Volume 10
Issue 2 *Selected Papers from NWAVE 32*

Article 16

2005

Global Faithfulness and Choice of Repair

Jennifer Nycz

Global Faithfulness and Choice of Repair

Global Faithfulness and Choice of Repair

Jennifer Nycz

1 Guy and Boberg's (1997) Observation

In all varieties of English, word-final consonant clusters are sometimes simplified through deletion of a final coronal stop. The likelihood of deletion depends on several factors, including the dialect of the speaker, style of speech, morphological class, stress, and the preceding and following phonological environment. Guy and Boberg (1997)—henceforth "G&B"—analyze the effect of the preceding segment in terms of cumulative OCP violations, construing the OCP as a prohibition on adjacent identical features as in Yip (1988). The relevant features for t/d deletion are [coronal], [continuant], and [sonorant]; the more features a final t/d shares with the preceding segment, the more likely it is that simplification of the offending cluster will occur. An interesting implication of G&B's cumulative analysis of the preceding segment effect is the fact that it unites a variable process and a categorical one. Greater numbers of shared features increase the likelihood of deletion, and the limit of this is categorical absence of t/d after segments sharing all three features—namely, other coronal stops. In this way, the variable process of t/d deletion after non-identical segments, and the categorical avoidance of coronal stops after identical segments both fall out under the same analysis. In the Variable Rule framework (Labov 1969, Cedergren and Sankoff 1974), this is formalized as a difference in rule application probability: in many contexts, the probability of rule application is between 0 and 1, indicating a variable process, but when enough OCP violations accumulate, rule probability reaches 1, meaning categorical avoidance (see Table 1, reproduced from G&B, for an illustration).

An issue which is sidestepped in G&B's account is the fact that there *is* a qualitative difference between the variable and categorical cases: each demands a different repair strategy. A coronal stop which is adjacent to a non-identical segment may delete to avoid featural OCP violations, but a coronal stop which would otherwise end up adjacent to another coronal stop is always saved by epenthesis. How are these different repairs dealt with in the rule model? There is no way to get epenthesis to come out of a deletion rule; instead, we predict that in cases like /weɪt+d/ "*wait+PAST*", where all relevant features are shared by the final coronal and its preceding segment, there should be categorical *deletion*, since rule probability has reached 1 in such a case. Generalizing the rule—to something like "Avoid OCP violations in coda clusters"—is not possible, since a rule must "uniquely determine the structural change in response to a structural condition" (Kager 1999:55). We must therefore posit a separate rule for epenthesis. By doing so, however, we give up a truly unified account of these two processes. There will simply be a categorical epenthesis rule which applies in some cases, and a variable deletion rule that applies elsewhere, and the fact that both are motivated by the OCP will be relegated to the realm of coincidence.

Preceding segment		N	% deletion	Factor Weight
/t,d/	[+cor,-son,-cont]	-	(categorical absence)	1
/s,z,sh,z/	[+cor,-son]	276	49	0.69
/p,b,k,g/	[-son,-cont]	136	37	0.69
/n/	[+cor,-cont]	337	46	0.73
/f,v/	[-son]	45	29	0.55
/l/	[+cor]	182	32	0.45
/m, ng/	[-cont]	9	11	0.33
/r/	?	86	7	0.13
vowels	-	-	(nearly categ. retention)	0

Table 1: Reproduced from Guy and Boberg 1997 (1994 corpus)

2 Toward an OT Analysis, and a Paradox

Optimality Theory (OT) is tailor-made for dealing with the kind of functional unity displayed by the t/d deletion and epenthesis processes. In this theory, statements such as "Avoid OCP violations" are perfectly acceptable, and take the form of markedness constraints. Precise repair strategies are not dictated by markedness constraints themselves, but fall out from the relative ranking of other constraints which enforce faithfulness to the input form. In the case of the English coronal stops, the fact that /tt/ and /st/ coda clusters are both subject to repair will be a result of both structures being issued marks by an OCP constraint, but the choice between deletion and epenthesis will be decided by the ranking of the faithfulness constraints which militate against each of these changes.

For this initial analysis, we will need the following constraints[1]:

[1]The faithfulness constraints used here are standard OT constraints (see e.g. Kager 1999). OCP is also generally used as a markedness constraint. However, as Guy (1997) points out, a single OCP (even one which is gradiently violable) will not be able to capture the cumulative effect that violations have on the likelihood of repair. In order to reflect this cumulativity, we will need to assume an OCP

Faithfulness:	MAX(t)	"Don't delete coronal stops"
	DEP(V)	"Don't insert vowels"
Markedness:	OCP-1 OCP-2 OCP-3	These assign marks to candidates containing sequences of segments sharing 1, 2, or 3 features. Note that violation of OCP-3 implies violation of OCP-2, which implies violation of OCP-1.

Let us first simply try to account for the variable repair facts (i.e. that OCP-2 and OCP-1 violations motivate deletion). In order for the deletion repair to come out over epenthesis, we need the ranking DEP >> MAX ("Epenthesis is worse than deletion"). Since deletion is a variable process, we also need MAX to be variably ranked with respect to OCP-2 and OCP-1. Finally, since epenthesis is never forced by OCP-2 or OCP-1 alone, DEP must outrank both of these constraints. The necessary ranking is illustrated in Tableau 1[2] below, using the underlying form /kɪs+d/ "*kiss*+PAST".

/ kɪs+d/	DEP	OCP-2	MAX	OCP-1
→ kɪst[3]		*!		*
→ kɪs			*!	
kɪsəd	*!			

Tableau 1

To include epenthesis in this analysis, OCP-3 must be added to the tableau. Since repair of OCP-3 violations always occurs, OCP-3 must strictly

subhierarchy (as in Padgett 2002), in which greater OCP violations are ruled out in higher ranked constraints. The subhierarchy is grounded, in the sense that it is based on a scale of "difficulty": OCP-3 violations arguably make production and/or perception more difficult that OCP-2 violations, which themselves create more difficult than OCP-1 violations. In this way, functional unity may be interpreted as resulting not only from violation of a single markedness constraint, but also from violation of a given markedness hierarchy.

[2]In tableaux, solid lines separating constraints indicate that those constraints are strictly ranked with respect to each other. Dotted lines indicate that constraints are mutually unranked. A rightward-pointing finger indicates an attested form, while a leftward-pointing finger indicates an unattested form which is wrongly deemed optimal.

[3]Throughout this paper, we assume an undominated AGREE(voice) constraint which prohibits obstruents in the same coda from differing in [voice].

dominate both faithfulness constraints. In order for epenthesis to be chosen over deletion, it must be the case that MAX outranks DEP. These facts are shown in Tableau 2, for the input form /weɪt+d/.

/ weɪt +d /	OCP-3	MAX	DEP
weɪtt	*!		
weɪt		*!	
→ weɪtəd			*

Tableau 2

However, this ranking of the faithfulness constraints is the opposite of that which was determined to be necessary for the deletion cases. In fact, given the constraints defined above, there is no way to account for both deletion and epenthesis in the same OT analysis. Tableaux 3 through 6 illustrate the problem (OCP-1 is omitted for the sake of space).

/ kɪs+d /	OCP-3	DEP	OCP-2	MAX
→ kɪst			*!	
→ kɪs				*!
kɪsəd		*!		

Tableau 3: The ranking DEP >> MAX correctly yields variable deletion in the *kiss*+PAST case…

/ weɪt+d /	OCP-3	DEP	OCP-2	MAX
weɪtt	*!		*	
→ weɪt				*
(→)weɪtəd		*!		

Tableau 4: …but incorrectly predicts deletion in the *wait*+PAST case.

/ weɪt+d /	OCP-3	OCP-2	MAX	DEP
weɪtt	*!	*!		
weɪt			*!	
→ weɪtəd				*

Tableau 5: MAX >> DEP correctly yields epenthesis in the *wait*+PAST case…

/ kɪs+d /	OCP-3	OCP-2	MAX	DEP
(→)kɪst		*!		
(→)kɪs			*!	
→ kɪsəd				*

Tableau 6: …but incorrectly results in epenthesis here.

Moreover, this problem cannot be dealt with through variable ranking of MAX and DEP, because the choice of repair in each situation is categorical: always epenthesis when three features are shared, always deletion (if any repair) when fewer than three features are shared. If there were variable ranking, then in evaluating the candidates for a form like /weɪt+d/ we could possibly find ourselves in a grammar that mandates deletion instead of epenthesis; likewise, when evaluating the possible outputs for a form like /kɪs+d/, we could end up epenthesizing instead of deleting or doing nothing.

It may help to consider possible functional motivations for each repair. Although epenthesis and deletion can both be used to fulfill a particular function such as cluster simplification, they differ in one key respect: while epenthesis simplifies a cluster and *preserves* an underlying coronal stop, deletion simplifies a cluster and *does not preserve* this segment. We might guess that epenthesis is chosen as a repair when its preserving property is especially needed. Why should epenthesis be required in a form like *waited* but not *kissed*? Given that consonant length is not contrastive within the word in English, a fully faithful form [weɪtt] will be indistinguishable from the form [weɪt] which lacks a second coronal stop. It would be impossible, without epenthesis, for an English speaker to recover the second stop. This same danger is not present in the case of [kɪst]. While simplification may be warranted in both cases, the need for preservation is more dire in the *waited* case, where the final coronal stop is obscured.

So, we hypothesize that epenthesis must occur in potential [tt] clusters in order for the second stop to be recoverable. But for this to happen, we must ensure that deletion does not apply, lest [weɪt] win out over epenthetic [weɪtəd]. This is easily done in a rule-ordering system: epenthesis applies first to /weɪt+d/, yielding the form [weɪtəd]. Since this resulting form no longer has a complex coda cluster, it does not meet the structural requirements of the variable deletion rule, and deletion does not apply. In a parallel OT system, we must remove the possibility of deletion some other way. This calls for some sort of Faithfulness constraint. However, we cannot rely on a traditional MAX constraint, since such a constraint will also rule out the possible deletion form [kɪs].

3 The Proposal: Global Faithfulness

In OT, faithfulness constraints enforce correspondence between elements in two strings. MAX-IO, for example, states that every element in the input should have a correspondent in the output. Correspondence between input and output is important because it increases the likelihood that underlying forms will be recoverable. We might restate our faithfulness constraints, then, in terms of recoverability: MAX-IO thus states that an element in the input form should be recoverable from the output form.

With respect to this constraint, the deletion candidates /weɪt+d/ → [weɪt] and /kɪs+d/ → [kɪs] are equally bad, since in both cases an underlying coronal stop is not recoverable. A difference between the two emerges when we expand our view to consider the unrepaired candidate. In the case of /weɪt+d/, the unrepaired form [weɪtt] does not allow for recoverability of the underlying coronal stop, while unrepaired [kɪst] for /kɪs+d/ does. It is clear that when deciding on a repair, we need to refer to the badness of the unrepaired form with respect to recoverability. However, MAX-IO constraints cannot do this; they can only look at individual candidates, to ensure that elements present in the underlying representations are recoverable from individual output forms.

Yet given the surface variability of language, this is a rather parochial notion of recoverability. Speakers can produce several variants of a particular word, only some of which may actually preserve the underlying contrast. Because of the t/d deletion "rule", for instance, speakers produce both [post] and [pos] for the underlying form /post/. But speakers are not thwarted by the existence of deletion tokens, and know that the coronal stop is part of the underlying form from other tokens that they have heard which do retain the final stop. Thus it seems that recoverability is not always a strict condition on every individual form, but can be a general constraint on total usage.

What if there are other, more global MAX constraints which take into account this variability? These constraints would say, in effect, "Make sure the underlying representation is recoverable from *general usage*." Such constraints could broaden their view beyond single output candidates,[4] and judge potential *alternations* between output forms—specifically, between a given candidate and the unrepaired candidate. This is just the sort of constraint we will propose to account for the different repairs described in this paper. We will call the constraint GLOBAL MAX, and it will take the following form:

> G-MAX(cor): A candidate *x* will be issued a mark by G-Max(cor) if, assuming surface variation between *x* and the unrepaired candidate *y*, a final coronal stop in the UR is never recoverable. If the final coronal stop is recoverable from at least one of {*x*, *y*}, then *x* will not violate G-MAX(cor).

The effect of this constraint will be the following: if the unrepaired form is bad enough with respect to recoverability, a repair that fixes it is required. The action of such a constraint is illustrated in the partial tableaux 7 and 8.

/ weɪt+d /	**G-MAX(cor)**
{[weɪtd], **[weɪtt]**}	*
{[weɪtd], **[weɪt]**}	*
{[weɪtd], **[weɪtəd]**}	√

Tableau 7

In the /weɪt+d/ example, each potential output form (in bold) is considered along with the unrepaired form. If the final coronal stop is not recoverable from either of the two forms, then the potential output form is issued a mark by G-MAX(cor). Since the unrepaired form does not allow for recoverability, the burden is put on the candidate itself; in this case, the only one which improves upon the unrepaired form is the epenthetic candidate. Deletion is ruled out, since while it may simplify the coda and make the output less marked, it does not aid in recoverability.

[4]Such an approach is precedented. In Comparative Markedness theory (McCarthy 2002), a markedness constraint can compare a given candidate to the fully faithful candidate, penalizing old and new markedness violations differently.

/ kɪs+d /	**G-MAX(cor)**
{[kɪsd], **[kɪst]**}	√
{[kɪsd], **[kɪs]**}	√
{[kɪsd], **[kɪsəd]**}	√

Tableau 8

In the /kɪs+d/ example, G-MAX(cor) applied vacuously: since the unrepaired form is already fine from a recoverability standpoint, the candidates do not have to bear the burden. Deletion form [kɪs] is allowed to pass through unstarred, since the unrepaired form [kɪst] already fulfills the constraint.

Tableaux 9 and 10 show how this constraint can be situated in the full analysis to yield the correct repair for each case. In tableau 9, a high-ranking G-MAX constraint rules out deletion for the /weɪt+d/ case, leaving epenthesis as the only repair option. This means that DEP can safely outrank MAX lower in the constraint hierarchy, correctly yielding deletion in the cases which are not affected by G-MAX (Tableau 10).

/ weɪt+d /	OCP-3[5]	G-MAX(cor)	DEP	OCP-2	MAX
weɪtt	*!	*		*	
weɪt		*!			*
→weɪtəd			*		

Tableau 9

/ kɪs+d /	OCP-3	G-Max(cor)	DEP	OCP-2	MAX
(→)kɪst				*	
(→)kɪs					*
kɪsəd			*!		

Tableau 10

[5]Note that in this case, OCP-3 is made redundant by G-Max(cor). Does this mean we don't need such a constraint at all? Not necessarily. Given the qualities of English, it so happens that an OCP-3 violation results in nonrecoverability. However, recoverability of contrasts in various contexts is language specific. In some other language—one which contrasts consonant length, for instance—it may not be the case that an OCP-3 violation for a cluster [tt] entails a G-MAX(t) violation.

This type of analysis can potentially be extended to any case in which violations of the same markedness hierarchy result in different repairs, depending on the severity of violation: less serious cases of markedness may result in one type of simplification, but once the degree of markedness becomes so great that recoverability is threatened, a new repair strategy will be mandated.

4 Quantitative Results

The previous section showed how an analysis using GLOBAL-MAX can derive the correct qualitative results with regards to choice of repair. In this section we will see how the same analysis can also yield the correct quantitative results, when embedded within a stochastic OT framework (Boersma 1997). While regular OT involves a strictly ordinal ranking of constraints, stochastic OT assumes that constraints are given real-number ranking values along a continuum; this enables constraints to be ranked relatively closely or far apart. Moreover, at any given evaluation, noise is added to the resting value of each constraint, such that the actual ranking of a given constraint varies normally over many evaluations. The closer that two constraints are on the continuum, the more likely it is that their relative ranking will be reversed at a particular evaluation. This feature enables stochastic OT to capture the fine-grained quantitative facts of variation.

To accurately reproduce the attested output frequencies discussed in G&B, we will need to expand the constraint set somewhat. First, we need featurally-specific OCP constraints. As shown in Table 1, the three features do not affect t/d deletion to exactly the same extent. So, we need to explode the OCP-2 and OCP-1 constraints to make them refer to specific features, to reflect the fact that, for instance, sharing the two features [coronal] and [continuant] is worse than sharing [coronal] and [sonorant] or [sonorant] and [continuant]. Moreover, we will have to add a NoCoda constraint which generally militates against codas. This constraint is added for completeness; it would cause t/d deletion after vowels. In the dialect discussed by G&B, this constraint will be low-ranked, since there is no t/d deletion after vowels. The complete list of constraints is embedded in an OT Grammar file, readable by the Praat software (Boersma 1997). The annotated contents of this grammar file are included in Appendix A.

This initial state grammar, along with a distribution file containing the token output frequencies from G&B, was fed into the Gradual Learning Algorithm (Boersma 1997) within Praat. After learning was complete, the resulting grammar was then used to generate predicted output frequencies. The results of one typical run (which generated 10,000 tokens per input type) are shown in Table 2, which compares the predicted output frequencies with those attested in G&B. For all cluster types, the predicted frequency of each token type either matches or closely approximates that of the attested data.

cluster	**repair?**	**input frequency (G&B '97)**	**GLA output frequency**	**Predicted token #'s**
t+t	retention	0	0	0
	deletion	0	0	0
	epenthesis	100	100	10000
s+t	retention	51	50	4984
	deletion	49	50	5016
	epenthesis	0	0	0
p+t	retention	63	62	6185
	deletion	37	38	3815
	epenthesis	0	0	0
n+t	retention	54	54	5417
	deletion	46	46	4583
	epenthesis	0	0	0
f+t	retention	71	71	7121
	deletion	29	29	2879
	epenthesis	0	0	0
l+t	retention	68	67	6679
	deletion	32	33	3320
	epenthesis	0	0	0
m+t	retention	89	90	9003
	deletion	11	10	997
	epenthesis	0	0	0
V+t	retention	100	100	10000
	deletion	0	0	0
	epenthesis	0	0	0

Table 2: GLA results

5 Conclusion

In the beginning of this paper, we pointed out a problem presented by the phenomena of English t/d deletion and epenthesis. Though both processes are conceptually united under an OCP analysis, they resist formal unification in both the rules-based and current OT frameworks. The problem is insurmountable in the first case, due to the nature of rules. However, there is nothing about OT which inherently forbids a unified account of these two processes. For this reason, we pursued an OT analysis of the problem, and ultimately posited a new type of constraint, GLOBAL-MAX, which essentially operates by comparing a given candidate to its unrepaired counterpart; if the latter does not allow for recoverability of an underlying contrast, and if the repaired candidate does not improve upon this state of affairs, the repair is deemed insufficient and the candidate cannot surface as an output form.

Using this constraint, we were able to unite the deletion and epenthesis facts in one analysis, capturing G&B's observation that the two processes are functionally related, while recognizing that the type of repair employed depends on the severity of the markedness violation. However, markedness and repairs remain autonomous in the way that OT requires them to be, since specific structural conditions do not directly trigger repairs, but merely rule out repairs that may be insufficient.

We also identified a new class of phenomena: those in which violations of the same markedness hierarchy result in different types of repairs. In the case discussed here, violations confined to the low end of the OCP-hierarchy result in a repair which eradicates an underlying segment, but violations of the higher end of this hierarchy result in a repair which preserves this segment.

After deriving the correct qualitative results regarding choice of repair, we implemented this analysis within a stochastic OT grammar and reproduced the token output frequencies reported by G&B.

Appendix A Initial State Grammar

! A Grammar which uses Global Max plus featural OCP constraints to derive correct repair strategies
! All constraints start out mutually unranked, i.e. all at value 100

11 constraints
constraint [1]: "O\s{CP-3}" 100 100 !
constraint [2]: "O\s{CP-2-cor-son}" 100 100 !
constraint [3]: "O\s{CP-2-son-cont}" 100 100 !
constraint [4]: "O\s{CP-2-cor-cont}" 100 100 !
constraint [5]: "O\s{CP-1-cor}" 100 100 !
constraint [6]: "O\s{CP-1-son}" 100 100 !
constraint [7]: "O\s{CP-1-cont}" 100 100 !
constraint [8]: "G\s{-MAX}" 100 100 !
constraint [9]: "M\s{AX}" 100 100 !
constraint [10]: "D\s{EP}" 100 100 !

constraint [11]: "N\s{OCODA}" 100 100 !

0 fixed rankings

8 tableaux
input [1]: "t+t" 3
  candidate [1]: "tt" 1 1 1 1 1 1 1 1 0 0 1
  candidate [2]: "t-" 0 0 0 0 0 0 0 1 1 0 1
  candidate [3]: "tet" 0 0 0 0 0 0 0 0 0 1 1
input [2]: "s+t" 3
  candidate [1]: "st" 0 1 0 0 1 1 0 0 0 0 1
  candidate [2]: "s-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "set" 0 0 0 0 0 0 0 0 0 1 1
input [3]: "p+t" 3
  candidate [1]: "pt" 0 0 1 0 0 1 1 0 0 0 1
  candidate [2]: "p-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "pet" 0 0 0 0 0 0 0 0 0 1 1
input [4]: "n+t" 3
  candidate [1]: "nt" 0 0 0 1 1 0 1 0 0 0 1
  candidate [2]: "n-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "net" 0 0 0 0 0 0 0 0 0 1 1
input [5]: "f+t" 3
  candidate [1]: "ft" 0 0 0 0 0 1 0 0 0 0 1
  candidate [2]: "f-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "fet" 0 0 0 0 0 0 0 0 0 1 1
input [6]: "l+t" 3
  candidate [1]: "lt" 0 0 0 0 1 0 0 0 0 0 1
  candidate [2]: "l-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "let" 0 0 0 0 0 0 0 0 0 1 1
input [7]: "m+t" 3
  candidate [1]: "mt" 0 0 0 0 0 0 1 0 0 0 1
  candidate [2]: "m-" 0 0 0 0 0 0 0 0 1 0 1
  candidate [3]: "met" 0 0 0 0 0 0 0 0 0 1 1
input [8]: "V+t" 3
  candidate [1]: "Vt" 0 0 0 0 0 0 0 0 0 0 1
  candidate [2]: "V-" 0 0 0 0 0 0 0 0 1 0 0
  candidate [3]: "Vet" 0 0 0 0 0 0 0 0 0 1 1

Appendix B Constraint rankings after learning (typical)

G-Max	122.961
DEP	111.757
OCP-3	105.350
MAX	98.494
OCP-1(cor)	97.038
OCP-1(son)	97.009
OCP-2(cor, cont)	96.771
OCP-2(cor, son)	95.840
OCP-1(cont)	95.033
OCP-2(son, cont)	94.949
NoCoda	82.878

In the continuous ranking system, the smaller the distance between two constraints A and B, the greater the likelihood that A and B will vary in their respective ranking. For instance, MAX and OCP-1(cor) have a high likelihood of switching their ranking, while G-Max and DEP are, for all practical purposes, strictly ranked with respect to one another.

References

Boersma, Paul. 1997. How we learn variation, optionality, and probability. *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* 21:43-58.

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 31(1):45-86.

Cedergren, Henrietta and David Sankoff. 1974. Variable rules: performance as a statistical reflection of competence. *Language* 50:333-355.

Guy, Gregory. 1980. Variation in the group and the individual: the case of final stop deletion. In *Locating language in time and space*, ed. William Labov, 1-36. New York: Academic Press.

Guy, Gregory. 1997. Violable is variable: Optimality Theory and linguistic variation. *Language Variation and Change* 9:333-347.

Guy, Gregory and Charles Boberg. 1997. Inherent variability and the Obligatory Contour Principle. *Language Variation and Change* 9:149-164.

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

Kisseberth, Charles. 1970. On the functional unity of phonological rules. *Linguistic Inquiry* 1:291-306.

Labov, William. 1969. Contraction, deletion, and inherent variability of the English copula. *Language* 45:715-762.

McCarthy, John. 2002. Comparative markedness. http://roa.rutgers.edu/view.php3?id=745.
McCarthy, John and Alan Prince. 1995. Faithfulness and reduplicative identity. http://roa.rutgers.edu/view.php3?roa=60
Padgett, Jaye. 2002. Constraint conjunction versus grounded constraint subhierarchies in Optimality Theory. http://roa.rutgers.edu/view.php3?id=731.
Prince, Alan and Paul Smolensky. 1993. *Optimality Theory: constraint interaction in generative grammar.* Ms., Rutgers University and the University of Colorado. Also http://roa.rutgers.edu/view.php3?roa=537.
Yip, Moira. 1988. The Obligatory Contour Principle and phonological rules: A loss of identity. *Linguistic Inquiry* 19:65-100.

Department of Linguistics
New York University
719 Broadway, 4th Floor
New York, NY 10003-6806
jrn218@nyu.edu