

Driving Derived Environment Effects with Alignment Constraints

Reed Blaylock

University of Southern California

1 Introduction

1.1 Derived environment effects Derived environment effects are phonological processes that only occur when the environment for those changes has been created by some other means (Kiparsky 1973, Mascaró 1976, Mohanan 1982, Iverson & Wheeler 1988). Traditionally, there are two types of derived environment effect: phonologically-derived environment effects, and morphologically-derived environment effects. This paper focuses on morphologically-derived environment effects.

Morphologically-derived environment effects are phonological changes that happen near morpheme boundaries due to morpheme concatenation. In Korean, for instance, the morpheme combination /mat + i/ (“the eldest”) undergoes palatalization to [maci] because the environment /ti/ is derived by morphological concatenation, whereas underlying /mati/ does not palatalize because its /ti/ environment is not derived (example from from Ahn 1985). In short, palatalization is blocked tautomorphemically, but possible at a morpheme boundary (Kim 1976, Ahn 1985).

(1)	Morphologically derived environment effect, classical derivation		
	/mat/ ‘eldest’	/mati/ ‘knot’	underlying representation
	[[mat] i]	_____	add morpheme (1st cycle)
	[[mac] i]	_____	t-palatalization
	[maji]	[madi]	voicing (post-lexical stage)
	[maji] ‘the eldest’	[madi] ‘knot’	final output

The example in (1) shows how this type of effect is analyzed in Lexical Phonology (Mohanani 1982). In Lexical Phonology, morphemes are added in successive cycles to the output of previous levels. In each cycle, certain phonological events are restricted to applying to structural descriptions already created within that cycle, represented by square brackets. In Korean, coronal palatalization is a process bound to a cycle.

In the case of /mat-i/, morpheme concatenation in the first cycle feeds palatalization in the same cycle, because the structure /ti/ was created in that level. In the post-lexical stage, exceptionless processes like inter-sonorant voicing occur, turning /maci/ into [maji].

Underlying /mati/, on the other hand, cannot feed palatalization because there is no morpheme concatenation in the first cycle to create the environment. Without adding a morpheme to create the environment, palatalization is blocked. The derivation then proceeds to the post-lexical stage, where voicing happens, turning /mati/ into [madi].

In summary, a morphologically-derived environment effect (hereafter, “derived environment effect”) is a phonological change which only applies in environments that have been derived through morphological concatenation.

This paper proposes an analysis in which morphologically-derived environment effects are driven by

* Thanks to the audiences at USC and AMP 2016 for their feedback, and especially to Charlie O’Hara and Karen Jesney for motivating and correcting me.

Alignment (or, rather, are triggered by misalignment). The remainder of this section introduces the Korean data that will be used to illustrate the efficacy of this approach. Section 2 presents a formal analysis of the data, and section 3 discusses how this approach may be extended to account for a wider array of morpho-phonological phenomena.

1.2 Case study: Korean Korean exhibits two separate derived environment effects, palatalization and [n]-insertion, occurring in one derived environment, [ti]. In each case, morpheme-final /t/ or /t^h/ are concatenated onto morpheme-initial /i/ or /j/ and separated by a morpheme boundary. However, the distribution of palatalization and [n] insertion is complementary: [n] insertion only occurs at prefix boundaries, and palatalization only occurs at suffix boundaries.

1.2.1 Palatalization At stem-suffix boundaries (2c-d), coronal stops palatalize to [c] or [c^h]. Within roots (2a-b), palatalization does not apply (Kim 1976, Ahn 1985). (This is an extended version of the data and source from section 1.1.)

- (2)
- | | | | |
|----|-----------------------|-----------------------|-------------|
| a. | /canti/ | [candi] | ‘grass’ |
| b. | /t ^h i/ | [t ^h i] | ‘dust’ |
| c. | /mat-i/ | [mac-i] | ‘firstborn’ |
| d. | /kat ^h -i/ | [kac ^h -i] | ‘together’ |

Note that intervocalically, [c] becomes [j]. This intervocalic voicing is not relevant to the discussion of prefix and suffix derived environment effects, and will be ignored.

1.2.2 [n]-insertion In the examples in (3), [n] is inserted between a prefix and a stem when the prefix ends in an obstruent and the stem begins with /i/ or /j/ (Jun 2014, 2015). As in (2a-b) above, tautomorphic [ti] sequences do not trigger [n] insertion.

- (3)
- | | | | |
|----|--------------------------|--------------|---------------------|
| a. | /hot ^h -ipul/ | [honnipul] | ‘unlined comforter’ |
| b. | /təs-jaŋmal/ | [tənnjaŋmal] | ‘anklet socks’ |

In words that undergo [n] insertion, the preceding consonant assimilates to [n] (Jun 2015). That the prefixes in the examples above are /hot^h/ and /təs/ is independently justified. The assimilation to [n] is motivated by an independent phonological process, and applies to more environments than the /ti/ cases presented here.

Notice that the prefix in (3b) ends with /s/ rather than /t/ or /t^h/. The [n] insertion illustrated above occurs with many consonants, not just coronal stops; therefore, [n] insertion is not the prefixation version of palatalization. In fact, as discussed in section 3, [n] insertion may also apply across a range of boundary strengths, whereas palatalization applies only at stem-suffix boundaries. However, a subset of these cases includes /t-i/ sequences, which means palatalization should be able to apply; but instead of palatalization, the chosen repair is [n] insertion.

1.2.3 Lack of overlap Despite occurring in the same structural context (i.e. [t-i]), palatalization and [n] insertion never overlap as derived environment effects. Palatalization never occurs at prefix-stem boundaries, and [n] insertion never occurs at stem-suffix boundaries, even though the phonological conditions are the same. Any account of Korean derived environment effects must accommodate palatalization in suffixes without leaving the possibility of [n] insertion in suffixes; likewise, the same account must motivate [n] insertion, but not palatalization, in environments derived by prefixation.

Even though [n] insertion occurs in more derived environments than [ti] or [tj] (for example, see (3b) with derived environment [sj]), a complete grammar must provide an account for why coronal stops palatalize before suffix [i] and [j], but never at prefix boundaries. Likewise, [n] insertion never occurs with suffixes, and must also be accounted for.

1.3 What drives derived environment effects? There have been many approaches to accounting for derived environment effects, including OT-based (Prince and Smolensky 1993) parallel approaches like Local Conjunction of Markedness and Faithfulness constraints (Łubowicz 2002) and Comparative

Markedness (McCarthy 2002). Each approach takes its own stance on what the driving force behind derived environment effects is. For Comparative Markedness, tracking the origin of marked sequences is key: change occurs because a new marked sequence appears, whereas old marked sequences do not merit change.

Local Conjunction, at least in its best-known analyses, drives derived environment effects by penalizing poor alignment between stems and syllable boundaries. Łubowicz (2002) points out that suffix derived environment effects like palatalization feature a misalignment of the right edge of a stem and a syllable. She goes on to argue that every suffix derived-environment effect is the result of a misalignment. The analysis proposed in this paper builds on that premise, incorporating prefix derived environment effects as well. Rather than assigning violations to misaligned stems, the proposed analysis uses Alignment constraints to assign violations to affixes that are misaligned to syllable boundaries.

This approach is advantageous because it has no need for environment-specific Markedness constraints—alignment itself drives the derived environment effect. Furthermore, the Alignment approach has the flexibility to account for not only prefix and suffix derived environments, but derived environment effects at other boundaries as well.

2 Formal Account

The account proposed here uses Alignment constraints (McCarthy and Prince 1993) to drive palatalization and [n] insertion. Section 2.1 defines and illustrates the constraints; section 2.2 shows how the constraints interact to yield the correct typology.

2.1 Constraint definitions and examples In this proposal, the distinct patterning of prefix-based and suffix-based derived environment effects arises from two Alignment constraints: ALIGN(prefix)-R and ALIGN(suffix)-L. ALIGN(prefix)-R drives derived environment effects when prefixes are misaligned, and ALIGN(suffix)-L drives derived environment effects when suffixes are misaligned. In Korean, ALIGN(prefix)-R drives [n] insertion, while ALIGN(suffix)-L drives palatalization.

- (4) ALIGN(prefix)-R — Assign a violation mark when a component (i.e. feature or segment) that is aligned with the right edge of a prefix is not aligned to the right edge of a syllable in the output.
- (5) ALIGN(suffix)-L — Assign a violation mark when a component (i.e. feature or segment) that is aligned with the left edge of a suffix is not aligned to the left edge of a syllable in the output.

ALIGN(prefix)-R will be used to motivate [n]-insertion; ALIGN(suffix)-L will accommodate palatalization.

Using Alignment to drive [n]-insertion is not novel. Jun (2015) proposes ALIGNMORPHEME-R (“the right edge of a morpheme coincides with the right edge of a syllable”), citing Shin (1997), Park (2005), Lee (2006), Lee and Lee (2006), Hwang (2008), and Ahn (2009). The primary difference between ALIGNMORPHEME-R and ALIGN(prefix)-R is evident in the names of the constraints: whereas ALIGNMORPHEME-R assigns violations when any morpheme is misaligned on the right side, ALIGN(prefix)-R only assigns violations to prefixes. ALIGN(prefix)-R will therefore assign violations more restrictively.

To understand how the constraints assign violations, consider the input forms /hoth-ipul/ ‘unlined comforter’ and /kath-i/ ‘together’. Where it is relevant in future illustrations, prefixes and suffixes are underlined to distinguish them from stems. The input form /hoth-ipul/ can be realized in at least three different ways: [ho.thipul], [ho.^{ch}ipul], and [hon.nipul] (laid out (6) below).

Both the fully faithful candidate [ho.thipul] and the palatalization candidate [ho.^{ch}ipul] incur violations of ALIGN(prefix)-R because the right edge of the prefix is not aligned to the right edge of a syllable. Instead, the right edge of the prefix ([th] or [^{ch}]) is in onset position, and therefore aligned to the left edge of a syllable.

(6)

<i>Output form</i>	<i>Description</i>	<i>Prefix aligned to right edge</i>
a. <u>h</u> o.t ^h ipul	Fully faithful	No
b. <u>h</u> o.c ^h ipul	Palatalizes	No
c. <u>h</u> on.nipul	Epenthesizes	Yes

The epenthesis candidate [hon.nipul] satisfies ALIGN(prefix)-R because the right edge of the prefix ([n], after assimilation to the following [n]) is aligned to the right edge of a syllable. This alignment is possible because of an epenthetic [n] which, when inserted to the right of the prefix, prevents the right edge of the prefix from becoming an onset.

ALIGN(suffix)-L assigns violations in a similar manner. For an input form /kat^h-i/ ‘together’, there are three relevant possible outputs: [ka.t^hi], [ka.c^hi], and [kan.ni].

(7)

<i>Output form</i>	<i>Description</i>	<i>Suffix aligned to left edge</i>
a. ka.t ^h i	Fully faithful	No
b. ka.c ^h i	Palatalizes	Yes
c. kan.ni	Epenthesizes	No

(8) UR: /kat^h-i/ a. *ka.t^hi b. ka.c^hi c. *kan.ni
 | | \ |
 [pal] [pal] [pal] [pal]

The fully faithful candidate [ka.t^hi] (8a) and the epenthesis candidate [kan.ni] (8c) fail to satisfy ALIGN(suffix)-L because the suffix [i] is in the nucleus of a syllable, preceded by an onset, and therefore not aligned to the left edge of a syllable. In the fully faithful candidate, onset position is filled by stem-final [t^h]; in the epenthesis candidate, the stem-final consonant is displaced to coda position by [n]-insertion, but because the epenthetic [n] was not part of the suffix to begin with, it has the same misalignment as the fully faithful candidate.

The winning candidate is [ka.c^hi] (8b), which crucially undergoes palatalization. In this case, palatalization can be thought of as spreading the [palatal] feature of [i] onto the consonant [t^h]. The resulting [c^h] corresponds to the input stem-final consonant, but shares the [palatal] component of the suffix. ALIGN(suffix)-L requires the alignment of any component of the left edge of a prefix to the left edge of a syllable—since a “component” could be a feature or a segment, spreading the feature [palatal] to the left edge of a syllable boundary satisfies ALIGN(suffix)-L.

Having established how ALIGN(prefix)-R and ALIGN(suffix)-L assign violations, the discussion now turns to the necessary rankings and resulting interactions of these constraints and the relevant Faithfulness constraints.

2.2 Constraint interactions and typology Alignment constraints are Markedness constraints, and therefore they potentially motivate violations of Faithfulness constraints. [n] insertion violates DEP, and palatalization violates IDENT(pal).

(9) DEP — Assign a violation for each segment in the output that does not have an input correspondent.

- (10) IDENT(pal) — Assign a violation for each segment that differs in the feature [palatal] between the input and the output.

2.2.1 Assigning violations ALIGN(prefix)-R and ALIGN(suffix)-L interact with DEP and IDENT(pal) according to the tableaux in (11) and (12).

(11)

<u>hot</u> ^h -ipul ‘unlined comforter’	ALIGN(prefix)-R	ALIGN(suffix)-L	DEP	IDENT(pal)
a. <u>ho</u> .t ^h ipul	*!			
b. <u>ho</u> .c ^h ipul	*!			*
☞ c. <u>hon</u> .nipul			*	

In the tableaux of /hot^h-ipul/ in (11), ALIGN(prefix)-R assigns crucial violations to output candidates (11a) and (11b). To satisfy ALIGN(prefix)-R, some component of the output segment corresponding to /t^h/ must be present at the right edge of a syllable. For each of those candidates, the right side of the prefix sits fully in onset position of the next syllable, and therefore cannot also be at the right edge (nucleus or coda position) of a syllable. Candidate (11c) satisfies ALIGN(prefix)-R because the content corresponding to the input segment /t^h/ is on the right edge of a syllable—even though it has changed its form to [n] by assimilation to the following epenthetic [n].

Violations of ALIGN(suffix)-L are never incurred for any output candidate of /hot^h-ipul/ because there is no suffix in the input. ALIGN(suffix)-L will never assign to prefixes or to prefix-stem interactions. Instead, ALIGN(suffix)-L assigns violation marks for stem-suffix concatenation, as described below.

(12)

kat ^h -i ‘together’	ALIGN(prefix)-R	ALIGN(suffix)-L	DEP	IDENT(pal)
a. ka.t ^h i		*!		
☞ b. ka.c ^h i				*
c. kan.ni		*!	*	

The tableau in (12) shows how ALIGN(suffix)-L assigns violations to the output candidates of the stem-suffix combination /kat^h-i/. Candidate (12a) receives a violation mark because no component—segment or feature—of the left side of the suffix, [i], is aligned with the left side of a syllable. Candidate (12c) violates ALIGN(suffix)-L for the same reason; epenthesis only keeps the suffix [i] farther from the left edge of its syllable. The winning candidate is (12b) because the [palatal] feature of [i] spreads left to the left edge of the syllable, palatalizing the underlying /t^h/ to [c^h]. The [palatal] component is aligned with the left edge of a syllable, satisfying ALIGN(suffix)-L at the expense of violating IDENT(pal). Notice that ALIGN(prefix)-R has no bearing on the candidate selection of this tableau; ALIGN(prefix)-R only assigns violations when a prefix is involved.

2.2.2 Alignment drives derived environment effects A key feature of this analysis is that the Markedness constraints motivating derived environment effects apply to a particular type of misalignment, rather than to particular segment sequences. They assign violations to certain morphological contact situations instead of assigning violations to specific strings of segments.

Consider a constraint like PAL, which is used in the Local Conjunction account to motivate palatalization. It can be defined as follows:

- (13) PAL — Assign a violation to the sequence of a coronal followed by a front vowel (Łubowicz 2002:256; definition changed to accommodate coronal palatalization rather than velar palatalization).

PAL is a Markedness constraint that targets a particular structure, like [ti]. PAL is a sequence-marking constraint, because it assigns violations to an environment in which a particular structure is followed by another structure—a marked sequence. Sequence-marking constraints like PAL are too narrow to motivate derived environment effects in general, because they only apply to one segmental sequence, and too broad to focus on a derived environment to the exclusion of non-derived environments.

PAL can only target environments that have a specific sequence of features. While the sequence [ti] is common among derived environments, it is not the only derived environment (nor is it even the only derived environment in Korean). An analysis that uses PAL to account for one derived environment effect must commit to including as many sequence-marking constraints as there are derived environment sequences. Furthermore, if each derived environment effect uses its own distinct repair, then the full analysis requires a particular and likely complicated arrangement of interleaving Markedness and Faithfulness constraints. PAL alone does not capture the generalization that derived environment effects all share misalignment, nor is it useful for any derived environment effects that do not feature a derived [ti] sequence.

While sequence-marking constraints like PAL are too narrow to capture generalizations about derived environment effects, they are also too broad to assign violations to derived environments specifically. In the case of PAL, all [ti] sequences are targeted, whether they are heteromorphemic or tautomorphemic. In order to preserve tautomorphemic [ti] sequences and assign violations to derived [ti] sequences, an additional constraint like R-ANCHOR(Stem; Syllable) is required.

- (14) R-ANCHOR(Stem; Syllable) — the rightmost segment of a stem in the input has a correspondent at the right edge of a syllable in the output (Łubowicz 2002:257).

A local conjunction between PAL and R-ANCHOR(Stem; Syllable) creates [PAL & R-ANCHOR(Stem; Syllable)], which only assigns a violation when both PAL and R-ANCHOR(Stem; Syllable) have both been violated. Local Conjunction, or similar constraint augmentations (e.g. McCarthy 2002), are the only way to make sequence-marking constraints work on derived environments while excluding non-derived environments.

In contrast, the Alignment approach developed here is broad enough to capture many derived environments, and does not intrinsically over-apply in non-derived environments. An analysis using Alignment only requires two Markedness constraints: ALIGN(prefix)-R and ALIGN(suffix)-L. These two constraints will apply to any derived environment in which affixes do not align appropriately with their syllables. Because they only assign violations to the edges of affixes, they cannot over-apply violations within morphemes.

The choice of specific repair, then, is determined by the interaction of Faithfulness constraints such as in Korean, where violating DEP and IDENT(pal) are the least costly ways of avoiding violations of ALIGN(prefix)-R and ALIGN(suffix)-L.

Removing sequence-specific Markedness constraints also eliminates the redundant “coupling” that exists between Markedness and Faithfulness constraints in other OT-based accounts of derived environment effects.

The typical Local Conjunction analysis of derived environment palatalization relies on PAL and IDENT(pal), in addition to R-ANCHOR(Stem; Syllable) and the conjunction of [PAL & R-ANCHOR(Stem; Syllable)]. The motivation for IDENT(pal) is clear: when palatalization occurs, a segment is gaining a [palatal] feature its input correspondent did not have. However, there is no motivation for a constraint like PAL, except as the formal impetus for violations of IDENT(pal). It has no independent motivation, aside from the fact that IDENT(pal) is violated in many languages. The primary reason to posit PAL is to drive violations of IDENT(pal). In some sense, PAL and IDENT(pal) are coupled together: PAL only exists because IDENT(pal) is violated. When PAL is used in Local Conjunction accounts, it only works when conjoined to other constraints that target misalignments—by itself, PAL is inadequate for correctly driving derived environment effects.

Removing PAL from the constraint system allows other constraints like Alignment to pull more weight, which they can do with independent motivation (McCarthy and Prince 1995). The nature of the derived

environment effect then comes, not from a coupling between Markedness and Faithfulness constraints, but from ranking of a particular Faithfulness constraint against an independently postulated Markedness constraint—an Alignment constraint. The driving factor in a derived environment effect is misalignment, rather than an ad-hoc sequence-marking constraint.

Using Alignment instead of sequence-marking constraints like PAL, the limits of derived environment effects emerge from the interaction of Alignment constraints with other constraints in the system. Consider, for example, /kap-i/, which would manifest on the surface as [ka.pi]. In this case, [pi] is a derived environment in which the suffix /i/ is not aligned to the left edge of a syllable, violating ALIGN(suffix)-L. If /kap-i/ were a Korean word, then because ALIGN(suffix)-L dominates IDENT(pal), in principle the output should be *[kapʲi], with a palatalized labial stop (an illicit combination). An easy way to restrict this sort of segment is to rely on high-ranked Markedness constraints like *pʲ, which correctly rules out [pʲ] in general.

In summary, four constraints are required to achieve a palatalization derived environment effect with a sequence-marking constraint: PAL, R-ANCHOR(Stem; Syllable), IDENT(pal), and the local conjunction [PAL & R-ANCHOR(Stem; Syllable)]. The Alignment approach requires just two constraints, ALIGN(suffix)-L and IDENT(pal). Using Alignment to drive palatalization reduces the number of constraints, generalizes appropriately to many different derived environments, and targets derived environments without also targeting non-derived environments.

2.2.3 Prefix repairs and suffix repairs cannot overlap ALIGN(prefix)-R and ALIGN(suffix)-L never interact with each other. By definition, they apply to different morpheme types, and therefore can never assign violations in the same conditions.

In both tableaux (11) and (12), the candidate that uses the non-optimal repair is harmonically bounded by the fully faithful candidate. In the case of prefixed /hoth-ipul/, the fully faithful candidate [ho.thipul] violates just ALIGN(prefix)-L, whereas the non-optimal repair [ho.^{ch}ipul] violates both ALIGN(prefix)-L and IDENT(pal). As a result, there is no ranking of these constraints in which [ho.^{ch}ipul] can be the winning output candidate. The winner in any language with these constraints will always be epenthetic candidate if ALIGN(prefix)-L outranks DEP, or the fully faithful candidate if DEP outranks ALIGN(prefix)-L (assuming DEP is the lowest ranking Faithfulness constraint whose violation would yield a form at satisfies the alignment constraint).

The same harmonic bounding occurs with ALIGN(suffix)-R. The non-optimal repair [kan.ni] violates both ALIGN(suffix)-R and DEP, whereas the fully faithful candidate violates just ALIGN(suffix)-R. The epenthetic output candidate [kan.ni] is harmonically bounded by the fully faithful candidate [ka.ti]. In any language with these constraints, the winning candidate will be palatalization candidate if ALIGN(suffix)-R outranks IDENT(pal), or the fully faithful candidate if IDENT(pal) outranks ALIGN(suffix)-R.

The empirical predictions of this harmonic bounding are straightforward: since palatalization is harmonically bounded in prefix conditions, palatalization should never appear as a derived environment effect after prefixation; likewise, since consonant epenthesis is harmonically bounded in suffix conditions, epenthesis should never occur as a derived environment effect after suffixation. These predictions seem to hold true in Korean, but other languages with similar derived environments should show a similar complementary distribution of prefix derived environment effects and suffix derived environment effects.

3 Discussion

Section 3.1 demonstrates the use of the Alignment approach with Turkish. Section 3.2 proposes that the Alignment approach can be extended to account for effects derived from compound or phrase boundaries.

3.1 Application to another derived environment effect This section examines a related case study from Turkish. Section 3.1.1 demonstrates the pattern; section 3.1.2 addresses the pattern using Alignment constraints; and, section 3.1.3 summarizes how the Turkish data validates and expands the predictions of the Alignment approach to derived environment effects.

3.1.1 Turkish velar deletion Turkish velar deletion is another stem-suffix derived environment effect. In these examples, velar consonants are deleted between vowels, but only if the second vowel comes from a different morpheme than the first vowel and velar consonant (data from Chong 2017, citing Inkelas 2011).

- (15) Suffix boundary velar deletion
- /bebek-In/ → [bebein] ‘baby-GEN’
 - cf. /bebek/ → [bebek] ‘baby-NOM’
 - /ipek-A/ → [ipee] ‘cotton-DAT’
 - cf. /ipek/ → [ipek] ‘cotton-NOM’
- (16) Non-deletion morpheme-internally
- /hareket/ → [hareket] ‘motion’
 - /sigorta/ → [sigorta] ‘insurance’
 - /sokak-A/ → [sokaa] ‘street-DAT’
 - /mekik-A/ → [mekie] ‘(weaver’s) shuttle-DAT’
- (17) Non-deletion when target velar is suffix-initial
- /sene-ki/ → [seneki] (*senei) ‘year-REL’
 - /ada-dA-ki/ → [adadaki] (*adadai) ‘island-LOC-REL’

The patterns in these data are similar to the suffixation patterns in Korean, but with a different repair (deletion instead of palatalization), and the need to block deletion from overapplying when the velar consonant is suffix-initial.

3.1.2 Applying Alignment to Turkish velar deletion This velar deletion fits neatly into an Alignment account using ALIGN(suffix)-L. For instance, consider /bebek-In/ in tableaux (19) below.

- (18) MAX — Assign a violation for each segment in the input that does not have an output correspondent.

(19)

/bebek-In/ ‘baby-GEN’	ALIGN(suffix)-L	IDENT(pal)	MAX
a. bebe.kin	*!		
b. bebe.in			*
c. bebe.cin		*!	

In this example, ALIGN(suffix)-L penalizes [bebe.kin] because the left edge of the suffix /In/ is not aligned to the left edge of a syllable. Whereas Korean repaired this violation with palatalization, Turkish repairs it with MAX. Velars will not delete from monomorphemic words or the ends of prefixes because ALIGN(suffix)-L can only assign violations when suffixes are not aligned properly. All that is needed to drive deletion instead of palatalization is to rank IDENT(pal) above MAX.

Turkish examples like /sene-ki/ offer another situation not addressed in the Korean data above: when a suffix begins with a consonant, a repair is not necessary (see (20) below).

To satisfy ALIGN(suffix)-L, some component of the /k/ in the suffix /ki/ must be aligned to the left edge of a syllable boundary. Crucially, ALIGN(suffix)-L is satisfied by the fully faithful candidate (assuming high-ranking Onset, as always). If velar deletion applied, the result would be an unnecessary violation of MAX. As in (19), palatalization would also be redundant.

(20)

/sene-ki/ ‘year-REL’	ALIGN(suffix)-L	IDENT(pal)	MAX
a. sene.ki			
b. sene.i			*!
c. sene.ci		*!	

3.1.3 Implications Taken together with Korean, Turkish velar deletion illustrates that Alignment constraints are sufficiently general to apply to multiple derived environments, without the need for Markedness constraints that target a specific environment. Different effects will appear based on the rankings of Faithfulness constraints above and below ALIGN(prefix)-R and ALIGN(suffix)-L.

Words like /sene-ki/ reveal another empirically verifiable prediction: derived environment effects will never occur if a suffix begins with a consonant, in a language with phonotactics like those of Korean or Turkish. Consonant-initial suffixes will satisfy ALIGN(suffix)-L because the initial suffix will be placed in onset position. Cases in which a derived environment effect changes an affix but not the stem should be typologically rare, or nonexistent.

3.2 Beyond affixes This section introduces a possible extension to the Alignment approach, operating over more boundaries than just prefix-stem and stem-suffix boundaries. Section 3.2.1 introduces the motivation for this extension; section 3.2.2 illustrates how the extension would apply; and, section 3.2.3 discusses the typological implications of such an extension.

3.2.1 Compound and phrase boundaries While [n] insertion does not apply between stems and suffixes, it is not restricted to applying solely between stems and prefixes. In fact, [n] can be inserted between two morphemes at prefix-stem boundaries (21a-b), compound boundaries (21c-e), and phrase boundaries (21f-g) (Jun 2014, 2015).

- (21)
- a. /hot^h-ipul/ [honnipul] ‘unlined comforter’
 - b. /təs-jaŋmal/ [tənnjaŋmal] ‘anklet socks’
 - c. /pat^h-ilaŋ/ [pannilaŋ] ‘ridge of a field’
 - d. /com-jak/ [comnjak] ‘mothball’
 - e. /som-ipul/ [sønnipul] ‘cotton sheet’
 - f. /han # il/ [hannil] ‘thing that (someone) has done’
 - g. /mæk-il # jəs/ [mækiljət] ‘taffy that (someone) will eat’

One way to extend the Alignment approach to different boundary types would be to substitute different boundary identities into the Align framework. ALIGN(prefix)-R would account for (21a-b), and constraints like ALIGN(stem)-R and ALIGN(phrase)-R would accommodate (21c-e) and (21f-g).

An advantage of this extension of Alignment over other approaches is the ability to account for outputs that are heterogenous on the surface composed of the same string of segments in the input. Consider /pat^h-ilaŋ/ in (22):

- (22)
- a. /pat^h-ilaŋ/ [pannilaŋ] ‘ridge of a field’
 - b. /pat^h-ilaŋ/ [paç^hilaŋ] ‘field and...’

In this case, the same derived environment receives two repairs: palatalization if the second morpheme is a suffix, and [n]-insertion if the second morpheme is a stem. The Alignment approach fits here, using ALIGN(stem)-R and DEP to account for (22a), and ALIGN(suffix)-L and IDENT(pal) to account for (22b).

3.4.2 Illustration of the extension The extension of this approach to compound words would look like the tableau in (24).

- (23) ALIGN(stem)-R — Assign a violation mark when a component (i.e. feature or segment) that is aligned with the right edge of a stem in the input is not aligned to the right edge of a syllable.

(24)

/pat ^h -ilan/ ‘ridge of a field’	ALIGN(suffix)-L	ALIGN(stem)-R	IDENT(pal)	DEP
a. pa.t ^h ilan		*!		
b. pa.c ^h ilan		*!	*	
☞ c. pan.nilan				*

ALIGN(stem)-R assigns violations to both candidates (24a) and (24b) because the rightmost correspondent of the stem /pat^h/ is not aligned to the right edge of a syllable. Candidate (24c) uses [n] insertion as a repair, keeping the right edge of the stem aligned with the right edge of its syllable. IDENT(pal) is harmonically bound by the fully faithful candidate, predicting that palatalization will never occur at a compound boundary. Notice that ALIGN(stem)-L cannot achieve the same effect, since [n] insertion prevents the left edge of /ilan/ from being aligned to the left edge of a syllable.

ALIGN(suffix)-L assigns violations with the same outcome as before (see (18)).

(25)

/pat ^h -ilan/ ‘field and...’	ALIGN(suffix)-L	ALIGN(stem)-R	IDENT(pal)	DEP
a. pa.t ^h ilan	*!	*		
☞ b. pa.c ^h ilan		*	*	
c. pan.nilan	*!			*

In the tableau in (25), ALIGN(suffix)-L assigns violations to candidates (25a) and (25c) because no component of the left edge of the suffix /ilan/ is aligned to the left edge of a syllable. The epenthetic candidate (25c) is harmonically bound by the fully faithful candidate (25a), ensuring that [n] insertion will never occur at stem-suffix boundaries (the same conclusion drawn in section 2 above).

If ALIGN(stem)-R assigns violations when a component of the right edge of a stem is not aligned to the right edge of a syllable, then candidate (25b) should violate ALIGN(stem)-R. This candidate, [pa.c^hilan], violates ALIGN(stem)-R because the output correspondent of the right edge of the stem, [c^h], is not aligned to the right edge of a syllable. Unlike the ranking between ALIGN(suffix)-L and ALIGN(prefix)-R, the ranking between ALIGN(suffix)-L and ALIGN(stem)-R is crucial. If ALIGN(stem)-R outranked ALIGN(suffix)-L, then the epenthetic candidate (26a) would win, and palatalization would never occur in Korean suffixes (see the tableau in (26)). Therefore, ALIGN(suffix)-L must outrank ALIGN(stem)-R.

(26)

/pat ^h -ilan/ ‘field and...’	ALIGN(stem)-R	ALIGN(suffix)-L	IDENT(pal)	DEP
a. pa.t ^h ilan	*!	*		
⊖ b. pa.c ^h ilan	*!		*	
☞ c. pan.nilan		*		*

3.4.3 Predictions of the extension The introduction of ALIGN(stem)-R introduces some new predictions for this account. In particular, it predicts that some languages will have the same derived environment effect at both prefix-stem boundaries and stem-suffix boundaries (as in 26 above), while other languages will have different, non-overlapping derived environment effects at prefix-stem and stem-suffix boundaries (like Korean).

The first type of language, with a single derived environment effect at all boundaries, would be those in which ALIGN(stem)-R outranks ALIGN(suffix)-L. The combined efforts of high-ranking ALIGN(prefix)-R and ALIGN(stem)-R would ensure that potential right-edge misalignments are always repaired, even if the result is a misaligned suffix's left edge. There would be only one possible repair, because the same Faithfulness constraint that is most optimally violated to satisfy ALIGN(prefix)-R would be the best constraint to satisfy ALIGN(stem)-R. The second type of language, in which prefix-based and suffix-based derived environment effects are not the same, would arise from a ranking in which ALIGN(suffix)-L dominates ALIGN(stem)-R. There would be two distinct repairs, because the Faithfulness constraint most optimally violated to satisfy ALIGN(suffix)-L would not be the same constraint whose violations most optimally satisfy ALIGN(prefix)-R or ALIGN(stem)-R.

(27)

<i>Ranking</i>	<i>Prefix misalignments</i>	<i>Stem misalignments</i>	<i>Suffix misalignments</i>
ALIGN(stem)-R, ALIGN(prefix)-R >> ALIGN(suffix)-L	[n] insertion	[n] insertion	No repair
ALIGN(suffix)-L >> ALIGN(stem)-R	[n] insertion	No repair	Palatalization

There are other ways to account for /pat^h-ilan/ without using Alignment. One could posit two different /i/s, one appearing in the suffix /ilan/ 'and...', and one appearing in the stem /ilan/ 'ridge' (for example, see Kaplan's (1981) analysis of Barrow Inupiaq /i/). Similarly, one could suggest two different morphemes /pat^h/, each meaning 'field', but each with a different reaction to morpheme concatenation. Both of these approaches are prohibitively abstract.

A local conjunction approach could work if there were multiple conjunctions, each applying over a different boundary domain (i.e. suffix-stem, stem-stem). However, the implications of multiple domains are similar to the predictions of ALIGN(stem)-R; and because the local conjunction/sequence-marking constraint approach relies on more constraints and fails to generalize to as many derived environments, the Alignment approach makes these predictions more elegantly.

4 Conclusion

Derived environment effects can be driven by Alignment constraints. Using Alignment provides a more unified account of derived environment effects, since each derived environment is targeted by the same set of constraints rather than by a specific sequence-marking constraint for every single environment. The approach can be extended to account for morpho-phonological phenomena beyond morpheme boundaries, including at compound boundaries and phrase boundaries.

References

- Ahn, Mee-Jin. 2009. /n/-insertion and onset simplification in Kyungsang Korean. *Studies in Phonetics, Phonology and Morphology* 15. 263–282.
- Ahn, Sang-Cheol. 1985. *The interplay of phonology and morphology in Korean*. University of Illinois at Urbana-Champaign.
- Beckman, Jill. 1997. Positional faithfulness, positional neutralization and Shona vowel harmony. *Phonology*, vol 14, p. 1-46.
- Beckman, Jill. 1999. *Positional Faithfulness*. New York: Garland.

- Cho, T. 2001. Effects of morpheme boundaries on intergestural timing: Evidence from Korean. *Phonetica*, 58, 129–162. <http://doi.org/10.1159/000056196>
- Chong, Adam. 2017. Examining the lexicon in derived-environment effects: Korean and Turkish. *Proceedings of the 2016 Annual Meeting on Phonology*.
- Hayes, Bruce and Wilson, Colin. 2008. A Maximum Entropy Model of Phonotactics and Phonotactic Learning. *Linguistic Inquiry*, vol. 39 (3), p. 379-440.
- Hwang, Sangjin 2008. Korean speakers' knowledge of /n/-insertion: p-map approach. MA thesis, Seoul National University.
- Inkelas, Sharon. 2011. Another look at velar deletion in Turkish, with special attention to the derived environment condition. In E. E. Taylan & B. Rona (Eds.), *Puzzles of Language: Essays in honour of Karl Zimmer*. Wiesbaden: Harrassowitz.
- Iverson, Gregory K. and Wheeler, Deirdre W. 1988. Blocking and the Elsewhere Condition, in *Theoretical Morphology: Approaches in Modern Linguistics*, eds. Michael Hammond and Michael Noonan. Academic Press, Inc.
- Jun, Jongho. 2014. Phonological variation in Seoul Korean n-insertion. The 45th annual meeting of the North East Linguistic Society. Retrieved from http://ling.snu.ac.kr/jun/work/2014_n-ins_nels.pdf
- Jun, Jongho. 2015. Korean n-insertion: a mismatch between data and learning. *Phonology*, 32(3), 417–458. <http://doi.org/10.1017/S0952675715000275>
- Kaplan, Lawrence. D. 1981. *Phonological Issues in North Alaskan Inupiaq*. Alaska Native Language Center Research Papers (Vol. 6).
- Kim, Soo-Gon. 1976. Palatalization in Korean. Ph.D. Dissertation, The University of Texas at Austin.
- Kiparsky, Paul. 1973. Abstractness, opacity, and global rules, in Osamu Fujimura (ed.), *Three Dimensions of Linguistic Theory*, TEC, Tokyo, 57-86.
- Lee, Minkyung. 2006. Gyeongsang Korean n-insertion revisited. *Studies in Phonetics, Phonology and Morphology* 12. 623–641.
- Lee, Yongsung & Minkyung Lee. 2006. n-insertion as y-devocalization in Korean. *Korean Journal of Linguistics* 31. 413–440.
- Lubowicz, Anna. 2002. Derived environment effects in Optimality Theory. *Lingua*, 112, 243–280. [http://doi.org/10.1016/S0024-3841\(01\)00043-2](http://doi.org/10.1016/S0024-3841(01)00043-2)
- Mascaró, Joan. 1976. Catalan phonology and the phonological cycle. Ph.D. dissertation, MIT. Distributed 1978, Indiana University Linguistics Club.
- McCarthy, John J. 2002. Comparative markedness (long version). *Linguistics Department Faculty Publication Series*. Retrieved from http://scholarworks.umass.edu/linguist_faculty_pubs/30
- McCarthy, John J., and Alan Prince. 1993. Generalized Alignment. In: G. Booij and J van Marle (eds) *Yearbook of Morphology*, 79-153. Dordrecht: Kluwer.
- McCarthy, John J., and Alan Prince. 1995. "Faithfulness and reduplicative identity." 249.
- Mohanan, Karuvannur. 1982. *Lexical Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Park, Sunwoo. 2005. Positional markedness and /n/-insertion in Korean. [Wichicek yuphosengkwa hankwukeyu niunsapip.] *Studies in Phonetics, Phonology and Morphology* 11. 323–335.
- Prince, Alan and Smolensky, Paul. 1993. *Optimality Theory: Constraint interaction in generative grammar*. New Brunswick, NJ: Rutgers University Center for Cognitive Science. [Reprinted in *Optimality Theory in Phonology: A Reader*, ed. by John J. McCarthy, Malden, MA and Oxford, Blackwell (2003). Available on Rutgers Optimality Archive, ROA-537.]
- Shin, Seung-Hoon. 1997. Constraints within and between syllables: syllable licensing and contact in Optimality Theory. PhD dissertation, Indiana University.