^w | [lab]$_2$ | [back] Feature change (deletion of [lab] and [back] & Insertion of [cor])		*!	*	

In this case, Dep[cor] is logically expected to rank higher than Uniformity[lab], however, candidate (b) in tableau (15) could be ruled out by Max[lab]. Therefore, there is no empirical evidence for establishing a ranking between Dep[cor] and the two constraints to its left.

(16) A constraint ranking in Ponapean

OCP[lab] Max[lab]
|___________|
|
Uniformity[lab], Max-IO, Dep[cor]

Thus, the crucial ranking in Ponapean is that OCP[lab] and MAX[lab] outrank UNIFORMITY[lab]. This is exactly what is expected for the ranking in a Type 2 language. Recall that there are two possible structures for the surface form [p^wap^w] or [pap] as follows.

(17) two possible structures for [p^wap^w] (repeated from (5)):

```
(a)   p^wap^w                        (b)   p^wap^w
       |    |                                \  /
    [lab] [lab]          OR                  [lab]
      |     |                                  |
  [back][back]                               [back]
```

(18) two possible structures for [pap](repeated from (6)):

```
(a)   p a  p                         (b)   p a  p
       |    |                                \  /
    [lab] [lab]          OR                  [lab]
```

The ranking given in (16) explains why the structures for [p^wap^w] or [pap] are not (17a) nor (18a), but (17b) and (18b). It also makes it clear why an OCP violation, featural deletion, segmental deletion, or featural insertion do not take place in this language.

3.1.2.5 Impossible Sequence : *[p^wap]

A sequence of two adjacent labials which do not share the same value for backness is not possible in Ponapean. For instance, *[p^wap] is never observed in the language. This section will discuss how the constraint ranking can account for this fact.

First, let us recall from section 3.1.2.1, that OCP[lab] outranks UNIFORMITY [lab].

(19)

/p^wa p/ | | $[lab]_1$ $[lab]_2$ | [back]	OCP[lab]	UNIFORMITY [lab]
☞a. p^w a p^w \ / $[lab]_{1,2}$ | [back] Feature Fusion		*
b. p^w a p | | $[lab]_1[lab]_2$ | [back] Faithful to Input	*!	

In candidate (a), two labial features are fused, and two segments share the same backness. Candidate (b) loses because OCP[lab] outranks UNIFORMITY[lab].

We should consider another possible candidate in which both feature deletion and feature fusion take place. Such a candidate should lose because it violates two kinds of faithfulness constraints: one is UNIFORMITY[F]; and the other is MAX[F].

(20)

/ p^{w}a p/ | | [lab]$_1$ [lab]$_2$ | [back]	OCP[lab]	UNIFORMITY [lab]	MAX[back]
☞a. p^w a p^w \\ / [lab]$_{1,2}$ | [back] Feature Fusion		*	
b. p^{w}a p | | [lab]$_1$[lab]$_2$ | [back] Faithful to Input	*!		
c.[1] p a p \\ / [lab]$_{1,2}$ Feature Deletion & Feature Fusion		*	*!

As tableau (20) shows, both candidate (b) and (c) lose. Candidate (b) loses due to the violation of the higher ranked constraint OCP[lab], and candidate (c) is penalized because it violates MAX[lab] in addition to UNIFORMITY[lab]. The ranking of MAX[lab] is not determined yet from this tableau, because candidate (c) loses regardless of the ranking of MAX[lab].

This section has established the ranking in (21).

---

1 I assume /p^w.p/ → [p^w.p^w], but there are actually no alternations in the data. Since the input is the ill-formed sequence, it never surfaces in this language. There is another possibility that /p^w.p/ is realized as [p . p] by deleting the [back] feature. Both [p^w.p^w] and [p . p] would be better than [p^w.p]. We cannot tell which would be the real alternation. MAX[back] would choose [p^w.p^w], while a markedness constraint *[back] would choose [p. p]. Therefore, it depends on ranking of those two constraints.

(21)

```
                OCP[lab]
                    |
          UNIFORMITY[lab],     MAX[back]
```

The ill-formedness of the sequence of two labial segments which do not share the same value for backness is derived from the ranking in (21).

3.1.2.6 Summary

The discussion from sections 3.1.2.1 to 3.1.2.5 has established the following ranking in Ponapean.

```
(22) OCP[lab]   MAX-IO
     |__________|
           |
     UNIFORMITY[lab],   MAX[lab], DEP[cor],  MAX[back]
```

All the possible candidates for each of the sequences, [p^{w}ap^w], [pap], or [p^{w}ap] with this ranking are presented in (23).

(23) $[p^w a p^w]$

<table>
<tr><th>/ $p^w a\ p^w$/
| |
$[lab]_1[lab]_2$
| |
[back] [back]</th><th>OCP [lab]</th><th>MAX-IO</th><th>MAX [lab]</th><th>DEP [cor]</th><th>UNIFORMITY [lab]</th><th>MAX [back]</th></tr>
<tr><td>☞a. $p^w a\ p^w$
\ /
$[lab]_{1,2}$
|
[back]
F-Fusion</td><td></td><td></td><td></td><td></td><td>*</td><td></td></tr>
<tr><td>b. $p^w a\ p^w$
| |
[lab] [lab]
| |
[back][back]
Faithful</td><td>*!</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>c. $p\ a\ \ p^w$
| |
[lab][lab]
|
[back]
Del [back]</td><td>*!</td><td></td><td></td><td></td><td></td><td>*</td></tr>
<tr><td>d. [cor]
|
$ta\ p^w$
|
[lab]
|
[back]
F- change</td><td></td><td></td><td>*</td><td>*!</td><td></td><td>*</td></tr>
<tr><td>e. $a\ p^w$
|
[lab]
|
[back]
F&Seg-Del</td><td></td><td>*!</td><td>*</td><td></td><td></td><td></td></tr>
<tr><td>f. p a p
\ /
$[lab]_{1,2}$
F-del & fus.</td><td></td><td></td><td></td><td></td><td>*</td><td>*!</td></tr>
</table>

When an input contains two labials with the same values for backness, the candidate in which the two labials are fused as in (a). All the other candidates lose due to the postulated ranking. Next, let us review the impossible sequence [pʷap] with the same constraint ranking.

(24) *[pʷap]

/ pʷa p/ | | [lab]$_1$ [lab]$_2$ | [back]	OCP [lab]	MAX-IO	MAX [lab]	DEP [cor]	UNIFORM-ITY [lab]	MAX [back]
a. pʷa pʷ | | [lab] [lab] | | [back][back] F- Insertion	*!					
b. pʷa p | | [lab][lab] | [back] Faithful	*!					
☞c. pʷapʷ \ / [lab]$_{1,2}$ | [back] F-share/fus.					*	
d. [cor] | ta p | [lab] F-change			*	*		*!
e. a p | [lab] | [back] F&Seg-Del		*!	*			
f. p a p \ / [lab]$_{1,2}$ F-deletion & fusion					*	*!

The sequence [p^wap] is never observed in the language. The ill-formedness of the sequence is explained by the proposed ranking. What penalizes candidate (b) is its violation of OCP[lab]. The two labial features cannot be fused in (b), because they do not share the same value for backness.

In this section, I have shown that the proposed constraint ranking for Type 2 in section 2.3.3.2 can account for OCP effects when realized as a Morpheme Structure Constraint (MSC) by identifying a valid ranking which accounts for both the possibility of the well-formed structure, and the impossibility of the ill-formed structures.

3.2 Type 3: Coronal Dissimilation and The Emergence of The Unmarked (TETU) in Dakota Reduplicated Verbs

In this section, I will analyze Dakota coronal dissimilation as an example of a Type 3 language. When a feature deletes from a segment to satisfy an OCP constraint, it creates a potentially ill-formed structure in a language. To avoid this, additional repairs may take place at the expense of violating other faithfulness constraints.

In a Type 3 language, a feature deletes in order to satisfy an OCP constraint. In addition to featural deletion, featural insertion takes place so as not to give rise to an ill-formed structure. The ranking proposed for Type 3 in section 2.3.3.3 is as follows.

(25) The ranking for Type 3

```
OCP[F],  HAVEPLACE, UNIFORMITY[F], MAX-IO
  |__________|__________|_____________|
              ______|________
             |               |
          MAX[F]          DEP[F]
```

To examine this ranking in an actual language, I will discuss data from Dakota.

3.2.1 Coronal Dissimilation in Dakota Reduplicated Verbs

In Dakota, when a coronal is followed by another coronal in a reduplicated verb form, it turns into a dorsal (Shaw 1980, 1985). According to Shaw, this alternation is not observed in lexical compounds. I will discuss the asymmetry between the reduplicated verbs and the lexical compounds in more detail in section 3.3. In this section, I will concentrate only on the reduplicated forms.

(26) Dakota Coronal Dissimilation: Reduplicated Verb forms (Shaw 1985:184)

/**s**u**t**/	→	*[su**ts**úta]	[su**ks**úta]	'strong'
/**ž**a**t**/	→	*[ža**tž**áta]	[ža**gž**áta]	'curved'
/**t**he**č**/	→	*[the**čt**héča]	[the**kt**héča]	'be new'
/**č**he**č**/	→	*[čhe**čč**héča]	[čhe**kč**héča]	'to look like'
/**n**i**n**/	→	*[ni**nn**ína]	[ni**gn**ína]	'very'

When a stem both begins and ends with coronals, the two coronal features are adjacent in the reduplicated forms. This would result in an OCP[cor] violation without any repair. To avoid this violation, the first coronal in the sequence becomes dorsal.

3.2.2 An Analysis within the OT Framework

3.2.2.1 Faithfulness Constraints for Base and Reduplicants: Correspondence Theory

As mentioned in section 2.3.2.1, McCarthy and Prince's (1995) Correspondence Theory generalizes to various kinds of linguistic relationships. All the types of correspondence relations, such as input-output (IO), output-output (OO), base-reduplicant (BR), etc. are instances of one general relation, each involving a comparable set of faithfulness constraints. Therefore, Input-Output has a full set: {MAX-IO, DEP-IO, ...}; similarly, Base-Reduplicant has a full set: {MAX-BR, DEP-BR,....}; Output-Output has a full set: {MAX-OO, DEP-OO, ...} and so on. I will not go into a detailed discussion of Correspondence Theory here, but leave that task for chapter 5. This section will use Base-Reduplicant (BR) constraints, following McCarthy and Prince (1995).

3.2.2.2 OCP[cor] >> MAX[cor]-BR

Let us begin the analysis by establishing the relative ranking of OCP[cor] and MAX[cor]-BR. I take the Dakota data to indicate that OCP[cor] is respected at the expense of MAX[cor]-BR.

(27) OCP[cor] >> MAX[cor]-BR

/**RED**+ **s** u **t**/ | | [cor]$_1$**[cor]$_2$**	OCP[cor]	MAX[cor]-BR
a. a. **s** u **t** **s** u **t** |		| [cor]$_1$**[cor]$_2$**[cor]$_1$**[cor]$_2$**	*!	
☞ b. [dor] | **s** u **k** **s** u **t** |	| [cor]$_1$ [cor]$_1$ **[cor]$_2$**		*

The ranking in (27) entails that candidate (b), in which the place feature changes, is a better candidate than (a), in which OCP is violated.

I will examine other possible candidates and additional constraints in the next sections.

3.2.2.3 HAVEPLACE >> DEP[dor]-BR

In section 3.2.2.2, I have stated that deletion of a coronal feature is preferred to an OCP violation. However, the deletion of a coronal feature results in an ill-formed placeless segment. Some place feature must be inserted to repair this illicit structure. Thus, the ranking, HAVEPLACE >> DEP[dor]-BR is needed.

(28)

/**RED**+ **s** u **t** / | | $[cor]_1$**$[cor]_2$**	OCP[cor]	HAVE PLACE	MAX[cor]-BR	DEP[dor]-BR
a. a. **s** u **t** **s** u **t** | | | | $[cor]_1$**$[cor]_2$**$[cor]_1$**$[cor]_2$**	*!			
☞b. [dor] | **s** u **k** **s** u **t** | | | $[cor]_1$ $[cor]_1$**$[cor]_2$**			*	*
c. c. **s** u **t** **s** u **t** | | | $[cor]_1$ $[cor]_1$**$[cor]_2$**		*!	*	

Neither candidate (b) nor (c) violates OCP[cor]; therefore, both are better than (a). Nevertheless, candidate (c), which has a placeless segment, loses to (b). Therefore, HAVEPLACE outranks DEP[dor]-BR.

3.2.2.4 UNIFORMITY[cor] >> MAX[cor]-BR, and MAX-BR >> DEP[dor]-BR

In this section, other possible repair strategies, such as featural fusion and segmental deletion, are analyzed. Since neither of these strategies is observed in the language, I assume that UNIFORMITY[cor] and MAX-BR are higher-ranked than MAX[cor]-BR and DEP[dor]-BR.

(29)

/**RED**+ **s** u **t** / | | [cor]$_1$**[cor]$_2$**	OCP [cor]	HAVE PLACE	UNI-FORM-ITY [cor]	MAX-BR	MAX [cor]-BR	DEP [dor]-BR
a. **s** u **t** **s** u **t** | | | | [cor]$_1$**[cor]$_2$**[cor]$_1$**[cor]$_2$**	*!					
☞b. [dor] | **s** u **k** **s** u **t** | | | [cor]$_1$ [cor]$_1$**[cor]$_2$**					*	*
c. **s** u **t** **s** u **t** | \\ / | [cor]$_1$ **[cor]$_{1,2}$** **[cor]$_2$**			*!			
d. **s** u **s** u **t** | | | [cor]$_1$ [cor]$_1$**[cor]$_2$**				*!	*	

In order to rule out candidate (c) or (d), UNIFORMITY[cor] must outrank MAX[cor]-BR, and MAX-BR is necessarily higher-ranked than DEP[dor]-BR.

Let us summarize the ranking so far.

(30) The ranking in Dakota:

```
UNIFORMITY[cor]    OCP[cor]       HAVEPLACE        MAX-BR
   |_______________|  |_____________|_______________|
          |                         |
     MAX[cor]-BR                DEP[dor]-BR
```

This ranking is consistent with that initially formulated for Type 3 languages in (45) in chapter 2. The next section will examine this reduplication by adding further possible candidates.

3.2.2.5 *Phar]$_\sigma$ >> *Dor, *Lab >> *Cor >> *Phar

In the previous sections, I have discussed the candidates with: 1) OCP violation; 2) [cor] deletion and [dor] insertion; 3) [cor] deletion resulting in an ill-formed placeless segment; 4) featural fusion; and 5) segmental deletion. The ranking in (30) accounts for the optimality of the second.

In this section, I will consider why the inserted feature is [dorsal]. Prince & Smolensky (1993) propose that there is a universal ranking of markedness constraints for place feature.

(31) Universal Ranking of Markedness Constraints for Place Features:

*Dor, *Lab >> *Cor

Lombardi (1995b) extends this hierarchy by adding a markedness constraint for the pharyngeal feature.

(32) Revised Universal Ranking of Markedness Constraints for Place Features:

*Dor, *Lab >> * Cor >> *Phar

This ranking is assumed to be universal; therefore, the most unmarked feature should always surface unless some faithfulness constraints for some place features are higher-ranked. Consequently, the least marked feature [phar] would be the most appropriate default feature.

Now, let us consider the case of Dakota. Recall that the [cor] feature must be deleted due to the highly ranked constraint, OCP[cor]. HAVEPLACE which forbids a placeless segment is also ranked highly; hence, some place feature must be inserted as a default. According to Lombardi's hierarchy, the most unmarked feature [phar] rather than [dor] should have been inserted.

Then, /t/ would turn not into [k] but into [ʔ]. Following McCarthy (1989), Lombardi (1995b) indicates that [ʔ] and [h] each bears the place feature [phar]. Also,[ʔ] bears the manner feature [stop], while [h] bears [cont]. With respect to the manner feature, [ʔ] is the most unmarked segment .

Nevertheless, [dor] is actually inserted, resulting in [k]. Why is it not [ʔ] (or even [h]) but [k]? The phonemic inventory of this language contains both sounds; therefore, both should be possible candidates as the default. Shaw (1985) does not mention a reason for inserting the dorsal feature. However, I assume based on the data she provides that [ʔ] and [h] are not permitted in the coda. They freely occur in the onset position, but they are never observed in the coda.

I conclude that a coda condition constraint prohibiting pharyngeal codas is ranked above the markedness constraint for the dorsal feature.

The constraint for the coda condition is formalized as follows:

(33)

$$\begin{array}{c} *R]_{\sigma} \\ | \\ \text{[phar]} \end{array} \quad \text{(abbreviated as } *\text{Phar}]_{\sigma}\text{).}$$

The insertion of the dorsal feature is preferable to that of the pharyngeal. *Phar]σ outranks *Dor; therefore, the candidate with the dorsal insertion will be optimal despite the universal markedness hierarchy for place features.

(34) dorsal insertion

/**RED**+ su**t**/ | | [cor][cor]	*Phar]σ	*Dor	*Phar
a. [phar] | **suʔ sut** | [cor]	*!		*
☞ b. [dor] | **suk sut** | [cor]		*	

Tableau (34) describes the discussion in this section. Candidate (b) is optimal because *Phar]σ outranks the entire ranking for the markedness constraints.[2] Thus, the ranking in tableau (34) correctly accounts for the dorsal insertion.

(35) Ranking for dorsal insertion (1):[3]

*Phar]σ
|
*Dor
|
*Phar

We cannot determine yet how the constraint ranking in this section will interact with the one in (30) in section 3.2.2.4.

2 The interaction of the faithfulness constraints which prohibit insertion of the features could also account for the dorsal deletion. The ranking DEP[phar]-BR >> DEP[dor]-BR gives rise to dorsal insertion rather than pharyngeal insertion. However, once the ranking in (30) is determined, the ranking of those faithfulness constraints does not matter.

3 Smolensky (p.s.) suggests the possibility of the ranking *[lab] >> *[dor] in Dakota. I will consider this ranking with the more data in future investigation.

We need to discuss one more place feature in the hierarchy in (32). Once we conclude that the pharyngeal feature is inappropriate as the inserted place feature, there are two possible candidates for the inserted place feature, dorsal and labial. I will discuss this in the next section.

3.2.2.6 DEP[lab]-BR >> DEP[dor]-BR

Another possible candidate is one involving insertion of the labial feature. Unlike the case of [ʔ, h], the labial stop [p] can occur in the coda in Dakota. The universal ranking of the markedness constraint for the place features does not distinguish *[dor] from *[lab].

We could lead to one possible interpretation when we follow the idea discussed in section 3.2.2.5: [lab] cannot occur in the coda position due to the coda condition which prohibits [lab] from occurring in coda. However, this interpretation does not hold, because [lab] is observed in the coda position in this language. Thus, the coda-condition account is impossible.

I propose that an appropriate ranking of faithfulness constraints restricting the insertion of [dorsal] and [labial] explains the dorsal insertion. The faithfulness constraint which prohibits the insertion of [labial], DEP[lab]-BR ranks above that for [dor], DEP[dor]-BR.

(36)

/**RED**+ **sut**/ | | [cor][cor]	OCP[cor]	DEP [lab]-BR	DEP [dor]-BR	MAX [cor]-BR
a. **sut sut** | | [cor][cor]	*!			
☞b. [dor] | **suk sut** | [cor]			*	*
c. [lab] | **sup sut** | [cor]		*!		*

Candidates (b) and (c) both violate MAX[cor]-BR. Therefore, the rankings between MAX[cor]-BR and DEP[lab]-BR as well as MAX[cor]-BR and DEP[dor]-BR are not determined. What makes candidate (c) lose to (b) is the fatal violation of DEP[lab]-BR.

(37) Ranking for dorsal insertion (2)

DEP[lab]-BR
|
DEP[dor]-BR

Thus, the rankings given in (34) and (37) account for why the dorsal feature is inserted in preference to other place features.

Let us take a look at the entire ranking in Dakota so far:

(38) The revised ranking in Dakota:

```
(a)     UNIFORMITY[cor]  OCP[cor] HAVEPLACE MAX-BR
                |______________| |_____|__________|       DEP[lab]-BR
                      |                          |          |
                MAX[cor]-BR                      DEP[dor]-BR
```

```
(b)     *Phar]σ
            |
         *Dor
            |
         *Phar
```

Both of the rankings (a) and (b) in (38) belong to a single ranking in Dakota; however, the interaction of the two has not been determined yet. To determine the full ranking, further investigation of the data is necessary.

The next section discusses the asymmetry of the alternations between reduplicative verbs and lexical compounds.

3.2.3 Lexical Compounds

In section 3.2.1, I introduced the data in which coronal dissimilation is observed. Shaw (1985) points out that such coronal dissimilation takes place only in the reduplicative verb as in (26), and is not observed in lexical compounds.

(39) Lexical Compounds

/phe**t** + **n**akpa-kpa/	→	[phe**dn**akpakpa]	'sparks'
/sdo**t** + **č**hi-ya/	→	[sdo**dč**hi-ya]	'I know you'

In the lexical compounds, in contrast to reduplicative verb forms, two coronals can be adjacent. (Note that voiceless obstruents in the syllable coda position become voiced in the lexical compounds, while they do not in the reduplicative ones. This is not explained here.)

How can we account for this asymmetry between the two forms? Shaw (1985) explains it within the lexical phonology framework. She indicates that both reduplication and coronal dissimilation take place at level 1, while lexical compounds are focused on at level 2. Morpheme level differences can be analyzed in OT with the families of correspondence discussed earlier–IO, BR, OO– and different rankings of the parallel faithfulness constraints (e.g. MAX-IO, MAX-BR, MAX-OO...).

As Itô and Mester (1996), Alderete (1997), or Suzuki (1998) claim, the constraint for OCP is assumed to specify the domain in which the constraint is effective. If we adopt this idea, the asymmetry here can be explained as a difference of a domain. In this section, nevertheless, I try to account for these phenomena without specifying the domain, and assume the OCP constraint prohibits any two adjacent [cor] features. I analyze the asymmetry as constraint interaction rather than a stipulated refinement of a single constraint. However, this does not imply that no OCP constraints specify a domain. I will leave the domain of the OCP for future investigation.

I argue that the asymmetry between the two forms is derived from "The Emergence of The Unmarked (TETU)" (McCarthy and Prince 1995) as a result of constraint interaction. McCarthy and Prince claim that even in a language in which a marked structure is generally allowed, it is often the case that the same marked structure is banned in a special domain. In such a domain, an unmarked structure emerges. This is called TETU.

I claim that Dakota coronal dissimilation is a case of TETU. In Dakota, two coronals can usually be adjacent. However, in the domain of reduplication, the sequence [cor][cor] is banned. Therefore, the unmarked sequence of [dor][cor] emerges.

I propose that the constraint ranking which accounts for TETU in Dakota is the one in (40).

(40) The ranking for TETU

MAX[cor]-IO
|
OCP[cor]
|
MAX[cor]-BR

I will analyze the case of lexical compounds with the new ranking first (ignoring the voicing change)\.

(41) no coronal dissimilation in lexical compounds:

/sdo**t** + **č**hi-ya/ | | $[cor]_1[cor]_2$	MAX[cor]-IO	OCP[cor]	MAX[cor]-BR
☞a. sdo**d** + **č**hi-ya | | $[cor]_1[cor]_2$		*	
b. sdo**g** + **č**hi-ya | $[cor]_2$	*!		

Since there is no reduplication here, MAX[cor]-BR is vacuously satisfied. Candidate (b), in which coronal dissimilation takes place, loses to candidate (a) due to a fatal violation of MAX[cor]-IO.

Next, I will examine the reduplicated forms which I have already illustrated in section 3.1. with the ranking in (40).

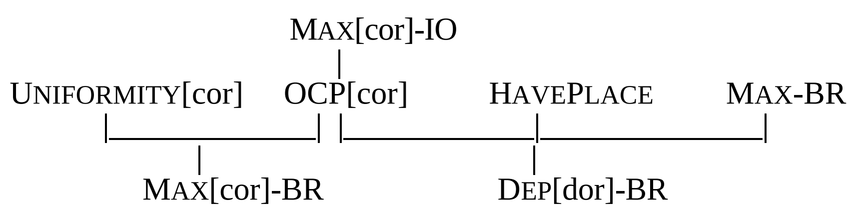
(42) coronal dissimilation in reduplicative verbs:

/**RED**+ **s** u **t** / [cor]$_1$**[cor]$_2$**	MAX[cor]-IO	OCP[cor]	MAX[cor]-BR
a. **s** u **t** **s** u **t** [cor]$_1$**[cor]$_2$**[cor]$_1$**[cor]$_2$**		*!	
☞ b. [dor] **s** u **k** **s** u **t** [cor]$_1$ [cor]$_1$**[cor]$_2$**			*

In tableau (42), neither of the two candidates violates MAX[cor]-IO. In the reduplicative form, it is not IO faithfulness constraints but rather BR faithfulness constraints that are violated.

Thus, the proposed ranking correctly accounts for both the lack of coronal dissimilation in the lexical compounds and coronal dissimilation in the reduplicative morphemes. Therefore we have the following ranking.

(43) The revised ranking in Dakota



The ranking in (43) accounts for TETU in Dakota. In general, the marked sequence of two adjacent coronals can surface due to the ranking of MAX[cor]-IO >> OCP[cor]. However, only in the domain of the reduplicated morphemes is the sequence banned, and the unmarked sequence surfaces due to the ranking of

OCP[cor] >> MAX[cor]-BR. The asymmetry can be explained by the whole ranking: MAX[cor]-IO >> OCP[cor] >> MAX[cor]-BR.

3.2.4 Coda Deletion Derived from TETU

In section 3.2.3, I have made it clear that the TETU ranking brings forth the phonological asymmetry between the verbal morphemes and the others in Dakota. In this section, I will discuss one more TETU phenomenon seen in this language.

Let us go back to verbal reduplication once again. Recall that the deletion of the coronal feature takes place in the coda position, and not in the onset position: [**s**u**ks**u**t**], and *[**s**u**tx**u**t**]. I have not yet considered the candidate in which coronal deletion is observed in the onset position, such as [**s**u**tx**u**t**]. In this section, I will reexamine the verbal reduplication with this additional candidate.

I claim that the alternation in the coda in Dakota is derived from the TETU ranking proposed in section 3.2.3. In Dakota, the two adjacent coronal features belong to distinct morphemes–the coda belongs to the reduplicant, while the onset is a part of the base. Therefore, the types of faithfulness constraints restricting feature deletion in the coda are different from those in the onset. IO faithfulness constrains the onset deletion, while BR faithfulness constrains the coda deletion.

Let us repeat the TETU ranking proposed in section 3.2.3.

(44) The ranking for TETU

MAX[cor]-IO
|
OCP[cor]
|
MAX[cor]-BR

Since the IO faithfulness constraint for the coronal feature is higher ranked than that for BR, the deletion of coronal in the coda is preferred to that in the onset.

(45)[4]

/**RED**+ **s** u **t** / | | [cor]$_1$**[cor]$_2$**	MAX[cor]-IO	OCP[cor]	MAX[cor]-BR
a. **s** u **t** **s** u **t** | | | | [cor]$_1$**[cor]$_2$**[cor]$_1$**[cor]$_2$**		*!	
☞ b. [dor] | **s** u **k** **s** u **t** | | | [cor]$_1$ [cor]$_1$**[cor]$_2$**			*
c. [dor] | **s** u **t** **x** u **t** | | | [cor]$_1$ **[cor]$_2$** **[cor]$_2$**	*!		

Candidate (c), in which [cor] deletion takes place in the onset, loses to (b) due to the TETU ranking in tableau (45).

Thus, the TETU ranking proposed in section 3.2.3 clearly accounts for deletion in the coda rather than the onset.

There are two other possible candidates in this sequence. One is that deletion and insertion take place in the onset of the reduplicant, and the other is that the alternations occur in the coda of the base. This is because there are actually four

4 A similar argument for TETU seems to be provided by the interaction of the faithfulness constraints which prohibit the insertion of dorsal, namely, DEP[dor]-IO and DEP[dor]-BR. However,
once the TETU interaction with MAX[cor]-IO and MAX[cor]-BR explains why deletion in the coda is preferred to that in the onset, it becomes unclear whether the ranking of these DEP constraints contributes to TETU with respect to determination of insertion of dorsal in the coda position. I assume that it is because the insertion of dorsal is triggered by the deletion of coronal.

possible adjacent [coronal] features involved in the sequence of the reduplicative form.

All relevant candidates are listed below. Some of them are relevant especially because since [cor] and [dor] are on different tiers, intervening dorsal does not prevent the [cor] OCP violation.

(46)

```
Input:    /RED+ s u  t/
                |    |
            [cor]1[cor]2
```

(a) Possible Candidate (a): three OCP violations

```
   [ s  u   t  s  u  t ]
     |      |  |     |
 [cor]1[cor]2[cor]1[cor]2
   |______| |___| |___|
       1      2     3
```

(b) Possible Candidate (b): two OCP violations, and alternations in the coda of the reduplicant (**optimal candidate**)

```
         [dor]
           |
  [ s  u   k  s  u  t ]
    |         |     |
 [cor]1     [cor]1[cor]2
   |__________| |___|
       1          2
```

(c) Possible Candidate (c): two OCP violations, and alternations in the onset of the base

```
           [dor]
             |
  [ s  u  t  x  u  t ]
    |     |        |
 [cor]1[cor]2   [cor]2
   |______| |_______|
       1         2
```

(d) Possible Candidate (d): two OCP violations, and alternations in the onset of the reduplicant

```
 [dor]
    |
  [ x  u   t   s  u  t ]
           |     |        |
        [cor]2[cor]1[cor]2
           |_____| |___|
               1       2
```

(e) Possible Candidate (e): two OCP violations, and alternations in the coda of the base

```
                    [dor]
                        |
  [ s  u   t  s  u  k ]
     |        |   |
[cor]1[cor]2[cor]1
   |_____| |___|
       1     2
```

Among these five candidates in (46), candidates (a), (b), and (c) have already been examined in tableau in (45) with the TETU ranking, and the analysis correctly accounts for why candidate (b) wins. Now, let us consider the remaining the two new candidates, (d) and (e), along with considering the four adjacent coronal features.

(47)

/**RED**+ **s** u **t**/ | | [cor]$_1$[cor]$_2$	MAX[cor]-IO	OCP[cor]	MAX[cor]-BR
a. su**ts**u**t** | | [cor][cor]		***!	
☞ b. actual winner [dor] | [**s** u **k** **s** u **t**] | | | [cor]$_1$ [cor]$_1$[cor]$_2$		**	*
c. [dor] | [**s** u **t** **x** u **t**] | | | [cor]$_1$[cor]$_2$ [cor]$_2$	*!	**	
*☞d. [dor] | [**x** u **t** **s** u **t**] | | | [cor]$_2$[cor]$_1$[cor]$_2$		**	*
e. [dor] | [**s** u **t** **s** u **k**] | | | [cor]$_1$[cor]$_2$[cor]$_1$	*!	**	

The proposed TETU ranking correctly accounts for why the alternations do not take place in the base. However, it does not distinguish between the two candidates (b) and (d), where the alternations take place in the onset and in the coda of the reduplicant, respectively. I claim that this is because a positional featural faithfulness constraint for the coronal feature, MAXONS[cor]-BR, is higher ranked than the general faithfulness constraint for [cor], MAX[cor]-BR.

Beckman (1995) suggests that faithfulness constraints are relativized to position on the basis of the observation that prosodic head positions tend to retain distinctions that are lost in other positions. She indicates that IDENTONS[F] >> IDENT[F] is a ranking which is generally observed. Lombardi (1995a) proposes a universal constraint ranking: "IDENTONSLar >> IDENTLar", which accounts for the fact that a laryngeal distinction is often retained in the onset when lost in other positions. Padgett (1995a) introduces MAXONS[Place][5] as a constraint which says that an input place feature in the onset must have an output correspondent.

Following the research on positional faithfulness constraints, I thus assume that there is an effect of positional featural faithfulness on [coronal] in the onset, and that deletion of a stop feature is observed not in the onset but in the coda of the reduplicant due to the constraint, MAXONS[cor]-BR. Let us see the analysis with this constraint.

(48)

/**RED**+ su**t**/	MAX[cor]-IO	OCP[cor]	MAXONS [cor]-BR	MAX[cor]-BR
a. **s**u**ts**u**t**		***!		
☞ b. **s**u**ks**u**t**		**		*
c. **s**u**tx**u**t**	*!	**		
d. **x**u**ts**u**t**		**	*!	*
e. **s**u**ts**u**k**	*!	**		

5 The actual constraint which Padgett (1995a) proposes is MAX_{REL}[Place] which has the same result as MAXONS[Place].

Thus, in tableau (48), candidate (d) is correctly ruled out due to the violation of the proposed positional faithfulness constraint, MAXONS[cor]-BR. The ranking of MAXONS[cor]-BR is not determined by this analysis. However, when we consider another candidate, [ʔuksut], in which the place features in both the onset and coda change, we obtain the ranking of MAXONS[cor]-BR. This candidate does not violate MAX[cor]-IO, and it commits only a single violation of OCP[cor]. Therefore, MAXONS[cor]-BR is necessarily higher-ranked than OCP[cor] as the following tableau illustrates:

(49)

/**RED**+ su**t**/	MAX[cor] -IO	MAXONS [cor]-BR	OCP[cor]	MAX[cor] -BR
a. **sutsut**			***!	
☞ b. **suksut**			**	*
c. **sutxut**	*!		**	
d. **xutsut**		*!	**	*
e. **sutsuk**	*!		**	
f. **ʔuksut**		*!	*	**

Thus, the ranking MAXONS[cor]-BR >> OCP[cor] is crucial in Dakota.

3.2.5 Summary

In section 3.2, I have analyzed the data from an actual language, Dakota. This language is Type 3 since featural deletion and insertion take place in response to the

OCP. The ranking discussed in this section is indeed consistent with the one I proposed in section 2.3.3.3.

(50) The basic constraint ranking in Dakota (repeated from 30):

```
                  MAXONS[cor]-BR
                         |
UNIFORMITY[cor]     OCP[cor]        HAVEPLACE        MAX-BR
   |____________________|  |_______________|_______________|
             |                             |
       MAX[cor]-BR                    DEP[dor]-BR
```

In addition to this ranking, I have introduced constraint interactions which give rise to TETU.

(51) TETU for coronal dissimilation

MAX[cor]-IO
|
OCP[cor]
|
MAX[cor]-BR

From this ranking, the marked sequence of two adjacent coronals is allowed in regular forms, while it is banned in the reduplicated forms. One of the coronals in the sequence becomes dorsal as an unmarked alternative in the reduplicative forms. Also, this alternation takes place not in the onset but in the coda of the reduplicant due to a positional featural faithfulness constraint, MAXONS[cor]-BR.

3.3 Type 4: Deletion and Spirantization in Basque Consonant Clusters

In section 2.3, I indicated that the typology of OCP effects on features can be explained on the basis of the four types of constraint rankings, and in sections 3.1

and 3.2, I analyzed actual languages for Type 2 and 3. In this section, I will analyze featural and segmental deletion in Basque, as a language of Type 4.

I have postulated that the following ranking will account for type 4 languages.

(52) Ranking for Type 4

OCP[F], HAVEPLACE, UNIFORMITY[F], DEP[F]

MAX[F]   MAX-IO

In Basque, we see deletion of a stop in a sequence of two stops, and spirantization of an affricate in a sequence of an affricate and a stop (Archangeli & Pulleyblank 1987, Hualde 1987, 1988, 1991). Lombardi (1990a, b) analyzes these phenomena as the result of delinking of [stop] from a segment due to the effects of the OCP.

If my prediction for the type 4 language in section 2.3.3.4 is correct, then, the Basque phenomena should follow from the ranking in (52).

3.3.1 Previous Analyses in Autosegmental Phonology

Hualde (1987) and Archangeli and Pulleyblank (1987) point out that a stop deletes when it precedes another stop in Basque, as the following data show:

(53) Basque stop + stop (from Hualde 1987, Archangeli and Pulleyblank 1987):

z = [s], s = [ś], tz =[c], ts = [ć], tx = [č]

/bai**t** **n**aiz/	bai**n**aiz	'since I am'
/oroi**t** + **m**en/	oroi**m**en	'remembrance'
/gu**k** **p**itzu/	gu **p**iztu	'we light'
/ardie**k** **n**ituen/	ardie **n**ituen	'we had sheep'
/ba**t** **p**aratu/	ba **p**aratu	'put one'
/ba**t** **t**raban/	ba **t**raban	'one stuck'
/ba**t** **k**urri/	ba **k**urri	'run one'

According to Hualde (1987), deletion and spirantization[6] can apply at both morpheme boundaries and word boundaries in rapid speech in Basque. Hualde (1987) and Archangeli and Pulleyblank(1987) formalize the rule of stop deletion in Basque as follows:

(54)

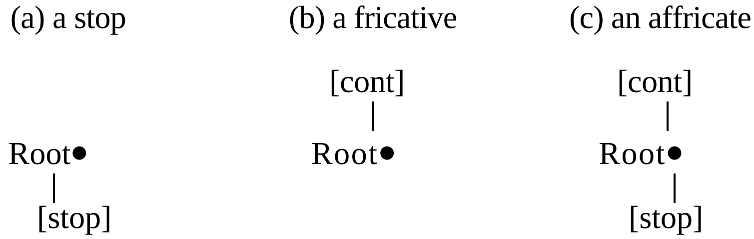
$$[-\text{cont}, -\text{son}] \rightarrow \varnothing \ / \ \underline{\hspace{2cm}} \ [-\text{cont}].$$

Lombardi (1990 a, b) analyzes this as delinking of [stop] from the first segment motivated by the OCP.[7] Lombardi further argues that [stop] and [cont] must be independent features on separate tiers to account for the Basque data. As (55) shows, both a stop and a fricative bear only one manner feature, while an affricate bears two manner features which are placed on two separate tiers.

6 The data for spirantization will be given later in this section.

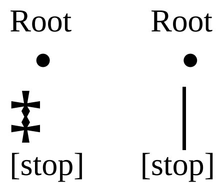
7 I assume the domain for the OCP in Basque is any two adjacent segments, following Hualde's and Archangeli and Pulleyblank's claims. The more detailed analysis of the domain for the OCP in Basque is left for future research.

(55)



According to Lombardi, the rule for the effects of the OCP in Basque is formalized as follows:

(56)



Lombardi claims that after delinking of the feature [stop] as the result of the OCP effects, the entire segment must delete, because loss of specification for manner of articulation gives rise to the impossibility of articulating the segment.

When an affricate precedes a stop, the effect of this rule is different. The affricate does not delete but becomes a fricative:

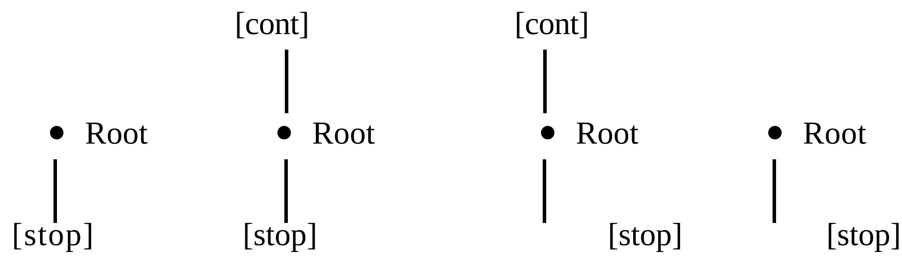
(57) Basque affricate + stop (from Hualde (1987) and Archangeli and Pulleyblank (1987)) z = [s], s = [ś], tz =[c], ts = [ć], tx = [č]

/hi**ts** + **t**egi/	hi**zt**egi	'dictionary'
/hi**ts** + **k**eta/	hi**zk**eta	'conversation'
/hari**ts** + **m**endi/	hari**zm**endi	'oak mountain'
/hari**ts** + **k**i/	hari**zk**i	'oak wood'
/ho**tz** + **t**u/	ho**zt**u	'a cold'

Lombardi argues that since the [stop] and [cont] features of an affricate belong to two separate tiers, both features are accessible from both the contexts which precede and follow the segment. When an affricate is preceded or followed by a stop , the two [stop] features are adjacent as in (58 a) and (58b):

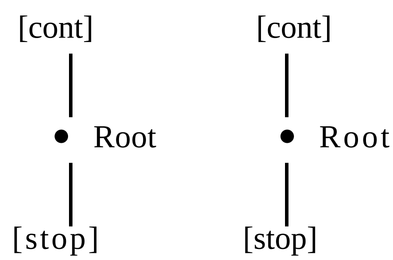
(58)

(a) a stop and an affricate: (b) an affricate and a stop



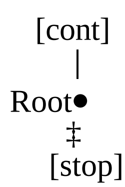
When an affricate is either preceded or followed by another affricate, the two [stop] features are also adjacent as in (59):

(59) an affricate and an affricate:



Thus, the structural description of the rule in (56) is met by an affricate-stop (or affricate) cluster. But the effect of deletion is different. When the [stop] feature is delinked from an affricate, it still has all the features for a fricative:

(60)



As shown in (60), an affricate is realized as a fricative after delinking of the feature [stop], which stands in contrast to the deletion of the entire stop segment after deletion of the [stop] feature.

3.3.2 Analyses within the OT Framework

Let us use the following constraints which have been proposed in section 2 to analyze the Basque deletion.

(61) Constraints:

(a) OCP[stop] = *[stop][stop]
Adjacent stop features are prohibited (Leben 1973; Goldsmith 1976; Mester 1986; McCarthy 1986).

(b) MAX[stop]: An input stop feature must have an output correspondent (no featural deletion) (Lombardi 1995a, Lamontagne and Rice 1995).

(c) DEP[cont]: An output continuant feature must have an input correspondent (no featural insertion) (Lombardi 1995a, Lamontagne and Rice 1995).

(d) UNIFORMITY[stop]: No feature of the output has multiple correspondents in the input (no featural fusion) (McCarthy and Prince 1995, Causley 1997).

(e) MAX-IO: Every segment of the input has a correspondent in the output. (no segmental deletion) (McCarthy and Prince 1995).

(f) HAVEMANNER: Every segment must bear some manner feature;

In addition to the constraints for the typology which I have already introduced in section 1, we also require another constraint in this case, namely (f) HAVEMANNER. Padgett (1994, 1995a) proposes a constraint against a placeless segment, **HAVEPLACE**. I assume that a constraint against a segment without Manner, **HAVEMANNER** is also a valid constraint on the basis of Padgett's (1994, 1995a) idea.

In Basque, the interaction of the basic constraints established for the typology alone results in an ill-formed structure a mannerless segment. In such a case, additional constraints affect the selection of the optimal candidate. In another case, for example, where the interaction of the basic constraints gives rise to a voiceless segment, or a stridentless segment, a similar effect would not be observed. Since a voiceless segment or a stridentless segment is not ill-formed, no constraint such as HAVEVOICE or HAVESTRIDENT need be introduced.

In the following sections, I will discuss how the ranking of these constraints gives rise to optimal candidates.

3.3.3 Deletion in the Sequence of [stop][stop]

Let us embark on the analyses of the phonological phenomena in Basque by analyzing the sequence of two stops.

3.3.3.1 OCP[stop] >> MAX-IO, MAX[stop]

The [stop] feature deletes to satisfy OCP[stop]; therefore, the ranking OCP[stop] >> MAX[stop] is expected, and deletion of the entire segment ensues; hence, MAX-IO must also be lower ranked than OCP[stop].

(62)

/ba**t** **t**raban/	OCP[stop]	MAX-IO	MAX[stop]
☞a. ba **t**raban		*	*
b. ba**t** **t**raban	*!		

Tableau (62) illustrates that OCP[stop] must outrank both MAX-IO and MAX[stop] to make candidate (a), in which deletion of the [stop] feature and the segment takes place, win. Thus, one part of the ranking postulated in (52) has been confirmed at this stage:

(63)

```
    OCP[stop]
      __|___
     |      |
MAX-IO  MAX[stop]
```

3.3.3.2 OCP [stop], HAVEMANNER >> MAX-IO, MAX[stop]

As mentioned in section 3.3.1, Lombardi (1990 a, b) indicates that deletion of the manner feature results in the deletion of the entire segment due to impossibility of articulation. This can be formalized within an OT analysis by the ranking HAVEMANNER >> MAX-IO, MAX[stop]:

(64) OCP[stop], HAVEMANNER >> MAX-IO, MAX[stop]:

/ba**t** **t**raban/ | | [stop][stop]	OCP [stop]	HAVE MANNER	MAX-IO	MAX [stop]
a. ba**t** **t**raban | | [stop][stop]	*!			
b. ba**t** **t**raban | [stop]		*!		*
☞ c. ba **t**raban | [stop]			*	*

As tableau (64) shows, OCP[stop] and HaveManner should be satisfied at the expense of a violation of MAX-IO in Basque. Since candidates (b) and (c) both violate

MAX[stop], the violation of that constraint is not decisive here. Thus, the following ranking is obtained:

(65)

```
  OCP[stop]  HAVEPLACE
        |______|
         __|___
        |       |
MAX-IO  MAX[stop]
```

This ranking explains why deletion of the entire segment is preferred to just the deletion of the [stop] feature.

3.3.3.3 MAXONS[stop]

In addition to the candidates which I have examined in tableaux (62) and (64), there is another candidate in which the second segment in the sequence deletes. The ranking given in the previous sections cannot rule out this candidate, because the violation which this candidate incurs is equal to that of the optimal candidate .

(66) OCP [stop], HAVEMANNER >> MAX-IO, MAX[stop]

/ba**t** **t**raban/	OCP [stop]	HAVE MANNER	MAX-IO	MAX [stop]
☞a. ba **t**raban			*	*
*☞b. ba**t** raban			*	*

Both of the candidates equally violate the lower-ranked constraint MAX-IO and MAX[stop]. In order to decide the optimal candidate, a positional featural faithfulness constraint MAXONS[stop] should be introduced.

(67)

MAXONS[stop]: An input stop feature in the onset must have an output correspondent (Beckman 1995, Lombardi 1995a, Padgett 1995a)[8]

A featural faithfulness constraint for the feature [stop], namely, MAX[stop] is relatively low-ranked in this language. On the other hand, I claim that the positional featural faithfulness constraint MAXONS[stop] is highly ranked.

As I already referred to in section 3.2.4, Beckman (1995) suggests that faithfulness constraints are relativized to position on the basis of the observation that prosodic head positions tend to retain distinctions that are lost in other positions. I proposed one positional faithfulness constraint MAXONS[cor] in my analysis of Dakota reduplication in section 3.2.4.

Following this concept of the positional faithfulness constraint, I thus assume that there is an effect of positional featural faithfulness on [stop] in the onset, and that deletion of a stop feature is observed not in the onset but in the coda due to the constraint, MAXONS[stop]. MAXONS[stop] is therefore needed, even though a positional faithfulness constraint for the specific [stop] features has never been proposed before.

An additional violation of MAXONS[stop] will correctly rule out candidate (b) in tableau (68):[9]

---

8 There is a conrovertial issue regarding the definition of this constraint. This definition assumes syllable structure in the input. I would like to deal with the issue in future investigation.

9 As long as we analyze the sequence of two stops, NOCODA or MAXONSIO can be replaced by MAXONS[stop] to correctly rule out the failed candidate. However, neither NOCODA nor MAXONSIO cannot provide the correct analysis for the sequence of a stop and an affricate which will be dissucssed in section 3.3.5.

(68) MAXONS[stop]:

/ba**t** **t**raban/	MAX-IO	MAX[stop]	MAXONS [stop]
☞ a. ba **t**raban	*	*	
b. ba**t** raban	*	*	*!

This tableau shows that deletion of the segment in the coda position is preferred to deletion in the onset position. From the analysis in this tableau, the ranking of MAXONS[stop] remains undetermined.

The following ranking has thus been established so far:

(69)

```
OCP[stop]            HAVEMANNER
        |________________|
            ___|__
           |      |
  MAX-IO  MAX[stop]                 MAXONS[stop]
```

Let us review what we have discussed so far with the ranking in (69) with the tableau in (70):

(70) OCP[stop], HAVEMANNER >> MAX-IO, MAX[stop], MAXONS[stop]

/ba**t** **t**raban/ | | [stop][stop]	OCP [stop]	HAVE MANNER	MAX-IO	MAX [stop]	MAXONS [stop]
a. ba**t** **t**raban | | [stop][stop]	*!				
b. ba**t** **t**raban | [stop]		*!		*	
c. ba**t** raban			*	*	*!
☞ d. ba **t**raban			*	*	

It has been observed that the ranking OCP[stop], HAVEMANNER >> MAX-IO, MAX[stop], MAXONS[stop] accounts for the deletion of the entire segment in coda position. In the next section, I will show why changing a segment into another segment is not a possible way to satisfy OCP[stop] in this language.

3.3.3.4 DEP[cont] >> MAX-IO

Deletion of one of the features is one way to satisfy the higher-ranked OCP constraint. This is what takes place in Basque. However, there are at least three other ways to satisfy the OCP constraint in the same situation. One of them which will be discussed in this section is changing the character of one of the two segments by insertion of some feature(s). In order for insertion of a feature to be possible, the faithfulness constraint which prohibits the insertion of a feature, DEP[F] must be relatively low-ranked in the language.

In Basque, as mentioned above, deletion (of both the segment and the feature) is preferred to changing the character of the segment; therefore, DEP[cont], a higher-ranked constraint, is satisfied at the expense of a violation of MAX-IO.

(71) DEP[cont] >> MAX

/ba**t** **t**raban/	OCP[stop]	**DEP[cont]**	MAX-IO	MAX[stop]
☞ a. ba **t**raban			*	*
b. ba**s** **t**raban		*!		*
c. ba**t** **t**raban	*!			

Since I assume, following Lombardi (1990a, b), that there are two separate features: [stop] and [cont] , candidate (b) in (71) violates both DEP[cont] and MAX[stop]: to change a stop to a fricative, the feature [stop] must delete, and the feature [cont] must be added. Therefore, the ranking which crucially rules out candidate (b) is:

(72)

DEP[CONT]
|
MAX-IO.

3.3.3.5 UNIFORMITY[stop] >> MAX-IO, MAX[stop]

Another possible way to satisfy OCP[stop], apart from deleting or changing a segment, is to fuse the two [stop] features into one so that the two stops share one feature. Coalescence of the two [stop] features is not allowed as a repair in Basque; therefore, I assume that UNIFORMITY[stop], which prohibits fusion of the features, is higher-ranked than MAX-IO and MAX[stop].

(73) UNIFORMITY[stop] >> MAX-IO, MAX[stop]

/ba**t** **t**raban/ | | $[stop]_1[stop]_2$	OCP[stop]	UNIFORMITY [stop]	MAX-IO	MAX[stop]
☞ a. ba **t**raban | $[stop]_2$			*	*
b. ba**t** **t**raban | | $[stop]_1[stop]_2$	*!			
c. ba**t** **t**raban \/ $[stop]_{1,2}$		*!		

Even though candidate (c) satisfies the higher-ranked constraint OCP[stop], it still loses which indicates that UNIFORMITY[stop] is also highly ranked. I am assuming that fusion of the features does not violate MAX[stop], because the stop features of the input are realized in the output in the case of fusion.

3.3.3.6 DEP-IO >> MAX-IO

I argued in section 1.2 that epenthesis does not repair OCP on features. In this section, let us observe the actual data in Basque to make it clear why epenthesis of a segment between the two segments with a [stop] feature is not a relevant repair.

Since epenthesis of a segment is not observed in Basque, I conclude that DEP-IO, which prohibits segmental insertion, is also higher-ranked than MAX-IO:

(74) DEP-IO >> MAX-IO

/ba**t** **t**raban/	OCP[stop]	DEP-IO	MAX-IO
☞ a. ba **t**raban			*
b. ba**t** **t**raban	*!		
c. ba**t** **V** **t**raban	*!	*!	

Tableau (74) shows that epenthesis of a segment is not possible in this language not only due to the fatal violation of DEP-IO but also due to the violation of OCP[stop]. The failed candidate (c) violates OCP[stop] because the stop features are still adjacent regardless of the existence of the inserted vowel. Thus, the Basque datum shows that epenthesis of a segment should be eliminated from the typology of repair strategies for the OCP effects on features.

3.3.3.7 Summary

The discussion so far has lead us to the conclusion that the first segment of a sequence of two stops deletes in Basque due to the following constraint ranking:

(75)

```
OCP[stop] HaveManner Uniformity[stop], Dep-IO  Dep[cont]
     |__________|____________|_______________|            |
         _______________|________________________    |
         |                    |                  |   |
     Max[stop],          MaxOns[stop],         Max-IO
```

Let us summarize using the following tableau:

(76) a stop and a stop

/ba**t** **t**raban/	OCP [stop]	HAVE MANNER	UNIFORMITY [stop]	DEP IO	DEP [cont]	MAX-IO	MAX [stop]	MAX ONS [stop]
a. ba**t** **t**raban | | [stop] [stop] (no alternation)	*!							
b. ba**t** **t**raban | [stop] (deletion of the manner feature)		*!					*	
☞ c. ba **t**raban (deletion of a segment in the coda)						*	*	
d. ba**t** raban (deletion of a segment in the onset)						*	*	*!
e. ba**s** **t**raban (Change of a segment in the coda)					*!		*	
f. ba**t** **s**raban (Change of a segment in the onset)					*!		*	
g. ba**t** **t**raban \/ [stop] (fusion of the feature)			*!					
h. ba**tVt**raban (epenthesis)	*!			*!				*

Tableau (76) stresses the following points: first, OCP[stop] should be satisfied in this language (excluding candidate (a)). Second, a segment without a manner feature is impossible; therefore, deletion of the feature [stop] results in deletion of the entire segment. (cf. candidates (b) and (c)). Third, deletion of a segment in the coda is preferred to deletion in the onset (cf. candidates (c) and (d)). Fourth, deletion of a segment is preferred to changing the segment, fusion of two features, or epenthesis of a segment (cf. candidates (e), (f), (g), and (h)).

The ranking obtained so far corresponds to the one proposed in section 2.3.3.4. In the next section, I will examine the spirantization of affricates in Basque affricate and stop sequences.

3.3.4 Spirantization in the Sequence [stop, cont][stop]

Unlike the deletion of the first segment in a sequence of two stops, in a sequence of an affricate followed by a stop, spirantization of the first segment is observed.

Recall that in a sequence of two stops, when the feature [stop] deletes, the entire segment must delete at the expense of a violation of MAX-IO so as to satisfy the higher-ranked constraint HAVEMANNER. However, in the case of an affricate and stop sequence, the deletion of the [stop] feature does not necessarily violate MAX-IO. This is because of the asymmetry between the manner features of an affricate compared with a stop. An affricate bears both [stop] and [cont], while a stop has only [stop]. Thus, in a sequence of an affricate and a stop, there is a candidate which does not violate MAX-IO despite deletion of the [stop] feature.

(77) an affricate and a stop (1)

/hi**ts** + **t**egi/	OCP [stop]	HAVE MANNER	MAX [stop]	MAX-IO
a. hi**ts** **t**egi	*!			
b. hi **t**egi			*	*!
☞ c. hi**s** **t**egi			*	

As candidate (c) in tableau (77) shows, deletion of only the feature [stop] from an affricate results in violation of neither HAVEMANNER nor MAX-IO, because there is still a segment which bears a manner feature [cont]. Then, what penalizes candidate (b) is violation of MAX-IO.

In contrast to the sequence of two stops, in a sequence of an affricate and a stop there are two possible candidates which violate MAX[stop]; one also violates MAX-IO, while the other does not.

Before going on further with the analysis spirantization in the sequence of an affricate and a stop, I will determine the ranking of MAXONS[stop] crucial to in the analysis of deletion in the sequence of a stop and an affricate in the following section.

3.3.5 Deletion in the Sequence [stop][stop, cont]: MAXONS[stop] >> MAX-IO

Even though Hualde (1987, 1988, 1991), Archangeli & Pulleyblank (1987), and Lombardi (1990a, b) claim that deletion of a stop should take place in the sequences of a stop and an affricate, no actual data on such sequences has been presented in those studies.

We expect that one of the stop features will delete in that type of sequence to satisfy OCP[stop] which is highly ranked in the language. However, the present analysis makes as yet no predictions on whether the first or the second one deletes.

In this section, I will analyze the outcome of stop-affricate sequences by crucially ranking MAXONS[stop] higher-ranked than MAX-IO.

The following data relative to stop-affricate sequences have been obtained from the native speakers of Basque:[10]

10 Thanks to Ricardo Echepare, Aitsiber Atutxa, Elixabete Murguia and Itziar San Martin for the data here.

(78) Basque stop + affricate:

/ba**t** **tz**at/ ba **tz**at "as one"[11]
/ba**t** **tz**en/ ba **tz**en "putting together" (this example is derived from /bat+ tu+ tzen/.)

Based upon the data above, it appears to be the case that the stop deletes in the stop-affricate sequence.

We established above that deletion is observed not in the second segment (or the onset) but in the first one (or the coda) in the sequence of two stops in Basque. I have analyzed this as the result of the appropriate ranking of a positional featural faithfulness constraint MAXONS[stop].

In this section, I will illustrate the evidence of the ranking of MAXONS[stop] by analyzing the sequence of a stop and an affricate. Let us look at the following tableau:

(79) a stop and an affricate:

/ba**t** **tz**en/	OCP[stop]	**MAXONS [stop]**	MAX-IO
a. ba**t** **tz**en	*!		
☞ b. ba **tz**en			*
c. ba**t** en		*!	*
d. ba**t** **s**en		*!	

11. According to my informants, /bat tzat/ is a really awkward word in Basque. It will never show up in natural speech. However, they make it clear that the first stop will definitely delete in the sequence, if /tzat/ is forced to attach to /bat/.

In this sequence, there is a candidate (d) which violates neither MAX-IO nor any other higher-ranked constraint. It is crucial that MAXONS[stop] outranks MAX-IO, otherwise candidate (d) would win. Note that violation of MAX[stop] is not relevant, because candidates (b), (c), and (d) all violate it.

(80) Revised Rankings of Constraints:

```
OCP[stop],HAVEMANNER UNIFORMITY[stop],DEP-IO  DEP[cont] MAXONS[stop]
      |_______|____________|______________|           |       /
              _______________|_____________________   |      /
              |                                    |  |     /
            MAX[stop],                            MAX-IO
```

I will now go back to the analysis of spirantization in the sequence of an affricate and a stop with the new ranking in (80) in the following sections.

3.3.6 Impossibility of Deletion of a Segment in the Sequence of an Affricate and a Stop

In section 3.3.4, I have analyzed the sequence of an affricate and a stop and considered the three possible candidates: one in which no alternation occurs; one in which deletion of the affricate takes place; and one in which spirantization of the affricate is observed. The ranking introduced in tableau (77) correctly predicts that the one with spirantization of the affricate is the optimal candidate.

In section 3.3.5, the ranking of MAXONS[stop] was determined to outrank MAX-IO in the analysis of deletion of the stop segment in a sequence of a stop and an affricate. The analysis in that section established the new ranking given in (80).

In this section, in addition to those three candidates for the sequence of an affricate and a stop discussed in section 3.3.4, let us consider another candidate in which deletion takes place in the onset position with the established ranking in (80). This candidate should be ruled out because a positional featural faithfulness constraint, MAXONS[stop] is highly ranked in this language.

(81) No deletion in the sequence:

/hi**ts** **t**egi/	MAXONS [stop]	MAX-IO	MAX[stop]
☞ a. hi**s** **t**egi			*
b. hi**ts** egi	*!	*	*

Candidate (b) loses to (a) due to its fatal violation of MAXONS[stop]. In addition, it violates MAX-IO. Deletion of the stop feature is thus preferred in the coda position.

3.3.7 Impossibility of Spirantization of a Stop in the Sequence of an Affricate and a Stop

I have stated in section 3.3.3.4 that spirantization of a stop is not possible in Basque due to a high-ranked constraint DEP[cont]. However, in the sequence of an affricate and a stop, spirantization of an affricate is preferred to that of a stop. Let us also look at the phenomenon in the sequence of an affricate and a stop:

(82) No Spirantization of a stop:

/hi**ts** **t**egi/	DEP[cont]	MAXONS [stop]	MAX[stop]
☞ a. hi**s** **t**egi			*
b. hi**ts** **s**egi	*!	*!	*

Since an affricate originally bears [cont] as well as [stop], spirantization of an affricate does not violate DEP[cont] which is violated in the case of stop spirantization. Candidate (a) also does not violate another highly ranked constraint MAXONS[stop]. The ranking between DEP[cont] and MAXONS[stop] cannot be determined at this stage.

3.3.8 Summary of the Sequence of an Affricate and a Stop

The validity of the ranking proposed in the analysis of the sequence of two stops; hence the validity of the ranking predicted in section 3.3.3, has also been confirmed in the sequence of a stop and an affricate and that of an affricate and a stop with an additional ranking of MAXONS[stop] >> MAX-IO.

(83) Overall ranking in Basque:

```
OCP[stop] HaveManner UNIFORMITY[stop] DEP-IO DEP[cont] MAXONS[stop]
         |___________|_____________|__________|        |    /
              _______________|_________________        |   /
         |                                    |        |  /
         MAX[stop],                           MAX-IO
```

Let us summarize the analysis of sequence of an affricate and a stop and consider two additional candidates.

(84) an affricate and a stop:

/hi**ts** **t**egi/	OCP [stop]	HAVE MANNER	UNIFORM-ITY [stop]	DEP-IO	DEP [cont]	MAX ONS [stop]	MAX [stop]	MAX-IO
a. hi**ts** **t**egi | | [stop][stop] (no alternation)	*!							
b. hi **t**egi (deletion of a segment in the coda)							*	*!
☞ c. hi**s** **t**egi (deletion of the feature [stop]; spirantization of an affricate)							*	
d. hi**t** **t**egi (deletion of the feature [cont])	*!							
e. hi**ts** egi (deletion of a segment in the onset)						*!	*	*
f. hi**ts** **s**egi (spirantization of a stop)					*!	*!	*	
g. hi**ts** **t**egi \/ [stop] (fusion of the feature [stop])			*!					
h. hi**tsVt**egi (epenthesis)	*!			*!				
i. hi**ts** **t**egi | [stop] (mannerless: both [stop] & [cont] delete)		*!					*!	

Candidates (g) and (h) in tableau (84) indicate the impossibility of fusion of the two [stop] features and epenthesis of a segment, respectively.

Thus, the established ranking with an additional ranking "MAXONS[stop] >> MAX-IO >> NOCODA" correctly accounts for the data.

3.3.9 Summary of the OCP Effects on [stop]

Let us examine the validity of the new ranking in (83) by reanalyzing a sequence of two stops.

(85) Revised analysis of a stop and a stop:

/ba**t** **t**raban/	OCP [stop]	HAVE MANNER	UNIFORM-ITY [stop]	DEP-IO	DEP [cont]	MAX ONS [stop]	MAX [stop]	MAX-IO
a. ba**t** **t**raban | | [stop] [stop] (no alternation)	*!							
b. ba**t** **t**raban | [stop] (deletion of the manner feature)		*!					*	
☞ c. ba **t**raban (deletion of a segment in the coda)							*	*
d. ba**t** raban (deletion of a segment in the onset)						*!	*	*
e. ba**s** **t**raban (change of a segment in the coda)					*!		*	
f. ba**t** **s**raban (change of a segment in the onset)					*!	*!	*	
g. ba**t** **t**raban \/ [stop] (fusion the feature)			*!					
h. ba**tVt**raban (epenthesis)	*!			*!				

In section 3.3, I discussed deletion and spirantization as repair strategies to the OCP effect on [stop] in Basque consonant clusters. I conclude that Basque, which belongs to Type 4, has the ranking of constraints (83) which is repeated in (86):

(86) Revised Ranking of Constraints:

```
 OCP        HAVE      UNIFORMITY     DEP-IO   DEP     MAXONS
[stop]     MAMMER       [stop]                [cont]   [stop]
  |__________|____________|___________|______|          |
            _____________|_________________________     |
            |                                     |     |
         MAX[stop],                              MAX-IO
```

Let us compare this ranking with the ranking which was assumed for the Type 4 language in section 2.3.3.4, repeated here:

(87) The predicted ranking postulated for Type 4:

```
OCP[F], HAVEPLACE, UNIFORMITY[F], DEP[F]
  |_________|___________|____________|
            _____|_______
            |           |
         MAX[F]      MAX-IO
```

Thus, I conclude that the postulated ranking for Type 4 is consistent with the one in Basque. At the same time, I have introduced several additional rankings to account for the actual data.

The ranking in (86), has therefore accounted for the following four phenomena:

First, OCP[stop] is highly ranked in Basque; therefore, some phonological alternations must take place so as to satisfy this constraint at the expense of violations of the lower ranked constraints.

Second, deletion of a [stop] feature is preferred in Basque to other possible phonological alternations such as changing a stop into a fricative, fusion of the two [stop] features into one, or epenthesis of a vowel to break the sequence, because

MAX-IO and MAX[stop] are lower ranked than DEP[cont], UNIFORMITY[stop], and DEP-IO.

Third, the feature [stop] deletes not in the onset but in the coda in the sequences in Basque due to a higher-ranked positional featural faithfulness constraint on [stop] in the onset, MAXONS[stop].

Fourth, in the sequence of two stops or that of a stop and an affricate, the first segment (stop) deletes. On the other hand, in the sequence of an affricate and a stop, the first segment (affricate) spirantizes into a fricative. This asymmetry of appearance of the phonological alternations is based not upon the difference of the constraint ranking but upon the difference of the manner features between a stop and an affricate. To spirantize, a stop must violate a higher-ranked constraint DEP[cont], while an affricate does not have to. A single invariant ranking can account for this asymmetry.