CHAPTER 3

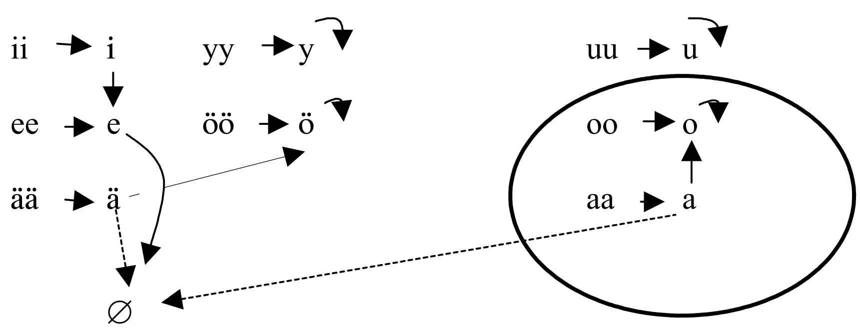
CASE STUDY: FINNISH

"No man is an Island, entire of itself;
every man is a piece of the Continent,
a part of the main…"
John Donne (1571 - 1631), Meditation XVII

3.1 Finnish Vowel Alternations[1]

The core of PC theory is the interaction between mappings. Mappings influence one another. Therefore, to understand the workings of a phonological system, we need to look at a set of vowel alternations in some environment. This chapter examines vowel alternations in Finnish before the plural suffix –i (Anttila 2000, Harrikari 2000, Keyser & Kiparsky 1984). This is illustrated below. The chain shift discussed so far is circled and represented in the context of the Finnish vowel space.[2]

(1) Vowel alternations in Finnish before -i



The following are examples of each mapping shown in the diagram. Examples are divided into sets of unrounded and rounded vowels, respectively.

[1] I would like to thank participants of the annual Finnish Festival held at the University of Minnesota in the summer of 2002 for verifying the data and providing examples.
[2] The vowel space, as shown here, represents only single segments. Diphthongs are not included.

(2) Examples of mappings in Finnish (due to Heli Harrikari (p.c.))

Unrounded vowels

	singular nom.	*gloss*	*plural essive*
/ii/→i	kallii- (kallis)	‘expensive (stem)’	kalli-i-na
/i/→e	lasi	‘glass’	lase-i-na
/ee/→e	essee	‘essay’	esse-i-nä
/e/→∅	lapse- (lapsi)	‘child’	laps-i-na
/ää/→ä	jää	‘ice’	jä-i-nä
/ä/ → ö	tekijä	‘author’	tekijö-i-nä
↘ ∅	ystävä	‘friend’	ystäv-i-nä
	kesä	‘summer’	kes-i-nä
/aa/→a	tehtaa- (tehdas)	‘factory’	tehta-i-na
/a/ → o	matka	‘trip’	matko-i-na
↘ ∅	1. Some words more than two syllables long		
	asema	‘station’	asem-i-na
	2. –u(V)Ca; -o(V)Ca		
	kuva	‘picture’	kuv-i-na

Rounded vowels

/yy/→y	revyy	‘revue’	revy-i-nä
/y/→ y	kyky	‘ability’	kyky-i-nä
/uu/→u	paluu	‘return (noun)’	palu-i-na
/u/→u	savu	‘smoke (noun)’	savu-i-na
/öö/→ö	miljöö	‘milieu’	miljö-i-nä
/ö/→ö	näkö	‘(eye)sight’	näkö-i-nä
/oo/→o	ehtoo	‘evening’	ehto-i-na
/o/→o	katto	‘roof’	katto-i-na

Let us now move on to generalizations. As shown above, long vowels shorten before the plural suffix. This is true for all vowels in Finnish, /ii/→[i], /yy/→[y] etc. The shortening of long vowels can potentially result in their neutralization with corresponding short vowels. To avoid the neutralization of length, in some cases, short vowels move away from their underlying position. They change in some feature(s). As a result, there is a chain shift effect, as the one discussed so far. To take an example, in the set of low vowels, the long low vowel shortens (e.g., /aa/→[a]) and the corresponding short vowel undergoes rounding (e.g., /a/→[o]). In consequence, the two vowels, /aa/ and /a/, do not neutralize.

Altogether, there are three chain shift effects in Finnish. They all take place in the set of unrounded vowels. I will now recall the examples with specific chain shifts indicated.[3]

(3) Finnish chain shifts in nouns

Unrounded vowels

/ii/→i	kallii-	'expensive (stem)'	kalli-i-na	**Chain shift 1:** non-low front unrounded vowels
/i/→e	lasi	'glass'	lase-i-na	
/ee/→e	essee	'essay'	esse-i-nä	
/e/→∅	lapse-	'child'	laps-i-na	

/ää/→ä	jää	'ice'	jä-i-nä	**Chain shift 2:** low front unrounded vowels
/ä/ → ö	tekijä	'author'	tekijö-i-nä	
↘ ∅	ystävä	'friend'	ystäv-i-nä	
	kesä	'summer'	kes-i-nä	

/aa/→a	tehtaa-	'factory'	tehta-i-na	**Chain shift 3:** low back unrounded vowels
/a/ → o	matka	'trip'	matko-i-na	
↘ ∅	1. Some words more than two syllables long			
	asema	'station'	asem-i-na	
	2. –u(V)Ca; -o(V)Ca			
	kuva	'picture'	kuv-i-na	

One of the chain shifts is the shift in the low back vowels discussed so far, indicated here as chain shift 3. There is also a mirror shift in the low front vowels (chain shift 2), and a shift in the high front vowels (chain shift 1). These are all illustrated below:

[3] In the discussion, I will focus on the nominal paradigm.

(4) Finnish vowel shifts

High front vowels	ii	→	i	→	e	→	∅
Low front vowels	ää	→	ä	→	ö	(or ä→∅)	
Low back vowels	aa	→	a	→	o	(or a→∅)	

This chapter accounts for each shift. (The low vowel shifts also include deletion as the last step.) [4]

Besides the three chains shifts in Finnish, there is another interesting fact about the dynamics of the Finnish vowel space. Namely, there is a clear difference between rounded and unrounded vowels. Rounded vowels do not participate in shifts, unlike unrounded vowels. That is, when long rounded vowels shorten, their short counterparts stay where they originally are and a merger takes place (e.g., /oo/ vs. /o/ → [o]). The difference in the behavior of rounded and unrounded vowels is summarized below.

[4] These are shifts in nouns. Similar mappings take place for verbs before the past tense marker –i, and for adjectives before the superlative –in. Just as in nouns, there are no shifts in the rounded vowels before the other two –i suffixes. But there are differences in the shifts themselves. The low back vowel shift (/aa/→a, /a/→o or /aa/→a, /a/→∅) is the same for nouns and verbs; adjectives do not show the rounding alternation (/aa/→a, /a/→∅ only). The low front vowel shift (/ää/→ä, /ä/→ö or /ää/→ä, /ä/→∅) is true only for nouns; verbs and adjectives do not include the rounding alternation (/ää/→ä, /ä/→∅ only). (Anttila argues that this is a tendency rather than a categorical change.) The high front vowel shift is the same for nouns and adjectives (/i/→e, /e/→∅) but in case of adjectives the long high vowel also lowers (/ii/→e); verbs show deletion of both high and mid short vowels but no shift (/i/, /e/→∅).

Overall, nouns represent the biggest variety of shifts. This supports the observation that nouns are often more faithful than other grammatical categories (Smith 1997); they preserve more contrasts. In the remainder of this chapter, I will concentrate on nouns.

(5) Rounding as factor in chain-shifting

	Unrounded				Rounded			
Short▶ Long▼	i	e	ä	a	y	ö	u	o
ii	*							
ee	√	*						
ää			*					
aa				*				
yy					√			
öö			√			√		
uu							√	
oo				√				√

√'s: length mergers are allowed
stars: length mergers are banned
shaded cells: length mergers are not attested

The length contrast is neutralized in the rounded vowel set (yy/y, öö/ö, uu/u, oo/o). But unrounded short vowels do not merge with long vowels of the same height (*aa/a, *ee/e, *ii/i, *ää/ä). Instead, they participate in shifts. I will deal with rounding in chain-shifting in section 3.4.

Finally, in each of the low vowel shifts, sometimes deletion takes place instead of rounding (Antilla 2000). When deletion takes place, the low vowel shift is of the form shortening *plus* deletion instead of shortening *plus* rounding.

(6) Schematically, deletion versus rounding

aa	→	**a**	→	∅	(deletion)	versus
aa	→	**a**	→	**o**	(rounding)	

The following are examples of deletion. Compare with rounding in (8).

(7) Deletion (/a/→∅)

singular	*gloss*	*plural essive*
asema	'station'	asem-i-na
ravintola	'restaurant'	ravintol-i-na
kukka	'flower'	kukk-i-na

(8) Rounding (/a/→o)

singular	*gloss*	*plural essive*
matka	'trip'	matko-i-na
raha	'money'	raho-i-na
aika	'time'	aiko-i-na

I will follow Antilla in assuming that deletion is a result of labial dissimilation. Deletion will be discussed in detail in section 3.2.

The goal of this chapter is to give a global view of vowel alternations in Finnish. The chapter is organized as follows. Deletion in the low vowel shift (chain shift 2 and 3) is discussed in section 3.2. The shift in the front high vowels (chain shift 1) is discussed in section 3.3. Finally, the chapter concludes with a discussion in section 3.4 of the difference between rounded and unrounded vowels and their participation in shifts in Finnish.

3.2 Chain Shift with Deletion

In the low vowel shift discussed so far, shortening forces rounding. The short low vowel undergoes rounding to avoid a merger with a long vowel that shortens, /a/→o because /aa/→a. But in some cases, the low vowel deletes instead of rounding, /a/→∅ instead of /a/→o. Deletion takes place if the preceding vowel is rounded ([o] or [u] or equivalent long vowel) or when the immediately preceding consonant is a labial ([m], [p], [b], [v], [f]).[5] Some examples of deletion follow. These are examples of deletion in both nouns and verbs. Next to each form, there is an incorrect output with a rounded vowel.

[5] Antilla (2000) shows that the effects of deletion are quantitative. The effects are strongest when both the consonant and the preceding vowel are rounded. Antilla also notices the effect of foot structure on rounding dissimilation. Dissimilation is more consistent when the trigger and target of dissimilation belong to the same foot.

(9) Deletion before –i (/a/→∅)

singular	*gloss*	*plural essive*	
ase**ma**	'station'	ase**m**∅-i-na	*ase**mo**-i-na
ravint**ola**	'restaurant'	ravint**o**l∅-i-na	*ravint**olo**-i-na
present		*past*	
matk**u**st**a**-n	'to travel'	matk**u**st-i-n	*matk**u**st**o**-i-n
ot**a**-n	'to take from'	**o**t-i-n	***o**t**o**-i-n

I will claim, following Antilla (2000), that deletion is a result of labial (rounding) dissimilation. Deletion takes place to avoid two adjacent labial articulations, which would be a result of rounding. In the examples above, the vowel that deletes and the labial that causes dissimilation are in bold font. In dissimilation, it is important to state the domain for co-occurrence restrictions. In Finnish, both vowels and consonants cause dissimilation. For dissimilation to take place, the labial consonant needs to be the immediately preceding consonant, whereas the vowel needs to be in the nucleus of the preceding syllable. In what follows, I will state the domain uniformly as adjacent segments, where adjacency is defined separately at the vocalic and consonantal tier (Gafos 1996, NiChiosain & Padgett 1997).[6]

The OCP constraint militates against having two labial (=round) segments in the same domain. There are various ways to formalize it. To use the existing tools of OT, one can formalize it as self-conjunction of markedness constraints (following Alderete 1997). I call it OCP_{Adj}(round) in traditional autosegmental terms. This does not have any effect on the analysis here. The relevant OCP constraint (Leben 1973, McCarthy 1986) is defined below:

(10) OCP_{Adj}(round)
Adjacent round (labial) articulations are prohibited.

Dissimilatory deletion is limited to derived labial sequences in Finnish (as will be discussed later, underlying /o/ and shortened /oo/ do not delete). Other cases of labial dissimilation are Palauan (Finer 1986), Akkadian (Von Soden 1969, McCarthy 1979) etc.

The argument is as follows. Deletion, just like rounding, preserves some length contrasts despite shortening, /asemaa-i-na/ vs. hypothetical /asema-i-na/ → [asema-i-na] (with shortening) vs. [asem-i-na] (with deletion). Rounding is a preferred way to preserve length contrasts (it takes place in more instances; after any vowel except for [o, u, oo, uu] and labial consonants) but deletion takes place if rounding would result in an OCP_{Adj}(round) violation.[7]

Let us review the argument for rounding. Formally, long vowels are avoided at the cost of merging some length contrasts. But length neutralizations are accumulated in Finnish at the cost of merging rounding. This is the ranking already established and recalled below:

(11) $*\sigma_{\mu\mu\mu}$ >> PC_{OUT}(lg), PC_{IN}(lg) >> PC_{OUT}(rd), PC_{IN}(rd)

---

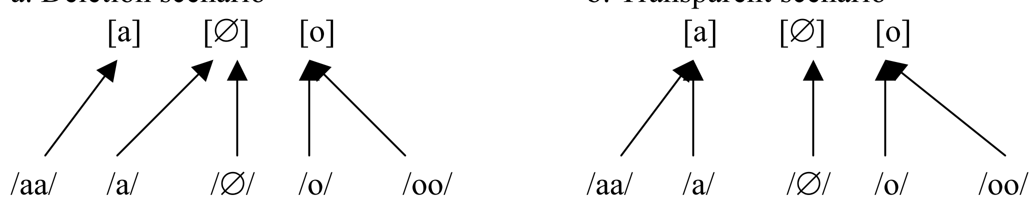
6 The presence of a labial further away in the stem does not cause dissimilation i.e., /**p**al**a**-i/→**p**al**o**i 'burn-PAST', /a**p**in**a**-i-ssa/→a**p**in**o**issa 'monkey-PL-INE', /**m**atk**a**-i-ssa/→**m**atk**o**issa 'trip', /l**u**sikk**a**-i-ssa/→l**u**sik**o**issa 'spoon'. Compare with deletion in: /m**u**n**a**-i-ssa/→m**u**nissa 'egg', /ase**ma**-i-ssa/→ase**m**issa 'railroad station'.

7 Antilla (2000) argues that both deletion and rounding are motivated to resolve OCP violations. Rounding is blocked after labials since it would create an OCP(round) violation but deletion is blocked after coronals since it would create an OCP(high) violation; a coronal would be adjacent to a high front vowel if deletion took place. Here I am taking a different route: rounding is a default and deletion takes place only in some instances. In this proposal, the fact that rounding is a default follows from constraint ranking. It is predicted by constraint permutation that there are languages where deletion is preferred to rounding. No claims are being made about the universality of the relative ranking of constraints on contrast preservation.

In short, high-ranked markedness $*\sigma_{\mu\mu\mu}$ forces some length mergers, but length mergers need to be accumulated in a scenario rather than distributed across outputs, and this is at the cost of rounding mergers. This explains why rounding takes place.

Like rounding, deletion accumulates length mergers in a scenario and thus improves on PC_{OUT}(long) constraint. Compare the following two scenarios. Both have shortening, but only the actual scenario, scenario A, shows deletion.

(12) a. Deletion scenario b. Transparent scenario



Scenario A has only one output ambiguous in length, the [o] output, whereas the competing transparent scenario has two such outputs, [a] and [o]. It is assumed here that deletion does not incur a length merger. Deletion results in a merger of contrast with words that end in a consonant to begin with, /asem**a**-i-na/ vs. /asem-i-na/ → asem-i-na. Some examples of mappings for consonant-final stems follow. [8]

(13) Consonant-final stems

singular nom.	*gloss*	*plural essive*	*stem*
kuus(i)	‘spruce tree’	kuus-i-na	kuus-
vuos(i)	‘year’	vuos-i-na	vuot-
huol(i)	‘trouble, worry’	huol-i-na	huol-
sisar	‘sister’	sisar-i-na	
present		*past*	
pur(e)n	‘to bite’	pur-i-n	
kävel(e)n	‘to walk’	kävel-i-n	
pääs(e)n	‘to get to’	pääs-i-n	
pan(e)n	‘to put’	pan-i-n	

[8] These are possible but not necessarily the existing forms of the language. This follows from the architecture of the theory; see chapter 1 for discussion. Consonant stems in Finnish can only end in an alveolar consonant: *l,r,n,s* and *t*. The input /asem-i-na/ mapping onto [asem-i-na] comes from richness of the base. It is used here to illustrate the merger with words where the /a/ vowel deletes.

I am assuming here that deletion is a violation of a separate PC constraint, $PC_{IN/OUT}(V/\varnothing)$.

For the scenario with deletion to win, it must be more important to have fewer outputs ambiguous in length than it is to avoid deletion. Therefore, constraints militating against mergers of the V/∅ contrast must be ranked lower than the PC_{OUT}(long) constraint. Altogether, the ranking is as follows:

(14) $*\sigma_{\mu\mu\mu}$ >> PC_{OUT}(lg), PC_{IN}(lg) >> $PC_{OUT}(V/\varnothing)$, $PC_{IN}(V/\varnothing)$

We now move a step further and try to account for the choice between deletion and rounding to aid length. Even though in Finnish both deletion and rounding accumulate length mergers, rounding mergers are preferred to deletion. Formally:

(15) $PC_{IN/OUT}(V/\varnothing)$ >> $PC_{IN/OUT}$(round)

It is more important to avoid mergers of the V/∅ type than it is to avoid merges of rounding.

Nevertheless, deletion is chosen as a repair when rounding would result in an OCP violation. Deletion avoids an OCP_{Adj}(round) violation by deleting the vowel. To ensure that deletion takes place when OCP_{Adj}(round) is at stake, the constraint militating against deletion must be ranked lower than the OCP_{Adj}(round) constraint. Thus:

(16) OCP_{Adj}(round) >> $PC_{IN/OUT}(V/\varnothing)$

It is more important to avoid an OCP_{Adj}(round) violations than it is to avoid deletion.

It is important to note that some OCP violations do exist due to shortening. When the long mid vowel shortens, as in /rokokoo-i-na/→[rokokoina] ‘rococo’, it incurs some

OCP_{Adj}(round) violations. To allow for these violations, it must be more important to shorten the vowel than it is to avoid OCP violations.

(17) $*\sigma_{\mu\mu\mu}$ >> OCP_{Adj}(round)

The ranking established so far is illustrated in the following tableaux. The first tableau contrasts a chain-shift scenario with deletion and a chain-shift scenario with rounding. The basic idea is that deletion takes place to avoid OCP_{Adj}(round) violations which would be a result of rounding.

(18) Deletion instead of rounding[9]

	Scenarios	OCP_{Adj}(round)	$PC_{IN/OUT}$(V/∅)	$PC_{IN/OUT}$ (rnd)
☞	A. CHS with deletion /asem**oo**+i+na/ → asem**o**+i+na /asem**o**+i+na/→ asem**o**+i+na /asem**aa**+i+na/ → asem**a**+i+na /asem**a**+i+na/ → asem∅+i+na /asem∅+i+na/ → asem∅+i+na	(**) oo, o → o	** PC_{IN}: {/a/,/∅/} PC_{OUT}: [∅]	
	B. CHS with rounding /asem**oo**+i+na/ → asem**o**+i+na /asem**o**+i+na/ → asem**o**+i+na /asem**aa**+i+na/ → asem**a**+i+na /asem**a**+i+na/ → asem**o**+i+na /asem∅+i+na/ → asem∅+i+na	(**)*! oo, o, a → o		*** PC_{IN}: {/o/,/a/}, {/oo/,/a/} PC_{OUT}: [o]

Both scenarios incur some OCP violations due to shortening and keeping underlying /o/ as such but the scenario with deletion, scenario A, avoids additional OCP violation, which would be a result of rounding. Therefore, the scenario with deletion is optimal. The relative ranking of PC constraints is irrelevant in this case, since the choice is already made on OCP_{Adj}(round) that outranks both of the PC constraints.

[9] It is important to note that the words shown in a scenario are not all the existing words of the language. As far as constraints go, for reasons of space, input and output-oriented PC constraints are represented together. Violations are explained in columns below the constraints.

If there were no OCP$_{Adj}$(round) violation, rounding would be chosen over deletion. In that case, the OCP constraint would be inactive and the choice would be made on the relative ranking of the PC constraints. This is illustrated in the following tableau.

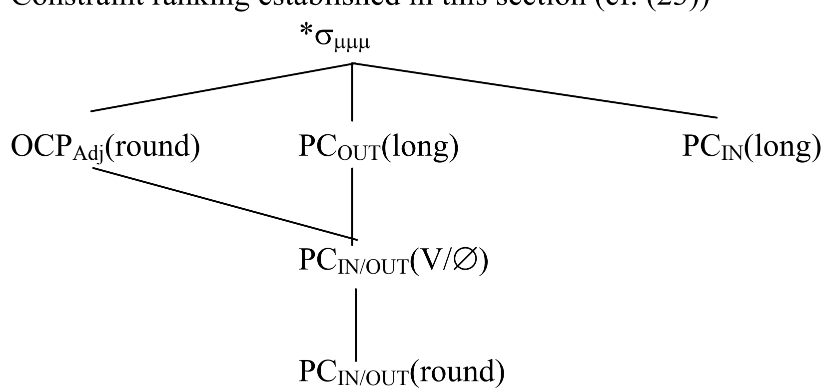
(19) Rounding otherwise

	Scenarios	OCP$_{Adj}$(round)	PC$_{IN/OUT}$(V/∅)	PC$_{IN/OUT}$ (round)
	A. CHS with deletion /val**oo**+i+na/ → val**o**+i+na /val**o**+i+na/ → val**o**+i+na /val**aa**+i+na/ → val**a**+i+na /val**a**+i+na/ → val∅+i+na /val∅+i+na/ → val∅+i+na		**! PC$_{IN}$: {/a/,/∅/} PC$_{OUT}$: [∅]	
☞	B. CHS with rounding /val**oo**+i+na/ → val**o**+i+na /val**o**+i+na/ → val**o**+i+na /val**aa**+i+na/ → val**a**+i+na /val**a**+i+na/ → val**o**+i+na /val∅+i+na/ → val∅+i+na			*** PC$_{IN}$: {/o/,/a/}, {/oo/,/a/} PC$_{OUT}$: [o]

The chain shift with rounding wins since it incurs a violation of a lower ranked PC constraint.

The ranking is as follows:

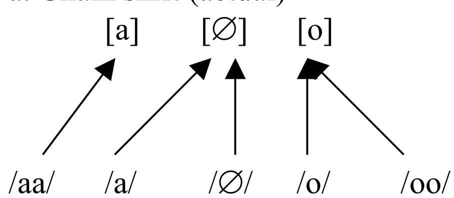
(20) Constraint ranking established in this section (cf. (23))



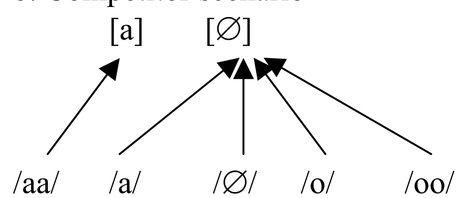
In essence, rounding neutralizations are preferred to neutralizations in presence versus absence of a segment but rounding is not an option when it would result in an OCP_{Adj}(round) violation. In that case, deletion takes place instead.

There are other possible alternatives to the actual scenario not considered so far. One of them is a scenario where the long mid vowel /oo/ and its short counterpart /o/ delete. Such a scenario satisfies both high-ranking markedness constraints, $*\sigma_{\mu\mu\mu}$ and OCP_{Adj}(round). It contains no long vowels and incurs no OCP_{Adj}(round) violations. Therefore, as the constraint ranking stands right now, such a scenario wins over the actual scenario. Consider the actual scenario versus its competitor.

(21) a. Chain shift (actual)



b. Competitor scenario

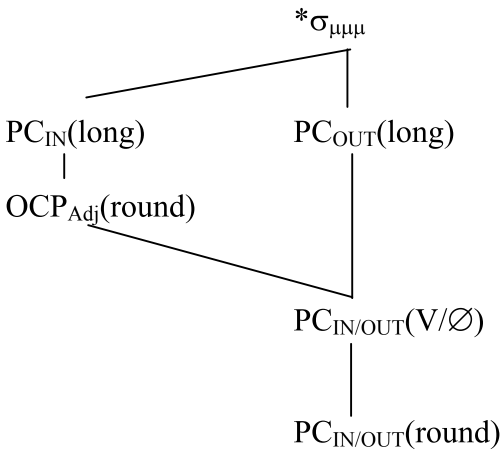


The competitor scenario wins on markedness since it does not violate either $*\sigma_{\mu\mu\mu}$ or OCP_{Adj}(round), but to do so, it merges two input pairs in length, {/a/, /oo/} and {/o/, /oo/}. In the actual scenario, on the other hand, there are two OCP_{Adj}(round) violations, /oo/, /o/ $\rightarrow$ [o], but only one length merger, {/oo/, /o/}. To ensure that the actual scenario wins, it must be more important to minimize the number of inputs that merge in length than it is to avoid OCP_{Adj}(round) violations. In terms of constraints, PC_{IN}(long) dominates OCP_{Adj}(round).

(22) PC_{IN}(long) >> OCP_{Adj}(round)

The logic is as follows. Even though OCP_{Adj}(round) violations are avoided, some violations are compelled by high-ranking PC_{IN}(long) constraint. Altogether, the constraint ranking is as follows:

(23) Final constraint ranking



(24) Ranking arguments

Ranking	Consequence
$*\sigma_{\mu\mu\mu}$ >> $PC_{IN/OUT}$(long)	Long vowels are avoided by shortening.
PC_{OUT}(long) >> $PC_{IN/OUT}$(V/∅)	Length mergers are accumulated by deletion.
PC_{OUT}(long) >> $PC_{IN/OUT}$(round)	Length mergers are accumulated by rounding.
$PC_{IN/OUT}$(V/∅) >> $PC_{IN/OUT}$(round)	Rounding is better than deletion.
OCP_{Adj}(round) >> $PC_{IN/OUT}$(round)	Deletion takes place to satisfy OCP.
PC_{IN}(long) >> OCP_{Adj}(round)	Some OCP violations exist to minimize the number of inputs that neutralize length.

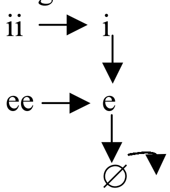
An alternative way to account for surface OCP_{Adj}(round) violations would be to ensure that a scenario can contain mergers of deletion or rounding but not both. The scenario that avoids OCP_{Adj}(round) violations, unlike the actual scenario, contains some mergers of both contrasts.

3.3 High Front Vowel Shift

There is also a shift in the high front vowels in Finnish. The long high vowel shortens (/ii/→i), the short high vowel lowers (/i/→e) and the short mid vowel deletes

(/e/→∅). Thus, there is a chain shift effect of the form: ii→i→e→∅, as represented below. There is also another mapping that participates in the shift: the long mid vowel /ee/ that shortens to [e].

(25) High front vowel shift



Some examples follow.

(26) Examples of the high front vowel shift (p.215)

	singular	*plural essive*	
/ii/→i	saalii- (salis)	saali-i-na	'a catch'
	kallii- (kalis)	kalli-i-na	'expensive, dear'
	kaunii- (kaunis)	kauni-i-na	'beautiful, lovely'
/i/→e	kuppi	kupe-i-na	'cup'
	teevati	teevate-i-na	'saucer'
	koti	kote-i-na	'home'
	lasi	lase-i-na	'glass'
/ee/→e	huonee- (huone)	huone-i-na	'room'
	kirjee- (kirje)	kirje-i-nä	'letter'
	lainee- (laine)	laine-i-na	'glove'
/e/→∅	nime- (nimi)	nim-i-nä	'name'
	lapse- (lapsi)	laps-i-na	'child'

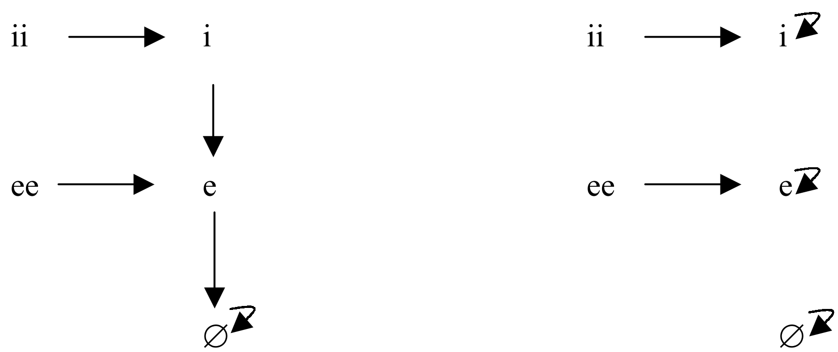
The shift originates with vowel shortening. The long vowels /ii/ and /ee/ shorten to avoid tri-moraic syllables (this is true of any other long vowel in Finnish in the same environment). In response, the short high vowel lowers (/i/→e) and the short mid vowel deletes (/e/→∅). In terms of contrasts, there is a merger in presence vs. absence of a segment (/e/ vs. /∅/ → ∅). This is where the shift terminates. There is also a merger of length and height at the point where the long mid vowel shortens (/ee/ vs. /i/ → e).

3.3.1 The Analysis

Let us now proceed to the analysis. In Finnish, long vowels are avoided due to high-ranked markedness constraint against tri-moraic syllables, $*\sigma_{\mu\mu\mu}$. Due to shortening, there are some length mergers in a scenario. But length mergers are minimized by lowering and deletion. If the short vowels [i] and [e] stayed as they are in the underlying form, there would be more length mergers in a scenario. The short vowels would merge with identical long vowels that shorten. As a result, there would be more input pairs that merge in length and there would be more outputs ambiguous in length. Compare the actual scenario to a scenario with no deletion and no lowering.

(27) Fewer length mergers

a. Chain shift (actual) b. Competitor scenario



Let us start with the competitor scenario, scenario B. In this scenario, there are two input pairs that neutralize in length, /ii/ vs. /i/→[i] and /ee/ vs. /e/→[e], and there are two outputs that correspond to inputs contrasting in length, [e] and [i]. No other mergers take place in this scenario. In the actual scenario, on the other hand, lowering and deletion result in some V/∅ and height mergers (/e/ vs. /∅/ → [∅], /ee/ vs. /i/ → [e]). But, due to lowering and deletion, there is only one input pair that neutralizes in length (/ee/ vs. /i/→[e]) instead of two such pairs as in the competitor scenario (/ii/ vs. /i/→[i], /ee/ vs. /e/→[e]). There is also only one output ambiguous in length in the actual scenario in comparison to two such outputs in the competitor scenario. To ensure that the actual scenario wins over the competitor scenario, it must be more important to minimize length neutralizations than to avoid height and V/∅ mergers. Altogether, the ranking is as follows:

(28) $*\sigma_{\mu\mu\mu}$ >> $PC_{OUT/IN}$(long) >> $PC_{OUT/IN}$(high), $PC_{OUT/IN}$(V/∅)

(29) Ranking arguments

Ranking	Consequence
$*\sigma_{\mu\mu\mu}$ >> $PC_{OUT/IN}$(long)	Long vowels shorten at the cost of length mergers
$PC_{OUT/IN}$(long) >> $PC_{OUT/IN}$(high), $PC_{OUT/IN}$(V/∅)	Length mergers are minimized by lowering and deletion

This is illustrated in the following tableau. The actual chain-shift scenario, scenario A, is compared to a competitor scenario with no lowering and no deletion, scenario B, and to an identity scenario with no shortening, scenario C.

(30) Illustration

	Scenarios	$*\sigma_{\mu\mu\mu}$	$PC_{IN/OUT}$ (long)	$PC_{IN/OUT}$ (high)	$PC_{IN/OUT}$ (V/∅)
☞	A. Actual /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na/ → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**∅**+i+na /las**∅**+i+na / → las**∅**+i+na		** PC_{IN}: {/ee/,/i/} PC_{OUT}: [e]	** PC_{IN}: {/i/,/ee/} PC_{OUT}: [e]	** PC_{IN}: {/e/,/∅/} PC_{OUT}: [∅]
	B. Shortening only /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las**i**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**e**+i+na /las**∅**+i+na / → las**∅**+i+na		****! PC_{IN}: {/i/,/ii/}, {/e/,/ee/} PC_{OUT}: [i], [e]		
	C. Identity /las**ii**+i+na/ → las**ii**+i+na /las**i**+i+na/ → las**i**+i+na /las**ee**+i+na / → las**ee**+i+na /las**e**+i+na / → las**e**+i+na /las**∅**+i+na / → las**∅**+i+na	**! [ii], [ee]			

The actual scenario wins. It avoids long vowels and thus satisfies the markedness constraint against tri-moraic syllables. It also minimizes length mergers and thus fares better on $PC_{IN/OUT}$(long) constraints than the competitor scenario with no lowering and no deletion, scenario B.

3.3.2 Problems

Lowering and deletion limit the number of length mergers in a scenario, but there are other scenarios in the same candidate set that win under the current constraint ranking. Consider the following scenarios: scenario B with no deletion and C-E, with no lowering. Three of the scenarios shown below, C-E, are problematic given the constraint ranking established so far.

(31) Competing scenarios

	Scenarios	*$\sigma_{\mu\mu\mu}$	$PC_{IN/OUT}$ (long)	$PC_{IN/OUT}$ (high)	$PC_{IN/OUT}$ (V/∅)
☞	A. Actual /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na/ → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na		** PC_{IN}: {/i/,/ee/} PC_{OUT}: [e]	** PC_{IN}: {/i/,/ee/} PC_{OUT}: [e]	** PC_{IN}: {/e/,/∅/} PC_{OUT}: [∅]
	B. No deletion /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**e**+i+na /las∅+i+na / → las∅+i+na		***! PC_{IN}: {/i/,/ee/}, {/e/,/ee/} PC_{OUT}: [e]	*** PC_{IN}: {/i/,/ee/}, {/i/,/e/} PC_{OUT}: [e]	
	C. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las**i**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na		** PC_{IN}: {/ii/,/i/} PC_{OUT}: [i]		** PC_{IN}: {/e/,/∅/} PC_{OUT}: [∅]
	D. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las∅+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**e**+i+na /las∅+i+na / → las∅+i+na		** PC_{IN}: {/ee/,/e/} PC_{OUT}: [e]		** PC_{IN}: {/i/,/∅/} PC_{OUT}: [∅]
	E. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las∅+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na			** PC_{IN}: {/i/,/e/} PC_{OUT}: [∅]	*** PC_{IN}: {/i/,/∅/}, {/e/,/∅/} PC_{OUT}: [∅]

Scenario B with lowering but no deletion loses on $PC_{IN/OUT}$(long). It merges too many length contrasts compared to the actual scenario, scenario A. Thus it is ruled out under the current constraint ranking. The other scenarios, scenarios C-E, are problematic. Scenarios C and D with deletion but no lowering, unlike the actual scenario, incur no height mergers. They satisfy $PC_{IN/OUT}$(high). Since they otherwise have the same violations as the actual scenario, given the current constraint inventory, they harmonically bind the actual scenario. They have a subset of the violation marks of the actual scenario. Scenario E, the one where both /i/ and /e/ undergo deletion, does not merge length at all and so wins under the current constraint ranking. It satisfies $PC_{IN/OUT}$(long). In what follows, I will first discuss scenario E (with no length mergers) and then scenarios C-D (with no height mergers). I will propose a possible solution to the problem.

3.3.2.1 Limit on Deletion

Let us start with scenario E. There are no length mergers in this scenario; both /i/ and /e/ delete. Under the current constraint ranking, scenario E should win over the actual scenario since it satisfies $PC_{IN/OUT}$(long). To rule out scenario E, we need a constraint ranked above $PC_{IN/OUT}$(long).

In comparison to the actual scenario, in scenario E there are more neutralizations of the V/∅ contrast; more input pairs merge in V/∅. As already established, deletion takes place to avoid neutralizations in length. Thus, to rule out scenario E, we need to put a limit on how many V/∅ contrasts can be neutralized to preserve length.

Among the constraints introduced so far, there is a constraint that plays just that role. It is the relational PC constraint (see (30), chapter 1). The relational PC constraint, as defined in chapter 1, demands that an output contrast in P correspond to at least one instance of an input contrast in P. In other words, it prohibits absolute neutralization (if there is an output P contrast, some input P contrasts need to be retained).

Let us take this idea a step further and assume that there is a family of relational PC constraints, where relational PC constraints differ on how many instances of a given P contrast need to be preserved in a scenario.

(32) PC-REL n (P)
An output contrast in P needs to correspond to at least n minimal instances of a given input contrast in P.

(33) PC-REL family (fixed ranking)
PC-REL (P) >> PC-REL 2 (P) >> PC-REL 3 (P) …

The constraints are in a fixed order, that is, if for a given property P, PC-REL 3 is satisfied, PC-REL 2 and PC-REL are satisfied as well. If three instances of a given contrast are preserved, then two instances are preserved as well etc. PC-REL 3, in particular, demands that an output contrast in P corresponds to at least three instances of input contrast in P. In other words, it demands preservation of at least three P contrasts from the input. To summarize, deletion is a good way to minimize length mergers but it cannot be overused. [10]

Consider the actual scenario versus the scenario with no length mergers. The constraint ranking is as follows:

(34) PC-REL3(V/∅) >> $PC_{IN/OUT}$(long) >> $PC_{IN/OUT}$(high), $PC_{IN/OUT}$(V/∅)

The following tableau illustrates the ranking.

[10] Another way to put limit on neutralizations of a given contrast is to propose locally-conjoined PC constraints. See chapter 1 for a discussion of this alternative. There exists a parallel between the two alternatives: locally-conjoined PC constraints and a family of PC-REL constraints.

(35) The role of relational PC

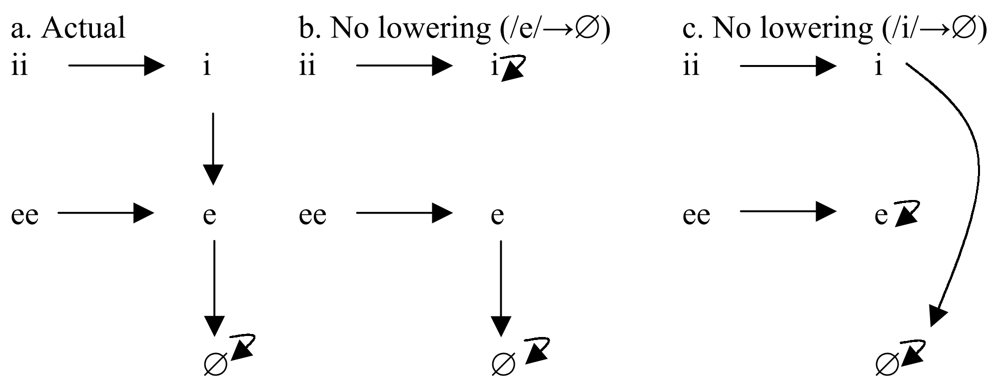
	Scenarios	*$\sigma_{\mu\mu\mu}$	PC-REL3(V/∅)	$PC_{IN/OUT}$ (long)	$PC_{IN/OUT}$ (high)	$PC_{IN/OUT}$ (V/∅)
☞	A. Actual /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na/ → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+I+na			**	** PC_{IN}: {/i/,/ee/} PC_{OUT}: [e]	** PC_{IN}: {/e/,/∅/} PC_{OUT}: [∅]
	B. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las∅+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na		*!		**	***

The scenario with no length mergers, scenario B, loses on the PC-REL3(V/∅) constraint since it preserves fewer than three V/∅ contrasts from the input. In this scenario, only /ii/ vs. /∅/ and /ee/ vs. /∅/ are preserved.

3.3.2.2 No Minimal Mergers

In tableau (31) we have seen two other scenarios that are problematic given our current inventory of constraints. These are scenarios with deletion but no lowering where only one pair of vowels undergoes deletion. We have seen that these scenarios harmonically bind the actual scenario. In the diagrams below, the two scenarios are represented together with the actual scenario.

(36)



So far, scenarios B and C win since they incur no height mergers and are otherwise identical to the actual scenario. They have a subset of the violation marks of the actual scenario.

But there is another property, besides no height mergers, that distinguishes the two competing scenarios from the actual scenario. The difference is that in the two "no lowering" scenarios there are minimal mergers in length (/ii/ vs. /i/ → [i], /ee/ vs. /e/ → [e]) whereas in the actual scenario length merger takes place between segments that are distinct on more than one dimension (/ee/ vs. /i/ → [e]).

PC constraints, as stated right now, do not distinguish between minimal and non-minimal neutralizations. They treat the two the same. In fact, merging non-minimal contrasts implies more types of mergers and thus violates more types of PC constraints.

It seems that in Finnish, merging minimal length contrasts is worse than merging length contrasts that are non-minimal. To take the idea a step further, it seems that merging within the set of P properties (here, [+/- high]) is worse than merging across sets (for example, [+high] with [-high]). One way to capture this observation is to propose domain-specific PC constraints in addition to general PC constraints. (In the next section we are going to see further support for this proposal.):

(37) PC_{IN}(Property/Domain)
inputs that belong to domain D, where D consists of inputs identical in some property Q, e.g., +/-high, that contrast in property P, e.g., +/-long, need to contrast in the output.

In Finnish, the relevant domain-specific PC constraint is as follows:

(38) PC_{IN}(long/high)
inputs of the same height (high vowels, mid vowels etc.) that contrast in length need to contrast in the output.
"do not merge length within the set of vowels of the same height"

This constraint prohibits both minimal mergers and mergers between vowels that have at least Q property in common (not necessarily minimally distinct in Q). For example, it prohibits length mergers among vowels that are of the same height but of different rounding etc. Section 3.4 discusses it in more detail.

Limiting our discussion to front vowels, the above constraint is satisfied by mergers of length among vowels of different height. Merging length among vowels of different height is better than merging length among vowels of the same height.

(39) No minimal length mergers

	Scenarios	$PC_{IN/OUT}$ (long)	PC_{IN} (long/high)	$PC_{IN/OUT}$ (high)	$PC_{IN/OUT}$ (V/∅)
☞	A. Actual /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na/ → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na	**		** PC_{IN}: {/i/,/ee/} PC_{OUT}: [e]	** PC_{IN}: {/e/,/∅/} PC_{OUT}: [∅]
	B. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las**i**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las∅+i+na /las∅+i+na / → las∅+i+na	**	*! {/ii/,/i/}		**
	C. No lowering /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las∅+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**e**+i+na /las∅+i+na / → las∅+i+na	**	*! {/ee/,/e/}		**

Scenarios B and C are ruled out since they violate the domain-specific PC constraint. They merge length between /ii/ vs. /i/ and /ee/ vs. /e/, respectively.

To summarize, long vowels are shortened in Finnish to avoid tri-moraic syllables. This results in some length mergers in a scenario, but length mergers are minimized by lowering and deletion. Two claims have been made about those processes: (i) deletion cannot avoid all length mergers (this is due to the relational PC constraint), and (ii) length mergers cannot take place within the set of vowels of identical height (this is due to the domain specific input-oriented PC constraint). Altogether, the ranking is as follows:

(40) Final constraint ranking

$PC_{REL}{}^{3}(V/\varnothing)$ $\quad$ $*\sigma_{\mu\mu\mu}$

$PC_{IN}(long/high)$ $\quad$ $PC_{IN/OUT}(long)$

$PC_{IN/OUT}(high)$ $\quad$ $PC_{IN/OUT}(V/\varnothing)$

(41) Ranking arguments

<u>Ranking</u>	<u>Consequence</u>
$*\sigma_{\mu\mu\mu} >> PC_{IN/OUT}(long)$	Shortening causes length mergers.
$PC_{IN/OUT}(long) >> PC_{IN/OUT}(high), PC_{IN/OUT}(V/\varnothing)$	Length mergers are minimized by lowering and deletion.
$PC_{IN}(long/high) >> PC_{IN/OUT}(high)$	Lowering avoids minimal length mergers.
$PC_{REL}\ 3\ (V/\varnothing) >> PC_{IN/OUT}(long)$	There is a limit put on deletion at the cost of length mergers.

3.4 Rounded Vowels

Finally, in Finnish there is a difference between rounded and unrounded vowels. Only unrounded vowels participate in shifts.

(42) Rounding as factor in chain-shifting (recalled from (5))

	Unrounded				Rounded			
Short► Long▼	i	e	ä	a	y	ö	u	o
ii	*							
ee	√	*						
ää			*					
aa				*				
yy					√			
öö			√			√		
uu							√	
oo				√				√

√'s: length mergers are allowed
stars: length mergers are banned
shaded cells: length mergers are not attested

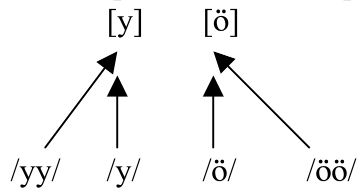
In terms of contrasts, length contrasts are merged within the set of rounded vowels (yy/y, öö/ö, uu/u, oo/o), and in some cases across rounded/unrounded vowel set (ä/öö, a/oo), but preserved within the set of unrounded vowels of the same height (ii/i, ee/e, ää/ä, aa/a).[11]

The logic is as follows. Long vowels are avoided at the cost of merging length contrasts. But length contrasts need to be re-distributed in a scenario and this is at the cost of merging rounding. That is what accounts for the chain shift effect (/a/→[o] because /aa/→[a]). This is the core of the analysis proposed so far. But rounded vowels behave differently. Rounded vowels shorten, just like unrounded vowels, but they do not re-distribute length contrasts (/y/→[y] even though /yy/→[y]).

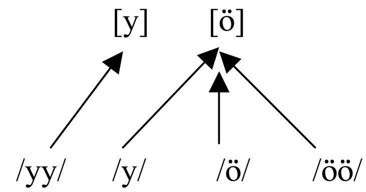
[11] In Finnish, /ee/ merges with /i/ (both unrounded vowels), so there is a merger of length in unrounded vowels but only if the two differ in height.

Consider the following scenarios with rounded vowels. Scenario A is the actual scenario in Finnish with shortening. Scenario B is a competing chain-shift scenario with shortening and lowering in which contrast is preserved between /yy/ and /y/.

(43) a. Transparent scenario (optimal)



b. Chain-shift scenario



In both, long vowels shorten, but in the chain-shift scenario, the mid vowel also lowers (/y/→[ö]). Consequently, in the chain-shift scenario, there is only one output ambiguous in length, the [ö] output. In the actual scenario, there are two such outputs. In Finnish, it is important to minimize the number of outputs ambiguous in length and therefore in the set of unrounded vowels, a chain shift wins over the competing transparent scenario. So why is the chain shift not optimal for rounded vowels?

One way to approach this problem is to propose that height contrasts need to be maintained within the set of rounded vowels in Finnish unlike in the set of unrounded vowels (for similarities with positional faithfulness theory see Beckman 1997). Formally, this can be achieved by proposing a high-ranking specific PC constraint on preserving height contrasts within the set of rounded vowels that would dominate output-oriented PC(long). Thus, it is more important to maintain height distinctions within the set of rounded vowels than it is to limit the number of outputs ambiguous in length.[12]

(44) PC_{IN}(high/+round)
Height contrasts need to be maintained within the set of rounded vowels.

(45) PC_{IN}(high/+round) >> $PC_{IN/OUT}$(long) >> $PC_{IN/OUT}$(high)

In effect, height mergers are not allowed within the set of rounded vowels but permitted elsewhere.

(46) The role of rounding in chain-shifting

	Scenarios	PC$_{IN}$ (high/+round)	PC$_{IN/OUT}$ (long)	PC$_{IN/OUT}$ (high)
☞	A. Chain shift (actual) /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na/ → las**e**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**∅**+i+na /las∅+i+na / → las∅+I+na		** PC$_{IN}$: {/i/,/ee/} PC$_{OUT}$: [e]	** PC$_{IN}$: {/i/,/ee/} PC$_{OUT}$: [e]
	B. No shift /las**ii**+i+na/ → las**i**+i+na /las**i**+i+na / → las**i**+i+na /las**ee**+i+na / → las**e**+i+na /las**e**+i+na / → las**e**+i+na /las∅+i+na / → las∅+I+na		****! PC$_{IN}$: {/ii/,/i/}, {/ee/,/e/} PC$_{OUT}$: [i], [e]	
	Scenarios	PC$_{IN}$ (high/round)	PC$_{IN/OUT}$ (long)	PC$_{IN/OUT}$ (high)
	C. Chain shift /las**uu**+i+na/ → las**u**+i+na /las**u**+i+na / → las**o**+i+na /las**oo**+i+na / → las**o**+i+na /las**o**+i+na / → las**∅**+i+na /las∅+i+na / → las∅+I+na	*! PC$_{IN}$: {/u/,/oo/}	** PC$_{IN}$: {/u/,/oo/} PC$_{OUT}$: [o]	** PC$_{IN}$: {/u/,/oo/} PC$_{OUT}$: [o]
☞	D. No shift (actual) /las**uu**+i+na/ → las**u**+i+na /las**u**+i+na / → las**u**+i+na /las**oo**+i+na / → las**o**+i+na /las**o**+i+na / → las**o**+i+na /las**∅**+i+na / → las**∅**+i+na		***** PC$_{IN}$: {/uu/,/u/}, {/oo/,/o/} PC$_{OUT}$: [u], [o]	

In the set of unrounded vowels, the chain-shift scenario wins. It minimizes the number of length mergers. In the set of rounded vowels, on the other hand, the no-shift scenario is the winner. The domain-specific PC constraint, PC$_{IN}$(high/+round), blocks lowering.

[12] To ensure that rounded vowels do not have to differ in height when merging length, PC(high/+round) needs to dominate PC(long/high).

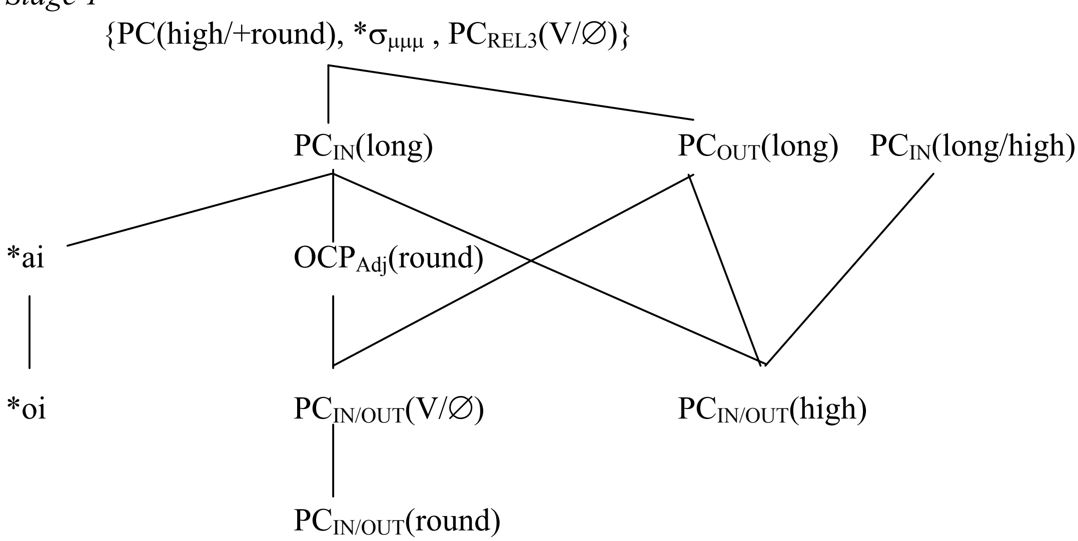
Domain-specific PC constraints resemble generalized faithfulness. Just like generalized faithfulness, they specify a set of segments for which contrast is evaluated. The question is whether there are restrictions on what sets can constitute the domain of PC constraints, for example, perceptually prominent positions only. The other question is whether there needs to be a relation between the domain of a PC constraint and the P property that is evaluated. Are rounded vowels special with respect to height mergers? I leave those questions open for further research.

3.5 Conclusion

Let us summarize the ranking established for Finnish and recall ranking arguments. This is by no means an exhaustive treatment of Finnish vowel alternations but it is an attempt at one.

(47) Mega ranking (combines (36), (59), and (70))

Stage 1



Stage 2

(+rd)-FAITH >> (-rd)-FAITH

(48) Ranking arguments

Ranking	Consequence
$*\sigma_{\mu\mu\mu}$ >> $PC_{IN/OUT}$(long)	Some length neutralizations take place due to shortening.
PC_{OUT}(long) >> $PC_{IN/OUT}$(round)	Rounding takes place to accumulate length neutralizations.
PC_{OUT}(long) >> $PC_{IN/OUT}$(V/∅)	Deletion takes place to accumulate length neutralizations.
$PC_{IN/OUT}$(V/∅) >> $PC_{IN/OUT}$(round)	Rounding is preferred to deletion.
OCP(round) >> $PC_{IN/OUT}$(V/∅)	Deletion takes place to satisfy OCP(round).
PC_{IN}(long) >> OCP(round)	But OCP(round) is violated in some forms to minimize length mergers.
PC_{REL3}(V/∅) >> $PC_{IN/OUT}$(long)	There is a limit put on deletion.
PC_{OUT}(long) >> $PC_{IN/OUT}$(high)	Height neutralizations take place to accumulate and minimize length mergers.
PC_{IN}(high/+round) >> $PC_{IN/OUT}$(long)	No height neutralizations within the set of rounded vowels.
PC_{IN}(long/high) >> $PC_{IN/OUT}$(high)	Height mergers take place to avoid minimal length mergers.

This chapter has investigated contrast in Finnish vowel alternations. The core of the proposal is vowel shortening that triggers the chain shift effects. The next step in the shift is rounding/deletion for the low vowels and lowering with deletion for the high vowels. Both shifts reduce the number of outputs ambiguous in length. Formally, the shifts are initiated by a high-ranking markedness constraint against tri-moraic syllables and then perpetuated by high-ranking PC constraints.

Furthermore, there are various conditions on the shifts. First, rounded vowels do not participate in height shifts that unrounded vowels do. Second, within the set of unrounded vowels, there are no length neutralizations among vowels of the same height. Those conditions are captured by domain- specific PC constraints where the domain argument specifies what set of segments is being evaluated, i.e., rounded vowels, high vowels, mid vowels.

Moreover, although in Finnish deletion aids the length contrast, there is a limit put on deletion. Too much deletion is not allowed, and thus there are some length mergers in a scenario. This is achieved by relational PC, a constraint that originally prohibits absolute neutralization. To account for Finnish, a family of relational PC constraints is proposed. Such constraints demand preservation of a particular number of input-length contrasts in the output.