Precedence Constraints and Opacity

Yongsung Lee
(Pusan University of Foreign Studies)

Yongsung Lee (2006) Precedence Constraints and Opacity. Journal of Language Sciences 13-3, xx-xxx. Phonological change in candidate chain theory is motivated by two things, gradual divergence in faithfulness and improving harmony for markedness. These two principles work with candidate chains to rule out any meaningless or radical changes. And candidate chains with all their violation marks for faithfulness constraints are subject to evaluation that includes new type of constraints: Precedence Constraints. Precedence constraints specify the possible order in the violation of faithfulness constraints. Cumulative violations of faithfulness may or may not result in the violation of precedence constraints. These precedence constraints can deal with both types of opacity found in phonology: underapplication opacity or overapplication opacity. This approach can explain the commonalities and differences of the two types of opacity and provide a uniform analysis of opacity in general. Thus the proposal on precedence constraints can dispense with any theory internal problems in such proposals as Comparative Markedness, Local Constraint Conjunction, Sympathy Theory or Base-identity Proposal. (Pusan University of Foreign Studies)

Key Words: candidate chains, precedence constraints, opacity, Local Conjunction, Sympathy Theory, counterfeeding, counterbleeding.

1. Introduction

McCarthy (2006) has proposed a major revision in OT. It gives up the persistently-held notion of Freedom of Exponence and focuses on the revision of Generator Component (=Gen). In essence, it argues that Gen’s capacity for generation is severely restricted in that it does not freely do anything to the input, but follows step-by-step derivational procedure to produce wellformed candidate chains. Candidate chains carry the violation
of faithfulness constraints in each step of the derivation and they keep record of the chronological order of faithfulness violation.

Truly, it incorporates the derivational model of phonological explanation into the classical OT system. And in a sense, it is a combined model of derivation and constraint evaluation. The basic motivation for Optimality Theory with Candidate Chains (=OT-CC) comes from the observation that there are crucial evidences in the phonology, revolving around on the opacity effect that we need to refer to the intermediate level for the sake of explanatory adequacy.

Opacity arises when the traditional markedness and faithfulness constraint interactions fail to produce the surface form. In the earlier version of Generative Phonology, opacity on the surface comes from the opaque interaction of phonological rules. Opaque interaction here means either counterfeeding or counterbleeding order of rule application. Such opacity effect cannot be explained by looking at the surface wellformedness (markedness constraints) or at the correspondence between input and the surface (faithfulness constraints).

This problem has long been observed and discussed in the literature. Accordingly, many proposals are made in the literature that include Multi-level Serial OT (Kiparsky 1997, Roca 1997), Base Identity (Kenstowicz 1994), Local Constraint Conjunction (Itô & Mester 2003, Moreton & Smolensky 2002), Sympathy Theory (McCarthy 1999, 2003), Comparative Markedness (McCarthy 2002a) and many more. Many of these approaches, however, are not satisfactory because they fail to provide comprehensive and uniform analysis for both types of opacity and still suffering from fundamental theoretical flaws.

Against this backdrop, this paper examines the OT-CC
proposal to see how it deals with opacity problems in OT, without resorting to problematic proposals introduced above. To this end, we will briefly talk about candidate chains and precedence relations in section 2. Then we see how the precedence constraints consistently deal with opacity problems in section 3. Two types of opacity, underapplication opacity and overapplication opacity, will be briefly discussed in connection with the precedence constraints to prove that precedence constraints can consistently explain both types of opacity. Section 4 wraps up the discussion to conclude the paper.

2. Candidate Chains and Precedence

In OT-CC, candidates are presented in the form of chains, which show the derivational stages along with the cumulative violations of faithfulness constraints. The assumption is well represented in the flowchart given in (1):

(1) Flowchart of OT-CC (McCarthy 2006: 17)

```
input ——> Gen

a. Compare f_n with f_{n-1}
   b. Full set of Candidate chains
       Eval

output
```

The striking difference of OT-CC in comparison to the

---

1) McCarthy’s original lecture note shows that (1a) is written as “Compare f_i with f_{i+1}.” This means that the faithfulness constraints are input-relative. However, the assumption was proven to be problematic in the course of the lecture and McCarthy later changed that both markedness and faithfulness constraints refer to the immediately preceding chain member (f_n).
classical OT is the loop that connects Eval to Gen. Gen works in a limited way. Gen takes the input and produces the most faithful candidate chain \(<f_1>\).\(^2\) \(f_1\), therefore, is identical to the input. Then \(<f_1>\) is fed back to Gen to produce next chain, \(<f_1, f_2>\), then the chain \(<f_1, f_2>\) is looped to Gen to produce \(<f_1, f_2, f_3>\) chain. Note that each step from \(f_n\) to \(f_{n+1}\) means one violation of a faithfulness constraint. Since faithfulness constraints are, by definition, limited in number, the chain cannot be added infinitely. This way we see that the candidates are finite and there is no radical change, since adding one more chain member means just one violation of a faithfulness constraint. Note also that the chain starts from an input, which is language particular. As a result, the chains are input-dependent and therefore, are language-particular.

The candidate chains must be bound by the following wellformedness conditions:

(2) Wellformed candidate chains (McCarthy 2006: 14)

a. All chains are faithfully initiated.

b. Chains are gradually divergent.

c. Chains are harmonically improving.

(2a) says that the first chain is identical to the input, no faithfulness violations at all, then the second chain \(<f_1, f_2>\) must be wellformed in the sense that there is one faithfulness

\(^2\) \(<f_1>\) is identical to the input. Here "f" is the element in the chain, supposedly means a "faithful chain element." With the assumption that prosodic structures are not underlyingly present, \(f_1\) does not have any prosodic structure at all. Prosodic structures are posited by the requirement of structural wellformedness constraints that belongs to markedness constraint family, which are ranked over related faithfulness constraints, such as NoStruc as in Prince and Smolensky (1993: 25).
violation, and that the change should result in the improvement of harmony. Consider the following graphic representation:

(3) Principle of candidate chain formation

(2c) Harmonic improvement
   (obeying markedness)

\[ f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow \ldots \rightarrow f_n \]

(2b) Gradual divergence
   (violating faithfulness)

The chain formation as such is wellformed only when there is harmonic improvement at the cost of faithfulness violation. Therefore, the chain stops when there is no room for markedness improvement or when there is no more faithfulness constraint to violate. To make more clear illustration, let’s take an example. Take a look at the following hypothetical chains for a language, similar to Axininca Campa, where a prosodic word should be at least two syllables due to the high ranking nature of the markedness constraint, FootBinarity (=FtBin):

(4) Hypothetical candidate chains
   a. *<pa, pata>
   b. *<pa, pat, pata>
   c. <pa, paa, pata>

(4) illustrates the change from /pa/ to [pata]. (4a) shows that the direct pa→pata change is not allowed. Truly the chain shows harmonic improvement in that it conforms to FtBin. But, in the process, two segments are inserted at the same time. This is two violations of Dep. But the gradual divergence clause in (2b) says that there should be just one violation at a time.3)
The change is too radical and thus it is not gradual. In (4b), we see \( \text{pa} \rightarrow \text{pat} \rightarrow \text{pata} \) chain. The problem here is the change from \( \text{pa} \) to \( \text{pat} \). It is gradually diverging, but the result does not show any improvement in the markedness. With \textit{NoCoda} type of constraint, visibly working, we see that in the change \( \text{pa} \rightarrow \text{pat} \), we see the decrease of harmony, not conforming to harmonic improvement clause given in (2c). On the other hand, the chain given in (4c) is wellformed. First, a vowel is added, violating \textit{Dep} to obey \textit{FtBin}. Then a consonant is inserted, violating \textit{Dep} again to conform to syllable wellformedness constraint, \textit{Onset}. There we see gradual divergence from the input and also continuous improvement of harmony.

This approach can actually record the derivational history of an input and an output is not just a form but a history of derivational stages, or specification of chronological recording of violated faithfulness constraints.

For clear understanding of opacity problem, we will start with the classification of the different constraint families. McCarthy (2006) argues that there are only two types of constraints in OT, markedness constraints and faithfulness constraints. Alignment constraints are divided into either markedness or faithfulness group depending on the nature of the constraints. The difference between the two groups of constraints is shown

---

3) Multi-segment deletion in general are prohibited. But it may be allowed in some prosodic morphology as in reduplication and truncation. Here, we see that multi-segment deletion is to conform to the template constraints. The gradual divergence, however, may be problematic in actual implementation. Suppose there is a \(-i\) change in a language (cf. (9a).) This change involves the violation of at least two constraints at the same time, \textit{Ident}(low) and \textit{Ident}(back). One may argue that this is a radical change that should not be allowed by (2b). Further research should elaborate the meaning of gradual divergence or should rephrase the \textit{Prec} constraints in terms of markedness constraints.
in (5):

(5) Two families of constraints
   a. Faithfulness constraint : Any constraint that prohibits changes in the input.
   b. Markedness constraint : Any constraint that causes the input to change.

The classification of the constraint families given in (5) is enough to dichotomize all the classical OT constraints. Take \textit{Onset} for example. It requires that an onset segment be inserted, if a syllable does not have one. Sometimes we have to insert a consonant to satisfy \textit{Onset}. Therefore this constraint is change-inducing and is classified as a markedness constraint. On the other hand, the constraint \textit{Max} does not allow deleting any input segment. This constraint, therefore, is input-preserving and accordingly classified as a faithfulness constraint. If an alignment constraint causes the input to change, then that one is a markedness constraint. Meanwhile, any input preserving alignment constraint is classified into faithfulness constraint family.

Now, a word on opacity is in order here. Opacity arises when the interaction of two families of constraints, alone, fails to produce the right result. There are times when we have to consider the derivational history (intermediate stages of derivation) for consistent analysis of phonological alternations. In these cases, we need the third type of constraints that refer to the derivational history or chronological order of violation. McCarthy (2006: 25) proposes the following constraints that interact with other constraints that might produce the opacity effect in phonological derivation:
(6) Precedence constraint \(= \text{Prec}(A, B)\)

Let \(A'\) and \(B'\) stand for forms that add violations of the faithfulness constraint \(A\) and \(B\), respectively.

a. To any chain of the form \(<X, B', Y>\), if \(X\) does not contain \(A'\), assign a violation mark, and
b. to any chain of the form \(<X, B', Y>\), if \(Y\) contains \(A'\), assign a violation mark.

(7) Illustration of \(\text{Prec}\)

<table>
<thead>
<tr>
<th></th>
<th>(\text{Prec}(A, B))</th>
<th>(A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>(&lt;... A'&gt;) (&lt;... A&gt;)</td>
<td>√</td>
<td>*</td>
</tr>
<tr>
<td>b</td>
<td>(&lt;... B'&gt;) (&lt;... B&gt;)</td>
<td>*</td>
<td>√</td>
</tr>
<tr>
<td>c</td>
<td>(&lt;... A', B'&gt;) (&lt;... A, B&gt;)</td>
<td>√</td>
<td>*</td>
</tr>
<tr>
<td>d</td>
<td>(&lt;... B', A'&gt;) (&lt;... B, A&gt;)</td>
<td>**</td>
<td>*</td>
</tr>
</tbody>
</table>

(The italicized \(A\) and \(B\) are faithfulness violations and \(A'\), \(B'\) are forms that include \(A\) or \(B\) violation.)

The definition in (6) is represented in a simplified version of tableau in (7). Suppose for the sake of simple exposition that the "..." part contains no violation of \(A\) or \(B\) in (7). Violating \(A\) alone as shown in (7a) is not the violation of \(\text{Prec}(A, B)\) by definition given in (6). But if there is \(B\) violation without preceding \(A\) violation as shown in (7b), it is a violation of \(\text{Prec}(A, B)\), according to the definition given in (6a). (7c), though it violates both of the constraints \(A\) and \(B\), still we see that \(B\) violation is preceded by \(A\) violation and therefore there is no \(\text{Prec}(A, B)\) violation involved. In (7d), we see the worst case; \(B\) is violated without preceding \(A\) violation. (6a) will assign a violation mark for this representation. Further, the
violation of $B$ is followed by $A$ violation. Therefore the clause in (6b) will assign another violation mark.

Without precedence constraints, (7a) or (7b) will be chosen as optimal over (7c) or (7d). But $Prec(A, B)$ does an interesting work. (7c) may show up as optimal over (7a) or (7b), if $Prec$ is placed higher than $A$ and $B$, and if there is another high ranking constraint that (7a) violates. What is interesting to observe is that (7d) is rendered suboptimal due to the presence of harmonically bounding candidate chain (7c), and even (7b) may turn out to be a looser due to the presence of (7c).

This is the basic tenet of McCarthy’s proposal on candidate chains and precedence constraints. In the next section, we will see how opacity can be dealt with within the framework of OT-CC.

### 3. Opacity and Precedence Constraints

As mentioned earlier, opacity occurs when markedness and faithfulness constraint interaction fails to produce the correct surface form. Traditionally, in early version of Generative Phonology, opacity is defined as different rule interactions as in (8):


A process $P$ of the form $A \rightarrow B/ C\_D$ is opaque, if there are surface structure with either of the following characteristics:

a. instances of $A$ in the environment $C\_D$.

b. instances of $B$ derived by $P$ that occur in the environment other than $C\_D$. 

Here we see two different types of opacity. First (8a) shows that CAD→CBD may not be realized on the surface. The rule appears to have not been applied at all even in the perfect environment for the change. McCarthy (1999) names this type underapplication opacity. Another type given in (8b) is that the derived form B is present even when the environment is not apparent on the surface. It seems that the rule is applied to the string not specified in the environment, which is a case of overapplication opacity.4)

We will now look into these two types of opacity and how precedence constraints deal with them.

### 3.1 Underapplication opacity

The typical example of underapplication opacity can be found in Bedouin Arabic vowel raising and vowel epenthesis. Consider the following sketchy data.

(9) Raising and Epenthesis in Bedouin Arabic

(McCarthy 2006: 4)

a. Raising (a→i/ __CV)
   /bagar/→bi.gar (cows)

b. Epenthesis (Ø→V/C__C#)
   /himl/→hi.mil (load)

c. opaque interaction
   /gabr/→ga.bur, *gi.bur (a grave)

Low vowels are raised before an open syllable, excluding final consonants as in (9a) and a vowel may be inserted between two

---

4) Traditionally, underapplication opacity is thought to be the result of counterfeeding rule ordering and overapplication, the result of counterbleeding rule ordering. (cf. Kiparsky, 1973)
consonants, presumably to break up consonant clusters. But if the input is /gabr/, the output is [gabur], not *[gibir]. The vowel in the first syllable is not raised, though there is perfect match in the environment. (10) shows the problem and wrong tableau in classical OT:

(10) OT analysis of Raising and Epenthesis

a. Constraints

M1) Raise : No [low] vowel in a non-final open syllable.
M2) *Complex-Coda (=*CC) : No syllable final clusters.
F1) Ident(low) : No raising.
F2) Dep : No epenthesis.

(M: Markedness constraint F: Faithfulness constraint)

b. Ranking

i) Raise (M1) \(\gg\) ID(low) (F1) (/bagar/ \(\rightarrow\) bigar, *bagar)
ii) *CC (M2) \(\gg\) Dep (F2) (/himl/ \(\rightarrow\) himil, *himl))

c. Wrong evaluation

<table>
<thead>
<tr>
<th>input</th>
<th>Raise</th>
<th>*CC</th>
<th>ID(low)</th>
<th>Dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>? i) *gibur</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>• ii) gabur</td>
<td>*!</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>• iii) *gabr</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

The constraints and ranking come directly from McCarthy (1999). Two markedness constraints, Raise and *CC, trigger phonological changes to the input, whereas two faithfulness

5) M here means a markedness constraint and F, a faithfulness constraint. The subscript means the interaction of constraints. Thus we see that M1 interacts with F1, having crucial ranking M1 \(\gg\) F1, to produce the right result.

6) The question mark (?) in the tableau represents a candidate wrongly chosen as optimal. The filled dot (•) denotes an actual surface form. We will use check marks (✓) for correctly chosen optimal forms.
constraints, *Dep and *ID(low), try to keep the input intact. The ranking is well motivated as in (10b), but the result is devastating. The intended output (10cii) is ruled out and an unwanted form *gibur in (10ci) shows up as optimal. In essence, what appears to be problematic in (10cii) is that the change motivated by surface forms is not realized. Earlier proposal, Local Constraint Conjunction, may deal with such opacity but applying it to underapplication opacity is quite problematic.7)

Consider the following cumulative violation in the derivational history:

(11) Cumulative violations and Precedence constraint
a. violation chains
i) gabr→*gabr: *CC
ii) gabr→gabur: *Raise, *Dep
iii) gabr→*gibur: *Dep, *ID(low)
b. Proposed Precedence constraints
Prec (ID(low), Dep)

The precedence constraint given in (11b) says that what is wrong with the unwanted form *gibur is that it has two violations of faithfulness *Dep and *ID(low) in that order. In line with precedence constraints, we may say that the violation history has something wrong, which leads us to propose the Prec(ID(low), Dep) constraint given in (11b). To rule out this form, we may posit the proposed precedence constraint over *Raise, so that Prec can crucially eliminate *gibur. Now consider the following evaluation tableau:8)

---

7) See McCarthy (2006) for detailed discussion on the proposed Local Conjunction of constraints in line with Smolensky (1995) and the theoretical problems that the proposal has.
(12) Evaluation with precedence constraint

<table>
<thead>
<tr>
<th>/gabr/</th>
<th>*CC</th>
<th>Prec</th>
<th>Raise</th>
<th>ID(low)</th>
<th>Dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. &lt;gabr&gt;</td>
<td>✓</td>
<td>*!</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>✓ b. &lt;gabr, gabur&gt; &lt;Dep&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓ c. &lt;gabr, gabur, gibur&gt; &lt;Dep, Ident(low)&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(Prec=Prec(ID(low), Dep))

Note also that *CC should dominate Prec(ID(low), Dep) so as to render <gabr> as suboptimal. With this provision, we see that the unwanted form *gibur as in (12c) is eliminated with proper ranking of Prec(ID(low), Dep) over Raise.

The ranking logic goes as follows. As shown in (10c), the wrongly chosen optimal form (10ci) has the violation of *F_i and *F_j;^9 and the actual output form (10cii) show the violation of *M_i and *F_j. And the phonological change shows the ranking, M_i ≫ F_j. In this situation, (10cii) cannot be optimal.

But, with the introduction of Prec, ranked higher than M_i, has the effect of blocking the change as shown in (12). Since the expected change is blocked by Prec(ID(low), Dep), we see underapplication opacity on the surface.

8) In (10c), the ranking between *CC and Raise was not decided and it was not relevant at all. But, as can be shown in (12), Prec must be dominated by *CC and still dominating Raise. Therefore, in line with the logic of ranking transitivity as in Prince & Smolensky (1993: 51), we see that *CC must dominate Raise.

9) The subscript represents the interaction of the related constraints. Thus M_i and F_i are interacting constraints, while M_i and F_j are not related at all in the evaluation. Here, interaction means that the change in the rank of the related constraints results in the change in output forms.
Crucial ranking in underapplication opacity

\( Prec; \gg M_i \gg F_i \)

The ranking, \( M_i \gg F_i \), is responsible for the change, but the ranking, \( Prec; \gg M_i \), makes it sure that, in some cases, certain changes are rendered illegitimate. This ranking, therefore, is responsible for the lack of expected change on the surface.

Finally, we will have to check whether the postulation of \( Prec \) constraint does not pose any derivational problems for transparent interaction. Consider the following tableaux:

(14) Precedence constraint and transparent outputs

<table>
<thead>
<tr>
<th>/bagar/, /himl/</th>
<th>*CC</th>
<th>Prec</th>
<th>Raise</th>
<th>ID(low)</th>
<th>Dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. &lt;bagar&gt; &gt;</td>
<td>✓</td>
<td>!</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>b. &lt;bagar, bigar&gt; &lt;ID(low)&gt;</td>
<td>✓</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>c. &lt;himl&gt; &lt;</td>
<td>✓</td>
<td>!</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>d. himil &lt;Dep&gt;</td>
<td>✓</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The effect of \( Prec(ID(low), Dep) \) is not witnessed in /bagar/ since the crucial constraint that is involved in \( Prec(ID(low), Dep) \) evaluation is \( Dep \). Both (14a) and (14b) vacuously satisfy \( Prec(ID(low), Dep) \). But, for /himl/, we see that there must be \( Dep \) violation to get the right surface form. But the violation of \( Dep \) without preceding \( ID(low) \) leads to the violation of \( Prec(ID(low), Dep) \). That is true, but observe that the suboptimal form in (14c) is already got rid of, for its fatal violation of \( *CC \).

We have seen that \( Prec(ID(low), Dep) \) lurks behind without visible influence in evaluation for transparent derivation. It
shows up only when two faithfulness violations are involved in evaluation. Certain faithfulness violation must be made on condition of previous violation of a specific faithfulness constraint. All in all, \( \text{Prec(ID(low), Dep)} \) can explain the underapplication opacity without sacrificing its explanatory adequacy for transparent interactions.

### 3.2 Overapplication opacity

In this subsection, we will look at another type of opacity. In overapplication opacity, what we actually get is the surface form in which changes are made, which are not motivated on the surface. The classic example of this type comes from Tiberian Hebrew.\(^{10}\) Consider the following rule interaction:

(15) Epenthesis and Deletion in Tiberian Hebrew

a. Epenthesis (Ø→V/C__C#)
   /melk/→me.lex (king)

b. ?-Deletion (?→Ø/__  )
   /qaraʔ/→qara (he called)

c. Opaque interaction
   /defʔ/→defe, *def (tender grass)

Just like in Bedouin Arabic, underlying consonant cluster in Tiberian Hebrew is broken up by inserting a vowel in between the consonants as in (15a). And a particular consonant /ʔ/ is not allowed in the coda position of a syllable, which is represented

\(^{10}\) McCarthy’s (2006) example for overapplication opacity was from Bedouin Arabic, the interaction between palatalization and syncope. But since Tiberian Hebrew is better known for overapplication opacity and for the original proposal for Sympathy Theory, we will consider Hebrew examples for further comparison between OT-CC and Sympathy Theory.
by $\theta$-deletion rule as in (15b). The underlying form, /defʔ/, shows up as [defe], the second vowel is inserted, but seen from the surface, there does not seem to be any motivation for this change. Consider the following constraints and their interaction on a tableau to see the cause of problem in classical OT:

(16) OT analysis of Tiberian Hebrew

a. Constraints

M1) *Coda–Cond: No $\theta$ in coda.
M2) *Complex–Coda (=*CC): No syllable final clusters.
F1) Max–C: No consonant deletion.
F2) Dep–V: No vowel epenthesis.

b. Ranking

i) Coda–Cond (M1) $\gg$ Max–C (F1) (qaraʔ → qara, *qara?)
ii) *CC (M2) $\gg$ Dep–V (F2) (melk → melex, *melk)

c. Unfortunate evaluation

<table>
<thead>
<tr>
<th>/defʔ/</th>
<th>Coda–Cond</th>
<th>*CC</th>
<th>Max–C</th>
<th>Dep–V</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) *defʔ</td>
<td>*!</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ii) *defeʔ</td>
<td>*!</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

The constraints are from McCarthy (1999) and their ranking is properly motivated as in (16b). The evaluation in (16c), however, fails to single out the surface form as optimal. The actual output (16ciii) is harmonically bounded by the pseudo-optimal form in (16civ) in the sense of McCarthy (2002b: 23). The surface form has all the violations that the pseudo-optimal form has and more. In this situation, manipulating with ranking of these constraints does not help at all. The notion of harmonic bounding, however, can be broken with the introduction of a
precedence constraint. Consider the following observation:

(17) Cumulative violations and precedence constraint
   a. violation chains
      i) \(<\text{def}?>: ^*\text{CC}\)
      ii) \(?<\text{def}?, \text{def}>: ^*\text{Max-C}\)
      iii) \(<\text{def}?, \text{def}?, \text{def}e>: ^*\text{Dep-V}, ^*\text{Max-C}\)
   b. precedence constraint
      \(\text{Prec (Dep-V, Max-C)}\)

Observing (17a), we came to know that (17aiii), a harmonically bounded form has the violation history of \(<\text{Dep-V}, \text{Max-C}>\), while (17aii) shows only one violation of \(\text{Max-C}\). A direct translation of the observation into precedence constraint is given in (17b). We legitimize the history of violations in (17aiii) while penalizing \(\text{Max-C}\) violation without preceding \(\text{Dep-V}\) violation in (17aii). Now suppose \(\text{Prec}\) is placed higher than \(\text{Dep-V}\), a constraint violated by the surface form but not by the pseudo-optimal form. Then \(\text{Prec}\) violation cost more than \(\text{Dep}\) violation. Consider the following evaluation to see the result:

(18) OT–CC evaluation of Tiberian Hebrew

<table>
<thead>
<tr>
<th>/defʔ/</th>
<th>Coda–Cond ^*CC</th>
<th>Max–C</th>
<th>Prec</th>
<th>Dep–V</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. (&lt;\text{def}?&gt;)</td>
<td>^*</td>
<td>^*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. (&lt;\text{def}?, \text{def}e?&gt;)</td>
<td>^*</td>
<td>^*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;\text{Dep}&gt;)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ c. (&lt;\text{def}?, \text{def}e?, \text{def}e&gt;)</td>
<td>^*</td>
<td>^*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;\text{Dep}, \text{Max}&gt;)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. (&lt;\text{def}?, \text{def}&gt;)</td>
<td>^*</td>
<td>^*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;\text{Max}&gt;)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\((\text{Prec}=\text{Prec(Dep-V, Max-C)})\)
It is clearly shown that the harmonic bounding is broken with the introduction of $Prec(\text{Dep-V}, \text{Max-C})$ and placing it over $\text{Dep-V}$. The logic given by the precedence constraint is that if a candidate chain violates $\text{Max}$, it may as well violate $\text{Dep}$ to get a better chance to be chosen as an optimal form. The crucial ranking of $Prec$ relative to $M$ and $F$ is given in (19):

$$M_i \gg Prec_i \gg F_i$$

The ranking, $M_i \gg F_i$, is responsible for phonological change. But, in some cases, more violation of faithfulness is allowed to meet the requirement of $Prec$, given the ranking, $Prec_i \gg F_i$. Here we see that $Prec_i$ works to change the input. Compare this with underapplication opacity ranking given in (13). In underapplication cases, $Prec_i$ dominates $M_i$ so as to prevent markedness improvement. But here in overapplication cases, $Prec_i$ is dominated by $M_i$, inducing phonological alternations. And, at the same time, it dominates $F_i$ so that more violation of faithfulness constraints may be allowed, if and only if, such violation works to satisfy $Prec_i$.

Again, we will have to check whether the proposal in this paper causes problems with transparent interactions found in Tiberian Hebrew. Consider the tableaux given in (20):

---

11) One might argue that another candidate chain $<\text{deʃʔ}, \text{deʃʔe}>$ might be the winner in the tableau. This seems to show that $Prec(A, B)$ should be placed higher than both $A$ and $B$. This is an interesting research question open to further research in this paper. But, in any case, it should not be placed higher than any markedness constraint that dominates both $A$ and $B$. See the discussion that follows. This may be an evidence that we have to think about $Prec$ in connection with markedness constraints.
(20) Transparent outputs in Tiberian Hebrew

<table>
<thead>
<tr>
<th>/melk/, /qaraʔ/</th>
<th>Coda-Con</th>
<th>+CC</th>
<th>Max-C</th>
<th>Prec</th>
<th>Dep-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. &lt;melk&gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>✓ b. &lt;melk, melex&gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>&lt;Dep&gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>c. &lt;qaraʔ&gt;</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓ d. &lt;qaraʔ, qara&gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>&lt;Max&gt;</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Theoretically, we can easily see that transparent outputs, the forms that do not show any interaction between two faithfulness constraints via the constraint, \(\text{Prec}(<\text{Dep-V, Max-C})\), can freely pass through \(\text{Prec}(<\text{Dep-V, Max-C})\), vacuously satisfying the constraint. Words like /qaraʔ/ that involves the violation of \(\text{Max}\) are not problematic at all. As shown in (20c) and (20d), the actual output shows the violation of \(\text{Prec}(<\text{Dep-V, Max-C})\), since there is \(\text{Max-C}\) violation without preceding \(\text{Dep-V}\) violation. It, however, shows up as optimal, because violating \(\text{Prec}\) costs less than violating even higher constraint \(\text{Coda-Con}\).

### 3.3 Brief discussion on previous proposals

Before concluding this paper, we may compare the theory with previous proposals to see the superiority of OT-CC over previous proposals. Local Constraint Conjunction was introduced to explain the underapplication opacity. Constraint conjunction, as a way to make complex constraints by combining simple constraints (Smolensky 1995) or preventing chainshift (Moreton and Smolensky 2002) is a very effective way to explain the interaction of constraints. But employing it to deal with opacity has several undesirable effects.
First, it cannot deal with overapplication opacity. The locally conjoined constraint \([A, B]_D\) disallows combined violation of A and B in a domain D. But overapplication opacity results from the tolerance of combined violation in a certain domain. Second the domain itself is a problem. Consider the following data from Bedouin Arabic:

(21) Vowel raising and vowel epenthesis

\[ /t^\prime \arad \, \ranam-\, \mathrm{ih}/ \rightarrow [t^\prime \mathrm{a}, \mathrm{ra}, \mathrm{di}, \mathrm{n}, \mathrm{mhi}] \]

("he pursued his sheep")

We see that a vowel is inserted in the third syllable \(\mathrm{dir}\) and in the next syllable we find vowel raising. These two constraints don’t interact with each other. But still it shows a combined violation of \(ID(\, \mathrm{low})\) and \(Dep\). This would block the surface form to emerge as an optimal output.

Sympathy Theory can deal with both underapplication and overapplication opacities. The theory, as a matter of fact, is too strong and there is nothing it cannot explain. One of the problems as noted by Kiparsky (1997) is the multiple sympathy cases found commonly in natural languages. Allowing multiple sympathy, however, leads to undesirable effects. McCarthy (2006:12) shows that even an unattested change, \(/p\mathrm{am}/ \rightarrow [\mathrm{pa} \, \mathrm{a}]\) can be allowed, which is not possible under \(OT-CC\). Consider the following chains:

(22) Candidate chains to derive \(/p\mathrm{am}/ \rightarrow [\mathrm{pa} \, \mathrm{a}]\)

a. \(\ast <\mathrm{pam}, \mathrm{pa} \, \mathrm{a}>\): radical diversion
b. \(\ast <\mathrm{pam}, \mathrm{pa}, \mathrm{paa}>\): no harmonic improvement in \(\mathrm{pa} \rightarrow \mathrm{paa}\).  
c. \(\ast <\mathrm{pam}, \mathrm{paa}, \mathrm{paa}>\): no harmonic improvement in \(\mathrm{pam} \rightarrow \mathrm{paa}\).
(22) illustrates that \textit{pam}→\textit{paə} change cannot be allowed because there is no legitimate candidate chain to derive the surface [\textit{paə}] from /\textit{pam}/, though the surface form can show up using multiple sympathy constraints as in McCarthy (2006: 12)

Admitted that the discussion here is sketchy given the limit of space in a single paper and the lack of any discussion on this newly emerging theory. But the proposal seems to open a new possibility to tackle the old problems on opacity in a more comprehensive and consistent manner.

4. Conclusion

We have seen that the introduction of \textit{Prec} in the frameworks of OT–CC can consistently deal with opacity phenomena. A more complete analysis may require more detailed theoretical and practical comparison between OT–CC and previous approaches that include Sympathy Theory, Local Constraint Conjunction and others. Still, however, we can see that OT–CC can give a comprehensive and very simple explanation for both types of opacity: underapplication opacity and overapplication opacity.

Underapplication, as exemplified with Bedouin Arabic, is caused by \textit{Prec} that dominates the change–inducing markedness constraint. There are cases where violating markedness constraint is the only way to avoid the violation of \textit{Prec}. OT–CC argues that such ranking results in non–surface true forms to be optimal.

Overapplication, on the other hand, is the result of placing \textit{Prec} between related markedness and faithfulness constraints. Markedness constraints favor the one with proper surface
phonological alternation. When Prec dominates related faithfulness constraints, what we get is a form with more violation of faithfulness constraint to satisfy Prec. This leads to apparently excessive change which is non-surface apparent.

As such, candidate chain implementation in OT is a great improvement in the theory. The candidate chain with its history of faithfulness violation can be employed to explain the opacity by subjecting candidate chains to the evaluation by Prec family of constraints.

References

Bakovič, E. 2006. A note on overapplication opacity. Ms., University of San Diego. [also in ROA-856]


McCarthy, John J. 1999. Sympathy and phonological opacity. Phonology 16, 331-399. [also in ROA 252]


McCarthy, John J. 2006. Gen, Eval, and phonological opacity. Lecture Notes for Indiana PhonologyFest 2006. Indiana University,
Bloomington, IN.


Smolensky, P. 1995. On the structure of the constraint component Con of UG. Handout of the talk presented at UCLA, Los Angeles. [also in ROA–86]

Yongsung Lee
Department of English
Pusan University of Foreign Studies.
55-1 Uam Dong, Nam Gu, Pusan, Korea.
(608–738)
Tel: 051–640–3054
Email: yslee@pufs.ac.kr

Received:
Accepted: