The constraint inventory shows that seven of the 16 scenarios in (1) are harmonically bounded. That is, they will never come out optimal. Scenarios are harmonically bounded either on PC in stage one of Eval (this is when they merge too many types of contrasts, involve too many input pairs in a merger, or contain too many ambiguous outputs), or on faithfulness in stage 2 of Eval (this is when they involve too much movement). Arguments for harmonic bounding are given below.

(3) Arguments for harmonic bounding

(i) The cross-corner scenarios, G and H, and multi-directional scenarios, O and P, are harmonically bounded by the opaque scenarios, I through L. They involve the same PC and markedness violations as the competing opaque scenarios but contain too much movement. That is, in these scenarios outputs and their corresponding inputs are too different from each other. Thus, these scenarios are harmonically bounded on FAITHFULNESS in stage 2 of Eval. The relevant tableaux follow. (Scenarios are grouped into the relevant tableaux according to the directionality of movement in a scenario, reflected by markedness violations, *ai and *oi.) The scenarios with rounding are presented first, followed by the ones with lowering.

Fewer unfaithful [ai]'s (=rounding)

	*$\sigma_{\mu\mu\mu}$	PC_{REL} (rd)	PC_{OUT} (lg)	PC_{IN} (lg)	PC_{OUT} (rd)	PC_{IN} (rd)	*ai	*oi	(-rd) Faith	(+rd) Faith
H. Cross-corner			*	**	*	**	*	***	**	***
I. CHS			*	**	*	**	*	***	*	**
K. DEE			*	**	*	**	*	***		***
O. Multi-direct.			*	**	*	**	*	***	*	****

Fewer unfaithful [oi]'s (=lowering)

	*$\sigma_{\mu\mu\mu}$	PC_{REL} (rd)	PC_{OUT} (lg)	PC_{IN} (lg)	PC_{OUT} (rd)	PC_{IN} (rd)	*ai	*oi	(-rd) Faith	(+rd) Faith
G. Cross-corner			*	**	*	**	***	*	***	**
J. CHS (reverse)			*	**	*	**	***	*	**	*
L. DEE(reverse)			*	**	*	**	***	*	***	
P. Multi-direct.			*	**	*	**	***	*	****	*

This shows that in PC theory, when scenarios tie on PC and markedness in stage 1 of Eval (see above), faithfulness in stage 2 chooses in favor of a scenario where inputs are overall closer to their outputs. In this case, those are the opaque (non-shaded) scenarios.

(ii) Similarly, the multi-directional scenario N is harmonically bounded by the transparent scenario A on FAITHFULNESS.

	*$\sigma_{\mu\mu\mu}$	PC_{REL} (rd)	PC_{OUT} (lg)	PC_{IN} (lg)	PC_{OUT} (rd)	PC_{IN} (rd)	*ai	*oi	(-rd) Faith	(+rd) Faith
A. Transparent			**	**			**	**	*	*
N. Multi-direct.			**	**			**	**	***	***

The two scenarios incur the same PC and markedness violations (stage 1 of Eval), but the multi-directional scenario violates faithfulness too much (stage 2), and thus will never win over the competing transparent scenario.

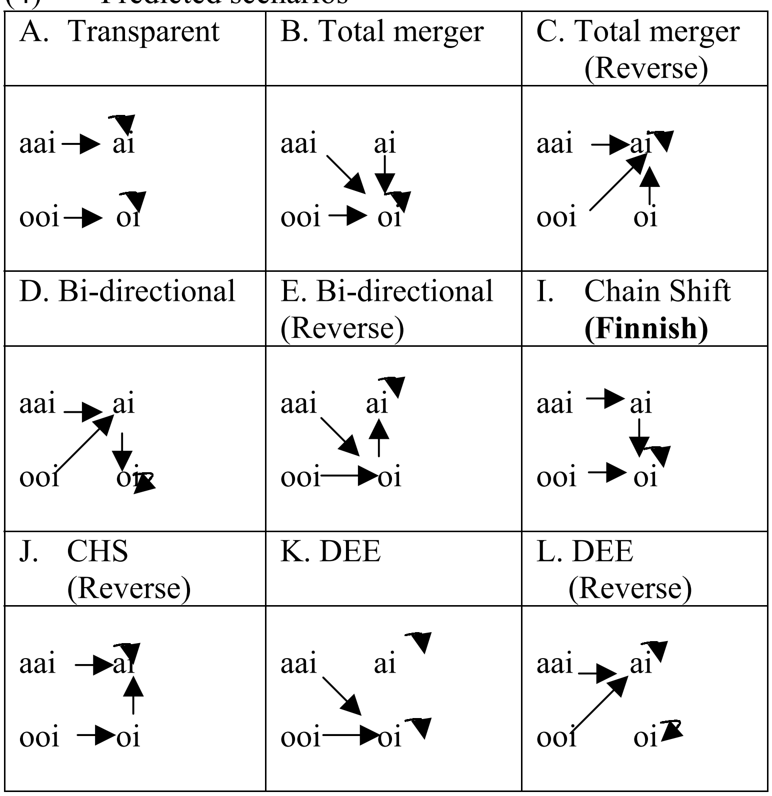
(iii) Finally, the cross-corner scenario F and the multi-directional scenario M are harmonically bounded by the transparent scenario A even before FAITHFULNESS comes into play. They fare the same on markedness but violate too many types of PC constraints.

	*$\sigma_{\mu\mu\mu}$	PC_{REL} (rd)	PC_{OUT} (lg)	PC_{IN} (lg)	PC_{OUT} (rd)	PC_{IN} (rd)	*ai	*oi	(-rd) Faith	(+rd) Faith
A. Transparent			**	**			**	**	*	*
F. Cross-corner			**	**	**	**	**	**	**	**
M. Multi-direct.			**	**	**	**	**	**	**	**

The transparent scenario merges along fewer dimensions of contrast (it does not violate PC(round) constraints at all), and thus wins over the two competing scenarios, regardless of the constraint ranking.

The following table shows the set of predicted scenarios. Harmonically bounded scenarios have been excluded. To the best of my knowledge, this prediction coincides with the set of empirically attested phenomena (see next section for examples).[1]

(4) Predicted scenarios



The above table includes a transparent scenario with no rounding, a total merger scenario where rounding applies across the board, a bi-directional scenario with rounding and lowering, and two opaque scenarios, the chain-shift scenario from Finnish and a

1 The claim here is about mapping interactions represented in (4) and not about the exact mappings.

competing derived-environment effect. For each scenario (except for the transparent scenario), there is a corresponding mirror-image effect, which I call a reverse scenario.

In the next section, I will give examples of attested scenarios. The scenarios that will be illustrated in the following section are not exactly like the scenarios given in (4), but they contain the same types of mappings, and thus represent the same types of scenarios. The overall geometry of the mappings is the same even though the features involved might be different.

2.4 Examples of Attested Scenarios

This section gives examples of attested scenarios. Scenarios differ in contrast preservation. In some, contrast is preserved transparently: a given output contrast corresponds to an identical input contrast. In others, contrast is preserved in an opaque way: a given input contrast is transformed into a different output contrast (contrast transformation takes place).

2.4.1 Transparent Scenarios

In a transparent scenario (see (4a)), a process applies to all forms in the language that are subject to it. For example, final devoicing in Polish applies to all voiced obstruents in word-final position (Rubach 1984). Thus, the underlying voicing contrast is neutralized on the surface for all voiced-voiceless input pairs in word-final context but preserved elsewhere.[2] Otherwise, contrasts are retained in the same way in which they are represented underlyingly.

Sometimes more than one process applies in a scenario, just as in the total-merger scenario. In a total-merger scenario, processes are in a feeding relation such that various inputs map onto one and the same output. The example given in (4) is of a language

[2] Neutralization also occurs in the environment of voicing assimilation.

where long vowels shorten and become rounded, and identical short vowels undergo rounding as well. Thus, many input contrasts are neutralized on the surface. The properties of transparent scenarios versus opaque scenarios will be discussed throughout this chapter.

Let us now move on to opaque scenarios: chain shifts, derived-environment effect, and bi-directional scenarios, in which contrasts are preserved but in a different way than in the input.

2.4.2 Opaque Scenarios

2.4.2.1 Derived-Environment Effect

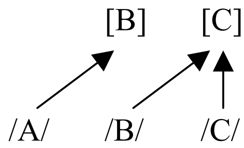
In the previous chapter we discussed one example of an opaque mapping, a chain shift effect. A derived-environment effect (Mascaró 1976, Kiparsky 1982, Rubach 1984) provides another example. It was observed in chapter 1 that opacity involves contrast transformation. In an opaque mapping, some underlying contrast is preserved on the surface but manifested as a different surface contrast. Some original contrast is lost as a result. Both chain shifts and derived-environment effects have this property.

Given an A-B-C scale of similarity, in a chain shift, /A/→[B] and /B/→[C]. In a derived environment, on the other hand, /A/→[C] and /B/→[B].[3] In both scenarios, contrast between /A/ vs. /B/ is preserved and manifested as [B] vs. [C]. The only difference between chain shifts and derived-environment effects is what inputs map onto particular outputs. This is represented schematically below.

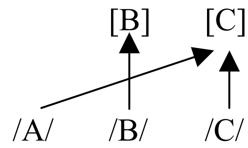
---

[3] This is called a derived-environment effect, since [B]'s are avoided only when they are "derived" and not underlying. In OT, "derived" means originating from a different source than itself.

(5) a. Chain shift



b. Derived-environment effect



In chain shifts, /A/ vs. /B/ → [B] vs. [C], whereas in a derived-environment effect, /A/ vs. /B/ → [C] vs. [B]. Differences between chain shifts and derived-environment effects are discussed in more detail in section 2.5.4.

We have already seen examples of a chain shift (see chapter 1 for Finnish and Kashubian). I will now illustrate two cases of derived-environment effects: the interaction of final devoicing and vowel raising in Polish, and the interaction of final devoicing and vowel lengthening in Friulian.

In Polish, obstruents devoice word-finally (see (6a)). But when the obstruent is preceded by a mid back vowel, devoicing is accompanied by vowel raising (shown in (6b)) (Grzegorczykowa *et al.* 1984, Gussmann 1980). Crucially, there is no raising before an underlyingly voiceless obstruent (see (6c)). Unlike mid vowels, high vowels remain high before both "devoiced" and underlyingly voiceless obstruents (see (6d)).

(6) Raising and final devoicing in Polish[4]

a. Final devoicing

nom.sg. or nom.pl	*gen.pl. or nom.sg.*	
te[z]a	te[s]	'hypothesis'
wa[d]a	wa[t]	'fault'
chle[b]y	chle[p]	'bread'

[4] The data presented here only include nouns, but raising also occurs in other morphological categories: verbs, adjectives, pronouns (see Grzegorczykowa et al. 1984, pp. 85-87, Gussmann 1980, pp. 113-130). Raising also occurs in closed syllables before sonorants: –r, -l, -w, -j, but since in those cases it is restricted to native vocabulary only (see Gussmann 1980), I will treat it as a separate process.

b. Vowel raising

k[oz]a	k[us]	‘goat’
l[od]y	l[ut]	‘ice’
r[og]i	r[uk]	‘horn’

c. No raising

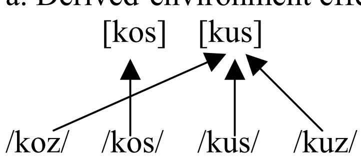
k[os]a	k[os]	‘scythe’
l[ot]y	l[ot]	‘flight’
r[ok]u	r[ok]	‘year’

d. No lowering

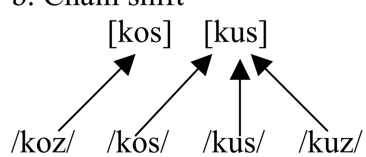
b[ud]a	b[ut]	‘kennel, shed’
b[ut]y	b[ut]	‘shoe’

The interaction of raising and final devoicing in Polish is an example of a derived-environment effect. Underlying /koz/ becomes surface [kus] with devoicing and raising, skipping over a more similar output [kos], even though there are surface instances of [kos] in Polish. The actual set of mappings is given below (it is compared to a hypothetical chain shift effect in (7b)):

(7) a. Derived-environment effect

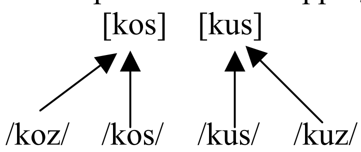


b. Chain shift



Polish raising and other derived-environment effects, similar to chain shifts discussed in the previous chapter, are not admitted to classic Optimality Theory (Prince and Smolensky 1993). OT is output-oriented. If [kos] is an acceptable output in Polish, then we expect /koz/ to map onto [kos] and not onto a more distant form [kus] (/koz/→[kos], *[kus]). Instead, in the actual mapping, final devoicing is accompanied by raising, causing the fell-swoop of /koz/ to [kus]. The problem is to explain why raising takes place. The expected set of mappings is given below.

(8) The expected set of mappings



Derived-environment effects are part of the synchronic grammar. Another example comes from Friulian (Baroni and Vanelli 2001, Hualde 1990, Repetti 1992, 1994, 2001). Friulian, like Polish, has a process of final devoicing, and final devoicing interacts with vowel lengthening, just as it interacts with vowel raising in Polish. In Friulian, vowels lengthen before devoiced obstruents but remain short before underlyingly voiceless obstruents. Some examples follow. Each voiced pair is followed by a voiceless pair.

(9) Lengthening and final devoicing in Friulian (cf. Polish in (6))

[ˈl**ad**e]	[ˈl**a:t**]	‘gone (f.) / (m.)’
[l**aˈt**a]	[ˈl**at**]	‘to breast-feed / milk’
[fiˈn**id**e]	[fiˈn**i:t**]	‘finished (f.) / (m.)’
[ˈm**at**e]	[ˈm**at**]	‘crazy (f.) / (m.)’
[l**uˈz**o:r]	[ˈl**u:s**]	‘diffuse light / light’
[ˈr**ɔs**e]	[ˈr**ɔs**]	‘red (f.) / (m.)’

This is problematic for standard OT. Why should devoicing be accompanied by vowel lengthening since there are surface instances of voiceless obstruents with a short vowel in the same context? We expect voiced obstruents to devoice word-finally, but there is no reason for the preceding vowel to lengthen (/ad/→[at], *[a:t]). In terms of mappings, there is no reason to map onto a more distant output when a more similar output is available. The question then is why lengthening takes place.

I will argue that lengthening allows preservation of the contrast between words that differ underlyingly in voiced-voiceless obstruent. Lengthening keeps the contrast, even though the obstruent devoices (see section 2.5).

Some other cases of derived-environment effects come from Campidanian Sardinian (Bolognesi 1998), Makassarese (McCarthy and Prince 1994), Polish (Gussmann 1980, Rubach 1984), Tiberian Hebrew (Prince 1975), Slovak (Rubach 1993).

2.4.2.2 Bi-directional Scenario

We are left with one other opaque scenario from the set of attested scenarios in (4). This is the bi-directional scenario. An example of a bi-directional scenario comes from the history of nasal vowels in Polish (Comrie and Corbett 1993, Rospond 1971, Rothstein 1993).

In the history of Polish, some nasal vowels lowered, others raised. As reported, lowering applied to underlying long nasal vowels (10a), and raising targeted their short counterparts (10b).

(10) a. Lowering (and shortening) (nom.sg.masc.)

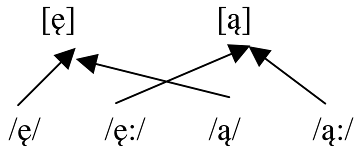
ę: > ą	*r[ę:]d	>	rz[ą]d	"row"
ą: > ą	* m[ą:]ż	>	m[ą]ż	"husband"

b. Raising (gen.sg.masc.)

ę > ę	* r[ę]du	>	rz[ę]du	"row"
ą > ę	* m[ą]ża	>	m[ę]ża	"husband"

As a result, contrast was preserved between underlying short and long nasal vowels. The two kinds mapped onto distinct outputs. Long nasal vowels mapped onto low vowels (10a). Short nasal vowels mapped onto mid vowels (10b). This is shown schematically below.

(11) Bi-directional scenario



The bi-directional scenario is opaque since it contains an output height contrast that does not correspond to any minimal instance of an input height contrast. It posits an underlying opposition that does not have any surface correlates (absolute neutralization). In the bi-directional scenario, every instance of the input-length contrast is preserved in the output and realized as height contrast. For a more detailed discussion of a bi-directional scenario see section 2.5.*2 Limits on Transformation*.

2.5 The Analysis: A Comprehensive View

We are now ready to present a comprehensive analysis of the low-vowel shift in Finnish, considering all the empirically attested scenarios in (4) that compete with the actual scenario. In the course of the discussion, I will establish a complete constraint ranking, under which the chain-shift scenario comes out optimal. The basic argument has been already developed in chapter 1, section 4. It will be recalled below and augmented so that the additional empirically attested scenarios shown in (4) are eliminated in favor of the actual scenario.

2.5.1 Chain Shift as Contrast Transformation

Recall that chain shifts involve *contrast transformation.* A given contrast is preserved on the surface but manifested in a different way than in the original form. This is at the expense of neutralizing some other contrast (contrast transformation). Contrast transformation minimizes the number of ambiguous outputs in a scenario (see chapter 1, section 4).

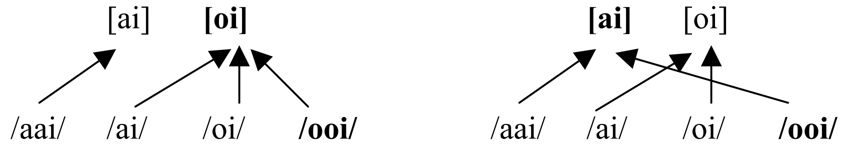
In Finnish, long vowels shorten, incurring some length mergers in a scenario, but despite shortening, length contrast is preserved on the surface for one minimal pair of inputs and is realized as a surface contrast in rounding. Some original rounding contrast is lost as a result. This improves on the distribution of length mergers in a scenario. Due to rounding, there are fewer outputs ambiguous in length. The ranking for Finnish is recalled below.

(12) $*\sigma_{\mu\mu\mu}$ >> PC_{OUT}(long), PC_{IN}(long) >> PC_{OUT}(round), PC_{IN}(round)
(See chapter 1, section 4 for discussion.)

2.5.2 Limits on Transformation

However, under the current ranking, there is another scenario that comes out optimal. This is the scenario that does not incur any length mergers at all and thus satisfies the high-ranking constraints against length mergers, PC_{OUT}(long) and PC_{IN}(long). It is called a bi-directional scenario. The actual scenario with some length mergers and the competing bi-directional scenario with no length mergers are represented below.

(13) a. Chain-shift scenario b. Bi-directional scenario



The two scenarios differ in mapping for underlying /ooi/. Underlying /ooi/ shortens in both scenarios, but in the bi-directional scenario, it also undergoes lowering. Thus, in the bi-directional scenario, contrast is preserved between long and short vowels for each input pair.

The bi-directional scenario is optimal under the current constraint ranking. The following tableau shows it formally.

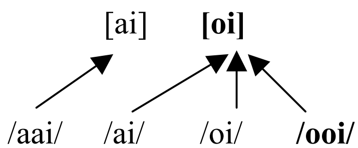
(14) Current constraint ranking – wrong result

Scenarios		$*\sigma_{\mu\mu\mu}$	PC_{IN} (long)	PC_{OUT} (long)	PC_{IN} (round)	PC_{OUT} (round)
A. Chain shift ☜ aai → ai ↓ ooi → oi	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi		** {/ai/, /ooi/} {/oi/, /ooi/}	*! [oi]	** {/ai/,/oi/} {/ai/,/ooi/}	* [oi]
B. Bi-directional ☞ aai → ai ↓ ooi oi	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → ai				** {/ai/,/oi/} {/aai/,/ooi/}	** [oi] [ai]

The bi-directional scenario, scenario B, satisfies high-ranking PC(long) constraints, PC_{IN}(long) and PC_{OUT}(long), since it does not merge length at all. The chain-shift scenario, scenario A, loses since it merges some input-length contrasts. This result is wrong. We need to ensure that it is the chain-shift scenario that surfaces in Finnish.

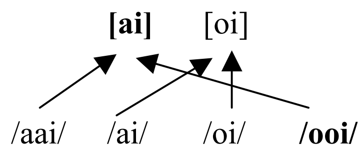
In the bi-directional scenario (14b), the length contrast is preserved for each pair of inputs and realized as a surface rounding contrast; long and short vowels map onto outputs distinct in rounding. As a result, each minimal rounding contrast from the input is neutralized, /aa/ vs. /oo/ → [a] and /a/ vs. /o/ → [o]. In a chain-shift scenario (14a), on the other hand, contrast is preserved for one minimal pair of inputs distinct in length and realized as a surface rounding contrast, /aa/ vs. /a/ → [a] vs. [o]. The other minimal length contrast, /oo/ vs. /o/, neutralizes. This results in the neutralization of one minimal rounding contrast, /a/ vs. /o/ → [o]. The other rounding contrast, /aa/ vs. /oo/, is retained in the output. The diagrams below show what minimal input contrasts are preserved in both types of scenarios.

(15) a. Chain-shift scenario



Preserved contrasts:
length: /aa/ vs. /a/
rounding: /aa/ vs. /oo/

b. Bi-directional scenario



length: /aa/ vs. /a/, /oo/ vs. /o/
rounding: none

In the chain-shift scenario, the output rounding contrast corresponds to both input minimal rounding and length contrasts. In the bi-directional scenario, on the other hand, the output rounding contrast corresponds to every input-length contrast but none of the minimal rounding contrasts. (The relation between contrasts is evaluated for minimal contrasts only.)

For chain shifts to be optimal, it must be more important to retain some minimal rounding contrasts from the input (to prevent absolute neutralization) than it is to avoid neutralizations of all length contrasts. This is the role of relational PC. The constraint is recalled below. The relevant constraint ranking follows.

(16) PC_{REL}(round) (cf. (30), chapter 1)
Assign a violation mark for a pair of outputs minimally contrasting in rounding that does not correspond to a pair of inputs minimally contrasting in rounding.

(17) PC_{REL}(round) >> PC_{OUT}(long), PC_{IN}(long)

The bi-directional scenario violates the PC_{REL} constraint, since in this scenario, none of the minimal rounding contrasts from the input are preserved in the output. The pair of outputs that contrast in [round], i.e., [ai] and [oi], does not correspond to any inputs that contrast for this feature. A chain-shift scenario satisfies this constraint since in a chain shift one of the minimal rounding contrasts is preserved on the surface. This is at the cost of merging some length contrasts. This is illustrated in the following tableau.

(18) CHS is preferred over bi-directionality

Scenarios		PC_{REL} (rnd)	PC_{IN} (lg)	PC_{OUT} (lg)	PC_{IN} (rnd)	PC_{OUT} (rnd)
A. Chain shift ☞ aai → ai ↓ ooi → oi ↺	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi		** {/ai/, /ooi/} {/oi/, /ooi/}	* [oi]	** {/ai/,/oi/} {/ai/,/ooi/}	* [oi]
B. Bi-directional aai → ai ↗ ↓ ooi oi ↺	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → ai	*!			** {/ai/,/oi/} {/aai/,/ooi/}	** [oi] [ai]

The chain-shift scenario wins under this ranking. Unlike the bi-directional scenario, it merges length contrasts but preserves a minimal rounding contrast from the input. This is preferred to avoiding length mergers altogether, as in the bi-directional scenario.

The augmented constraint ranking is given below.

(19) $*\sigma_{\mu\mu\mu}$ PC_{REL}(round)

PC_{OUT}(long) PC_{IN}(long)

PC_{OUT}(round) PC_{IN}(round)

In short, even though the bi-directional scenario satisfies PC(long) constraints, since it does not cause any length mergers in the system, it loses to the chain-shift scenario that incurs some violations of those constraints. The bi-directional scenario, unlike the chain-shift scenario, violates the high-ranking relational PC constraint, PC_{REL}(round). Since PC_{REL}(round) dominates PC constraints against length mergers (PC_{REL}(round) >> PC_{OUT}(long), PC_{IN}(long)), the bi-directional scenario is ruled out.[5]

2.5.3 Mirror Image Scenarios

Under the current constraint ranking, there are two scenarios that tie. This is the actual chain-shift scenario and its mirror image, reverse shift. The two scenarios differ only in the directionality of movement. The actual scenario contains rounding, while the reverse shift contains lowering. The two scenarios are represented below.

(20) a. Chain-shift scenario b. Reverse shift

Rounding, in the actual scenario, implies more tokens with rounded vowels than unrounded vowels. Lowering has the opposite effect. Our constraint ranking so far does not distinguish between the two scenarios.

In the rounding scenario (20a), both underlying round vowels, /oi/ and /ooi/, and a derived round vowel, /ai/, map onto a rounded output. In the reverse shift scenario, on the

5 Another way to rule out the bi-directional scenario is to use self-conjunction of output-oriented PC constraints. In this case, it would be self-conjunction of PC_{OUT}(round) (see Kirchner 1996; see also Smolensky 1993, 1997 for definition of local constraint conjunction). The bi-directional scenario would violate this constraint since it violates PC_{OUT}(round) twice (there are two outputs ambiguous in rounding). The chain-shift scenario would not violate the locally-conjoined constraint since it violates PC_{OUT}(round) only once. Since conjunction results in proliferation of constraints, I will not pursue it further. The self-conjunction analysis also leads to ranking ambiguity in the choice of the actual scenario. The actual scenario would win under either PC^2(rd) >> PC(lg) >> PC(rd) or PC^2(lg) >> PC(rd) >> PC(lg). We lose our original observation that in the opaque scenario, length is preserved at the cost of rounding.

other hand, there are more instances of unrounded vowels since they correspond to underlying low vowels, /ai/ and /aai/, and a derived low vowel, /oi/. For the actual shift to win, a markedness constraint against unrounded vowels must dominate a markedness constraint against rounded vowels. It is more important to avoid [ai]'s than it is to avoid [oi]'s. The relevant constraints and the ranking are given below.

(21) a. *ai
Assign a violation mark for every token [ai] in a scenario, where the number of tokens equals the number of inputs that map onto this output.

b. *oi
Assign a violation mark for every token [oi] in a scenario, where the number of tokens equals the number of inputs that map onto this output.

c. Markedness ranking
*ai >> *oi

For a discussion of tokenized markedness, see chapter 1, section 2.2.2. The ranking is illustrated in the following tableau.

(22) The role of tokenized markedness

Scenarios		*ai	*oi
A. Actual ☞ aai → ai ↓ ooi → oi ↻	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi	*	***
B. Reverse aai → ai ↻ ↑ ooi → oi	/aai/ → ai /ai/ → ai /oi/ → ai /ooi/ → oi	***!	*

The actual shift wins since it contains fewer unrounded tokens. The competing scenario with lowering contains more such tokens since there are more inputs in this scenario mapping onto [ai].

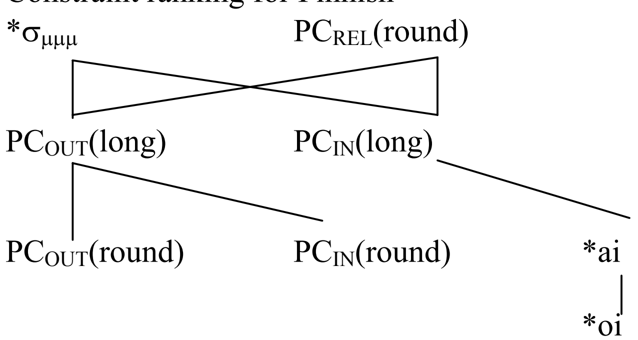
It is important to notice that not all instances of [ai]'s are avoided. This would be the case if all inputs merged onto rounded vowels, as in the total merger scenario. To avoid it, *[ai] must be ranked lower than a constraint against the total merger scenario. In Finnish, *[ai] must be dominated by PC_{IN}(long). The total merger scenario violates input-oriented PC too much.

(23) PC_{IN}(long) >> *ai

Scenarios		PC_{IN}(long)	*ai
A. Actual ☞ aai → ai ↓ ooi → oi ↩	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi	** {/ai/, /ooi/} {/oi/, /ooi/}	*
B. Total merger aai ai ↘ ↓ ooi → oi ↩	/aai/ → oi /ai/ → oi /oi/ → oi /ooi/ → oi	****! {/ai/, /ooi/} {/oi/, /ooi/} {/ai/, /aai/} {/oi/, /aai/}	

The complete constraint ranking is given below.

(24) Constraint ranking for Finnish cf. (12)



Ranking arguments are recalled in section 2.5.5, *Summary Ranking*.[6]

[6] An alternative to tokenized markedness is an enriched theory of faithfulness or PC. Tokenized markedness seems a good alternative since it is needed in PC theory independently to initiate a chain shift effect and it is defined using the core property of PC theory - mappings evaluated as a system.

2.5.4 Chain Shift and Derived-Environment Effect

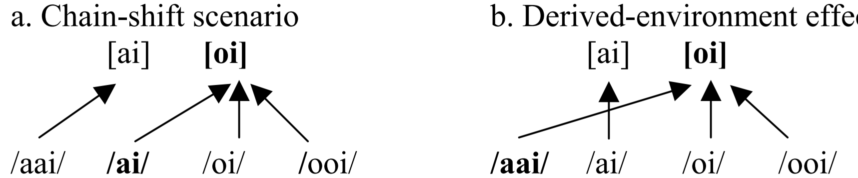
We now come to stage 2 of Eval. There are two scenarios that tie after stage 1. This is the actual chain-shift scenario and a competing derived-environment effect. We need to make the choice between the two. The two scenarios have the same outputs and involve the same types of movement, but they differ on which segments move and how much. The following tableau compares the two scenarios.

(25) A tie

Scenarios	PC_{IN}(long)	PC_{IN}(rd)	PC_{OUT}(long)	PC_{OUT}(rd)	*ai	*oi
A. CHS /aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi	** {/ai/, /ooi/} {/oi/, /ooi/}	** {/ai/, /ooi/} {/ai/, /oi/}	* [oi]	* [oi]	*	***
B. DEE /aai/ → oi /ai/ → ai /oi/ → oi /ooi/ → oi	** {/aai/, /oi/} {/oi/, /ooi/}	** {/aai/, /ooi/} {/aai/, /oi/}	* [oi]	* [oi]	*	***

The two scenarios fare the same on PC and markedness constraints. However, they are distinct and we should be able to express that formally.

(26) a. Chain-shift scenario b. Derived-environment effect



In a chain-shift scenario, movement is distributed among segments. The long low vowel shortens and the short low vowel rounds. Both segments undergo movement. In a derived-environment effect, on the other hand, movement is accumulated in one mapping in a scenario. The long low vowel both shortens and rounds. The short low vowel remains faithful.

Assume that outputs are divided into sets based on their value for a given P property, in this case, rounding. Assume furthermore that sets of properties differ in their resistance to unfaithful mappings. This is the idea behind the generalized faithfulness constraints introduced in chapter 1, section 2. The two scenarios differ in faithfulness of rounded and unrounded vowels. In the chain-shift scenario, both rounded and unrounded vowel outputs are unfaithful. In the derived-environment effect, on the other hand, the unrounded vowel output is fully faithful, but as a result, the rounded vowel output is worse on faithfulness than in the competing chain shift effect. Thus, for the chain shift to win, it must be more important to reduce unfaithfulness of rounded vowels than their unrounded counterparts.

In terms of constraints, the chain-shift scenario is selected when (+rd)-FAITH outranks (-rd)-FAITH, as shown below. (A competing derived-environment effect would be obtained with the opposite ranking.)

(27) a. (+rd)-FAITH
A (+rd) output is identical to its input correspondent in every property. Assign a violation mark for any type of disparity.

b. (-rd)-FAITH
A (-rd) output is identical to its input correspondent in every property. Assign a violation mark for any type of disparity.

c. The ranking
(+rd)-FAITH >> (-rd)-FAITH

For clarity of exposition, I list mappings that incur a violation of relevant faithfulness constraints. The number of stars indicates how many faithfulness violations a given mapping incurs. (For a detailed discussion of generalized faithfulness see chapter 1, section 2.2.3.)

(28) The role of generalized faithfulness

Scenarios		(+rd)-FAITH	(-rd)-FAITH
A. CHS ☞ aai → ai ↓ ooi → oi	/aai/ → ai /ai/ → oi /oi/ → oi /ooi/ → oi	** /ooi/ → oi (*) /ai/ → oi (*)	* /aai/ → ai (*)
B. DEE aai ai ↘ ooi → oi	/aai/ → oi /ai/ → ai /oi/ → oi /ooi/ → oi	***! /aai/ → oi (**) /ooi/ → oi (*)	

The chain-shift scenario wins as it minimizes faithfulness violations within the rounded vowel set. This is at the cost of faithfulness to unrounded outputs, (-rd)-FAITH. (Recall that rounding needs to take place, implying some (+rd)-FAITH violations. This is determined in stage 1 of Eval.)

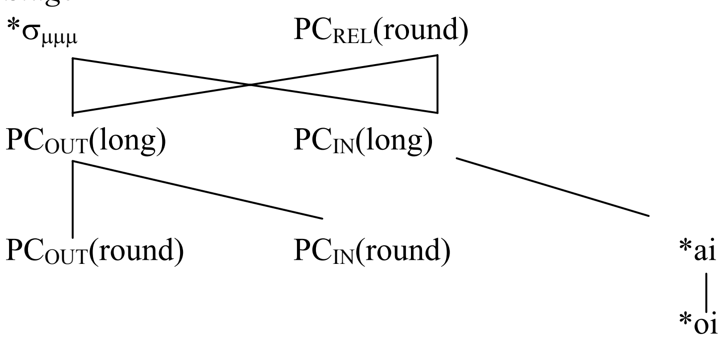
The direction of movement is determined by markedness ranking in stage 1 of Eval. In Finnish, *ai >> *oi, thus rounding takes place. When the ranking of generalized faithfulness coincides with markedness ranking (*ai >> *oi and (-rd)-FAITH >> (+rd)-FAITH), we get a derived-environment effect. In case of conflict between the two (*ai >> *oi but (+rd)-FAITH >> (-rd)-FAITH), a chain shift results. The logic behind it is as follows. In a chain-shift scenario, despite rounding (mapping towards [oi]), we still want to make rounded vowels as faithful as possible. We cannot avoid mappings onto rounded vowels altogether, since rounding takes place, but we can control how distant the mappings are. In a derived-environment effect, faithfulness agrees with the directionality of movement.

2.5.5 Summary Ranking

Altogether, the constraint ranking for Finnish is as follows. The two stages of Eval are shown separately. Stage 1 contains markedness and PC constraints. Stage 2 is generalized faithfulness.

(29) Constraint ranking for Finnish

Stage 1



Stage 2

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

Ranking arguments are recalled below.

(30) Ranking arguments

Ranking	Consequence
Stage 1 of Eval	
$*\sigma_{\mu\mu\mu}$ >> PC_{OUT}(long), PC_{IN}(long)	Shortening takes place to avoid tri-moraic syllables. It incurs some length mergers.
PC_{OUT}(long) >> PC_{OUT}(round), PC_{IN}(round)	Rounding takes place to achieve better distribution of length mergers in a scenario.

PC_{REL}(round) >> PC_{OUT}(long), PC_{IN}(long)	Bi-directional scenario (with no length mergers) is ruled out since it does not retain any identity between output and input contrasts.
*ai >> *oi	Mirror image scenario is ruled out. It contains too many [ai]'s.
PC_{IN}(long) >> *ai	Total merger is not an option. It merges every pair of inputs distinct in length.

Stage 2 of Eval

(+rd)-FAITH >> (-rd)-FAITH	CHS is favored over DEE.

To recall, the main argument is as follows. Shortening takes place to avoid tri-moraic syllables. This results in some length mergers. But length mergers need to be accumulated in a scenario rather than distributed among outputs and this is at the cost of merging rounding. However, even though preserving length contrasts is important in Finnish, and it takes place at the cost of rounding, not all length mergers are avoided. This would result in neutralizations of all minimal rounding contrasts from the input, and this is militated against by a high-ranking relational PC constraint. Finally, tokenized markedness constraints choose between mirror images of scenarios and segmental faithfulness makes the choice between chain shifts and derived-environment effects.

The relevant tableau is given below. Scenarios correspond to the ones established in (4). The established ranking selects the actual chain-shift scenario as optimal.

(31) Ranking for Finnish

	$*\sigma_{\mu\mu\mu}$	PC_{REL} (round)	PC_{OUT} (long)	PC_{IN} (long)	PC_{OUT} (round)	PC_{IN} (round)	*ai	*oi	(+rd)-FAITH
A. Transparent			**!	**			**	**	*
B. Total merger			*	****!	*	****		****	****
C. Total merger (Reverse)			*	****!	*	****	****		
D. Bi-directional		*!			**	**	**	**	*
E. Bi-directional (Reverse)		*!			**	**	**	**	***
I. CHS ☞ (Finnish)			*	**	*	**	*	***	**
J. CHS (Reverse)			*	**	*	**	***!	*	*
K. DEE			*	**	*	**	*	***	***!
L. DEE (Reverse)			*	**	*	**	***!	*	

Let us discuss the ranking. The bi-directional scenarios, scenarios D and E, are ruled out on relational PC. They transform each underlying length contrast into a surface rounding contrast. In consequence, in those scenarios none of the input rounding contrasts are preserved in the output. This is a violation of a high-ranked PC_{REL}(round) constraint (See section 2.5.2 for discussion.) The two total merger scenarios, scenarios B and C, lose on input-oriented PC, as they both merge too many input pairs distinct in length. In a total merger scenario, each input maps onto one and the same output. The transparent scenario, scenario A, loses on output-oriented PC, PC_{OUT}(long). It contains too many outputs ambiguous in length. (See chapter 1, section 4 for discussion.) The opaque scenarios, scenarios I-L, tie so far.

There are four competing opaque scenarios. Two of them, scenarios J and L, contain lowering. The other two, scenarios I and K, involve rounding. The opaque scenarios with lowering are ruled out on markedness since they contain too many tokens of unrounded vowels. The other two scenarios, scenario I and K, pass on to stage 2 of Eval.

In stage 2, the choice is made between the two opaque scenarios that tie so far. The scenario where rounded vowels are more faithful wins. This is the chain-shift scenario, scenario I. (See section 2.5.4.)

Thus, we have seen that the constraint ranking established for Finnish chooses the actual scenario as optimal. In the next section, I will show that each of the empirically attested scenarios from the candidate set is predicted to win in some language.

2.6 Discussion

This section discusses each of the attested scenarios shown in (4). Let us start with the actual scenario. The actual scenario neutralizes both length and rounding contrasts. This has been described in detail in chapter 1, section 4. But not every scenario in the same candidate set merges along both dimensions. In particular, there are two scenarios, transparent and bi-directional, that merge only one dimension of contrast. The transparent scenario (4a) merges length but not rounding, and the bi-directional scenario (4d-e) merges rounding but not length. The transparent scenario, therefore, wins if rounding mergers are not permitted as a result of shortening, whereas the bi-directional scenario is optimal if length mergers are ruled out (see below).

(32)

Type	Property
Transparent	No rounding mergers
Bi-directional	No length mergers

Each of the two scenarios would be optimal in a language that bans the relevant kind of merger.

The total merger scenario (4b-c) is similar to the chain-shift scenario, in that it merges both length and rounding. But, unlike the chain-shift scenario, it merges length and rounding for each input pair and thus violates input-oriented PC too much in comparison to other scenarios. As a result, the total merger scenario would never win solely on PC constraints. But it would be optimal if markedness against some output type were high-ranking in the language. The total merger scenario would be favored in this situation, since it contains only one output type. In fact, it is the only scenario in the candidate set that violates only one markedness constraint. Each of the remaining scenarios contains two output types and thus performs worse on markedness (assuming that there are markedness constraints against each output type). The properties of the total merger scenarios are given below.

(33)

Type	Property
Total merger	No [ai]'s
	No [oi]'s

Here is a summary of constraint rankings established so far. The dividing lines in the "rankings" column show distinctions already made between scenarios.

(34) Constraint rankings

Scenarios	Stage 1 of Evaluation
Transparent	Saves rounding: $*\sigma_{\mu\mu\mu}$, PC_{OUT}(rd), PC_{IN}(rd) >> PC_{OUT}(lg), PC_{IN}(lg), *ai, *oi
Total merger	Leader on markedness: *ai (or *oi) >> PC
Bi-directional	Saves length: $*\sigma_{\mu\mu\mu}$, PC_{OUT}(lg), PC_{IN}(lg) >> PC_{OUT}(rd), PC_{IN}(rd), *ai, *oi, PC_{REL}(rd)
Opaque	Minimizes the number of outputs ambiguous in length: $*\sigma_{\mu\mu\mu}$, PC_{REL}(rd) >> PC_{OUT}(lg), PC_{IN}(lg) >> PC_{OUT}(rd), PC_{IN}(rd), *ai, *oi

So far major distinctions have been made between scenarios, but we still need to distinguish between mirror images (reverse scenarios) and the two types of opaque scenarios, chain shift and derived environment effect. This is necessary to make sure that each scenario can win under some ranking.

Distinctions between mirror image scenarios are made on either tokenized markedness or segmental faithfulness. In cases when directionality results in a different number of tokens of a particular type, tokenized markedness makes the choice between scenarios (compare the two chain-shift scenarios and derived environment scenarios; I refer to them as opaque scenarios). Tokenized markedness makes the choice between opaque scenarios that contain rounding and the same scenarios that contain lowering. However, when there is no difference in the number of tokens between mirror images, segmental faithfulness comes into play (compare the two bi-directional scenarios). The role of tokenized markedness and segmental faithfulness is illustrated below:

(35)

Type	Ranking
Total merger:	*ai >> *oi
	*oi >> *ai
Bi-directional:	(+rd)-FAITH >> (-rd)-FAITH
	(-rd)-FAITH >> (+rd)-FAITH
Opaque w/rounding:	*ai >> *oi
Opaque w/lowering:	*oi >> *ai

(36) Augmented constraint rankings

Stage 1 of Evaluation	Stage 2 of Evaluation
Transparent: $*\sigma_{\mu\mu\mu}$, $PC_{OUT/IN}$(rd) >> $PC_{OUT/IN}$(lg), *ai, *oi	
Total merger onto [oi]: *ai >> PC, *oi	
Total merger onto [ai]: *oi >> PC, *ai	
Bi-directional: $*\sigma_{\mu\mu\mu}$, $PC_{OUT/IN}$(lg) >> $PC_{OUT/IN}$(rd), *ai, *oi, PC_{REL}(rd)	Lg vowels map onto [ai]: (+rd)-FAITH>>(-rd)-FAITH Lg vowels map onto [oi]: (-rd)-FAITH>>(+rd)-FAITH
Opaque (with rounding): $*\sigma_{\mu\mu\mu}$,PC_{REL}(rd)>>$PC_{OUT/IN}$(lg)>>$PC_{OUT/IN}$(rd),*ai>>*oi	Needs to be determined
Opaque (with lowering): $*\sigma_{\mu\mu\mu}$,PC_{REL}(rd)>>$PC_{OUT/IN}$(lg)>>$PC_{OUT/IN}$(rd),*oi>>*ai	Needs to be determined

We still need to make distinctions between chain shifts and derived environment effects of the same direction of movement. Those distinctions are made in stage 2 of Eval by segmental faithfulness.

(37)	Rounding pair:	(+rd)-FAITH >> (-rd)-FAITH	Chain shift
		(-rd)-FAITH >> (+rd)-FAITH	DEE
	Lowering pair:	(-rd)-FAITH >> (+rd)-FAITH	Chain shift
		(+rd)-FAITH >> (-rd)-FAITH	DEE

The final constraint ranking is given below.

(38) Final rankings

Stage 1 of Evaluation	Stage 2 of Evaluation
Transparent: $*\sigma_{\mu\mu\mu}$, $PC_{OUT/IN}$(rd) >> $PC_{OUT/IN}$(lg), *ai, *oi	
Total merger onto [oi]: *ai >> PC, *oi	
Total merger onto [ai]: *oi >> PC, *ai	
Bi-directional: $*\sigma_{\mu\mu\mu}$, $PC_{OUT/IN}$(lg) >> $PC_{OUT/IN}$(rd), *ai, *oi, PC_{REL}(rd)	Lg vowels map onto [ai]: (+rd)-FAITH>>(-rd)-FAITH Lg vowels map onto [oi]: (-rd)-FAITH>>(+rd)-FAITH
Opaque (with rounding): $*\sigma_{\mu\mu\mu}$,PC_{REL}(rd)>>$PC_{OUT/IN}$(lg)>>$PC_{OUT/IN}$(rd),*ai>>*oi	Limit mergers on [oi]: (+rd)-FAITH>>(-rd)-FAITH Save [ai]: (-rd)-FAITH>>(+rd)-FAITH
Opaque (with lowering): $*\sigma_{\mu\mu\mu}$,PC_{REL}(rd)>>$PC_{OUT/IN}$(lg)>>$PC_{OUT/IN}$(rd),*oi>>*ai	Limit mergers on [ai]: (-rd)-FAITH>>(+rd)-FAITH Save [oi]: (+rd)-FAITH>>(-rd)-FAITH

All distinctions are made among scenarios. This shows that our constraint inventory generates all of the attested scenario types – a basic goal of OT.