Unifying Morphologically-Derived Environment Effects and Blocking*

Reed Blaylock

University of Southern California

1. Overview

This paper proposes an account that unifies morphologically-derived environment effects and derived environment blocking. Section 2 gives an overview of derived environment effects and blocking, as well as the motivation for combining them in Harmonic Serialism. Section 3 offers the formal account. Section 4 briefly discusses this approach in comparison with some alternatives, and Section 5 concludes.

2. Introduction

Morphologically-derived environment effects and morphologically-derived environment blocking have been treated as separate phenomena, but are in fact quite similar. I propose a mechanism for uniting morphologically-derived environment effects and derived environment blocking in a single framework.

Derived environment effects and derived environment blocking are phonological changes 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). For instance, morphologically-derived environment effects are those that happen near morpheme boundaries due to morpheme concatenation. In the Korean example below (taken from Ahn 1985), palatalization is blocked tautomorphemically, but mandated across a morpheme boundary (Ahn 1985, Kim 1976a).

*I am grateful to the PhonLunch group at USC for their insight, and especially to Karen Jesney and Charlie O'Hara.

(1) *Morphologically derived environment effect, classical derivation*

/mat/ 'eldest' *(condition not met)*	/mati/ 'knot' *(blocked by strict cyclicity)*	*0th cycle*
[[mat] i]	-----	*add morpheme (1st cycle)*
[[matʃ] i]	-----	*t-palatalization*
[madʒi]	[madi]	*voicing (post-lexical stage)*
[madʒi] 'the eldest'	[madi] 'knot'	*final output*

The example in (1), borrowed from Ahn (1985), shows a morphologically-derived environment effect in the Lexical Phonology framework (Mohanan 1982, Kiparsky 1982). In Lexical Phonology, morphemes are added in successive cycles by adding layers of brackets around the new word, as [mat] → [[mat] i] in the example on the left. Certain phonological processes only occur within these cycles—in Korean, coronal palatalization is one of those processes. In the first cycle, the nominalizer morpheme /-i/ is added, creating the opportunity for palatalization, which then applies in the same cycle (as [[mat] i] → [[matʃ] i]). The internal brackets are removed before the post-lexical stage, which is where the exceptionless processes like inter-sonorant voicing occur ([matʃi] → [madʒi]).

The example on the right side, underlying /mati/, will never palatalize because the /ti/ sequence is within the root, not divided across brackets. 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] → [madi].

This is a morphologically-derived environment effect: an effect (i.e. palatalization) that only applies in environments derived through morphological concatenation.

Derived environment blocking is the inverse of a derived environment effect. Whereas in a morphologically-derived environment effect the process only applies after morpheme concatenation, a morphologically-derived environment blocking applies in all environments *except* after morpheme concatenation. Northern Irish dentalization is an example of morphologically-derived environment blocking (Benua 1997).

(2) *Morphologically derived environment blocking, classical derivation*

/leyt/ 'late' *(condition not met)*	/lædər/ 'ladder' *(blocked by strict cyclicity)*	*0th cycle*
[[leyt] ər]	-----	*add morpheme (1st cycle)*
-----	[læd̪ər]	*dentalization*
[leytər] 'later'	[læd̪ər] 'ladder'	*final output*

In (2), /leyt/ and /lædər/ are inputs to the phonological derivation. In the first cycle, the /ər/ morpheme is added to /leyt/, and nothing is added to /lædər/. By the middle of the first cycle, both forms have a marked coronal-rhotic sequence (/tər/ in /leytər/, /dər/ in /lædər/). Dentalization is the repair of choice for Northern Irish, but it never applies across morpheme boundaries. When dentalization is applied in (2), /leytər/ does not participate because the internal brackets surrounding /leyt/ create a boundary between the /t/ and /ər/;

/lædər/, on the other hand, has no morpheme juncture, and so dentalization can apply. The internal brackets of /leytər/ are removed before the final output, but the opportunity to dentalize has already passed.

The case above is morphologically-derived environment blocking: an effect (here, dentalization) is blocked in environments that have been derived through morphological concatenation, but not blocked elsewhere.

Descriptions of derived environment effects tend to characterize them as serial phenomena: one change happens that directly leads to another. Naturally, this is difficult to express formally in Optimality Theory (Prince and Smolensky 1993). Optimality Theory looks for the globally optimal candidate, one that satisfies the most high-ranking constraints. Partial neutralizations are impossible without a special constraint set (e.g. Positional Faithfulness constraints, Beckman 1998). Derived environment effects, being a particular kind of partial neutralization, cannot rely on basic Markedness and Faithfulness constraints—with normal constraints, the change will either happen all the time or never.

Consider the example of Korean palatalization from (1). In Optimality Theory, this push to palatalize or not palatalize could be expressed in terms of a Markedness constraint like PAL and a Faithfulness constraint like IDENT(cor).

(3) a. IDENT(cor) — The specification of a feature [coronal] of an input segment must be preserved in its output correspondent.
b. PAL — Assign a violation mark (*) to a velar segment followed by a non-low front vowel.

To get the palatalization we expect in derived forms, PAL must dominate IDENT(cor)—Markedness drives change, and must be high-ranked.

(4) *Korean derived environments: Markedness >> Faithfulness*

mat-i	PAL	IDENT(cor)
☞ a. maʧi		*
b. mati	*!	

Palatalization only occurs in derived environments. In non-derived environments, palatalization should not be possible. But ranking PAL over IDENT(cor) affects every segment that violates PAL, not just the ones at derived environments. To block this change in non-derived environments requires a ranking of Faithfulness over Markedness.

(5) *Korean non-derived environments: Faithfulness >> Markedness*

mati	IDENT(cor)	PAL
a. maʧi	*!	
☞ b. mati		*

This gives us a ranking paradox. Markedness must dominate Faithfulness in one circumstance, but the opposite ranking must be true at other times. Some additional constraint is necessary to solve this conundrum.

Derived environment blocking has a similar problem. Using the Northern Irish dentalization from (2), it is clear that in derived environments, where dentalization is blocked, Faithfulness (IDENT(dist)) must outranking Markedness (*TR).

(6) a. *TR — Assign a violation mark (*) to every alveolar segment followed by a rhotic.
b. IDENT(dist) — The specification of a feature [distributed] of an input segment must be preserved by its output correspondent.

(7) *Northern Irish derived environments: Faithfulness >> Markedness*

leyt	IDENT(dist)	*TR
a. leyt̪ər	*!	
☞ b. leytər		*

If this were true all the time, then dentalization would never occur. In non-derived environments, the opposite ranking is crucial to motivate dentalization.

(8) *Northern Irish non-derived environments: Markedness >> Faithfulness*

lædər	*TR	IDENT(dist)
☞ a. læd̪ər		*
b. lædər	*!	

Derived environment effects and blocking share the same type of ranking paradox, and both need some additional constraint to solve the problem. Notice, though, that the rankings for derived environment blocking are opposite to the rankings of derived environment effects: blocking requires Faithfulness over Markedness in derived environments, while the effect needs Markedness over Faithfulness. Unifying derived environment effects and derived environment blocking introduces a second kind of paradox.

The solution offered in the next section attempts to solve both of these paradoxes. Using Harmonic Serialism (McCarthy 2008) rather than classic Optimality Theory adds levels to the phonological derivation between initial input and final output; this makes it possible to reference the order in which phonological changes occur, and to settle on a locally optimal candidate rather than the globally optimal candidate. A new constraint is introduced that assigns violations differently in derived forms than in non-derived forms, and which can work with Faithfulness or Markedness constraints to produce derived environment effects and blocking.

3. Morphologically-derived environment effects and blocking, unified

Section 3.1 provides an overview of the theoretical assumptions for this proposal. Section 3.2 offers a new constraint, DE-BRACKET, which can be used to produce morphologically-derived environment effects and blocking. Section 3.3 applies this constraint to attested cases of morphologically-derived environment effects (Polish) and morphologically-derived environment blocking (Northern Irish).

3.1 Theoretical assumptions

3.1.1 Harmonic Serialism

Harmonic Serialism (Prince and Smolensky 1993, McCarthy 2006, 2007ab, 2008, Wolf 2008) is a variant of Optimality Theory (Prince and Smolensky 1993) that uses ranked constraints to select the locally optimal phonological output. Potential outputs are restricted to forms that deviate from the input by only one change—as a result, sometimes the globally optimal candidate of classic Optimality Theory is never available for selection. For simplicity, let us assume that one change violates one Faithfulness constraint (McCarthy 2007b). The optimal output from one step becomes the input to the next step, and multiple steps may be required before no more change are possible and the derivation converges.

In this approach, the phonological content of a morpheme is inserted as a Step in a Harmonic Serialism derivation. Because the input form is different from the output form (an abstract morpheme in the input vs its abstract phonological realization in the output), inserting a morpheme's phonological material constitutes a change. Likewise, failing to insert phonological material violates a Markedness constraint (MORPH-REAL).

3.1.2 Serial morpheme insertion

Under this approach, morphemes are introduced serially into phonology (Wolf 2008). For our purposes, roots are inserted first, followed by suffixes—that is, left to right. Each morpheme insertion counts as a single change in Harmonic Serialism, and so morphemes can only be inserted one at a time, once per each Step.

For example, the word [kæts] ("cats") is composed of the morphemes <cat> and <plural>. The <cat> morpheme would be inserted first, then the plural morpheme in the next Step. The insertion of the plural morpheme also introduces a marked /tz/ sequence; voicing assimilation in the third Step fixes this. After voicing assimilation, there are no more changes to be made, so the final output [kæts] is given. This is summarized in (9).

(9) <cat> + <plural> → kæt + <plural> → kæt + z → kæts

Inserting morphemes serially like this makes it possible to know which morpheme was just added. This is crucial for a serial understanding of derived environment effects that relies on phonological changes fed by prior morpheme insertion. Such an approach would not be possible in classic Optimality Theory where morphemes are all inserted at the same time, because when every environment is derived at the same time, there is no longer a distinction between derived and non-derived environments.

3.1.3 Brackets

Morphologically-derived environment effects and blocking require some reference to the level of morphological complexity of a phonological form. Morphologically-complex things can be targeted by derived environment effects or blocked by derived environment blocking, but derived environment effects should not be able to apply past a certain point.

The proposal in this paper uses the notion of morpheme brackets to encode opportunities for derived environment effects and blocking to occur. Morpheme brackets ({ and }) surround segments that constitute a single morphological unit. The phonological content of morphemes is initially inserted with brackets. Then, GEN automatically removes brackets in the next step. Crucially, bracket removal does not count as a change in Harmonic Serialism—it is automatic in GEN, and does not violate any Faithfulness constraints. However, because brackets are removed between some input and output, constraints can reference bracket removal.

To see how this might work, consider the previous example of “cats”:

(10) a. MORPH-REAL — A morpheme must have some phonological exponent in the output (Kurisu 2001, Samek-Lodovici 1993).
b. AGREE(voice) — Adjacent output segments have the same value of the feature voice (Baković 2007).
c. IDENT(voice) — The specification of a feature [voice] of an input segment must be preserved in its output correspondent.

(11) *Voicing assimilation in English plurals using serial morpheme insertion*

Step 1: insert <cat> phonology

<cat> + <plural>	MORPH-REAL	AGREE(voice)	IDENT(voice)
a. <cat> + <plural>	**!		
☞ b. {kæt} + <plural>	*		

Step 2: insert <plural> phonology

{kæt} + <plural>	MORPH-REAL	AGREE(voice)	IDENT(voice)
a. kæt + <plural>	*!		
☞ b. kæt + {z}		*	

Step 3: apply voicing assimilation

kæt + {z}	MORPH-REAL	AGREE(voice)	IDENT(voice)
a. kæt + z		*!	
☞ b. kæt + s			*

Final output: kæts

In Step 1, the phonological content of <cat> is inserted with morpheme brackets (1b). The alternative, (1a), inserts no phonological realization of any morpheme and violates MORPH-REAL twice. Therefore, inserting one morpheme is a harmonic improvement.

In Step 2, the input form is the same as the output of Step 1: the phonological content of the morpheme <cat>, and the non-filled morpheme <plural>. Based on this input, GEN returns a fully faithfully output candidate as well as candidates that differ from the input by one change. Notice, however, that the morphemes brackets around /kæt/ are completely gone in the output candidates of Step 2—bracket removal is an automatic component GEN, and happens to all morphemes that had brackets in the input. The winning candidate of Step 2 is (2b), where the content of <plural> is inserted, surrounded by morpheme brackets.

While this morpheme insertion introduces a marked /tz/ sequence, it is still more harmonic than inserting no morphological content at all.

In Step 3, the input form has brackets around the plural morpheme {z}. These brackets are removed by GEN when making the output candidates. Because no morpheme was inserted in Step 3, no part of the output candidates could have morpheme brackets. Candidate (3b) satisfies AGREE(voice), whose violation was introduced in Step 2. Although changing the [voice] specification of /z/ violates IDENT(voice), it is more harmonic to change voicing than to choose an output with such a marked sequence.

After Step 3, there are no more harmonically improving changes that can be made, and so the derivation converges with a final phonetic output of [kæts].

3.2 DE-BRACKET constraint

In addition to the morpheme brackets described above, I propose a constraint that references the removal of brackets: DE-BRACKET(X). It is a constraint function that takes a normal Markedness or Faithfulness constraint X and alters the way in which it assigns violations. For other examples of constraint functions, see Positional Faithfulness (Beckman 1998) or Local Conjunction (Smolensky 1993, Łubowicz 2002, 2005) (to name a few).

(12) DE-BRACKET(X) — Assign a violation mark (*) to a segment that violates constraint X without also removing an adjacent morpheme bracket to which the segment is external.

DE-BRACKET(X) assigns a violation mark when some constraint X is violated *but* an adjacent morpheme bracket has not been removed. Violating X in a non-adjacent position with respect to the morpheme juncture also violates DE-BRACKET(X)—violations of X are possible, but doing so also violates DE-BRACKET(X).

Because of the serial insertion of morphemes, the phonological content inside morpheme brackets is always the most recently inserted morpheme. In most derived environment effects, the segment that changes is in the word stem, not the suffix (Inkelas 2014). Since suffixes are inserted after stems, the content from the stem is necessarily outside the brackets of the suffix. Therefore, the locus of violation is a segment with "an adjacent morpheme bracket to which the segment is external".

To understand DE-BRACKET(X) better, let us first see a toy example before applying it to real data in Section 3.3.

(13) DE-BRACKET(IDENT(cor)) — Assign a violation mark (*) to a segment that violates IDENT(cor) without also removing an adjacent morpheme bracket to which it is external.

In (14), IDENT(cor) assigns a violation to /paʧi/ because the output segment [ʧ] differs from its input correspondent /t/ in the [coronal] feature. Because IDENT(cor) is violated without having removed a bracket next to /t/, DE-BRACKET(IDENT(cor)) also assigns a violation.

(14) *Violation of* IDENT(cor) *without bracket removal violates* DE-BRACKET

{pati}	DE-BRACKET(IDENT(cor))	IDENT(cor)
a. patʃi	*	*
b. pati		

In (15), on the other hand, while IDENT(cor) still assigns a violation to the changed [ʧ], DE-BRACKET(IDENT(cor)) is satisfied because [ʧ] is adjacent to the place where a morpheme bracket was just removed—the juncture between [ʧ] and [i].

(15) *Violation of* IDENT(cor) *with bracket removal satisfies* DE-BRACKET

pat + {i}	DE-BRACKET(IDENT(cor))	IDENT(cor)
a. paʧ + i		*
b. pat + i		

3.3 Application of DE-BRACKET

3.3.1 Different results using Markedness and Faithfulness constraints

By definition, violations of Faithfulness constraints result in phonological change. When constraint X of DE-BRACKET(X) is a Faithfulness constraint, then the DE-BRACKET(FAITH) constraint prevents change except when a bracket is removed—that is, change occurs at a morpheme juncture from which a bracket was just taken away. When change is restricted to morpheme junctures like this, it is a morphologically-derived environment effect.

When X is a Markedness constraint, the opposite is true. Markedness constraints motivate change between input and output forms, so violations of Markedness constraints happen when there is no change. When X of DE-BRACKET(X) is a Markedness constraint, change cannot happen where a morpheme bracket was just removed. When change is blocked at a morpheme juncture like this, the phenomenon is morphologically-derived environment blocking.

Section 3.3.2 shows how DE-BRACKET(FAITH) produces a morphologically-derived environment effect found in Polish, and Section 3.3.3 illustrates morphologically-derived environment blocking in Northern Irish using DE-BRACKET(MARK).

3.3.2 Morphologically-derived environment effect: Polish

A well-known morphologically-derived environment effect is Polish First Velar Palatalization (e.g. Rubach 1984, Łubowicz 2002). In this rule, /k/ → /ʧ/ preceding a morpheme that begins with a front vowel (/i/ or /ɛ/). Some examples follow (from Rubach 1984 and Gussmann 2007):

(16) *Polish velar palatalization in morphologically-derived environments*

krok	[krok]	‘step’	kroczek	[krɔʧ-ɛk]	‘step *dim.*’
strach	[strax]	‘fright’	straszyć	[straʃ-ɨʨ]	‘frighten’
bok	[bɔk]	‘side’	boczek	[boʧ-ɛk]	‘side *dim.*’
brzuch	[bʒux]	‘belly’	brzuszysko	[bʒuʃ-ɨskɔ]	‘belly *aug.*’

The examples on the left in (16) show the surface forms of these Polish roots; the words on the right of (16) are their suffixed forms, illustrating that root-final velars palatalize before suffixes beginning with non-low front vowels. This type of palatalization, only occurs when morphemes have been concatenated; it does not happen morpheme-internally.

This can be described with four constraints: MORPH-REAL, abbreviated to MR, IDENT(cor), PAL, and DE-BRACKET(IDENT(cor)), abbreviated to DB(ID(cor)). Example (18) shows blocking in non-derived environments, and (19) shows how the morphologically-derived environment effect emerges.

(17) a. IDENT(cor) — The specification of a feature [coronal] of an input segment must be preserved in its output correspondent.
b. PAL — Assign a violation mark (*) to a velar segment followed by a non-low front vowel.
c. DE-BRACKET(IDENT(cor)) — Assign a violation mark (*) to a segment that violates IDENT(cor) without also removing an adjacent morpheme bracket to which it is external.

The tableaux (18) are an example of how DE-BRACKET(FAITH) blocks change outside a morpheme boundary. In Step 1, the phonological content of <waiter> is inserted along with morpheme brackets, giving the locally-optimal output [kɛlnɛr]. In Step 2, although the /k/ of /kɛlnɛr/ is in the environment for palatalization, it cannot palatalize because no morpheme bracket was removed between it and its palatalizing environment, /ɛ/. The output that palatalizes is a loser because it violates DE-BRACKET(IDENT(cor)).

(18) *No palatalization within a morpheme*

Step 1: insert <waiter> phonology

<waiter>	MR	DB(ID(cor))	PAL	IDENT(cor)
a. <waiter>	*!			
☞ b. {kɛlnɛr}			*	

Step 2: remove brackets; palatalization is impossible

{kɛlnɛr}	MR	DB(ID(cor))	PAL	IDENT(cor)
a. ʧɛlnɛr		*!		*
☞ b. kɛlnɛr			*	

Final output: [kɛlnɛr]

The derivation in (19) is a morphologically-derived environment effect, the natural counterpart to (18). Whereas in (18) the /k/ segment could not palatalize because there was no morpheme bracket removed, the second /k/ in /krɔk+ɛk/ does palatalize. In Step 1, the phonological content of <step> is inserted, along with brackets; Step 2 removes those brackets, and adds the diminutive suffix with its own brackets. Finally, Step 3 removes the brackets on the diminutive morpheme, and in doing so, allows the stem-final /k/ to palatalize to [ʧ]. DE-BRACKET(IDENT(cor)) assigns no violation because the violation of IDENT(cor) happened at a morpheme boundary.

(19) *Palatalization occurs at a morpheme boundary*

Step 1: insert <step> phonology

<step> + <*dim*>	MR	DB(ID(cor))	PAL	IDENT(cor)
a. <step> + <*dim*>	**!			
☞ b. {krɔk} + <*dim*>	*		*	

Step 2: insert <*dim*> phonology, remove brackets on /krɔk/

{krɔk} + <*dim*>	MR	DB(ID(cor))	PAL	IDENT(cor)
a. krɔk + <*dim*>	*!			
☞ b. krɔk + {ɛk}			*	

Step 3: palatalize /k/ while removing brackets on /ɛk/

krɔk + {ɛk}	MR	DB(ID(cor))	PAL	IDENT(cor)
a. krɔk + ɛk			*!	
☞ b. krɔʧ + ɛk				*

Final output: **[**krɔʧɛk]

3.3.3 Morphologically-derived environment blocking: Northern Irish

Northern Irish (Benua 1997) blocks coronal dentalization in morphologically-derived environments.

(20) *Northern Irish: coronals are dentalized preceding rhotics*

‘train’	[t̪reyn]
‘matter’	[mæt̪ər]
‘ladder’	[læd̪ər]
‘pillar’	[pɪl̪ər]

(21) *Northern Irish: at morpheme boundaries, coronals are not dentalized*

‘later’	[leytər]
‘louder’	[laudər]
‘cooler’	[kulər]
‘bedroom’	[bedrʊm]

Before rhotics, coronals /t, d, l, n/ become dentalized (20). However, dentalization is blocked when the coronal comes at the end of a root in a morphologically complex word (21). In this environment derived by morpheme concatenation, a normally ubiquitous process (dentalization) is blocked—hence, morphologically derived environment blocking.

The constraints used here are MORPH-REAL (abbreviated MR), *DENTAL (abbreviated *D), *TR, IDENT(dist) (abbreviated ID(dist)), and DE-BRACKET(*TR) (abbreviated DB(*TR)). Example (23) shows how the derived-environment blocking applies, and (24) shows that change is allowed to happen in non-derived environments.

(22) a. *DENTAL — Assign a violation mark (*) to every dental segment in the output.
b. *TR — Assign a violation mark (*) to every alveolar segment followed by a rhotic.
c. IDENT(dist) — The specification of a feature [distributed] of an input segment must be preserved by its output correspondent.
d. DE-BRACKET(*TR) — Assign a violation mark (*) to a segment that violates *TR without also removing an adjacent morpheme bracket two which it is external.

Example (23) shows morphologically-derived environment blocking. In Steps 1 and 2, the contents of <wait> and the nominalizer morpheme are inserted serially, along with their morpheme brackets. The brackets around <wait> are removed in Step 2, and the brackets around <*nom*> are removed in Step 3. Also in Step 3, /t/ fails to dentalize before /r/. Although the winning candidate [wet+ər] violates *TR, it does not violate DE-BRACKET(*TR) because the violation of *TR occurs at a morpheme juncture. In this derived environment, dentalization is blocked.

(23) *Dentalization is blocked at a morpheme juncture*

Step 1: insert <wait> phonology

<wait> + <*nom*>	MR	DB(*TR)	*D	*TR	IDENT(dist)
a. <wait> + <*nom*>	**!				
☞ b. {wet} + <*nom*>	*				

Step 2: insert <*nom*> phonology, remove brackets on /wet/

{wet} + <*nom*>	MR	DB(*TR)	*D	*TR	IDENT(dist)
a. wet + <*nom*>	*!				
☞ b. wet + {ər}				*	

Step 3: remove brackets on /ər/, do not dentalize

{wet} + <*nom*>	MR	DB(*TR)	*D	*TR	IDENT(dist)
a. wet̪ + ər			*!		*
☞ b. wet + ər				*	

Final output: [wetər]

Example (24) has the natural opposite behavior of (23). The phonological content of <ladder> is inserted in Step 1, and its morpheme brackets are removed in Step 2. Because [d] preceding a rhotic is marked, the candidate [lædər] violates *TR; and because it violates *TR away from a morpheme boundary, it also violates high-ranking DE-BRACKET(*TR). The next best candidate satisfies these constraints by dentalizing, and so [læd̪ər] is the optimal candidate. DE-BRACKET(MARK) allows change to occur everywhere except derived environments, creating morphologically-derived environment blocking.

(24) *Dentalization occurs within a morpheme*

Step 1: insert <ladder> phonology

<ladder>	MR	DB(*TR)	*D	*TR	IDENT(dist)
a. <ladder>	*!				
☞ b. {lædər}		*		*	

Step 2: apply dentalization

{lædər}	MR	DB(*TR)	*D	*TR	IDENT(dist)
a. lædər		*!		*	
☞ b. læd̪ər			*		*

Final output: [læd̪ər]

4. Discussion

DE-BRACKET(X) unifies morphologically-derived environment effects and blocking by making use of the relationship between Markedness and Faithfulness constraints. There have, however, been other approaches to derived environment effects, many of which work in standard parallel Optimality Theory. Notable examples of other accounts include Local Conjunction (Smolensky 1993, Łubowicz 2002, 2005) and Comparative Markedness (McCarthy 2002). These approaches are generally successful at capturing derived environment effects, and can sometimes manage derived environment blocking as well (though it is tried less often).

Compared to these previous attempts, the approach outlined here has the advantage of using the intuitive relationship between Markedness and Faithfulness directly. In this way, it brings them together as a single phonological phenomenon whose manifestation depends solely on the class of constraint offered to the DE-BRACKET(X) constraint function.

In addition, this type of analysis does not risk encompassing phonologically-derived environment effects and phonologically-derived environment blocking. Though types of effects have been analyzed along with morphologically-derived environment effects in the past, more recent work has suggested that they should be considered separate phenomena (Hammond 1992, Kiparsky 1993, Łubowicz 2002, McCarthy 2002). The DE-BRACKET account does not risk creating phonologically-derived environment effects because it depends heavily on a morpheme boundary, which does not always seem to be true of phonologically-derived environment effects.

5. Conclusion

Morphologically-derived environment effects and derived environment blocking are two sides of the same phonological coin. They can be grouped together in a single account that takes advantage of their oppositional Markedness and Faithfulness constraint rankings. This type of approach seems to make good predictions about the overlap (or lack thereof) of morphologically-derived environment effects and phonologically-derived environment effects.

References

Ahn, S. C. 1985. The interplay of phonology and morphology in Korean. University of Illinois at Urbana-Champaign.

Baković, Eric. 2007. Local assimilation and constraint interaction. In: de Lacy, Paul (ed), The Cambridge Handbook of Phonology. Cambridge University Press.

Beckman, Jill N. 1998. Positional Faithfulness. Ph.D. dissertation, University of Massachusetts Amherst. [ROA-234].

Benua, Laura. 1997. Transderivational Identity: Phonological Relations Between Words. Ph.D. thesis, University of Massachusetts, Amherst.

Gussmann, Edmund. 2007. The phonology of Polish. Oxford University Press.

Hammond, M. 1992. Deriving the Strict Cycle Condition. CLS, 28, 126–140.

Inkelas, Sharon. 2014. The Interplay of Morphology and Phonology. Oxford University Press.

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.

Kim, Soo-Gon. 1976a. 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.

Kiparsky, Paul. 1982. Lexical morphology and phonology. In Linguistic Society of Korea (ed.) Linguistics in the morning calm, 3–91. Seoul: Hanshin.

Kiparsky, P. 1993. Blocking in nonderived environments. Studies in Lexical Phonology, 277–313.

Kurisu, Kazutaka. 2001. The phonology of morpheme realization. Ph.D. thesis, University of California Santa Cruz.

Łubowicz, A. 2002. Derived environment effects in Optimality Theory. Lingua, 112, 243–280.

Łubowicz, Anna. 2005. Locality of Conjunction. In J. Alderete, C. Han and A. Kochetov (eds.) Proceedings of the 24th West Coast Conference on Formal Linguistics. Somerville, MA: Cascadilla Press. 254-262.

Mascaró, Joan. 1976. Catalan phonology and the phonological cycle. Ph.D. dissertation, MIT. Distributed 1978, Indiana University Linguistics Club.

McCarthy, J. J. 2002. Comparative markedness (long version). Linguistics Department Faculty Publication Series.

McCarthy, J. J. 2006. Restraint of analysis. In E. Baković, J. J. McCarthy & J. Itô (eds.), Wondering at the Natural Fecundity of Things: Essays in Honor of Alan Prince, 195-219. [ROA-844.4].

McCarthy, J. J. 2007a. Hidden Generalizations: Phonological Opacity Optimality Theory. London: Equinox.

McCarthy, J. J. 2007b. Slouching toward optimality: Coda reduction in OT-CC. Linguistics Department Faculty Publication Series. 74.

McCarthy, J. J. 2008. The gradual path to cluster simplification. Phonology, 25, 271.

Mohanan, Karuvannur. 1982. Lexical Phonology. Ph.D. thesis, Massachusetts Institute of Technology.

Prince, Alan and Smolensky, Paul (1993) Optimality Theory: Constraint interaction in generative grammar. New Brunswick, NJ: Rutgers University Center for Cognitive Science.
Rubach, Jerzy. 1984. Cyclic and lexical phonology: the structure of Polish. Studies in Generative Grammar. Volume 17. Walter de Gruyter.
Samek-Lodovici, Vieri. 1993. A unified analysis of crosslinguistic morphological germination. Pages 265-283 of: Proceedings of CONSOLE, vol. 1.
Wolf, M. (2008). Optimal Interleaving: serial phonology-morphology interaction in a constraint-based model. University of Massachusetts Amherst.

Reed Blaylock
reed.blaylock@gmail.com