Input “Clusters” and Contrast Preservation in OT

Anne-Michelle Tessier
University of Massachusetts Amherst

1. Introduction

The starting point of this paper is the Contrast Preservation Theory (PCT) of Lubowicz (2002), and its novel approach to phonological opacity in OT. A core claim of PCT is that various opaque mappings, which have been problematic in OT, can be readily explained as the effect of systemic contrast preservation (CP). Thus, PCT is an OT model which evaluates not only the Markedness and Faitfhfulness of its candidates, but also their preservation of phonological contrasts. To do so, the EVAL component of PC theory assesses *multiple* input forms, in an input *scenario*.

The goal of this paper is to provide an alternative to scenarios, which still captures the contrast-preserving patterns suggested by PCT. My alternative is a grammar-based algorithm that builds finite, language-specific sets of input forms called input *clusters*. In building clusters, the algorithm relies crucially on the existing core of OT: the language-specific ranking of Markedness and Faithfulness constraints, and the decision-making powers of EVAL. Working loosely within the framework of PCT, I use the algorithm and its resulting clusters to analyze a derived environment effect (one opaque pattern explained under PCT). The success of this analysis provides initial support for the algorithm, and for the broader claim that such an algorithm’s clusters will contain all the input forms necessary to capture contrast-preserving opacity.

The paper is organized as follows. Section 2 provides some minimal theoretical background, on PCT and its notion of input scenarios. In section 3, I propose my cluster-building algorithm. Section 4 introduces the derived environment effect (DEE), and uses data from a Campidanian Sardinian

* Thanks especially to John McCarthy and Joe Pater for many challenges and improvements, and also to Kathryn Flack, Maria Gouskova, John Kingston, Ania Lubowicz, Naz Merchant, Jaye Padgett, Alan Prince, Lisa Selkirk, Bruce Tesar, and the audiences at WCCFL 23 and HUMDRUM 2004. All progress not yet made remains my sole responsibility. Comments, questions and kindly criticism are welcome: tessier@linguist.umass.edu.

DEE to demonstrate how the algorithm builds clusters. Section 5 puts those clusters to work in the analysis of Campidanian Sardinian, using a contrast-preserving constraint based on those of Lubowicz (2002). The last section summarizes the results, and raises questions for future work.

2. Background: Contrast Preservation Theory and input scenarios

A core claim of PCT is that opaque mappings such as chain shifts and derived environment effects (DEEs) are driven by an explicit grammatical pressure to preserve contrasts. (This notion of protecting contrast across multiple mappings in a language is also central to recent OT models of Dispersion Theory (Flemming, 2001): see Padgett (2003a,b), Bradley (2001) and others.) To implement this pressure with OT constraints, any such theory requires a revised notion of input – one which includes *multiple* input strings and corresponding outputs within a single evaluation. This revision is necessary because knowing whether a contrast has been preserved requires the comparison of multiple input-output pairs in the same language. To determine for example whether /t/ and /d/ merge, we must be able to compare two mappings like those in (1) below, and check whether their outputs are identical:

(1)	*Inputs*		*Outputs*	*The contrast-preserving question*
	/pa**t**/	→	[??]	**Are these outputs identical?**
	/pa**d**/	→	[??]	

Note that this type of contrast preservation is *not determined by the lexicon* i.e. it is not homophony avoidance. In this model, phonological contrasts are preserved or merged regardless of whether or not /pat/ and /pad/ are real (or even *possible*) words of the language.

In Lubowicz's (2002) statement of PCT, the input to EVAL is an input scenario[1]. The procedure for building a scenario takes a single input string, which I will refer to as the *base* input form (or *base*), and builds a very large, language-independent scenario of other input strings, with the aim of detecting all the possible mergers that might drive any contrast-preserving mappings. To see this procedure at work using the hypothetical base /pat/, consider Lubowicz's illustration in (2), which provides a descriptive template for all the forms in /pat/'s scenario:

(2) __ _p_ __ _a_ __ _t_ __

[1] Thanks to John McCarthy for insight into this section.

As (2) shows, scenario-building is roughly an operation that fills each one of these 'slots' with any possible segment, or leaves it blank.

Input scenarios are 'contrast catch-alls'. Created by a very general formula (similar in scale to GEN[2]), the scenario throws its net so wide as to include every form whose contrast might be preserved in some language. The only restriction on scenarios is one which limits the template in (2) to 7 slots (and in general, to 2n+1 slots for any base input string of n segments). This stipulation creates an upper bound on the amount of epenthesis, relative to the base, in each of the scenario's forms. This restriction on epenthesis is the reason that input scenarios are finite: without it, the scenario would contain an infinite number of strings (including /pat/, /pata/, /pataa/, /pataaa/, /pataaaa/, etc.) As Lubowicz notes, this guarantee of finiteness is required in PCT (beyond any computational or aesthetic concerns) to assess the theory's contrast-preserving constraints.

Given its riches of input forms, input scenarios clearly must contain more forms than the attested patterns of CP require – so many more so that they must be arbitrarily restrained from being infinite. However, many of the unnecessary forms in an input scenario can be ruled out by OT machinery already in place: both the language-specific rankings that constitute the grammar itself, and the general OT principles of evalution, which derive economy of epenthesis. Even in an OT with explicit contrast preservation, Markedness and Faithfulness are still powerful, in determining both which mergers can be at issue and which forms will successfully report them.

These observations suggest an OT-grammatical alternative to the scenario, and they form the basis of the proposal I spell out below.

3. Proposal: Replacing scenarios with grammar-based *clusters*

The cluster-building algorithm starts with a base input form, /base/, and a language-specific ranking H_{lang}. The algorithm's core has three steps; finding a complete input cluster requires taking every markedness constraint *X in CON, and running the same three steps for each *X. I provide the full algorithm below in (3), which I will then exemplify step-by-step:

---

[2]Compared to the candidate set created by the GEN of 'classic' OT (Prince and Smolensky, 1993), input scenarios are smaller because they do not have correspondence relations or unbounded epenthesis.

(3) *The cluster-building algorithm*

> To find an input cluster, given a /base/ and a ranking of constraints H_{lang}, repeat the following three steps for each markedness constraint *X:
>
> 1. Ignoring all markedness constraints except *X, run /base/ through the grammar and find its winning output [B].
>
> 2. If [B] is not a fully-faithful candidate, include [B] in the cluster.
>
> 3. If [B] is a fully-faithful candidate, inspect the violations of *step 1's losing candidates*, and include as neighbors all the forms which:
>   (a) violate *X exactly once and
>   (b) are otherwise most harmonic (i.e. most faithful)

The resulting input cluster has one base, whose optimal output is being chosen, and a set of *neighbors* – the forms chosen in steps 2 and 3 – whose potential to merge may influence the choice of optimal base output.

3.1 A first walk through the algorithm

The role of contrast preservation in PCT starts with Markedness. High-ranking Markedness (indirectly) compels mergers across multiple mappings, which CP constraints may counteract. To see how this algorithm builds clusters that detect such mergers, we can start with a simple ranking:

(4) *VoicedObstruent >> Ident[voice]

This ranking of M >> F rules out a marked structure (voiced obstruents) and thereby neutralizes a contrast (the contrast between obstruents which differ only in voicing, like /d/ and /t/).

Let us see how to determine the input clusters for two bases – /pad/ and /pat/ – given the ranking in (4). We begin with just the bases:

(5) <u>Starting point:</u>
- Cluster for /pat/: /pat/ $_{base}$
- Cluster for /pad/: /pad/ $_{base}$

*Step 1: Ignoring all markedness constraints except *VoicedObstruent, run the base through the grammar and find its winning output*

(6a)

/pat/	*VcdObs.	Id[vce]
☞ pat		
pad	*!	*

(6b)

/pad/	*VcdObs.	Id[vce]
☞ pat		*
pad	*!	

(7) Results of Step 1:
/pat/ → [pat] (faithful)
/pad/ → [pat] (unfaithful)

The tableau that results from this step 1 mapping serves as the basis for the next two steps, both of which choose neighbor(s) for the cluster.

*Step 2: **If the base did not map a the fully-faithful candidate in step 1**, include the winning output as a neighbor in the cluster*

This step is relevant to base /pad/, since its optimal output [pat] is not fully faithful to input voicing, so we add it to the cluster:

(8) After Step 2:
- Cluster for /pat/: /pat/ $_{base}$
- Cluster for /pad/: /pad/ $_{base}$ **/pat/**$_{neighbor}$

*Step 3: **If the base did map a the fully-faithful candidate in step 1**, inspect the violations of step 1's losing candidates, and include as neighbors all the forms which (a) violate *X exactly once and (b) are otherwise most harmonic (i.e. most faithful)*

This step is relevant to base /pat/, since its optimal output [pat] is indeed a fully-faithful candidate. So, we look back at the losers from Step 1:

(6b)

/pat/	*VcdObs.	Id[vce]
☞ pat		
pad	*!	*

Of all the losing candidates, the one which satisfies both criteria is [pad], since it violates *VcdObs. exactly once, and is otherwise as harmonic as possible. (As we will see soon: a more realistic tableau, which contains various losing candidates and multiple faithfulness constraints, will make these selection criteria less vacuous.) For now, we can simply add this loser [pad] to /pat/'s cluster:

(9) Final results, after Step 3:
- Cluster for /pat/: /pat/ $_{\text{base}}$ **/pad/** $_{\text{neighbor}}$
- Cluster for /pad/: /pad/ $_{\text{base}}$ /pat/ $_{\text{neighbor}}$

In a real grammar, these three steps are run for every markedness constraint, and all resulting neighbors are included in the final cluster.

3.2 A little analysis of the algorithm and its clusters

Given the ranking in (4) and either of the above forms as a base, (9) shows that the algorithm can detect the possible merger and adds the other form as a neighbor. This is our first hint that the algorithm does what it ought.

With this toy grammar of only two constraints, the resulting cluster is very small – one base, and one neighbor. Given the discussion of finiteness with respect to input scenarios earlier, it is important to point out that this result is general; that is, input clusters built by this algorithm are guaranteed to be finite. Any proof of why is beyond the scope of this paper, but roughly: for each of the (finite number of) *X in CON, the cluster will contain at most the (finite number of) forms which contain *single* violations of **X* and are otherwise maximally faithful to the base, and summing finite sets of finite sets will necessarily yield a finite final set.

In the following two sections, I demonstrate how input clusters allow for a simple contrast-preserving analysis of derived environment effects (DEEs). There we will see that it is step 2 of the algorithm that provides the necessary neighbors for a DEE. Due to space constraints, the real purpose of step 3 cannot be addressed in this paper, but see Tessier (in prep.) for its crucial role in driving another opaque mapping, the chain shift.[3]

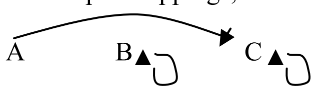
4. Derived Environment Effects and their input clusters

4.1 Anatomy of a DEE, in contrast-preserving terms

A generalized version of the Derived Environment Effect (Mascaro 1976; Lubowicz 1999, 2002) is given in (10) below. It involves three similar inputs, schematised here as /A/, /B/ and /C/, whose relative position in the diagram indicates the featural 'distance' (or number of faithfulness violations) between them. Thus: /A/ and /B/ differ in one feature; /B/ and

[3] See Lubowicz (2002)'s account of Finnish vowel chain shifts as contrast preservation, whose effects I also aim to derive with the present algorithm.

/C/ differ in another; /A/ and /C/ differ on both. (The next section will provide a concrete example.) The picture in (10) shows these three inputs and their output mappings, indicated by arrows as in PCT.

(10) 

/A/ → [C] due to *A, *B >> Faith
/B/ → [B] due to ???
/C/ → [C] due to Faith

In this analysis, the DEE pattern begins with two Markedness constraints, *A and *B, which are both ranked high enough to demand merger of all three inputs. While /A/ obeys markedness, /B/ mysteriously resists mapping to [C]. In the CP account, it is to preserve the contrast between /B/ and /C/ that marked /B/ is unexpectedly faithful[4].

For this analysis to be sucessful in the current model: what clusters must our algorithm build? According to (10), choosing the optimal outputs for /A/ and /C/ does not require any input form other than the base: /A/ is unfaithful for pure markedness reasons, and /C/ is faithful because there's no reason not to be. Thus, it is only base /B/ that requires a neighbor to be mapped correctly. Given the account above, /B/'s necessary neigbor is /C/ - that is, /C/ must appear in base /B/'s input cluster so that /B/ can "see" the danger of merging with /C/.

4.2 The Campidanian Sardinian DEE, in contrast-preserving terms

The DEE example I focus on here comes from Campidanian Sardinian (Bolognesi 1998; Lubowicz 1999). The data in (11a) show the expected /A/ to [C] mapping in Campidinian Sardinian: underlying voiceless stops become both voiced and lenited post-vocalically[5]. However, (11b) shows /B/ mapping faithfully to [B]: underlying voiced stops in the same position do not lenite[6]:

(11a) /A/ → [C]

[**p**]iːʃi	'fish'	bɛːlu[**β**]iːʃi	'nice fish'
[**t**]rintadus	'thiry-two'	sːu[**ð**]rintadus	'(the) thirty-two'
[**k**]uatro	'four'	dɛ [**ɣ**]uatro	'of four ...'

---

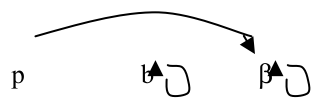
[4] The constraint that preserves this contrast will be introduced in §5.1.
[5] The voiceless affricate [tʃ] also undergoes voicing and lenition to [ʒ].
[6] Input /b/ is also reported to optionally delete.

(11b) /B/ → [B]

[**b**]ia	'road'	s:a[**b**]ia	'the road'
[**d**]ɔminiɣu	'Sunday'	don:ja[**d**]ɔminiɣu	'every Sunday'
[**g**]atu	'cat'	s:a[**g**]atu	'the cat'

Focusing just on the labial segments, the DEE mapping is as in (12):

(12)



In this account, the grammar of Campidanian Sardinian includes the ranking in (13):

(13) *V[-voice] >> *V[-cont.] >> Ident[voice], Ident[cont]

This ranking has two high-ranking markedness constraints, *V[-voice] and *V[-cont], which can both affect post-vocalic consonants (driving voicing and lenition, respectively.) As we know, this ranking creates the right environment for a DEE: the conspiring markedness constraints drive /p/ → [β], but the CP pressure to preserve the /b/ vs /β/ contrast keeps the /b/ → [b] mapping exceptionally faithful.

We saw the DEE cluster requirement in the previous section: that a base /B/ must contain /C/ in its cluster. The mapping in (12) shows us that for Campidanian Sardinian, this means that base /b/ must contain /β/ in its cluster. So to check whether the algorithm succeeds at its task, we will construct a cluster for the labial base from (11b), /s:a bia/.

4.3 Building the input cluster for base /s:a bia/

(14) Starting point:

- Cluster for /s:a bia/: /s:a bia/ $_{base}$

*Step 1. Ignoring all markedness constraints except *X, run /base/ through the grammar and find its winning output.*

Since we have two *X constraints, we run this step twice.

(15a) Step 1, *X = *V[-voice]:

/s:a bia /	*V [-vce]	Id [vce]	Id [cont]
s:apia	*!	*	
☞s:abia			
s:aβia			*!

(15b) Step 1, *X = *V[-cont.]:

/s:a bia /	*V [-cont]	Id [vce]	Id [cont]
s:apia	*!	*	
s:abia	*!		
☞s:aβia			*

(16) Results of Step 1:
- For *V[-voice]: /s:a bia/ → [s:abia] (faithful)
- For *V[-cont.]: /s:a bia/ → [s:aβia] (unfaithful)

Step 2: If the base did not map to a fully-faithful candidate in step 1), include step 1's winning output as a neighbor in the base's cluster

This step applies to the mapping in (15b), since*V[-cont.] forced lenition, so we add its winning candidate to the cluster:

(17) After Step 2:
- Cluster for /s:a bia/ /s:abia/$_{base}$ **/s:aβia/**$_{neighbor}$

*Step 3: If the base did map to a fully-faithful candidate in step 1), inspect the violations of step 1's losing candidates, and include as neighbors all the forms which: (a) violate *X exactly once and (b) are otherwise most harmonic (i.e. most faithful)*

This step applies to (15a) – since the base's post-vocalic stop is voiced, *V[-voice] doesn't prefer any unfaithfulness, and the fully faithful candidate wins. So now we examine an expanded version of (15a), with more of the losers that *V[-voice] ruled out, using the Step 3 criteria:

(18)

/s:abia/	*V [-vce]	Id [vce]	Id[cont]	*Commentary*
s:apia	*!	*		***the neighbor***
s:apiap	*!*	*		violates *V[-vce] twice
s:aɸia	*!	*	*	also violates Ident[cont]
☞s:abia				does not violate *V[-vce]
s:aβia			*!	does not violate *V[-vce]

(19) After step 3 – final result
- Cluster for /s:a bia/: **/s:a pia** /$_{neighbor}$ /s:a bia/$_{base}$ /s:a βia/$_{neighbor}$

We now know that given the ranking in (13), a base with a post-vocalic /b/ will return a cluster including neighbors with post-vocalic /p/ and /β/. Since our DEE cluster requirement was that base /b/ include neighbor /β/, we can happily conclude that the algorithm has succeeded in its task, and proceed to use this cluster in the analysis.

5. Using input clusters to analyze Campidanian Sardinian

5.1 A contrast-preserving constraint: PRESERVE(F)

The remaining ingredient to this contrast-preserving account is a contrast-preserving constraint. The constraint schema I have chosen to use, Preserve(F), is formulated in (20) below, modeled after PCT's PC(Input) constraints[7]:

(20) PRESERVE(FEATURE)

For each pair of forms /X/ and /Y/ in the input cluster that are identical except that they contrast for the feature F at one locus, assign a violation mark to any candidate in which /X/ and /Y/ map onto the same output form.

"If otherwise-identical inputs contrast for F, they must map to distinct outputs"

In the current case, Preserve(F)'s role will be to preserve contrasts between input forms with /b/ vs. /β/. Since these differ only in continuancy, the version of Preserve(F) we need is:

(21) PRESERVE (CONTINUANT)

"If otherwise-identical inputs contrast for continuancy, they must map to distinct outputs"

When presented with the input cluster created in (19), Preserve(cont.) will penalize the merger of inputs /aba/ and /aβa/ to the same output form, as in candidate (22ii):

(22)

/apa/$_{N1}$ /a**b**a/$_{B}$ /a**β**a/$_{N2}$	Preserve (cont)	*V[-cont]
. (i) apa$_{N1}$ a**b**a$_{B}$ a**β**a$_{N2}$		*(p) *(b)
(ii) apa$_{N1}$ a**β**a$_{B}$ a**β**a$_{N2}$	*	*(p)

The table above shows how output candidates are to be evaluated in this model. The output candidates in (22) are created with the basic GEN machinery of standard OT: each is a set of output forms with the usual

[7] This definition is simplified for present purposes. One of the necessary revisions deals with input forms that have more than one instance of the feature (F); such a revision makes Preserve(F) more complicated but not incoherent.

correspondence relations (indicated here with numerical subscripts, as well as the tags B(ase) and N(eighbor)). Constraint violations are accumulated across ALL forms in a candidate, and summed in each cell.[8]

5.2 The Campidanian Sardinian mappings

5.2.1 The DEE mapping: base /s:a bia /

As we have seen, choosing the base with post-vocalic /b/ must include reference to Preserve(cont), meaning we will need to consider the full cluster in our tableau.

Our first question is where to rank Preserve(cont). Since in the winning candidate Preserve drives exceptional *faithfulness,* we know it must rank above the *markedness* pressure that would otherwise require unfaithfulness. The table in (22) already provides the ranking argument, for Preserve(F) >> *V[-cont], and tableau (23) shows that this result holds in the full ranking. (To begin simply, (23)'s input has only base /b/ and relevant neighbor /β/):

(23) /B/ → [B]: the work of Preserve(cont) >> *V[-cont]

/sa bia/$_B$ /sa βia/$_{N2}$	*V [-vce]	Preserve (cont)	*V [-cont]	Id [vce]	Id [cont]
☞ (i) sa bia$_B$ sa βia$_{N2}$			*		
. (ii) sa βia$_B$ sa βia$_{N2}$		*!			*

Now we can consider the whole cluster, and the mappings for all three input forms. Three of the more promising candidates are shown below in tableau (24). High-ranking markedness rules out any post-vocalic /p/ (e.g. fully-faithful candidate i); Preserve(cont) rules out the merger of the base's /b/ with /β/ (candidate ii). The winning candidate (iii) obeys markedness, except where Preserve overrides it and protects /b/:

(24) /B/ → [B]: The full DEE effect

/sa **p**ia/$_{N1}$ /sa **b**ia/$_B$ /sa βia/$_{N2}$	*V [-vce]	Preserve (cont)	*V [-cont]	Id[vce], [cont]
(i) sa **p**ia$_{N1}$ sa **b**ia$_B$ sa βia$_{N2}$	*!		**	
. (ii) sa βia$_{N1}$ sa βia$_B$ sa βia$_{N2}$		*!		*
. ☞ (iii) saβia$_{N1}$ sa **b**ia$_B$ sa βia$_{N2}$			*	**

[8]It should be noted that these assumptions are not all shared by Lubowicz's (2002) PCT model – in particular, that her output candidates are mappings of the input scenario onto itself, with GEN providing only the differing correspondence relations. See Lubowicz (2002), especially chapter 2, for the details.

5.2.2 The pure markedness mapping: base /bɛ:lu **p**i:ʃi/

For a base with post-vocalic p, like /bɛ:lu **p**i:ʃi/, markedness alone explains the fell-swoop mapping to [bɛ:lu **β**i:ʃi], so no reference to the input cluster is necessary. That is: while in such a model, we assume that all of the input cluster's forms are always present when submitting a base to EVAL, with this ranking those neighbors will not affect base /p/'s mapping. (What ensures this result is that no Preserve(F) constraints referring to /p/'s merger with [β] rank high enough to have any effect.[9])

(25) /A/ → [C]: markedness alone

/bɛ:lu **p**i:ʃi/	*V [-vce]	Preserve (cont)	*V [-cont]	Id[vce], [cont]
(i) bɛ:lu**p**i:ʃi	*!		*	
(ii) bɛ:lu**b**i:ʃi			*!	*
☞ (iii)bɛ:lu**β**i:ʃi				**

5.2.3 The pure faithfulness mapping: base /aβa/

According to our DEE schema from §4.1, we know that base /C/ -- i.e. a base with post-vocalic /β/ -- should not be affected by any neighbors in the input cluster, since faithfulness alone determines its winner. There is now the danger that Preserve could undo this result, preserving the /b/~/β/ contrast by affecting the *latter* and not the former, as in (26ii) below[10]. To prevent this result, we must rank faithfulness to all of /β/'s properties above *V[-cont], the constraint that the winning candidate violates:

(26) /C/ → [C]: high-ranking faithfulness alone

$/aba/_N$ $/aβa/_B$	Preserve (cont)	"Faith β"	*V [-cont]	Id[vce], [cont]
☞ (i) aba_N $aβa_B$			*	
(ii) $aβa_N$ $aʔa_B$		*!		*

6. Summary and Open Questions

In this paper, I presented a method for providing a contrast-preserving brand of OT, like Lubowicz's Preserve Contrast Theory, with multiple

[9] In fact, since /p/ and /β/ differ for more than one feature – voicing as well as continuancy – the present CP constraint will *never* prevent this merger.
[10] Due to space constraints, I will merely state here that step 3 of the algorithm will ensure that /aba/ is included as a neighbor for base /aβa/.

input forms in a single evaluation. The proposed algorithm uses language-specific rankings to detect potential mergers with an input base and build a input clusters of forms, which can then be assessed using a standard CON augmented with contrast preserving constraints. To demonstrate the proposal, I analyzed a derived environment effect in Campidanian Sardinian, through the combination of input clusters and constraint ranking.

The ultimate test of the proposal is mostly an empirical one: does the algorithm build clusters with all the necessary neighbors? Of course, the answer to this question depends on which patterns are determined to require a CP analysis. Using PCT as our guide, another pattern which our clusters should predict is the chain shift (Kiparsky, 1973; Lubowicz 2002). The present algorithm was indeed designed to build clusters that can derive chain shifts; proving that it succeeds is a crucial next step in this work.

Even among DEEs, however, more complicated patterns than the Campidanian Sardinian example already raise questions about the algorithm's ability to provide all the necessary neighbors. For example: how does the cluster provide neighbors that prevent merger of forms with *multiple* violations of *X? (Consider the base input /a**b**a**b**a/, which should be prevented from merging with not only /a**p**a**b**a/ and /a**b**a**p**a/, but also /a**p**a**p**a/?) Another question is how we can drive processes initiated by the interaction of *multiple Markedness constraints*, since step 1 of the current algorithm can only see the effect of one *X constraint at a time? Further refinements of the algorithm will determine the extent to which the initial success of the algorithm can be extended to cover more complicated patterns of opacity. In the meantime, the current proposal offers one sketch of how the analytic insights of PCT might be implemented in a more standard OT model.

7. References

Bolognesi, Roberto (1998). *The Phonology of Campidanian Sardinian: A Unitary Account of Self-Organizing Structure.* Ph.D. dissertation, HIL. [Dordrecht: ICG Printing]

Bradley, Travis (2001). A Typology of Rhotic Duration Contrast and Neutralization. In M..J. Kim & U. Strauss (eds.) *Proceedings of NELS 31.* Amherst, MA: GLSA.

Flemming, Edward (2001). Contrast and perceptual distinctiveness. In B.Hayes, R.Kirchner & D. Steriade (eds.) *Phonetically-Based Phonology.* Cambridge: CUP.

Kiparsky, Paul (1973). Abstractness, opacity, and global rules. In O. Fujimora (ed.) *Three Dimensions in Linguistic Theory* , 57-86. Tokyo: TEC.

Lubowicz, Ania (1999). Derived Environments Effects in Optimality Theory. *Lingua* 112, 243-280.

Lubowizc, Ania (2002). *Contrast Preservation in Phonological Mappings.* Ph.D. dissertation, University of Massachusetts Amherst. Amherst: GLSA Publications.

Mascaro, Joan (1976). *Catalan Phonology and the Phonological Cycle.* Ph.D. dissertation. MIT.

Padgett, Jaye (2003a). Contrast and post-velar fronting in Russian. In *Natural Language and Linguistic Theory 21* pp. 39-87.

Padgett, Jaye (2003b). Systemic contrast and Catalan rhotics. Ms., University of California at Santa Cruz.

Prince, Alan & Paul Smolensky (1993/2001). *Optimality Theory: Constraint Interaction in Generative Grammar.* Ms. Rutgers University, New Brunswick, and University of Colorado, Boulder. [To appear, Cambridge, MA: MIT Press]

Tessier, Anne-Michelle (in preparation). Using Input Clusters to Preserve Contrast in OT: Chain Shifts and Derived Environment Effects. Ms., University of Massachusetts Amherst.