

## CHAPTER 4

### PREDICTIONS OF THE THEORY

#### 4.1 Introduction

Chain shifts are problematic for standard OT. Consider Finnish. In Finnish, underlying short low vowels undergo rounding but underlying long low vowels do not round. The standard interaction of markedness and faithfulness constraints does not admit chain shifts. If markedness outranks faithfulness, we expect every segment to undergo a process, whether underlying or derived. Thus, under this ranking in Finnish, both segments should undergo rounding. If faithfulness outranks markedness, on the other hand, we predict that none of the segments will undergo the process. In this case in Finnish, none of the segments should round.

There have been several approaches to chain shifts developed in OT so far, among them the local conjunction (LC) approach of Kirchner (1996), the scale approach of Gnanadesikan (1997), and the Sympathy Theory of McCarthy (1999). This last approach is a general approach to opaque interactions in OT, of which chain shifts are an example. All of those approaches propose special faithfulness constraints to handle chain shifts (see section 4.2 for discussion).

PC theory proposes a different explanation for chain shift mappings. The key observation is that chain shifts involve *contrast transformation*: some underlying contrast is preserved on the surface in a different form than underlyingly at the expense of neutralizing some other contrast. To capture this observation, PC theory proposes that contrast preservation exists as an imperative in a phonological system, formalized as constraints on preserving contrast. Given its architecture, PC theory makes different

predictions than previous OT approaches with respect to chain shifts. This chapter compares the predictions of PC theory with those of previous approaches.

I will show that, unlike previous OT approaches to chain shifts, PC theory predicts there exist shifts with a high-ranked markedness constraint against the initial stage in the shift, but no high-ranked markedness against the intermediate stage – push shifts (section 4.3.1). But, like other OT approaches, PC theory rules out shifts with no high-ranked markedness against the initial stage – pull shifts (section 4.3.2), and shifts with no termination point – circular shifts (section 4.3.3).<sup>1</sup>

## 4.2 Previous Approaches to Chain Shifts in OT

To account for chain shifts, Kirchner (1996) and Gnanadesikan (1997) propose special faithfulness constraints. Kirchner postulates locally-conjoined (Smolensky 1995) faithfulness constraints. Gnanadesikan formulates IDENT-ADJACENT type constraints in addition to classic IDENT-type constraints.

Let us look at Finnish under the two approaches. In both approaches, shortening and rounding are forced by high-ranking markedness constraints. The relevant rankings are given below.

(1)	<u>Ranking</u>	<u>Consequence</u>
	* $\sigma_{\mu\mu\mu}$ >> IDENT(long)	Shortening takes place.
	*ai >> IDENT(round)	Rounding takes place.

Shortening takes place to avoid tri-moraic syllables. Rounding avoids diphthongs of type [ai]. Those two processes are illustrated below.

---

<sup>1</sup> “High-ranked” markedness refers to markedness constraints ranked higher than conflicting PC constraints (and conflicting markedness constraints).

## (2a) Shortening

	/aai/	* $\sigma_{\mu\mu\mu}$	IDENT(long)
☞	ai		*
	aai	*!	

## (2b) Rounding

	/ai/	*ai	IDENT(round)
☞	oi		*
	ai	*!	

In (2a), the candidate with shortening, candidate [ai], wins. It satisfies high-ranked markedness \* $\sigma_{\mu\mu\mu}$ . Similarly, in (2b), the candidate with rounding, candidate [oi], is optimal. It satisfies high-ranked markedness \*ai.

But those rankings predict that underlying long low vowels should undergo rounding as well to satisfy high-ranked markedness against [ai]. This is illustrated in the following tableau.

## (3) Rounding of underlying long vowels – wrong result

	/aai/	* $\sigma_{\mu\mu\mu}$	*ai	IDENT(long)	IDENT(round)
☞	oi			*	*
☞	ai		*!	*	
	aai	*!			

Candidate [oi] wins as it satisfies both high-ranked markedness constraints. This is the wrong result for Finnish. In Finnish, the optimal form is the one with no rounding, [ai], indicated here with the forward-pointing hand, and not [oi], as predicted by the current constraint ranking.

To achieve the chain shift effect, in Finnish (/aai/→[ai], /ai/→[oi]), we need to block rounding from applying to underlying long vowels (/aai/→[ai],\*[oi]) but ensure it applies to underlying short vowels (/ai/→[oi],\*[ai]).

In both Kirchner's and Gnanadesikan's proposals, this is the role of high-ranking special faithfulness constraints. In the LC approach of Kirchner, a high-ranked locally conjoined faithfulness constraint blocks rounding from applying if it would result in a double violation of faithfulness in the same segment. In effect, due to local conjunction, a segment cannot both shorten and round. Local conjunction is defined below.

- (4) Def. of local conjunction (Smolensky 1993)  
 The Local Conjunction of  $C_1$  and  $C_2$  in domain  $D$ ,  $C_1 \&_D C_2$ , is violated when there is some domain of type  $D$  in which both  $C_1$  and  $C_2$  are violated.

Thus, if the domain of local conjunction is a segment, both  $C_1$  and  $C_2$  cannot be violated together in the same segment.

As indicated by the following constraint ranking, the role of local conjunction is to block rounding from applying to vowels that shorten, that is to underlying long low vowels in the shortening context.

- (5) The role of local conjunction  
 $[IDENT(long) \& IDENT(round)]_{seg} \gg *ai \gg IDENT(round)$

This is illustrated in the following tableaux.

- (6) /ai/ undergoes rounding

/ai /	$[IDENT(long) \& IDENT(round)]_{seg}$	&	*ai	IDENT(round)
ai			*!	
oi				*

- (7) /aai/ does not round

/aai /	$[IDENT(long) \& IDENT(round)]_{seg}$	&	*ai	IDENT(round)
ai			*	
oi	*!			*

The short vowel /ai/ in (6) undergoes rounding (/ai/→[oi]) but an identical long vowel

/aai/ in (7) does not round in the same environment (/aai/→[ai], \*[oi]). Consider first tableau (6) with an underlying short vowel. In this case, the locally conjoined constraint is satisfied by both candidates (there is no shortening, thus no violation of the LC constraint). The choice between candidates then goes to the markedness constraint \*ai. The candidate that satisfies \*ai, the rounding candidate, candidate [oi], is optimal. Consider now tableau (7) with shortening. In this case, the locally conjoined constraint is activated. It bans a candidate with both shortening and rounding, candidate [oi], even though it satisfies markedness \*ai. Thus, it is the candidate with no rounding, candidate [ai], that wins in (7).<sup>2</sup>

We now move on to Gnanadesikan (1997). In Gnanadesikan's approach, chain shifts are viewed as one-step movement on some scale of similarity. Thus, to account for the Finnish shift, we need to combine the two steps, shortening and rounding, into a single scale. Assuming that this is possible, then there is a high-ranking faithfulness constraint that prohibits two step movements on some scale of similarity, IDENT-ADJACENT [X scale].<sup>3</sup>

(8) Def. of IDENT-ADJACENT [X scale]

Given an input segment  $\alpha$  and its correspondent output segment  $\beta$ , then  $\alpha$  and  $\beta$  must have related values on scale X, where the defined relations are identity and adjacency. (In other words, the output may not have moved more than one step on the scale.)

In Finnish, the role of high-ranking IDENT-ADJACENT is to prohibit rounding in segments that also undergo shortening.

<sup>2</sup> Also, in Finnish it is more important to avoid tri-moraic syllables than to avoid the [ai] diphthong. Thus,  $*\sigma_{\mu\mu\mu} \gg *ai$ .

<sup>3</sup> Combining the two steps into a single scale is quite improbable, since they refer to very different dimensions. For types of possible scales, see Gnanadesikan (1997).

- (9) The role of IDENT-ADJACENT  
 IDENT-ADJACENT >> \*ai >> IDENT(round)

The tableaux below are analogous to the ones in the local conjunction approach discussed earlier. For clarity of exposition, they are recalled below.

- (10) /ai/ undergoes rounding

/ai/	IDENT-ADJACENT	*ai	IDENT(round)
ai		*!	
oi			*

- (11) /aai/ does not round

/aai/	IDENT-ADJACENT	*ai	IDENT(round)
ai		*	
oi	*!		*

IDENT-ADJACENT is active only in the latter case, tableau (11). It chooses in favor of the candidate with one-step movement, even though it implies a violation of a high-ranked markedness constraint \*ai. Otherwise, when IDENT-ADJACENT is satisfied by both candidates, as in (10), the markedness constraint \*ai is decisive. It chooses in favor of the candidate with rounding, candidate [oi].

We now come to the final approach to chain shifts and other opaque interactions discussed in this section, Sympathy Theory. In the Sympathy Theory of McCarthy (1999), there are special inter-candidate faithfulness constraints in addition to standard faithfulness. In case of chain shifts, the role of inter-candidate faithfulness is to explain why an otherwise regular process fails to apply in some forms of the language. Inter-candidate faithfulness constraints are like standard faithfulness, in that they evaluate identity between forms, but the dimension along which identity is evaluated is now different (no longer input-output). Inter-candidate faithfulness constraints demand that a

candidate resemble a designated failed candidate in some property or set of properties.<sup>4</sup>

In Finnish, the relevant inter-candidate faithfulness constraint demands faithfulness in rounding of the actual candidate to a candidate that did not undergo shortening, candidate [aai]. The candidate that sets the standards for identity, in Finnish [aai], is chosen on some designated IO faithfulness constraint, called the selector. In Finnish, the selector is the faithfulness constraint that militates against shortening, IDENT(long). The failed candidate that influences the actual output, in Finnish [aai], is indicated with a ✱-symbol. The faithfulness constraint that demands identity of the actual output to this candidate is the one with a subscript, IDENT(round)<sub>IDENT(long)</sub>. The subscript indicates the faithfulness constraint that chooses the ✱-candidate.

The role of inter-candidate faithfulness is to block rounding from applying to forms, where the input has the long low vowel [aai]. In terms of constraint ranking, the inter-candidate faithfulness constraint outranks the markedness constraint against diphthongs of type [ai]. This is illustrated below.

(12) The role of inter-candidate faithfulness  
IDENT(round)<sub>IDENT(long)</sub> >> \*ai >> IDENT(round)

As shown in the following tableaux, the high-ranking inter-candidate faithfulness constraint prohibits rounding in cases where the ✱-candidate is the one with a long low vowel [aai], as in (13). Yet it demands rounding when the ✱-candidate is the one with a short rounded vowel [oi], as in (14).

---

<sup>4</sup> For a more recent approach, cumulativity, see McCarthy (to appear).

## (13) No rounding of long vowels

/ai/	* $\sigma_{\mu\mu\mu}$	IDENT(round) <sub>IDENT(long)</sub>	*ai	IDENT(long)	IDENT(round)
oi		*!		*	*
ai			*	*	
$\text{⊛}$ aai	*!				

## (14) Rounding of short vowels

/ai/	* $\sigma_{\mu\mu\mu}$	IDENT(round) <sub>IDENT(long)</sub>	*ai	IDENT(long)	IDENT(round)
$\text{⊛}$ oi					*
ai		*!	*!		

In (13) the inter-candidate faithfulness constraint, IDENT(round)<sub>IDENT(long)</sub>, rules out the candidate with rounding, candidate [oi], since it is unfaithful in rounding to a candidate that did not shorten, candidate [aai]. In (14), the actual output is itself the  $\text{⊛}$ -candidate and since it is identical to itself, it vacuously satisfies the inter-candidate faithfulness constraint.

### 4.3 Predictions of PC theory

In this section I will discuss the predictions of PC theory with respect to chain shifts and compare them to the predictions of other OT approaches (Kirchner 1996, Gnanadesikan 1997). The predictions will be evaluated with respect to four types of chain shift mappings:

- (a) *push shifts*, with high-ranked markedness constraint against the initial stage in the shift but no high-ranked markedness against the intermediate stage,
- (b) *pull shifts*, with high-ranked markedness constraint against the intermediate stage but no high-ranked markedness against the initial stage,
- (c) *circular shifts*, with no termination point, and
- (d) *regular shifts*, with high-ranked markedness against each stage in the shift.


These are illustrated below. Mappings incurred by markedness are indicated with a solid arrow. Mappings that are a consequence of something other than markedness are shown with a dotted line. In the circular shift emphasis is on the type of movement rather than on markedness violations, and thus markedness violations are not indicated.

(15) Types of shifts

a. Push shifts (\*A)

A → B -> C

c. Circular shifts

A → B → C  


b. Pull shifts (\*B)

A -> B → C

d. Regular shifts (\*A, \*B)

A → B → C

Not all shifts are predicted to exist under every approach. The goal is to make sure the predictions of the theory coincide with the set of empirically attested phenomena.

It will be shown here that the predictions of PC theory differ from previous approaches with respect to push shifts - PC theory admits push shifts, while previous approaches do not allow for them. The predictions with respect to other types of shifts - regular shifts, pull shifts and circular shifts, are the same in PC as in previous approaches, though their execution is different.

#### 4.3.1 PC Admits Push Shifts

In PC, a phonological process can take place solely to preserve contrast. Push shifts are an example. Assuming A-B-C scale of similarity, push shifts occur when the /B/ to [C] mapping (the latter step in the shift) is an indirect consequence of the /A/ to [B] mapping (the prior step in the shift). In terms of constraints, there is a high-ranked markedness constraint against the initial stage in the shift (\*A) but no high-ranked

markedness against the intermediate stage. Thus, the /B/ to [C] mapping (the latter step in the shift) has to be a result of something other than markedness.

The Finnish chain shift,  $aai \rightarrow ai \rightarrow oi$ , provides an example. In Finnish, there is a high-ranked markedness constraint against trimoraic syllables,  $*\sigma_{\mu\mu\mu}$ , but there is no high-ranked markedness against *ai*. This is not to say that there is no markedness constraint against *ai* at all. It only means that  $*ai$  is not the constraint responsible for rounding in Finnish.

There is empirical evidence for this claim. If  $*ai$  were responsible for rounding in Finnish, then segments that this markedness constraint refers to should be ruled out from the language in all environments. Yet they surface in some cases, e.g., in initial syllables *aidata* “bar, enclose”, *aie* “intention”, *aika* “time”. This is the environment where shortening does not take place. Otherwise, when shortening takes place, /ai/ undergoes rounding. Thus, in Finnish rounding takes place only in the environment where shortening takes place but not elsewhere. (It can be also seen as a positional faithfulness effect in other approaches.)

Formally, since [ai]’s are allowed in the language except for the environment of shortening, I propose that the markedness constraint  $*ai$  needs to be ranked lower than conflicting PC(round) constraints. Otherwise, if  $*ai$  were high-ranking, [ai] segments would be ruled out across the board. To explain why rounding takes place in the environment of shortening, I propose that it is a push-shift effect where shortening forces rounding. In terms of constraints, rounding is forced by output-oriented PC constraint,  $PC_{OUT}(long)$ , ranked above  $PC_{IN/OUT}(round)$ .

Previous OT approaches to chain shifts (and standard OT generally) do not admit

push shifts. In standard OT, the only way to account for a phonological mapping is by a high-ranked markedness constraint (cf. Moreton 1996/1999). Therefore, for a chain shift to take place, there has to be a high-ranked markedness constraint against each stage in the shift. In Finnish, if there is no high-ranked markedness constraint against *ai*, there is no way to force rounding (thus no shift). As shown in the previous section, the role of LC (and other special faithfulness constraints) is to block a process from applying, but they cannot force it.

Unlike previous approaches to chain shifts, PC theory predicts that there exist shifts without high-ranking markedness against the intermediate stage, push shifts. If /A/ maps onto [B] to satisfy markedness, then a constraint on preserving contrast between /A/ and /B/ can by itself force the /B/ to [C] mapping. We no longer need a high-ranking markedness constraint against B to force the latter step in the shift. This prediction is illustrated in the following tableau.<sup>5</sup>

(16) /B/→C as a consequence of /A/→B

Scenarios		PC <sub>IN</sub> (A/B)	*A	PC <sub>IN</sub> (B/C)	FAITH
A. No shift A → B D → C	/A/ → B /B/ → B /C/ → C /D/ → D	*!			*
B. Push shift A → B D → C	/A/ → B /B/ → C /C/ → C /D/ → D			*	**

Both scenarios avoid A-type segments (due to \*A) but it is only in the push-shift scenario, scenario B, that /A/ and /B/ map onto distinct outputs. Therefore, only the push-

<sup>5</sup> The actual scenario in the tableau above is slightly different from the one discussed in chapter 1. The difference is in the mapping for /D/. Underlying /D/ maps onto itself and not onto [C]. This makes the argument more transparent. The reasoning, however, stays the same, regardless of this modification.

shift scenario satisfies the high-ranking  $PC_{IN}(A/B)$  constraint.  $PC_{IN}(A/B)$  demands that /A/ and /B/ contrast on the surface in some way, which they do in scenario B.

Thus, as shown in (16), in PC a shift can take place as long as there is a high-ranking markedness constraint against the initial step in the shift, here \*A. The subsequent step, /B/→[C], is then a result of high-ranking PC,  $PC_{IN}(A/B)$ , and we no longer need high-ranking markedness, \*B, to force it. In the next section, I will show that the opposite situation – where there is no high-ranked markedness constraint against the initial stage in the shift but there is one against the intermediate stage - is not predicted.

#### **4.3.2 No Pull Shifts**

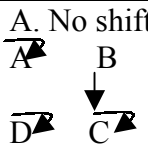
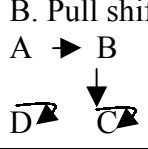
Even though the latter step in the shift can be forced by high-ranking PC, there needs to be a high-ranking markedness constraint against the initial stage in the shift. In other words, markedness is indispensable to ignite the shift. Therefore, in PC theory, as in previous approaches to chain shifts in OT, pull shifts are not admitted.

A pull shift, also known as a drag shift, takes place when the /A/ to [B] mapping (prior step in the shift) is an indirect consequence of the /B/ to [C] mapping (the latter step). In terms of constraints, in a pull shift there is no high-ranking markedness constraint against the initial stage in the shift but there is a high-ranking markedness constraint against the intermediate stage.

PC, like previous OT approaches to chain shifts, predicts no pull shifts. In previous approaches, if there is no high-ranked markedness against [A], there is no way to force /A/ to map onto a distinct output. In PC, similarly, if /B/ maps onto [C] and there is no high-ranking markedness against [A], contrast between /A/ and /B/ is preserved on the surface and there is nothing that forces the /A/ to [B] mapping. To put it differently, if

contrast is preserved and markedness is satisfied, then we do not move away from the input segment. This is illustrated in the following tableau.

(17) No /A/→B as a result of /B/→C

Scenarios		PC <sub>IN</sub> (A/B)	*B	PC <sub>IN</sub> (B/C)	FAITH
A. No shift 	/A/ → A /B/ → C /C/ → C /D/ → D			*	*
B. Pull shift 	/A/ → B /B/ → C /C/ → C /D/ → D		*!	*	**

Both scenarios satisfy PC<sub>IN</sub>(A/B), since in both, /A/ and /B/ map onto distinct outputs. But the pull shift scenario, scenario B, incurs a violation of the high-ranking \*B markedness constraint. Therefore, the pull shift scenario, scenario B, loses to the no shift scenario, scenario A.<sup>6</sup>

Thus, PC theory predicts there exist shifts without high-ranking markedness against the intermediate stage (push shifts – see section 4.3.1) but there are no shifts without high-ranking markedness against the initial stage (no pull shifts – current section). Push shifts improve on high-ranking PC, whereas pull shifts do not do so and in addition incur a violation of high-ranked markedness.

Though not allowing for pull shifts, PC theory admits a sequence of changes that resemble a pull shift effect. Take a situation where /A/→[B] “wants to happen” (\*A is high-ranking) but is blocked by /B/→[B] (to avoid neutralization). Then /B/→[C] comes along. Now the /A/→[B] map can emerge. Though this set of mappings resembles a pull shift, in the rest of this section, I will show that it can be understood as a regular shift. For

this shift to take place, there must be a high-ranking markedness constraint against each stage in the shift.

Labov provides an example of the Northern Cities Shift (Labov 1994:177-201). In the Northern Cities shift, /æ/ as in *cad* maps onto /ie/ as in *idea*, /a/ as in *cod* maps onto /æ/, and /ɔ/ as in *cawed* maps onto /o/. Thus, there is a chain shift effect of the form: [ɔ]→[a]→[æ]→[ie]. There are also other changes that take place in the Northern Cities Shift but we are going to concentrate on the ones mentioned above. Not all speakers reported by Labov show every mapping listed here. However, if a speaker shows evidence of the latter mapping in the shift, then the speaker also shows evidence of the earlier mapping. For example, if a speaker has the /o/→[æ] mapping in his grammar, then the speaker also shows raising of /æ/ to [ie], the subsequent mapping in the shift. Similarly, if a speaker shows the /ɔ/→[o] mapping, then the speaker also shows fronting of /o/ to [æ] and raising of /æ/ to [ie]. The mapping further back in the shift only takes place if the mapping(s) further front occur. Labov explains this effect as symmetry, called The Chain-Shifting Principle: “When the phonetic space between two members of a subsystem is increased by the shifting of one member (the leaving element), the other member will shift its phonetic position to fill that space (the entering element)” (also see Martinet 1955). In the following part of this section, I will argue that what Labov calls a pull shift effect should be understood as a regular shift in the theory of constraints. The logic is as follows: the initial mapping in the shift “wants to happen” due to markedness

---

<sup>6</sup> Assume that the markedness constraint \*A is dominated by \*B. Therefore, A’s are OK.

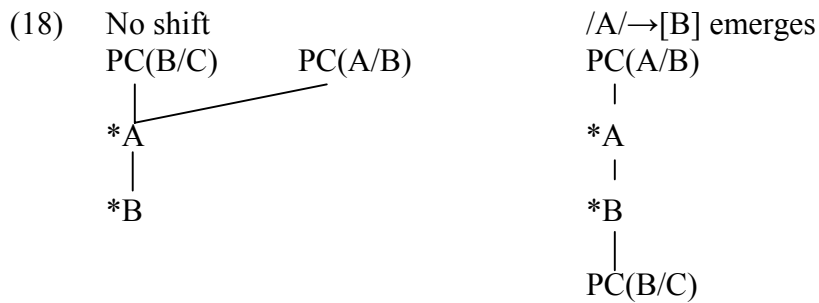
but it can only emerge once the subsequent mapping takes place. Crucially, there is a high-ranking markedness against each stage in the shift.

Let us first see whether the Chain-Shifting Principle stated by Labov (see also King 1969) can be expressed in OT. Labov's observation cannot be directly incorporated into the theory of constraints. In standard OT, phonological mappings follow from the interaction of markedness and faithfulness constraints, and there are no constraints on symmetry *per se*. But there is a version of OT, the Dispersion Theory of contrast (Flemming 1995; see section 4.5.2), in which principles of this type are present. In Dispersion Theory, there is a principle on the maximization of the distinctiveness of contrasts which refers directly to the shape of a particular space of contrasts. However, this principle has exactly the opposite effect from the Chain-Shifting Principle formulated by Labov (after Martinet). Namely, it prefers more distance between elements of a given space. Thus, if a segment deletes (or moves away), the segment that precedes it should stay where it originally is to take advantage of the increased distance between the two elements. There is another principle in the Dispersion Theory, called minimize articulatory effort, which resembles markedness constraints in other versions of OT. Minimizing articulatory effort can account for chain-shifting. This brings us back to the original observation that mappings like the Northern Cities Shift can be accounted for in OT if there are markedness constraints against each step in the shift.

Let us now give a PC account of the above mentioned sequence of changes: /A/→[B] takes place because /B/→[C] is introduced. Consider two competing languages. In both, there are markedness constraints against each stage in the shift, A and B, but \*A can only emerge in the language in which \*B is free to operate.

I will assume that the only difference between the two languages is in the relative ranking of the markedness constraint \*B versus a conflicting PC constraint, PC(B/C). In the language without the shift, PC(B/C) dominates \*B, thus blocking the /B/→[C] mapping. It is more important to avoid the B/C merger than it is to avoid segments of type B. As a result, the mapping /B/→[C] does not take place and the mapping /A/→[B] does not either. In the language with the shift, the ranking is reversed. Now \*B dominates PC(B/C). It is more important to avoid B's than it is to avoid the B/C merger. Consequently, the /B/→[C] mapping is introduced and /A/→[B] can emerge.

Altogether:

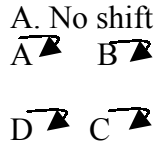
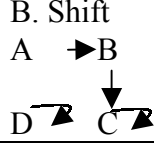
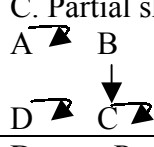
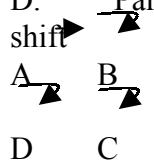


Not every ranking relation shown above can be proved for each language. I assume the same ranking of constraints, besides \*B versus PC(B/C), for the sake of uniformity.

In the tableaux below, I consider four scenarios. In scenario A, neither /A/ nor /B/ move (no shift). In scenario B, /B/ maps onto [C] and /A/ maps onto [B] (pull shift effect). In scenario C, /B/ maps onto [C] but /A/ stays where it is (no A movement). In scenario D, /A/ maps onto [B] even though /B/ stays where it is (no B movement).

Consider first the language in which /A/→[B] is blocked because /B/→[C] does not happen. In terms of constraint ranking, PC(B/C) outranks \*B.

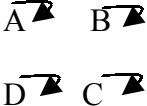

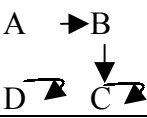
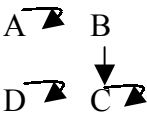
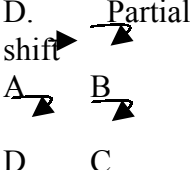
(19) /A/→[B] is blocked because /B/→[B]

Scenarios	PC <sub>IN</sub> (A/B)	PC <sub>IN</sub> (B/C)	*A	*B
A. No shift 			*	*
B. Shift 		*!		*
C. Partial shift 		*!	*	
D. Partial shift 	*!			**

Scenario A wins since it does not merge any contrasts, unlike any other scenario in the tableau. In scenario A, /B/ stays where it is, and so to preserve contrast with /B/, /A/ does not map onto [B], even though [A]'s are avoided in the language (\*A >> \*B). All the other scenarios violate high-ranking constraints on contrast. Scenarios B and C, where /B/ maps onto [C], merge B and C type segments, thereby violating PC(B/C). Scenario D merges A and B, violating PC(A/B).

We now discuss the language in which /B/ maps onto [C]. In terms of constraints, \*B outranks PC(B/C). In this situation, /A/ is free to map onto [B].

(20) /A/→[B] because /B/→[C]

Scenarios	PC <sub>IN</sub> (A/B)	*A	*B	PC <sub>IN</sub> (B/C)
A. No shift 		*!	*	
B. Shift  			*	*
C. Partial shift 		*!		*
D.  Partial shift	*!		**	

Scenario B is optimal. It avoids [A]'s, therefore satisfying markedness \*A, and it does not merge the contrast between /A/ and /B/ because /B/ maps onto [C]. Scenarios A and C, on the other hand, lose on markedness \*A, since they both contain A-type segments. Scenario D, in turn, is ruled out on contrast, since it merges A and B (thus a violation of PC(A/B)).

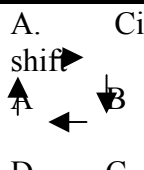
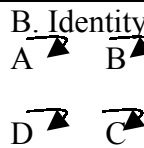
Therefore, in this case, one could say that /A/→[B] takes place because /B/→[C] is introduced. However, we still need a high-ranking \*A constraint (\*A >> \*B) for the /A/→[B] mapping to take place. Without this constraint ranked above \*B even if /B/→[C] were introduced /A/ would not map onto [B]. Scenario C would then win (see the above tableau for the violation profile of scenario C). Thus, what resembles a pull shift effect is really a regular shift. In the next section, we proceed to circular shifts.

### 4.3.3 No Circular Shifts

Circular shifts involve movement without termination point. This type of movement does not improve on either markedness or PC. It takes place for no apparent reason. PC theory rules it out. In PC, a circular shift is harmonically bounded by a competing identity scenario. In this section, I will show that circular shifts incur the following types of fatal violations: (i) fatal violation of relational PC constraints, (ii) fatal violation of the generalized faithfulness constraints, and (iii) fatal violation of too many PC constraint types.

Let us start with the violation of relational PC. The following table compares the circular shift scenario to an identity scenario.

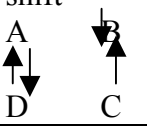
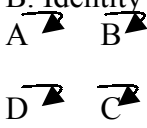
(21) No circular shifts – unnecessary PC-REL violations

Scenarios		PC <sub>IN/OUT</sub>	PC <sub>REL</sub>	MARK	FAITH
A. Circular shift 	/A/ → B /B/ → C /C/ → D /D/ → A		*!	A,B,C, D	**** A→B B→C C→D D→A
B. Identity 	/A/ → A /B/ → B /C/ → C /D/ → D			A,B,C, D	

Both scenarios satisfy input- and output-oriented PC constraints. In both, all input segments map onto distinct outputs and none of the outputs is ambiguous. Both scenarios incur the same violations of markedness since they contain the same outputs and the same number of them. But the circular shift scenario, scenario A, violates relational PC constraints by permuting contrasts. In the circular shift, none of the original input contrasts are preserved in the output (see (30), chapter 1).

Let us now consider a different type of circular shift, a shift with two symmetrical exchange processes. This shift incurs an unnecessary violation of generalized faithfulness.

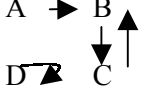
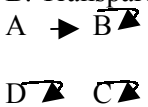
(22) No circular shifts – too many violations of faithfulness

Scenarios		PC <sub>IN/OUT</sub>	PC <sub>REL</sub>	MARK	FAITH
A. Circular shift 	/A/ → D /B/ → C /C/ → B /D/ → A			A, B, C, D	****! A→D B→C C→B D→A
B. Identity 	/A/ → A /B/ → B /C/ → C /D/ → D			A, B, C, D	

Both scenarios satisfy PC constraints: they do not neutralize input contrasts, have no ambiguous outputs and preserve some original input contrasts in the output. But the circular shift scenario involves unnecessary movement and thus is ruled out on generalized faithfulness in stage 2 of Eval.

Circular shifts can also result in more PC violations. Consider a set of mappings, part of which is an exchange process versus a corresponding transparent mapping. The two competing scenarios are shown in the following tableau. The regular mapping and the circular shift fare the same on markedness since they contain the same segments but the circular shift involves too many types of PC violations. Since in this scenario, /A/ and /C/ merge, in terms of mergers, it is a violation of both PC<sub>IN/OUT</sub>(A/B) and PC<sub>IN/OUT</sub>(B/C). The segments /A/ and /C/ are distinct on two dimensions. This is illustrated below.

(23) No circular shifts – too many types of PC violations

Scenarios		PC <sub>IN/OUT</sub> (A/B)	PC <sub>IN/OUT</sub> (B/C)	PC <sub>REL</sub>	MARK	FAITH
A. Circular shift 	/A/ → B /B/ → C /C/ → B /D/ → D	**!	**!		B,B,C,D	*** A→B B→C C→B
B. Transparent 	/A/ → B /B/ → B /C/ → C /D/ → D		**		B,B,C,D	* A→B

Scenario B wins as it violates fewer types of PC constraints.<sup>7</sup>

To sum up, in PC theory there is no movement unless it improves on PC or markedness. This shows that in PC theory, circular shifts are ruled out in favor of non-circular mappings (similarly exchange rules, see Anderson and Browne 1973). The same prediction is made in standard OT (Moreton 1996/1999).

#### 4.3.4 Regular Shifts

So far we have discussed cases where PC constraints activate a phonological process (see push shifts in 4.3.1). But PC constrains can also block a process. In cases like that, a process applies unless it would result in the loss of some contrast. Regular shifts are an example. In a regular shift, there is a high-ranking markedness constraint against each stage in the shift. Thus, each process in a regular shift is forced by a high-ranking markedness constraint. The key assumption of PC theory is that, despite high-ranking markedness, not all segments can get to the unmarked output, due to high-ranking PC constraints.

An example of a regular shift comes from language acquisition. As discussed by Kisseberth (1976) and Velten (1943), in child speech, there are processes of final

devoicing and denasalization (both due to high-ranking markedness constraints), but final devoicing targets only underlying non-nasals. De-nasalized voiced stops do not devoice. Thus:

- (24) Interaction of final devoicing and denasalization
- |               |              |                |              |
|---------------|--------------|----------------|--------------|
| <u>adult</u>  | <u>child</u> | <u>adult</u>   | <u>child</u> |
| ba <b>d</b>   | ba <b>t</b>  | br <b>ea</b> d | bu <b>t</b>  |
| broo <b>m</b> | bub <b></b>  | trai <b>n</b>  | du <b>d</b>  |

Since underlying nasals do not devoice, contrast is preserved between underlying nasal and non-nasal stops in word-final position, /n/ vs. /d/ → [d] vs. [t].<sup>8</sup>

Regular shifts like the one above are predicted to exist under any of the OT approaches to chain shifts discussed so far. In fact, in the non-PC approaches, where we need high-ranking markedness to force a phonological process, this is the only type of shift possible. In PC, this is one of two. (The other one is push shift – see 4.3.1.)

In PC, a regular shift is obtained under PC blocking, when a PC constraint outranks conflicting markedness, thus blocking an otherwise regular phonological process in contexts where it neutralizes contrast. In effect, a process applies unless it results in the loss of contrast.

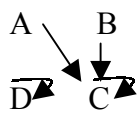
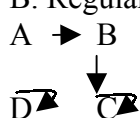
- (25) Chain shift ranking - PC blocking  
 $PC_{IN}(A/B) \gg *B \gg PC_{IN}(B/C)$

The process /B/→[C] applies due to high-ranked markedness \*B, but /A/ maps onto [B] and not [C] since contrast between /A/ and /B/ needs to be preserved on the surface. Consequently, in the optimal scenario, there is a violation of the \*B constraint. (Crucially, \*A dominates \*B, since [A] is not an acceptable output.)

---

<sup>7</sup> In the examples in (23), both scenarios satisfy PC-REL constraints. Both preserve each type of input contrast in the output.

## (26) Regular shift

Scenarios		PC <sub>IN</sub> (A/B)	*B	PC <sub>IN</sub> (B/C)	FAITH
A. No shift 	/A/ → C /B/ → C /C/ → C /D/ → D	*!		*	***
B. Regular shift <sup>8</sup> 	/A/ → B /B/ → C /C/ → C /D/ → D		*	*	**

Scenario B, in which /A/ and /B/ map onto distinct outputs, is optimal. Thus, even though [B]'s are avoided in the language, if /B/ maps onto [C], then /A/ stops at [B] instead of moving all the way to [C]. This preserves contrast with /B/.

#### 4.3.5 Summary

This section examined predictions of PC theory with respect to chain shifts. It has been shown that PC theory predicts push shifts and regular shifts but rules out pull shifts and circular shifts.

For A→B→C to exist as a chain shift under PC theory, there must be a high-ranking markedness constraint against the initial stage in the shift, \*A. The markedness constraint against the subsequent stage, \*B, must exist in CON but it is not necessarily high-ranking (see the difference between push shifts versus regular shifts). Thus, the difference between PC theory and its alternatives turns entirely on the matter of whether \*B must be high-ranking or not. This has empirical consequences. It predicts push shifts, where segments that \*B markedness constraint militates against exist but not in the chain-shifting environment.

<sup>8</sup> I assume that this is a regular shift and not a push-shift effect, since not all obstruents that device have corresponding nasals. Thus, not all instances of devoicing could be a result of denasalization.

In section 4.3.1 it was shown that PC constraints can by themselves activate a phonological process without recourse to high-ranking markedness constraints, as in standard OT. Therefore, push shifts are predicted in the PC approach to chain shifts. In section 4.3.2 we observed that although PC constraints can force a phonological mapping, there has to be a high-ranked markedness constraint to force the initial step in the shift. Thus, pull shifts with no high-ranked markedness against the initial step are not allowed in the theory. Section 4.3.3 showed, in addition, that PC theory rules out unnecessary movement, thus not allowing for circular shifts, and finally section 4.3.4 discussed cases where PC constraints block a phonological process, thereby accounting for regular shifts. The properties of PC theory and their predictions discussed in this section are summarized below.

(27)	Properties of PC theory	
	<u>Property</u>	<u>Effect</u>
(i)	PC can activate a process.	Push shifts exist (4.3.1)
(ii)	Markedness is needed to initiate a shift.	No pull shifts (4.3.2)
(iii)	PC does not allow for unnecessary movement.	No circular shifts (4.3.3)
(iv)	PC can block a process.	Regular shifts exist (4.3.4)

This concludes our discussion of possible chain shift mappings within PC theory. The following sections discuss other predictions of PC theory: diverse ways of preserving contrast (4.4.1), multiple opacity (4.4.2) and contrast in acquisition (4.4.3). This is followed by a discussion of the differences and similarities between PC constraints, standard faithfulness and previous approaches to contrast in OT, among those the Dispersion Theory of contrast (Lindblom 1986, Flemming 1995) (section 4.5.2).

## **4.4 Further Predictions**

### **4.4.1 Types of Transformations**

The main observation of PC theory is that in chain shift mappings and other opaque alternations, contrast is preserved but transformed into some other surface contrast. In PC theory the way in which contrast is preserved follows solely from the relative ranking of constraints on preserving contrast and markedness constraints. This allows for a very rich array of preserving contrasts. Through ranking permutations, it is predicted that contrast can be preserved in many diverse ways. For example, contrast in obstruent voicing can be preserved as contrast in vowel length, as will be discussed below, but also as nasality, ATR, labialization etc. In fact, any contrast can be preserved as any other contrast as long as contrast transformation is generated by the constraint ranking. This is problematic since it generates unattested ways of preserving contrast. In this section, I will examine this prediction of PC theory and try to come up with a possible solution to the problem.

Consider the obstruent voicing contrast. In PC theory, it is predicted that the obstruent voicing contrast can be preserved by vowel length, as in Friulian (see chapter 2), but also as any other contrast. Consider the following constraints: a markedness constraint against voiced obstruents syllable-finally,  $*\text{voiobs}]_{\sigma}$ , a conflicting PC constraint, PC(voice), as well as PC(long) and PC(nasal). By permutations of these constraints, assuming that no other conflicting constraints are ranked higher than those, four types of languages are predicted. Among them, there is a language with no final devoicing. In this language, markedness is ranked below PC constraints, thus unable to compel neutralizations. The other three languages show final devoicing. In all of them, markedness outranks one of the PC constraints. In one of those, the obstruent voicing contrast is neutralized on the surface. In the other two, the obstruent voicing contrast is

preserved but manifested as a different surface contrast, vowel length and vowel nasality, respectively. Let us now look at the languages and corresponding rankings.

- (28) Typological predictions
- a. No final devoicing (English)  
PC(voice), PC(nasal), PC(long) >> \*voiobs]<sub>σ</sub>
  - b. Final devoicing, voicing contrast neutralized (German)  
\*voiobs]<sub>σ</sub>, PC(nasal), PC(long) >> PC(voice)
  - c. Final devoicing, voicing contrast preserved as length (Friulian)  
\*voiobs]<sub>σ</sub>, PC(voice), PC(nasal) >> PC(long)
  - d. Final devoicing, voicing contrast preserved as nasality (unattested)  
\*voiobs]<sub>σ</sub>, PC(voice), PC(long) >> PC(nasal)

Three of the four ways of preserving contrast generated above (A-C) are found cross-linguistically but one of them, case D, is unattested. Case D represents a language where the obstruent voicing contrast is preserved as contrast in nasality, which is, to my knowledge, not attested.

Let us consider the problematic language in more detail. In this language voiced obstruents are ruled out syllable-finally due to a high-ranking markedness constraint \*voiced/obstruent]<sub>σ</sub>. Furthermore, avoiding voiced obstruents syllable-finally compels a merger in nasality (\*voiced/obstruent]<sub>σ</sub> >> PC<sub>IN/OUT</sub>(nasal)) and a merger in nasality is better than a merger in voicing or vowel length (PC<sub>IN/OUT</sub>(voice), PC<sub>IN/OUT</sub>(long) >> PC<sub>IN/OUT</sub>(nasal)). In this situation, if nothing else is involved, words that contrast underlyingly in obstruent voicing, contrast on the surface in nasality of the preceding vowel. This seems like a troubling prediction.

So far we have seen the following ways of preserving contrasts. Let us start with vowel length and vowel height contrasts in Finnish. In Finnish, length is transformed into

rounding, height and sometimes presence vs. absence of a segment. Undoubtedly, there is a relation between length, height and rounding (Ladefoged 2001). It has been reported that (i) lower vowels are longer than higher vowels, (ii) higher vowels are more rounded than lower vowels – low vowels are most often not rounded at all, and (iii) more peripheral vowels are both longer and if non-low also higher than less peripheral ones. Since there is a relation between length, height and rounding, it is predicted that the three properties form an alliance when contrast transformation comes into play.

Arabic (see chapter 5) will further support this observation. In Arabic, the epenthesis contrast is realized as stress contrast. That is, the presences vs. absence of a vowel in the input (V/∅ contrast) is realized as surface contrast in segmental prominence. This is predictable. The presence vs. absence of a vowel implies the presence vs. absence of a mora, and since dialects of Arabic are weight-sensitive, stress assignment also correlates with mora-hood. Therefore, transforming the V/∅ contrast into a stress contrast in Arabic is predictable. It takes place along the dimension of mora-hood.

Finally, the obstruent voicing contrast in Polish or Friulian (see chapter 2) is realized in predictable ways, by vowel height and vowel length, respectively. Both properties are correlates of the obstruent voicing contrast and thus those two ways of preserving the obstruent voicing contrast are predictable.

Is there a principled way to determine what P properties pair up in transformations? To explain the concept of a phonological feature/property, Kingston and Diehl (1994) propose that there is a set of surface correlates of a given contrast (phonological feature/opposition) and thus a phonological property P is not an abstract entity but rather results from co-occurrence of multiple P properties locally in the same

context. For example, obstruent voicing in word-final position is realized as (i) glottal pulsing during the production of the obstruent (closure voicing), (ii) shorter closure duration of the obstruent, (iii) longer preceding vowel, and (iv) lowering of  $F_1$  on the preceding vowel (vowel raising). Thus, a given opposition, in this case obstruent voicing, has multiple surface correlates. When one of the correlates is not present in the output, the remaining correlate(s) are often enhanced since they are now the sole exponent of the original contrast.

If we assume that certain types of contrast transformations are unattested, it is necessary to put restrictions on the set of satisfaction conditions of PC constraints. One way to do so is to establish a set of surface correlates for each P property, following Kingston and Diehl, and further propose that PC constraints are satisfied only when in addition to preserving contrast, the replacing contrast is chosen from the set of surface correlates of a given property P. For example, in case of the obstruent voicing contrast,  $PC_{IN}(\text{voice})$  is satisfied only when a pair of forms that contrasts in *voice*, contrasts on the surface in a property P that belongs to the set of surface correlates of *voice*. Therefore, a pair of forms /bat/ vs. /bad/, contrasting in voicing, that maps onto [bat] vs. [ba:t], contrasting in length, satisfies the constraint  $PC_{IN}(\text{voice})$ . But the same pair of forms that maps onto [bat] vs. [bãt], contrasting in nasality, violates  $PC_{IN}(\text{voice})$ . Nasality is not one of the correlates of voicing. This way of thinking about P properties would eliminate

impossible ways of preserving contrasts. It requires further research but that is the direction I would take.<sup>9</sup>

#### 4.4.2 Multiple Opacity

PC constraints are satisfied when contrast is minimally preserved for a given pair of inputs. Consider two inputs distinct in presence versus absence of a segment, /C<sub>1</sub>V<sub>2</sub>C<sub>3</sub>/ vs. /C<sub>1</sub>V<sub>2</sub>/. Assume furthermore that C<sub>3</sub> deletes to avoid a coda consonant. For contrast to be preserved between the two inputs, it is enough for the preceding vowel to lengthen when the consonant deletes. No other trace of the deleted consonant needs to be present in the output to mark a distinction between the two inputs. In fact, it is predicted that if contrast is preserved in one way, for example by vowel lengthening, as in the hypothetical example above, it should not be preserved in another way as well. This would create additional (unmotivated) violations of generalized faithfulness. But in some cases, it has been reported that contrast is preserved in multiple ways, called multiple opacity or more recently piling-up of faithfulness violations. In the rest of this section, I will show that multiple opacity is not admitted to PC theory if it were to follow solely from the interaction of input-oriented PC constraints, but it is predicted to occur if output-oriented PC is at stake.

Consider a Bantu language of East Africa, Luganda. In Luganda (Clements 1986, Rosenthal 1994) vowel length is contrastive, there are both long and short vowels, but length contrasts are neutralized in some environments, one of which is before a pre-

---

<sup>9</sup> This view of satisfaction conditions of PC constraints would also explain why contrast, when dislocated from the original locus, is kept locally rather than dislocated far away in the string of segments (see Friulian, Polish). In chapter 5, I will propose separate PC constraints that guard locality but it is possible that locality can be inscribed into the generic PC constraints themselves, once the component of PC theory that deals with predictability of contrast transformation is developed. This is the subject for further research.

nasalized consonant. In Luganda, before pre-nasalized consonants, vowels surface as long. This is traditionally described as compensatory lengthening before pre-nasalized consonants (Wetzels & Sezer 1986).<sup>10</sup>

In Luganda, there are no syllable codas except for geminates and there are no complex onsets except for the consonant/glide combination. Thus, a post-vocalic nasal avoids being in a coda position or in a complex onset by merging with the following consonant. This causes the following consonant to nasalize and the preceding vowel to lengthen.

(29) Prenasalization & lengthening

/mu+ntu/	→	mu: <sup>n</sup> tu	‘person’
/mu+lenzi/	→	mule: <sup>n</sup> zi	‘boy’
/ku+linda/	→	kuli: <sup>n</sup> da	‘to wait’
/ba+ntu/	→	ba: <sup>n</sup> tu	‘people’

As shown above, the nasal is retained in the output in the form of prenasalization and lengthening. This preserves contrast with words that do not have the nasal to begin with (/mutu/ vs. /muntu/ → [mutu] vs. [mu:<sup>n</sup>tu], /mu:tu/ vs. /mu:ntu/ → [mu:tu] vs. [mu:<sup>n</sup>tu]) but merges contrast with words that have a pre-nasalized consonant in the input (/mu<sup>n</sup>tu/ vs. /muntu/ → [mu:<sup>n</sup>tu], /mu:<sup>n</sup>tu/ vs. /mu:ntu/ → [mu:<sup>n</sup>tu]). Consider the following set of corresponding input and output contrasts in Luganda:

---

<sup>10</sup> For other environments of Luganda length neutralization, see Clements (1986).

(30) Luganda  
 a. Input contrasts

<i>Short vowels</i>	<i>Long vowels</i>		
mutu <sub>1</sub>	mu:tu <sub>4</sub>	<i>Non-nasals</i>	
muntu <sub>2</sub>	mu:ntu <sub>5</sub>	<i>Nasal C</i>	<i>Nasals</i>
mu <sup>n</sup> tu <sub>3</sub>	mu: <sup>n</sup> tu <sub>6</sub>	<sup>n</sup> <i>C</i>	

b. Output contrasts


<i>Short vowels</i>	<i>Long vowels</i>		
mutu <sub>1</sub>	mu:tu <sub>4</sub>	<i>Non-nasals</i>	
		<i>Nasal C</i>	<i>Nasals</i>
	mu: <sup>n</sup> tu <sub>2456</sub>	<sup>n</sup> <i>C</i>	

As shown above, contrast is preserved between nasal and non-nasal sets. But it is merged across the set of nasal and pre-nasalized consonants, and across the long-short vowel set.

Let us consider the problem formally. Following Clements (1986), McCarthy (1999), I will assume that nasal codas are banned in Luganda (\*mun.ta). Even though nasal codas are not allowed in the output, the nasal does not delete but is realized as pre-nasalization on the following consonant. I will claim that this is to preserve contrast with words that do not have the nasal to begin with. If the nasal dropped out completely, words with and without a nasal would be realized the same. Yet, retaining the nasal is at the cost of merging contrast between inputs with nasal and pre-nasalized consonants. The

preceding vowel also lengthens and so there is also a merger of the length contrast. (More on it below.) The following tableau shows the problem formally.<sup>11</sup>

(31) Reason for pre-nasalization

	NO-CODA	PC <sub>OUT/IN</sub> (N/∅)	PC <sub>OUT/IN</sub> (NC/ <sup>n</sup> C)	PC <sub>OUT/IN</sub> (VV/V)
A. Actual 			* * * * * PC <sub>OUT</sub> (NC/ <sup>n</sup> C): [mu: <sup>n</sup> tu]  PC <sub>IN</sub> (NC/ <sup>n</sup> C): /mu:ntu/,/mu: <sup>n</sup> tu/ /mu:ntu/,/mu: <sup>n</sup> tu/ /muntu/,/mu: <sup>n</sup> tu/ /muntu/,/mu: <sup>n</sup> tu/	* * * * * PC <sub>OUT</sub> (VV/V): [mu: <sup>n</sup> tu]  PC <sub>IN</sub> (VV/V): {/mu:ntu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu:ntu/} {/mu: <sup>n</sup> tu/,/mu: <sup>n</sup> tu/}
B. No pre-nasal.		* * * !  PC <sub>OUT</sub> (N/∅): [mu:tu]  PC <sub>IN</sub> (N/∅): /mu:tu/,/mu:ntu/ /mu:tu/,/muntu/		* * * * * PC <sub>OUT</sub> (VV/V): [mu: <sup>n</sup> tu], [mu:tu]  PC <sub>IN</sub> (VV/V): {/mu: <sup>n</sup> tu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu:tu/} {/muntu/,/mu:ntu/}
C. Nasal coda	* * ! /mu:ntu/→[mu:ntu] /muntu/→[muntu]			* * PC <sub>OUT</sub> (VV/V): [mu: <sup>n</sup> tu]  PC <sub>IN</sub> (VV/V): {/mu: <sup>n</sup> tu/,/mu: <sup>n</sup> tu/}

Scenario A is optimal. There is no nasal coda and words with and without a nasal are pronounced differently. This is not true of the other two competitor scenarios. In scenario B, there is a merger between words with and without a nasal since the nasal drops out

<sup>11</sup> As far as constraints go, I assume that both words with a separate nasal in the input (/muntu/) and a pre-nasalized consonant in the input (/mu:<sup>n</sup>tu/) “contain a nasal” and thus when they neutralize on the surface, there is no merger of the contrast in presence vs. absence of a nasal. The constraint PC(N/∅) is satisfied. Since this constraint is high-ranked in Luganda, this rules out any competing scenario in which words with and without a nasal map onto the same output.

completely. In scenario C, the nasal surfaces as a coda and so there is a violation of the NO-CODA constraint.

But to preserve contrast in presence vs. absence of a nasal, it would be enough to nasalize the following consonant. Why, in addition, lengthen the vowel?

(32) Possible competitor: the set of outputs (cf. (30b))

<i>Short vowels</i>	<i>Long vowels</i>		
mutu <sub>1</sub>	mu:tu <sub>4</sub>	<i>Non-nasals</i>	
↓	↓	<i>Nasal C</i>	<i>Nasals</i>
mu <sup>n</sup> tu <sub>24</sub>	mu: <sup>n</sup> tu <sub>56</sub>	<sup>n</sup> C	

In the competitor scenario above, words with and without a nasal are kept distinct and there are no mergers of the vowel length contrast since there is no lengthening.

Compare the actual scenario (30) to the competitor scenario above. Under the current constraint ranking, the competitor scenario comes out optimal.

(33) Wrong result: no lengthening

	NO-CODA	PC <sub>OUT/IN</sub> (N/∅)	PC <sub>OUT/IN</sub> (NC/ <sup>n</sup> C)	PC <sub>OUT/IN</sub> (VV/V)
<b>A. Actual</b> ☞ /mutu/ → mutu /mu:tu/ → mu:tu /muntu/ → mu: <sup>n</sup> tu /mu:ntu/ → mu: <sup>n</sup> tu /mu <sup>n</sup> tu/ → mu: <sup>n</sup> tu /mu: <sup>n</sup> tu/ → mu: <sup>n</sup> tu			* * * * * PC <sub>OUT</sub> (NC/ <sup>n</sup> C): [mu: <sup>n</sup> tu]  PC <sub>IN</sub> (NC/ <sup>n</sup> C): {/mu:ntu/,/mu: <sup>n</sup> tu/} {/mu:ntu/,/mu <sup>n</sup> tu/} {/muntu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu: <sup>n</sup> tu/}	* * * * *! PC <sub>OUT</sub> (VV/V): [mu: <sup>n</sup> tu]  PC <sub>IN</sub> (VV/V): {/mu:ntu/,/mu <sup>n</sup> tu/} {/muntu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu:ntu/} {/mu <sup>n</sup> tu/,/mu: <sup>n</sup> tu/}
<b>B. No lengthening</b> ☞ /mutu/ → mutu /mu:tu/ → mu:tu /muntu/ → mu <sup>n</sup> tu /mu:ntu/ → mu: <sup>n</sup> tu /mu <sup>n</sup> tu/ → mu <sup>n</sup> tu /mu: <sup>n</sup> tu/ → mu: <sup>n</sup> tu			* * * * PC <sub>OUT</sub> (NC/ <sup>n</sup> C): [mu: <sup>n</sup> tu],[mu <sup>n</sup> tu]  PC <sub>IN</sub> (NC/ <sup>n</sup> C): {/mu:ntu/,/mu: <sup>n</sup> tu/} {/muntu/,/mu <sup>n</sup> tu/}	

Both scenarios satisfy NO-CODA and do not merge the N/∅ contrast. But the “no lengthening” scenario in addition does not merge the contrast between words of different vowel length. If there is a short vowel, it stays short. If there is a long vowel, it remains long. It also fares better on input-oriented PC(NC/<sup>n</sup>C).

The question then is why both contrasts, pre-nasalization and length, are neutralized in Luganda to preserve contrast in presence vs. absence of a nasal. It seems that it would enough to neutralize only one dimension of contrast, in this case pre-nasalization, and leave length contrast intact. The contrast in presence vs. absence of a nasal would be preserved in that case and NO-CODA would be satisfied.

To explain lengthening, I will propose that there is a high-ranking output-oriented PC constraint. Compare the actual scenario (30b) with the alternative in (32). Both

scenarios merge some pre-nasalization contrasts but the actual scenario, in which there is lengthening, contains only one output that corresponds to inputs distinct in pre-nasalization. Therefore, the actual scenario improves on the  $PC_{OUT}(NC^nC)$  constraint. In the actual scenario, there is only one output that violates this constraint, whereas in the competitor scenario, with no length neutralizations, there are two such outputs. The two competing scenarios are presented in tableau (33). This is the core property of opacity. Opacity reduces the number of outputs ambiguous in property P at the cost of neutralization in some other property. In Luganda, opacity reduces the number of outputs ambiguous in  $NC^nC$  at the cost of length mergers.

For the actual scenario to win, the output-oriented PC constraint needs to outrank constraints against length mergers, as well as the input-oriented  $PC_{IN}(NC^nC)$ . The relevant ranking and tableau are given below.

- (34) Opacity ranking  
 NO-CODA,  $PC_{OUT/IN}(N/\emptyset)$ ,  $PC_{OUT}(NC^nC) \gg PC_{IN}(NC^nC)$ ,  $PC_{IN/OUT}(VV/V)$

(35) Opacity improves on output-oriented PC

	NO-CODA	PC <sub>OUT/IN</sub> (N/∅)	PC <sub>OUT</sub> (NC/ <sup>n</sup> C)	PC <sub>IN</sub> (NC/ <sup>n</sup> C)	PC <sub>OUT/IN</sub> (VV/V)
Actual $\Rightarrow$ /mutu/ → mutu /mu:tu/ → mu:tu /muntu/ → mu: <sup>n</sup> tu /mu:ntu/ → mu: <sup>n</sup> tu /mu <sup>n</sup> tu/ → mu: <sup>n</sup> tu /mu: <sup>n</sup> tu/ → mu: <sup>n</sup> tu			* [mu: <sup>n</sup> tu]	****	*****
No lengthening /mutu/ → mutu /mu:tu/ → mu:tu /muntu/ → mu <sup>n</sup> tu /mu:ntu/ → mu: <sup>n</sup> tu /mu <sup>n</sup> tu/ → mu <sup>n</sup> tu /mu: <sup>n</sup> tu/ → mu: <sup>n</sup> tu			***! [mu <sup>n</sup> tu] [mu: <sup>n</sup> tu]	**	

The actual scenario wins since it improves on the output-oriented PC<sub>OUT</sub>(NC/<sup>n</sup>C) constraint. In this scenario, there is only one output that corresponds to inputs distinct in pre-nasalization. This supports our observation that opacity reduces the number of outputs ambiguous in some property P. This is at the cost of incurring neutralizations of some other contrast in the system. In Luganda, the length contrast is neutralized to reduce the number of outputs ambiguous in prenasalization.

In conclusion, the Luganda case is admitted under PC theory, but this is not to say that any other case of multiple opacity would be allowed. Unless opacity improves on output-oriented PC or markedness, it should not take place. This is a restriction on what types of contrast transformations are possible under PC. This distinguishes the PC

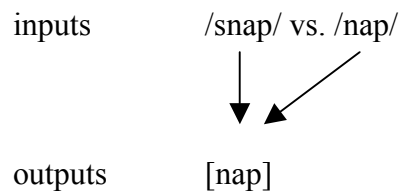
approach from other approaches to opaque process interactions like, for example, rule ordering.<sup>12</sup>

### 4.4.3 Contrast Preservation in Acquisition

In this section, the PC approach is investigated with respect to the acquisition of phonology. This is discussed on the example of cluster reduction.

Children often reduce consonant clusters in onsets (Gnanadesikan to appear, Pater 2002). Consider a case of cluster simplification in a sequence fricative *plus* nasal. The nasal is retained in the output while the fricative deletes (Pater 1997). In what follows, I am not going to discuss the pattern of deletion (see Pater 2002).

#### (36) Child mapping



(37) <i>Adult gloss</i>	<i>Child pronunciation</i>	<i>Age</i>
mommy sneeze	[mami+nis]	1;9.5
snake	[nek]	1;11.22
what (do) I smell?	[wʌs ai mɛʊ]	2;4.29

Formally, it must be the case that the child ranks markedness \*CLUSTER above conflicting PC constraints. It is more important for the child to satisfy the high-ranking well-formedness constraint than it is to preserve contrast in presence versus absence of a segment,  $PC_{IN/OUT}(C/\emptyset)$ , where C stands for a consonant. The child ranking is given below.

<sup>12</sup> Another example of multiple opacity comes from Arabic. In some Arabic dialects (John McCarthy (p.c.)), epenthetic words are special with respect to stress and emphasis harmony (or even palatalization). Thus, contrast is doubly-preserved.

- (38) Child ranking  
 $*\text{CLUSTER} \gg \text{PC}_{\text{IN/OUT}}(\text{C}/\emptyset)$

Let us now move on to adult speech. In adult language, the ranking is the opposite. Markedness is demoted below conflicting PC constraints, such that deletion is blocked.

- (39) Adult mapping  
 inputs            /snap/ vs. /nap/  
                               ↓            ↓  
 outputs            [snap] vs. [nap]

The ranking is given below.

- (40) Adult ranking  
 $\text{PC}_{\text{IN/OUT}}(\text{C}/\emptyset) \gg *\text{CLUSTER}$

The relevant tableaux follow.

- (41) Child language

			*CLUSTER	$\text{PC}_{\text{IN/OUT}}(\text{C}/\emptyset)$
a.	☞	/snap/ → nap /nap/ → nap		**
b.		/snap/ → snap /nap/ → nap	*!	

- (42) Adult language

			$\text{PC}_{\text{IN/OUT}}(\text{C}/\emptyset)$	*CLUSTER
a.		/snap/ → nap /nap/ → nap	**!	
b.	☞	/snap/ → snap /nap/ → nap		*

This is parallel to markedness demotion in language acquisition within the OT model with standard faithfulness, see Tesar (1998), Tesar & Smolensky (1998), also Hayes (1999).

So far it has not been made explicit how children learn the way PC constraints interact in adult language. Not all forms included in the scenarios are the existing forms of the language. Thus, what is in the scenarios that children compile? In standard OT, children learn constraint interaction based on the actual forms of the language. I will assume that the same is true of the PC model. The learning process is based on the mappings for the actual forms of the language (as shown above, 43-44), and only then extended (hypothesized) to non-existent forms.

This concludes our discussion of PC predictions that have to do with (i) constraint permutations, (ii) satisfaction conditions of PC constraints, and (iii) the role of contrast in acquisition. The next section compares the PC approach with other OT approaches to contrast.

#### **4.5 Comparison with Previous Approaches**

In this section I will discuss the differences between PC and faithfulness (4.5.1) and compare PC theory with previous approaches to contrast in OT (4.5.2).

##### **4.5.1 Faith in PC**

In standard OT, the role of faithfulness is to demand identity between inputs and their corresponding outputs. When faithfulness outranks conflicting markedness, a phonological process is blocked. With the opposite ranking, a phonological process takes place. Let us take final devoicing in standard OT. When markedness against voiced obstruents word-finally outranks conflicting faithfulness, final devoicing takes place. With the opposite ranking, there is no devoicing. The following tableaux illustrate the two cases: a language with final devoicing in (43) and one without devoicing in (44). In

each case, two types of inputs are considered: one with a voiced obstruent syllable-finally and one with a voiceless obstruent in the same position.

(43) Final devoicing

		/bud/	*VOICEDOBS] <sub>σ</sub>	IDENT(voice)
a.	☞	but		*
b.		<b>bud</b>	*!	

		/but/	*VOICEDOBS] <sub>σ</sub>	IDENT(voice)
c.	☞	but		
d.		<b>bud</b>	*!	*

(44) No final devoicing

		/bud/	IDENT(voice)	*VOICEDOBS] <sub>σ</sub>
a.		but	*!	
b.	☞	<b>bud</b>		*

		/but/	IDENT(voice)	*VOICEDOBS] <sub>σ</sub>
c.	☞	but		
d.		<b>bud</b>	*!	*

Let us consider forms with a voiced obstruent word-finally in both languages. Those are the top tableaux in each case, indicated here as (a-b). In the language with final devoicing, tableau (43), candidate (a) wins over candidate (b) since it satisfies high-ranked markedness. In a language with no final devoicing, on the other hand, tableau (44), candidate (b) is the winner. It does not change voicing specification and so satisfies high-ranked IDENT(voice). Let us now move to forms with a voiceless obstruent word-finally in both languages. Those are the bottom tableaux, indicated here as (c-d). In each case, it is the form with a voiceless obstruent word-finally, candidate (c), that wins. In consequence, in the language with final devoicing, there is neutralization of the voicing distinction on the surface. Inputs distinct in voicing word-finally, /bud/ and /but/, map

onto the same output. In the language with no final devoicing, on the other hand, the underlying voicing distinction is preserved on the surface.

PC theory takes those observations at face value and gives contrast preservation the status of an imperative in a phonological system. In PC, instead of input-output faithfulness constraints interacting with markedness constraints, neutralizations are accounted for by the relative ranking of markedness constraints and constraints on contrast. To evaluate constraints on contrast, in PC mappings are evaluated together as a system. Below I recall the PC rankings for languages with final devoicing and the lack of it (see also chapter 1, section 3). (Since in the following examples input and output-oriented PC constraints assign the same violation marks to each form, only input-oriented PC is shown in the tableaux.)

(45) Final devoicing

			*VOICEDOBS] <sub>σ</sub>	PC <sub>IN</sub> (VOICE)
a.	☞	/bud/→but /but/→but		*
b.		/bud/→bud /but/→but	*!	

(46) No final devoicing

			PC <sub>IN</sub> (voice)	*VOICEDOBS] <sub>σ</sub>
a.		/bud/→but /but/→but	*!	
b.	☞	/bud/→bud /but/→but		*

In the language with final devoicing, tableau (45), scenario (a) wins since it avoids voiced-obstruents word-finally, thus satisfying the high-ranked markedness constraint. This is at the cost of merging voicing. In the language with no final devoicing, on the other hand, tableau (46), scenario (b) is optimal since it does not merge the voicing

contrast on the surface and thus satisfies the high-ranked PC constraint. This is at the cost of having a voiced obstruent word-finally.

Both standard OT and PC theory are able to account for cases of neutralization and the lack of it but the execution is different. In standard OT, neutralizations follow solely from the interaction of markedness and faithfulness constraints. In PC theory, they are a result of the relative ranking of markedness constraints and constraints on contrast preservation. That is, while in standard OT, contrast follows solely from the way constraints interact, in PC theory it is inscribed in the constraints themselves. There are constraints on contrast in the theory, called PC constraints. Constraint ranking in PC theory is a way of blocking or activating constraints on contrast.

The two approaches differ in the range of phenomena they cover. While they both account for cases of simple neutralization and the lack of it (transparent processes), they differ with respect to opaque alternations, such as chain shifts and derived environment effects. It has been shown here that chain shifts and derived environment effects, the two types of opaque alternations, are not admitted under standard OT, but PC admits them. Therefore, by formulating contrast preservation as an imperative in a phonological system, PC theory accounts for transparent and opaque processes in a uniform manner with no additional mechanisms required. This is crucially different from standard OT.

PC constraints are the core of PC theory but they are not enough to evaluate contrast. There is more than one way to preserve contrast on the surface and all PC constraints require is to preserve contrast in some way. Therefore, to effectively compare scenarios we need something other than PC in the theory. For example, in the simple case of no neutralization in (46), contrast between forms distinct in voicing can be preserved

on the surface, as in the identity scenario, where inputs are identical to their outputs. But it can be also preserved by input-output permutations, as in the so-called permuted scenario (see chapter 1, section 3 for discussion). The permuted scenario is recalled below in (47b). As shown in the following tableau, the permuted scenario ties with the competing identity scenario on PC and markedness. I refer to this as the permuted scenario problem.

(47) Permuted scenario problem – a tie

			PC <sub>IN</sub> (voice)	*VOICED <sub>OBS</sub> ] <sub>σ</sub>
a.	identity	/bud/→bud /but/→but		*
b.	permuted	/bud/→but /but/→bud		*

The permuted scenario, scenario b, is like the identity scenario except that it switches the relation between inputs and their outputs. When compared to the identity scenario, in the permuted scenario, different outputs correspond to different inputs. The permuted scenario incurs the same markedness violations as the identity scenario since it contains the same outputs. It also satisfies contrast, just like the identity scenario (at least as far as PC(voice) goes). Therefore, it fares the same on PC and markedness as the identity scenario. But the two scenarios are clearly different.

In the permuted scenario, as opposed to the identity scenario, outputs and inputs are non-identical. However, PC constraints have nothing to say about this, unless it influences contrast preservation, which it does not in this case. Therefore, we need something else than PC to rule out the permuted scenario.

In PC theory this is the role of generalized faithfulness constraints that belong to stage 2 of Eval. Generalized faithfulness rules out unnecessary disparity between inputs

and outputs in a scenario. In this particular case, it rules out the permuted scenario. This is illustrated below.

(48) The role of generalized faithfulness

			$PC_{IN(voice)}$	$*VOICED_{OBS}]_{\sigma}$	FAITH
a.	identity	/bud/→bud /but/→but		*	
b.	permuted	/bud/→but /but/→bud		*	**!

Generalized faithfulness chooses the identity scenario as optimal. In the permuted scenario, there are two violations of generalized faithfulness, since there is voicing disparity in both mappings in the scenario. Thus, each mapping constitutes a violation.

Generalized faithfulness in PC can be seen as a descendant of standard faithfulness since it evaluates individual input-output mappings. But it is different from standard faithfulness in two respects. First, unlike standard faithfulness, it lumps together every type of faithfulness violation for a given mapping in a scenario, not differentiating between types of input-output disparity. In other words, any change in property P between inputs and their corresponding outputs constitutes as a violation of this constraint. Generalized faithfulness, therefore, can be thought of as a constraint on minimizing distance between inputs and their corresponding outputs. It evaluates disparity for outputs of a certain type.

The other difference between faithfulness in PC (generalized faithfulness) and standard faithfulness is its place in the constraint hierarchy and consequently its role in constraint interaction. Faithfulness in PC is limited in its force. It is different from other OT constraints since it is a non-permutable constraint, ranked below all other constraints

and activated only in case of a tie. This particular mode of activation of generalized faithfulness follows naturally from its placement in the constraint hierarchy. Generalized faithfulness, unlike other constraints, belongs to stage two of Eval.

The claim of PC theory is that PC constraints together with low-ranked generalized faithfulness subsume the role of standard faithfulness. They account for what standard faithfulness would and in addition cover more ground, thereby accounting for the full set of attested phenomena.

There are three types of PC constraints in the theory: input-oriented PC (also known as anti-neutralization PC), output-oriented PC and relational PC. To begin with, input-oriented PC constraints prohibit mergers of input contrasts in the output. They are satisfied even when an input contrast is realized as a different output contrast. Since input-oriented PC constraints are assigned a violation mark for any input pair distinct in property P that neutralizes in the output, they minimize the number of neutralizations in a scenario. Thus, overall, input-oriented PC constraints militate to retain underlying distinctions in some form on the surface.

Output-oriented PC constraints prohibit ambiguous outputs, by which they prohibit neutralizations (similar to input-oriented PC), but their primary role is to guard distribution of neutralizations in a scenario. Output-oriented PC constraints prefer a scenario where neutralizations are accumulated locally rather than distributed across outputs. This results in fewer ambiguous outputs. Under the assumption that fewer

ambiguous outputs increase recoverability of a scenario, output-oriented PC constraints increase recoverability.<sup>13</sup>

The third type of PC constraints proposed in the theory is relational PC. Relational PC works to retain some correspondence between output and input contrasts. An output contrast of a particular type needs to correspond to at least one instance of an identical input contrast.

Overall, PC constraints increase similarity between sets of inputs and their corresponding outputs, and in this they can be compared to standard faithfulness. But while standard faithfulness evaluates input-output mappings in isolation, PC constraints are novel in that they evaluate contrasts for pairs of underlying and surface forms. Unlike standard faithfulness, PC constraints allow underlying contrasts to be transformed into distinct output contrasts. In this way they admit phonological processes that involve contrast transformation (i.e., chain shifts). This is true for both input- and output-oriented PC. Relational PC sets limits on what types of transformations can take place.

But PC does not fully subsume faithfulness. As has become apparent, we do need some faithfulness in the theory despite PC. It is proposed that in addition to PC constraints, there are low-ranked generalized FAITHFULNESS constraints that evaluate input-output disparity for each mapping in a scenario (see chapter 1, section 2). Those constraints are more general than standard faithfulness.

In the next section I will discuss previous approaches to contrast in OT.

---

<sup>13</sup> The idea that opacity increases recoverability has been already discussed in the works of Gussmann (1976), Kaye (1974), (1975), Kisseberth (1976). Their understanding of recoverability is different from mine but the idea is similar.

#### 4.5.2 Previous Approaches to Contrast in OT

PC theory developed in this work recognizes contrast as an imperative in a phonological system. It is formalized as constraints on contrast preservation. Phonological systems are accounted for by the relative ranking of constraints on contrast preservation and conflicting markedness constraints. There are also low-ranked generalized faithfulness constraints in the theory.

Previous work on contrast in Optimality Theory includes the *Dispersion Theory of Contrast* (Flemming 1995, 1996; Padgett 1997, 2000). Both PC theory and the *Dispersion Theory* propose that contrast is an imperative in a phonological system and needs to be stated formally as a family of constraints. It does not always follow from the interaction of markedness and standard faithfulness constraints, as this interaction does not admit some empirically attested phenomena. Both theories also recognize that to evaluate contrast, candidates must be sets and not singletons as in standard OT.

The primary difference between the PC approach to contrast and previous OT approaches relates to the types of contrast constraints and the structure of candidate sets. In previous work, contrast constraints evaluate surface inventories of segments. They are formulated as competing constraints on the maximization of the number of contrasts, maximization of the distinctiveness of contrasts and minimization of the articulatory effort, all evaluated at the surface.<sup>14</sup>

In PC theory, on the other hand, constraints on contrast aim to account specifically for phonological processes. Since phonological processes either neutralize a particular contrast in some environments (transparent processes) or preserve a particular

contrast but express it in a different way than in the underlying form (opaque processes), PC theory proposes that constraints on contrast are really constraints on contrast preservation. These include anti-neutralization (input-oriented) PC, distribution (output-oriented) PC and relational PC. (See previous section for discussion.)

Since constraints on contrast are different in the two approaches, the structure of candidate sets is also different. In the *Dispersion Theory*, scenarios contain sets of possible outputs. In PC theory, on the other hand, scenarios contain sets of input-output mappings. The similarity here is that in both cases we are talking about possible words and not necessarily the existing words of the language. As for the size of scenarios, PC theory, unlike the *Dispersion Theory*, proposes that scenarios contain only a finite set of mappings, as only a finite set allows one to evaluate contrast.

#### **4.6 Conclusions**

This dissertation develops a theory of phonology in which phonological mappings are evaluated as a system and are accounted for by three distinct families of PRESERVE CONTRAST constraints (input-oriented PC, output-oriented PC and relational PC) interacting with conflicting MARKEDNESS constraints and low-ranked (non-permutable) FAITHFULNESS constraints.

The key observation is that an opaque phonological mapping, like a chain shift or a derived environment effect, involves *contrast transformation*. In an opaque mapping, a given underlying contrast is transformed into a different surface contrast at the cost of neutralizing some original instances of that surface contrast. In Finnish, an underlying

---

<sup>14</sup> But see more recent work of Padgett (2000) for the analysis of historical change in terms of contrast.

length contrast is transformed into a surface rounding contrast at the cost of neutralizing some original instances of the rounding contrast.

In cases of contrast transformation, a given underlying contrast cannot be preserved on the surface in the same way as in the underlying form due to the application of some phonological process. But, at the same time, it is required that this particular contrast be preserved on the surface in some way. In consequence, some other process is activated to retain this contrast. In Finnish, for example, shortening takes place and so the underlying length contrast cannot be preserved on the surface in the same way as in the underlying form. However, there is a high-ranked requirement in the language to preserve the length contrast in some way despite shortening. In consequence, rounding takes place, and so the underlying length contrast is transformed into surface rounding contrast.

Often the process that is activated in cases of contrast transformation, e.g., rounding in Finnish, is not due to high-ranked markedness. Therefore, we need some other way to force it. PC theory admits such mappings, because it allows a phonological process to take place solely to preserve contrast, as long as there is another process that takes place higher-up in the shift. In Finnish, in particular, PC theory admits rounding, as long as there is shortening in the same environment. Thus, rounding is indirectly forced by shortening together with the requirement on preserving contrast between forms distinct in length. We no longer need high-ranked markedness against unrounded vowels to force rounding.

In PC theory contrast preservation is an imperative in the grammar. It is formalized as competing constraints on minimizing neutralizations of input contrasts on

the surface (input-oriented PC), improving distribution of neutralizations in the system (output-oriented PC), and retaining correspondence between output and input contrasts (relational PC). Different aspects of contrast preservation take precedence in different languages.

Unlike standard faithfulness, preserve-contrast constraints allow one to retain contrasts in a way different than in the underlying form and thereby admit phonological mappings that involve contrast transformation. (This is true of both input and output-oriented PC.) The advantage of PC theory is that it accounts for both transparent and opaque processes (i.e., chain shifts) in a uniform manner with no additional mechanisms required.