To appear in *Kobe City University Journal* 65:5, 43-72, 2015.

Two Issues on Constraint Conjunction

Hideki Zamma and Seiichiro Kikuchi

1. Introduction

This paper discusses two issues on the conjunction of constraints in Optimality Theory: (i) how the locality function can formally be exercised in Local Conjunction; and (ii) if Self Conjunction should really be treated as the same mechanism as Local Conjunction. It is worth discussing these two issues because, although these two theoretical devices are quite useful and widely employed in many theoretical analyses, formal discussion on the devices themselves has seldom been elicited in the literature.

As for the first issue, three theoretical approaches have been proposed: that of (i) Smolensky (1995); (ii) Ito and Mester (2003) and Smolensky (2006); and (iii) Łubowicz (2005). Following the introduction of constraint conjunction in Section 2, we discuss in Section 3 which of these three approaches can best deal with the problem Local Conjunction tries to solve, observing the issue from a set-theoretic point of view. In the same section we also address two further issues that can arise from the discussion.

The formal status of Self Conjunction is even less well-known in the OT literature. It becomes evident in Section 4 that the function which Self "Conjunction" is supposed to exercise cannot actually be fulfilled solely from the same mechanism as Local Conjunction, and that the function needs additional or different mechanisms. We discuss several possible ways by which this additional mechanism can be best carried out.

2. Conjunction of Constraints

It is often the case that a phonological phenomenon can only be observed in a specific situation in a language; e.g. a phenomenon P occurs only in a situation S. This can pose a possible problem for Optimality Theory (Prince and Smolensky 1993/2004), where Universal Grammar is considered to be comprised of a finite number of universal constraints. If a constraint is highly specific in nature, it is quite unlikely that it is universally innate to humans.

A possible solution to this problem is a mechanism called Local Conjunction (Smolensky 1993, 1995, 1997), by which two (or more) constraints are conjoined, such that the conjoined constraint acts as a single one. Below is a definition given by Ito and Mester (2003:23):

(1) Definition of Local Conjunction

Let C_1, C_2 be constraints and δ be a (phonological or morphological) domain (segment, syllable, foot, prosodic word,…; root, stem, morphological word, …). Local Conjunction is an operation "&" mapping the triplet (C_1, C_2, δ) into the locally conjoined constraint donated by $C_1\&_\delta C_2$ (equivalently, $[_\delta\ C_1\&C_2]$), the δ-local conjunction of C_1 and C_2.

With this mechanism, the seeming complexity of a constraint can thus be derived from the conjoined nature of two different constraints. A conjoined constraint $C_1\&_\delta C_2$ is violated *only when a candidate violates both of the single constraints C_1 and C_2 at the same time.*

(2)

	C_1	C_2	$C_1\&_\delta C_2$
Cand_1	*	*	*
Cand_2	*		
Cand_3		*	

An interesting feature of a conjoined constraint is that it is always more specific than the two individual constraints, and thus should outrank the latter (cf. Panini's Theorem). Therefore, even in a situation where violations of individual constraints are not fatal due to the presence of an outranking constraint C_3 (as in (3)), the addition of a conjoined constraint $C_1\&_\delta C_2$ over C_3 can make the otherwise optimal candidate illegal: satisfaction of $C_1\&_\delta C_2$ overrides the violation of C_3 in another candidate (as in (4)).

(3)

	C_3	C_1	C_2
Cand_1	*!		
☞ Cand_2		*	*

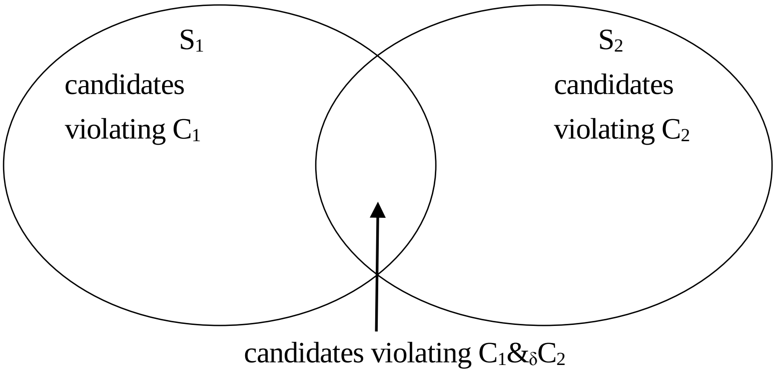
(4)

	$C_1\&_\delta C_2$	C_3	C_1	C_2
☞ Cand_1		*		
Cand_2	*!		*	*

In other words, the role of the conjoined constraint is to ban 'the worst of the worst': a violation of a constraint is fatal *only when* the candidate also incurs a violation of another constraint. It is notable that the mechanism of Local Conjunction allows the grammar to create a complex constraint from several simple, general ones which are present in the Universal Grammar.

It is easier to understand the scope of a conjoined constraint by means of a Venn diagram. Assuming that a finite number of possible candidate forms are available in an abstract space, those that C_1 aims to ban can be grouped into a set S_1 on the grounds that they violate C_1. Similarly, another set S_2 can be constituted on the grounds that the candidates violate C_2. Now, the candidates that would violate the conjoined constraint $C_1\&_\delta C_2$ belong to the intersection of S_1 and S_2 (i.e. $S_1 \cap S_2$).

(5)


S_1
candidates violating C_1
S_2
candidates violating C_2
candidates violating $C_1\&_\delta C_2$

In Boolean algebra, this is conjunction, but it should be noted that the set is defined in a negative way: the sets consist of elements that do *not* satisfy the relevant constraints. Crowhurst and Hewitt (1997) point out that it is in fact a disjunction if the set is defined in a positive way. However, we will set this problem aside here, and use the traditional terminology of Local Conjunction.

3. Local Conjunction

3.1 Locality of Conjoined Constraints

Another important property that Local Conjunction aims to have is the locality of violations: it is not that violations matter anywhere, only those in particular places. That is why both the definition of Local Conjunction in (1) and the conjunction operator "&" refer to the domain of the constraints. As we will see later in this section, however, formally exercising this function needs careful discussion.

Let us exemplify the problem by analyzing German Final Devoicing, as many studies of Local Conjunction do. The phenomenon has been well studied in the literature (see Rubach 1990 among others), where voiced obstruents (6a) become devoiced in syllable final position (6b).

(6)	a.	Tag-e	[g]	'days'	b.	Tag	[k]	'day'
		Bund-es	[d]	'union (gen.)'		Bund	[t]	'union (nom.)'
		Häus-er	[z]	'houses'		Haus	[s]	'house'

Note that underlyingly voiceless obstruents are not voiced in the onset position; compare *Welt* 'world' with *Welt-en* 'world (pl.)'. This is an example of Coda Condition (Ito 1986).

In Optimality Theoretic terms, this fact suggests the existence of one or more constraints that militate against voiced codas. Ito and Mester (1996, 2003) propose that this constraint is in fact a conjoined constraint involving NoCoda (7b) and *[voice] (7c). Note that the former is a well-established constraint observed in many languages and considered universal, while the latter is also a highly general constraint.

(7) a. NoCoda$\&_{\delta}$*[voice]: A coda consonant should be voiceless.
b. NoCoda: A syllable should not end with a consonant.
c. *[voice]: An obstruent should not be voiced.[1]

When these constraints are properly ranked against relevant Faithfulness constraints (i.e. Max and Ident), Final Devoicing can be properly analyzed as in the tableau below:.

[1] Ito and Mester (2003) name this constraint No-D. Accordingly they name its self-conjoined version No-D^2_m (cf. (24a)).

(8)

Bund	NoCoda&$_{\delta}$*[voice]	Max	Ident	NoCoda	*[voice]
bun[d]	W*		L	*	W**
☞ bun[t]			*	*	*
[p]un[t]			W**	*	L
bu		W**	L	L	*

Neither the devoicing of all obstruents to satisfy *[voice] (as in the third candidate) nor the deletion of all the coda segments to satisfy NoCoda (as in the final candidate) is more optimal than devoicing the coda obstruent (as in the second candidate), due to the higher-ranked NoCoda&$_{\delta}$*[voice]. Recall that this dominance of the conjoined constraint is guaranteed by the universal principle of Panini's Theorem.

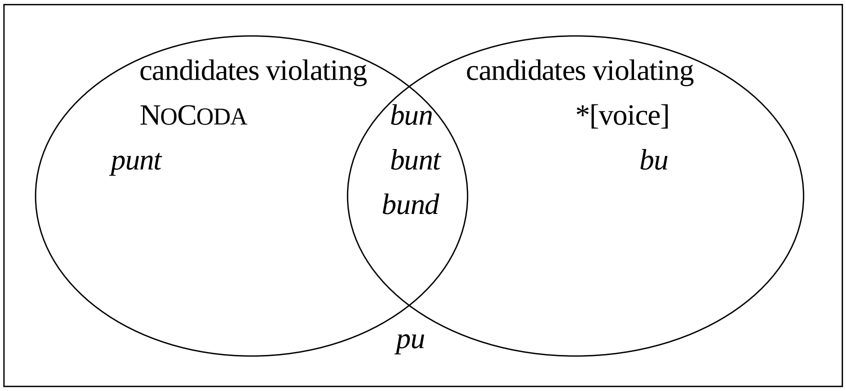
What is important in the calculation in (8) is that the optimal candidate does not violate NoCoda&$_{\delta}$*[voice], even though it does violate both of the individual constraints that comprise the conjoined one. This is where the locality function of Local Conjunction comes in. Although the optimal candidate violates *[voice] by virtue of having a voiced obstruent in the onset, the voiced obstruent is not regarded as violating NoCoda&$_{\delta}$*[voice] because that constraint is targeted at the coda segments.

A natural question that arises here is; how can this locality be formally obtained? Three proposals have been made in the literature, by (i) Smolensky (1995), (ii) Ito and Mester (2003) and Smolensky (2006), and (iii) Łubowicz (2005). In the next section, we will discuss which of these is the most adequate approach to perform the locality function.

3.2 How Can Locality Be Formally Obtained?

As mentioned in the previous section, it is necessary to guarantee the locality function for the analysis of Final Devoicing, whereby only a specific part of the candidate is evaluated by the conjoined constraints. Recall that the voicing of the onset segment is not relevant to NoCoda&$_{\delta}$*[voice], and is excluded from consideration. In contrast, the general constraint *[voice] militates against voiced obstruents in general wherever they are in the candidate. This problem can be expressed in the following diagram:

(9)

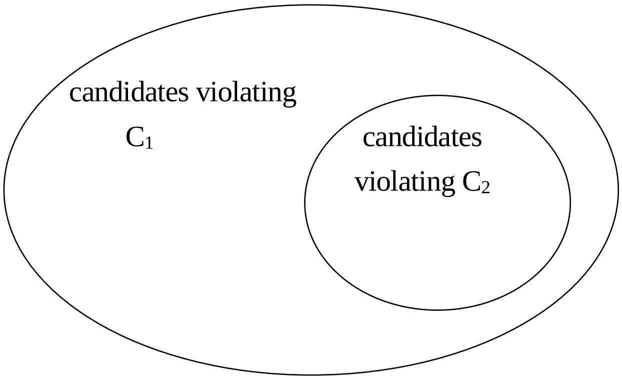


If Local Conjunction is just a simple conjunction (or a disjunction in Crowhurst and Hewitt's 1997 terms), candidates with a voiced obstruent in the onset but not the coda (e.g. *bunt*) would also be banned by NOCODA&$_δ$*[voice], as they appear in the intersection in the diagram in (9). It is somehow necessary to form a set of banned candidates that does not include those like this.

3.2.1 Smolensky (1995)

The first approach to the locality effect of Local Conjunction was proposed by Smolensky (1995). He specifies the domain to the conjunction operator by referring to the relevant constraint; e.g. *&$_1$* as in $C_1 \&_1 C_2$. He does not explicitly give the details of the mechanism, but the constraint can be understood as banning candidates which have a C_2 violation *among those which have a C_1 violation*. In other words, this approach tries to make a subset from the set of candidates violating C_1:

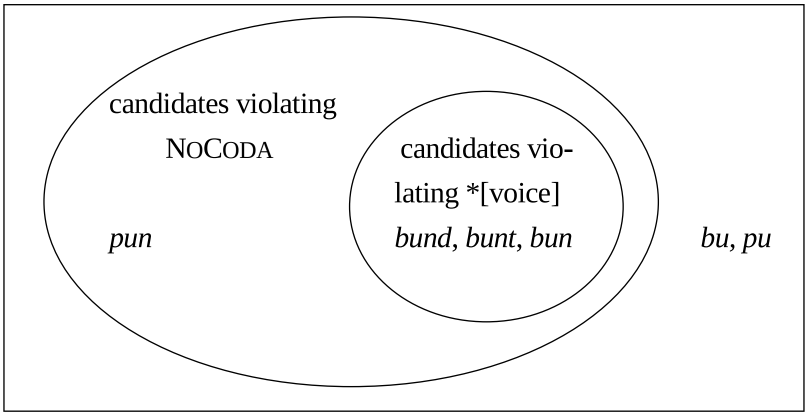
(10)



In Smolensky's (1995) approach, the locality function of Local Conjunction is thus obtained in set-theoretic terms as follows: in a domain where the first constraint is violated, violation of the second constraint is prohibited. This approach seems headed in the right direction, because what we are searching for is a way to define a subset of some kind among the candidates which violate the two constraints in question -- that is, for a way to include only *bund* (which violates *[voice] in the coda), but not *bun* and *bunt* (which violate *[voice] in the onset), into the intersection in (9).

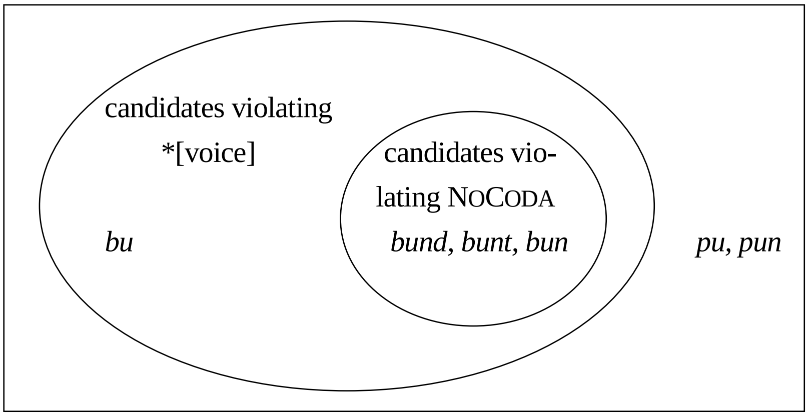
In order to capture this superset-subset relation among possible candidates, it is necessary to determine which group is the superset and which are the subset. In the case of Final Devoicing, then, we first need to define which is C_1 in the relevant conjoined constraint NOCODA&$_\delta$*[voice]. No matter which one is chosen as C_1, however, the required locality effect cannot be obtained in the case of Final Devoicing. Consider first the case where the superset constraint is NOCODA. This can make a subset of the *[voice] violation within the set of candidates that violate NOCODA, successfully excluding *bu*. Still it is impossible to properly exclude *bun* and *bunt*, which are actually legal candidates.

(11)



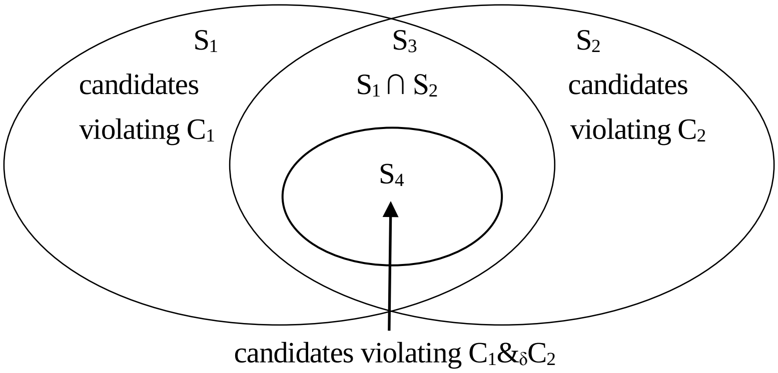
On the other hand, the same situation results when the superset is the *[voice] violating group. The subset successfully excludes *pun* from the fatal candidates, yet includes legal *bun* and *bunt*.

(12)



In sum, it is impossible to derive the locality function for the case of Final Devoicing we are considering just by making a subset from the violation set of a constraint. Making a subset, however, seems like the right direction, as discussed above. What we need is a way to make a subset *within an intersection of two violation sets*, as in (13).

(13)


S_1
S_3
S_2
candidates violating C_1
$S_1 \cap S_2$
candidates violating C_2
S_4
candidates violating $C_1\&_\delta C_2$

In set-theoretic terms, the situation depicted in (13) is thus a specific property of Local Conjunction.

3.2.2 Ito and Mester (2003), Smolensky (2006)

Ito and Mester (2003) and Smolensky (2006) propose the most popular way of performing the locality function, whereby a conjunction operator is assigned with a phonological or morphological domain where the conjoined constraint is operative. This assumption is in fact reflected in the definition of Local Conjunction in (2). In set-theoretic terms, the domain specification would delimit the range of the subset S_4 at the intersection S_3 in (13).

The problem for the present case of Final Devoicing then is what would be the appropriate domain. As NoCoda is a constraint on syllable structure (which is considered universal among human languages), the most natural assumption is that the domain for conjoined constraint would also be the syllable. If that were the case, however, the problem of excluding unnecessary candidates arises again: as the onset is also a syllable constituent, the voicing of an onset (as in the legal *bunt*) should also violate the conjoined constraint NoCoda&$_\sigma$*[voice].

Ito and Mester (2003) thus propose the following principle:

(14) Minimal Domain Principle

Let δ be a minimal domain shared by constraints A and B. Then their conjunction A&B has δ as a local domain.

Because NoCoda refers to both consonants and syllables, and *[voice] refers to obstruents in their respective definitions (see (7)), the minimal domain for NoCoda&$_\delta$*[voice] will be segments according to (14): i.e., NoCoda&$_{sg}$*[voice]. In this way, only the very segment that violates both constraints will become the target of the conjoined constraint.

Technically, this is a clever way to achieve the required delimiting effect for Final Devoicing. At the same time, it also reflects our intuition regarding the subset situation of prohibited candidates: a subset of possible coda consonants is prohibited. In sum, the principle (14) can be said to succeed in making the necessary delimitation for proper Local Conjunction depicted in diagram (13).

3.2.3 Łubowicz (2005)

Based on the McCarthy's (2003) notion of the locus of constraint violations, Łubowicz (2005) proposes that the domain of Local Conjunction be derived from the properties of constraints being conjoined, i.e. without specifying the domain to the conjunction operator. Her proposal is that, for Local Conjunction to be interpretable, the conjunct constraints must share a locus of violations. In other words, the domain of a conjoined constraint is the locus shared by the individual constraints. Based on McCarthy's (2003) theory, the markedness constraints specify their loci and define what constitutes a violation. In the case of Final Devoicing, for example, Loc Functions (Loc$_{\text{MARK}}$) for NoCoda and *[voice] are defined as in (15a) and (15b) respectively.

(15) a. Loc$_{\text{NoCoda}}$ ≡ Return every C, where C is final in some syllable.
(McCarthy 2003: 7)

b. Loc$_{\text{*[voice]}}$ ≡ Return every C, where C is [voice].

The Local Conjunction of NoCoda and *[voice] can now be explained as follows. Given that NoCoda and *[voice] share a locus of violation (i.e. a consonant), the conjoined constraint NoCoda&*[voice] is interpretable: it is violated when a consonant simultaneously violates both NoCoda and *[voice]. Therefore, *bund* violates NoCoda&*[voice] because NoCoda and *[voice] share a locus of violation: /d/. Contrastively, *bunt* does not violate the conjoined constraint, because NoCoda and *[voice] are violated in separate locations: /t/ for NoCoda, and /b/ for *[voice].

In sum, this approach can formally delimit the range of possible constraint conjunctions to cases where the domain of the individual constraints is shared, without recourse to arbitrary domain specification or other principle such as (14). In effect, the subset delimitation function (depicted in (13)) comes for free with the definition of the individual constraints. This represents thus the most restrictive approach to Local Conjunction at present.

Łubowicz's approach, however, might be too restrictive to account for various local interactions between constraints that have been discussed in the literature. Since the loci of violations are in most cases assumed to be individual segments (i.e. a consonant or a vowel; see McCarthy 2003:7), the domain of Local Conjunction is limited to a single segment. Phonological constituents larger than a segment (e.g. adjacent segments, syllable, or prosodic word) and morphological constituents (e.g. morpheme, stem) could not be the domain of a conjoined constraint. Disallowing the domain of "adjacent segments" in particular might cause a serious problem, because defining this domain based on a single segment is impossible: it is only possible as "two consecutive segments". The domain of "adjacent segments" has sometimes been proposed to constitute a domain of conjoined constraints in languages such as Polish (Łubowicz 2002) and Judeo-Spanish (Bradley 2007). In addition, Baertsch (2002) proposes to account for sonority restrictions in onset clusters, together with the Syllable Contact Law (Vennemann 1988) by locally-conjoined constraints whose domain includes "adjacent segments" (cf. Gouskova 2002 and Pons-Moll 2011).

Moreover, the theory is less restrictive in terms of the possible conjunction of constraints. If the loci of most constraints are either a consonant or a vowel, the range of possible conjunction becomes very huge.

3.2.4 Conclusion

While Smolensky's (1995) approach is not completely satisfactory in delimiting the range of impossible candidates for the problem at hand, both Ito and Mester's (2003) and Łubowicz's (2005) approaches properly include only the illegal candidates in the range of supposed impossible forms. Their approaches are similar in that they achieve the delimiting effect by means of the loci of the constraints. The difference between them is conceptual: the loci is determined by a principle independently imposed on conjoined constraints in Ito and Mester's (2003) approach, whereas it is automatically determined by the definition of constraints itself in Łubowicz's (2005) approach.

Łubowicz's (2005) approach can be said to be less costly in that it does not need anything special other than the constraints, as the loci are automatically and solely determined by Loc Functions. It is also more restrictive in that the loci are either a consonant or a vowel. However, various issues still remain unknown in her approach. Furthermore, future studies are needed to see if the two approaches lead to empirical differences. We remain neutral as to a definitive analysis, but conclude that loci of constraints is essential in Local Conjunction.

3.3 Further Issues

3.3.1 Onset Condition

In Section 3.2, we saw that Final Devoicing can be accounted for by means of Local Conjunction with a proper domain specification. It is logically possible then to assume that the same mechanism can apply to another syllable-structure-sensitive constraint: ONSET. Intuitively, we might imagine that an onset segment can also be restricted by the locally-conjoined constraint of ONSET and *[F] -- a featural condition on an onset segment -- in a mirror image of NOCODA&$_{\delta}$*[voice] for Final Devoicing, the latter being a kind of Coda Condition.

(16) a. ONSET A syllable should have an onset.
b. *[F] A feature [F] should not be present.
c. ONSET&$_{sg}$*[F] (A possible Onset Condition)

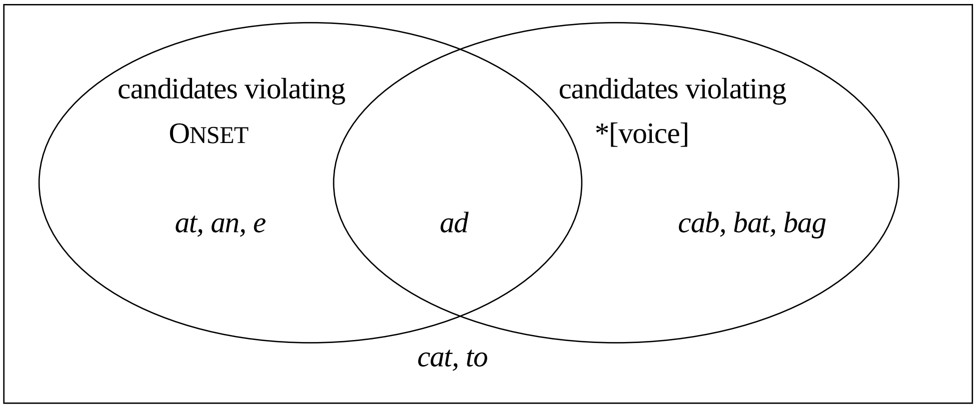
Simple logic tells us that this conjunction is impossible. Note that a violation of ONSET automatically implies satisfaction of *[F], since having no onset entails the absence of a segment to which *[F] is relevant. So, if we limit the range of featural restriction to onset segments (i.e. not on coda), there would never be any observed effect of the conjoined constraint of (16a) and (16b). Assuming that the relevant feature is [voice], we would expect an effect of Onset Devoicing from (16c). As shown in (17), however, not only candidates with an onset (17a, b), but also one that violates both of the individual constraints (17c) does not cause a violation of the conjoined constraint (the latter due to domain restriction).

(17)

	ONSET	*[voice]	ONSET&$_{sg}$*[voice]
a. dog		**	
b. cat			
c. ad	*	*	
d. at	*		

This fact can also be shown in the following diagram. Possible forms with an onset are all outside the intersection -- a situation equivalent to (17a, b). The only available forms in the intersection are those (i) without an onset and (ii) with a voiced coda (as in (17c)) -- but these do not share the same loci (and thus there is no subset in the intersection in (18)).

(18)



Note that the domain specification mechanism based on Loc Functions (Łubowicz 2005) shown in 3.2.3 also makes the same prediction: because the locus of violation for ONSET is a vowel -- as shown in (19) -- ONSET and *[voice] (15b) do not share the locus of violation and thus ONSET&*[voice] is not interpretable.[2]

(19) LOC$_{\text{ONSET}}$ ≡ Return every V, where V is initial in some syllable

(McCarthy 2003: 7)

From the theory of Local Conjunction it is thus predicted that there can be no condition on onset segments. Ito and Mester (2003) state that "in contrast to the rich set of coda conditions, few conditions are imposed on onset consonants (p.29)." In fact, many seeming conditions on onset segments are analyzable as those imposed on stem-initial segments; no word-initial /r/ in native Japanese vocabulary; no word-initial voiced consonants in Korean, etc.

Conjoining ONSET with a featural constraint itself is possible, however, if syllable-structure constraints are regarded as segmental, as in 3.2.2. On this approach, what constitutes an ONSET violation is a syllable-initial vowel.[3] Then, a constraint against a vowel feature can in principle be conjoined with ONSET. Consider the following hypothetical case where the relevant feature is [back], which can be both consonantal and vocalic.

(20)

	ONSET	*[back]	ONSET&$_{sg}$*[back]
yot		**	
yet		*	
ot	*	*	*
et	*		

In this hypothetical language, syllables would not start with a back vowel.

Still, a conjoined constraint like this is not what we expect for an Onset Condition -- a condition on syllable-initial consonants. Considered formally then, it is predicted that there will never be conditions on onset segments. Will this prediction be borne out?

[2] We assume that the locus of *[voice] is a consonant because vowels are underspecified for [+voice].

[3] Ito and Mester (2003) refer to Ito (1989), who states ONSET as "No syllable-initial moraic segments."

Conditions on onset segments are actually found in some languages. It has been cross-linguistically observed that less sonorous onsets are preferred. For example, in cluster simplification in child phonology, consonants with higher sonority are avoided in the resulting onset (Gnanadesikan 2004): e.g. *please* > [piz], *[liz]. In addition, a sonority restriction on an onset can be the trigger for a phonological process in some languages: in Argentinian Spanish, the palatal glide [j] becomes a less sonorous fricative [ʒ] in onset position, e.g. *mayo* 'may' [majo] → [maʒo] (Hualde 2005). There also exists a distributional asymmetry, by which the segmental inventory in the onset is more restrictive than that in the coda: in Chamicuro, the glottal consonants [h, ʔ] are contrastive only in the coda position, but are prohibited from appearing as an onset (Parker 2001).

Although these phenomena apparently show the need for locally-conjoined constraints ONSET&*[F], it is still possible to analyze them by other types of constraints. Sonority restrictions can be analyzed by *ONSET/X hierarchy (Smith 2002), and the Chamicuro case can be attributed to an onset-specific version of HAVEPLACE (Parker 2001). Thus, the existence of onset conditions is not necessarily inconsistent with the absence of ONSET&*[F]. However, if complex constraints such as onset-specific markedness (e.g. *HAVEPLACE/Onset) turn out to be derived by Local Conjunction as suggested by Smolensky (2006), it would also be possible that the relevant condition is a product of Local Conjunction of ONSET and other markedness constraints. This is obviously an issue that needs further consideration.

3.3.2 Conjunction of Declarative Constraints

When we look at the diagram in (18), we notice that the target candidate forms for the Onset Condition are all outside the range of those violating ONSET. This fact makes us think that the nature of the constraint might have a decisive role in allowing for the possibility of conjunction. That is, if the constraint(s) is/are declarative (i.e. not prohibitive) in nature, conjunction might not be possible, because a violation of a declarative constraint implies that the relevant segment (or structure) is absolutely absent. It is thus predicted that there would be no conjoined constraints one (or both) of which is declarative. Will this prediction be borne out?

One possible counterexample is WEIGHT-BY-POSITION (WXP; Hayes 1989). WXP is a declarative constraint which requires the coda consonant to be moraic. By conjoining WXP with featural markedness constraints, it is possible to derive a constraint that requires only coda consonants *with certain properties* to be moraic. For example, a conjoined constraint of WxP and *[son] (i.e. WxP&*[son]) would require only a sonorant, but not an obstruent, to be moraic if it is in coda position; e.g. *ban*$_\mu$ but *bat*:[4]

(21) a.

/ban/	WxP&*[son]	Dep-μ	WxP	*[son]
☞ ban$_\mu$		*		*
ban	*!		*	*

b.

/bat/	WxP&*[son]	Dep-μ	WxP	*[son]
bat$_\mu$		*!		
☞ bat			*	

Note that the moraicity of the coda obstruent is not only determined by WxP&*[son]: in (21b), the obstruent in the coda is nonmoraic, because WxP is dominated by Dep-μ, which prohibits non-underlying moras to appear in the output. If WxP dominated Dep-μ, coda consonants would always be moraic regardless of their sonority.

The question now is whether there are indeed languages in which the moraicity of coda consonants depends on their sonority or other featural properties. In languages such as Lithuanian and Kwakwala, syllables closed by sonorants pattern with heavy syllables, while those closed by obstruents behave as light ones (Zec 1988, 1995). Also, there are some languages where only a certain type of coda consonant contributes to syllable weight (Hayes 1995). In Cahuilla, closed syllables are heavy only when the coda consonant is a glottal stop; in Eastern Ojibwa, where coda consonants normally do not contribute to syllable weight, only syllables closed by nasals behave as heavy for a certain class of words. These facts indicate that WxP&*[F] is empirically necessary.

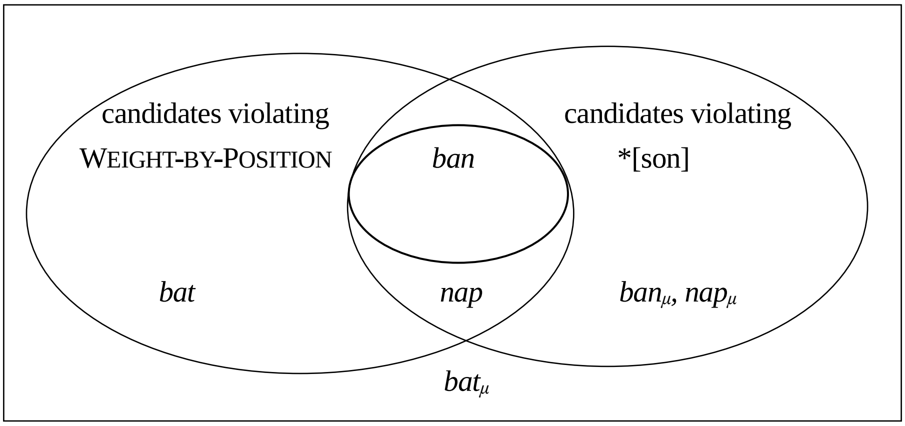
Given that WxP&*[F] is a possible construct that accounts for the sonority restriction on moraic codas, it has to be examined to see whether the relevant conjunction is possible according to the Minimal Domain Principle (14). For WxP to be properly conjoined with *[son], for example, WxP and *[son] must share their domain of evaluation (which corresponds to the loci of violations in Łubowicz's (2005)

[4] In (21) and below, moraic codas are represented by adding "μ" as a subscript. Underlying moras of vowels are not indicated because they are irrelevant to the evaluation.

analysis). The domain for *[son] is obviously the segment: *[son] assigns a violation mark to a consonant that is [son]. The domain for WXP, on the other hand, is less obvious because WXP is a prosodic well-formedness constraint like NOCODA. Still, it is quite natural to define WXP in terms of segments. WXP will then assign a violation mark to a consonant that is in the coda position and does not project a mora. The domain for both [son] and WXP is thus a consonant. Consequently, it can be concluded that WXP and *[F] share their domain of violation and that WXP&$_{sg}$*[F] is a valid conjoined constraint.

The subset situation of the candidates prohibited by the locally-conjoined constraint WXP&$_{sg}$*[son] can be properly diagramed in (22):

(22)



Only the candidates that share the loci of the violation constitute a subset in the intersection.

3.3.3 Conclusions

From the discussion of this section, it is clear that the conjoinability of constraints does not depend on the nature of the constraints themselves. The distinction between declarative and prohibitive constraints does not seem to matter in conjoinability -- rather it is the loci of the constraints that matters. If the loci of the constraints do not overlap, it is impossible to conjoin them. Still we must wait for future research to determine if it is really appropriate to exclude the possibility of Onset Conditions, which might be composed of constraints whose loci do not overlap.

4. Self Conjunction

4.1 OCP and Self Conjunction

The avoidance of the same/similar phonological structures within a domain has been widely observed in human languages, and has been studied in relation to the Obligatory Contour Principle (OCP; McCarthey 1986). In Optimality Theory, some researchers claim that OCP is actually the realization of a special case of Local Conjunction: Self Conjunction (Smolensky 1995, 1997, 2006; Alderete 1997; Ito and Mester 1996, 2003; etc.). In this section, we will investigate whether Self Conjunction can actually be implemented by means of Local Conjunction, or if it needs another mechanism other than that.

Many studies of Self Conjunction take the Rendaku phenomenon in Japanese as an example. Rendaku is a voicing process that affects the initial consonant of the second element in a compound (23a). The voicing, however, is blocked when the second element already has another voiced obstruent -- a clear case of OCP (23b):

(23) a. yaki 'grilled' + sakana 'fish' → yaki-**z**akana 'grilled fish'
b. yaki 'grilled' + soba 'noodle' → yaki-**s**oba 'stir-fried noodle'
*yaki-**z**oba

Alderete (1997) and Ito and Mester (1996, 2003) propose that OCP effects like this can be derived from Self Conjunction, a special case of Local Conjunction. The idea is that a self-conjoined constraint is violated when two instances of prohibited structure are present in a specified domain -- in parallel to usual Local Conjunction, where a conjoined constraint is violated when both of the independent constraints are violated. In the Japanese case at hand, multiple presence of a [voice] feature is prohibited in the second element of a compound (24a). Assuming that Japanese compounds have a voicing morpheme that triggers Rendaku (which is otherwise required to surface due to the constraint in (24b)), the blocking effect can be accounted for in the tableau (25):

(24) a. *[voice]$^{2}_{\text{Stem}}$: More than one voiced obstruent in a stem is prohibited.[5]
b. REALIZE-M: Every morpheme in the input has a nonnull phonological

[5] The original formulation by Alderete is: *[+voi, -son]. Here we take [-son] out just to save space. Note also that [-son] might be unnecessary in the formulation if we assume underspecification in the input: [+son] consonants are underspecified for [+voi].

exponent in the output (Kurisu 2001).

(25) a.

yaki + sakana	*[voice]$^{2}_{\text{Stem}}$	REALIZE-M	*[voice]
yaki**s**akana		*!	
☞ yaki**z**akana			*

b.

yaki + soba	*[voice]$^{2}_{\text{Stem}}$	REALIZE-M	*[voice]
☞ yaki**s**oba		*	*
yaki**z**oba	*!		**

Undergoing Rendaku incurs a violation of *[+voice]$^{2}_{\text{Stem}}$ only when the second morpheme has a voiced obstruent.

4.2 Problems of Self "Conjunction"

There is a potential problem, however, in applying the mechanism of Local Conjunction to Self Conjunction. The evaluation procedure of the self-conjoined constraint in (25) is actually different from the one for the locally-conjoined constraints in (2). The former actually counts the number of violations: only when the single constraint is violated twice is the conjoined constraint regarded as having been violated. If the same calculatory process as (2) applies to Self Conjunction -- so that violation of the conjoined constraint is calculated based on simultaneous violation of individual constraints -- even a non-Rendaku candidate would violate the conjoined constraint if the stem contains a voiced obstruent. This is because it would violate the individual *[voice] as shown in (26). The conjoined constraint would then act exactly in the same way as a single anti-voicing constraint. Consequently, devoicing would be predicted in the stem as shown in (27).

(26)

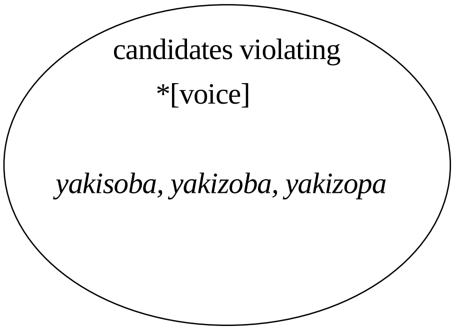
	*[voice]	*[voice]	*[voice]$^{2}_{\text{Stem}}$
yaki**z**oba	**	**	**
yaki**s**oba	*	*	*

(27)

yaki + soba	*[voice]$^{2}_{\mathrm{Stem}}$	REALIZE-M	*[voice]
yaki**z**oba	*!*		**
yaki**s**oba	*!	*	*
☞ yaki**s**opa		*	
yaki**z**opa	*!		*

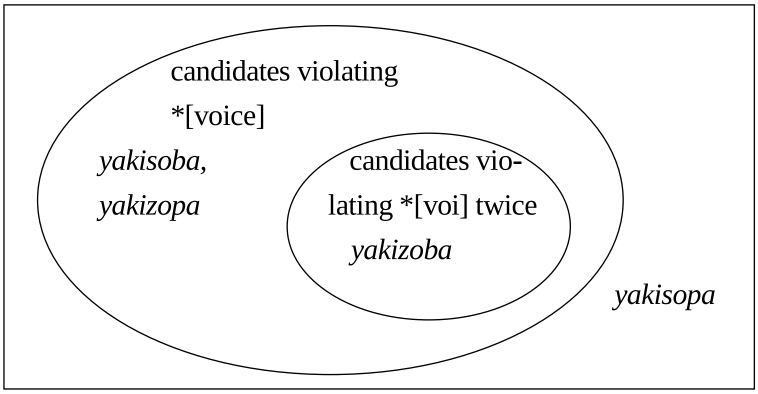
This problem can also be addressed in set-theoretic terms: as the idempotent law says, the intersection of the same set is the set itself (i.e. A ∩ A = A). This situation is depicted in diagram (28). As long as the conjoined sets are the same, there can never be a delimiting function among the candidates.

(28)



What is actually needed is a mechanism to delimit the subset of candidates with multiple violations of *[voice] from the superset of those with any number of *[voice] violations. See the diagram below:

(29)



We note in (29) that this situation is not the same as a typical Local Conjunction we observed in (13): it does have a proper subset, but the subset is not delimited by means of the locality function.

The impossibility of regarding OCP as a product of constraint conjunction is also obvious from the Minimal Domain Principle (14). In order for constraints to be conjoined, this principle requires them to have *the same locus*. In dissimilatory processes such as OCP, this will never be possible because *the (possible) locus of violations are necessarily in different places* (e.g. two consonants). In other words, the delimitation principles of Local Conjunction and Self "Conjunction" are completely contrary to each other.

The problem then is how we can achieve the delimiting effect for the OCP cases, such that it defines the subset in the diagram (29). Recall that Alderete (1997) and Ito and Mester (1996, 2003) assign the domain specification of stem (or morpheme) to the self-conjoined constraint. This is appropriate as the OCP effect can be observed within the second element of the compound. However, just by defining the locus as morpheme cannot achieve the delimiting effect in the present case, as the category does not constitute a subset of multiple *[voice] violations.

4.3 Possible Solutions

In this section, we discuss how the problem of the previous section can be solved. We will discuss the possibilities of: (i) analyzing the OCP case as a special case of general-specific conjunction (Section 4.3.1); (ii) assuming a different mechanism from constraint conjunction (Sections 4.3.2 and 4.3.3); and (iii) adding another mechanism on top of constraint conjunction (Section 4.3.4).

4.3.1 General-Specific Conjunction

It is technically possible to consider that the effect in question is achieved through the mechanism of conjunction. Let us assume here that the conjoined constraints are not exactly the same, but that one of them is a slightly specified version of the other, such as a positional markedness constraint (Smith 2004). In the case of Japanese Rendaku, conjoining general *[voice] with a specific version localized to the stem-initial position (*#[voice]) successfully targets the illegal candidate in question, as shown in (30). We assume Łubowicz's (2005) calculation method reviewed in Section 3.4 above.

(30)

	*#[voice]	*[voice]	*#[voice]&$_{\text{Stem}}$*[voice]
yakizoba	*z	~~*z~~, *b	*(z, b)
yakisoba		*b	
yakizopa	*z	*~~z~~	

This is only possible, however, with an additional mechanism of cancellation of violations, as indicated by the cross-out line in (30). The assumption is that in calculating a conjoined constraint, the violation of the general constraint is cancelled if it is incurred by the same segment that incurs a violation of the specific constraint. This mechanism is necessary to safely prevent the third candidate in (30), which has a voiced obstruent only in initial position, from undergoing a possible violation of the conjoined constraint. Note that the violation of a specific constraint automatically incurs the general constraint as well, irrespective of the specific details of the constraint.

It should be recalled that Smolensky's (2006) approach to Self Conjunction is similar to this one, in the sense that the conjoined constraints are not exactly the same. In his analysis of onset hierarchy, an alignment constraint ONS (= ALIGN-L(σ, C)) is used that requires consonantal features to appear at the beginning of a syllable. The consonant violates a self-conjoined version of ONS depending on the number of vocalic features it contains; for example, a liquid violates ONS^2 because it has [+approx] and [+son]. Technically speaking, this treatment can be viewed as dealing with different violations of different features; i.e. ALIGN-L(σ, [-approx]) and ALIGN-L(σ, [-son]).

This approach has the advantage of properly capturing the situation depicted in the diagram (29), where the target illegal candidates are in a proper subset of candidates violating the general constraints. At the same time, however, it has several disadvantages.

First, the supposed conjunction does not seem to satisfy the Minimal Domain Principle (14). According to this principle, the local domain should be the smallest one shared by the two constraints. In the present case -- as the constraints both refer to a consonantal feature -- the domain should be a segment, yet the supposed domain is the morpheme.

Second, it is still not obvious that this approach can be applied to other cases. The present case seems to work because the locus of the alternation happens to be at the morpheme edge. It might be difficult to analyze the OCP effect if it were not relevant to any domain boundary.

Third, OCP cases are structurally different from typical cases of Local Conjunction, as mentioned at the end of the previous section. While OCP cases always involve a situation where illegal structures lie in different locations, the Local Conjunction cases involve several different violations in one particular location.

4.3.2 Counting Operation

As observed in the diagram (29), what is needed is a way of delimiting the subset of candidates with two violations of *[voice]. Thus it might be reasonable to assume a new operation of counting the number of violations for OCP cases. This is proposed in (31) below:

(31) Counting Operation C^{n}_{δ}

Count the number of violations of a constraint C in a given domain δ. Only when *n* violations are found, C is regarded as violated.

With this operation, the relevant constraint for our problem is defined as $*[\text{voice}]^{2}_{\text{Stem}}$. In (32), only the first candidate violates this constraint, with two violations of *[voice].

(32)

	*[voice]	$*[\text{voice}]^{2}_{\text{Stem}}$
yakizoba	**	*
yakisoba	*	
yakizopa	*	

This solution might even be what many people assume in their analysis of Self Conjunction (Smolensky 1995, 1997, 2006; Alderete 1997; Ito and Mester 1996, 2003; etc.). Smolensky (2006) defines self-conjoined constraint as follows:

> [T]he self-conjunction $*A\&_{\mathcal{D}}*A$ is violated if (and only if) there are two distinct violations of *A in a single domain $\mathcal{D}$. (Smolensky 2006:43)

It is generally assumed, however, that constraints in Optimality Theory do not count the number of violations: the number of violations incurred by a candidate is relevant only in relation with other candidates (McCarthy 2002). The question then is how the counting function can be obtained.

One possible approach suggested by Smolensky (2006) is to pair violations of the relevant constraint. Whenever there are multiple violations in a designated domain, pair them and assign one violation mark for each pair. In mathematical terms, the number of violations of C^2_{Stem} corresponds to the number of sets with two elements within the power set of the violations of C. Let us show how the mechanism works in the case of *yakizoba* vs. *yakisoba*. *Yakizoba* incurs two *[voice] violations: {z, b}. From this set of violations, we establish a violation pair (z, b). $*[voice]^2_{Stem}$ assigns one violation mark to this pair; as a result, *yakizoba* incurs one violation of $*[voice]^2_{Stem}$. *Yakisoba*, contrastively, incurs only one *[voice] violation: {b}. In this case, no violation pair is detected, hence *yakisoba* does not violate $*[voice]^2_{Stem}$.

As Smolensky (2006) mentions, violations of Local Conjunction can also be determined by this pairing mechanism: $C_1\&_\delta C_2$ assigns one violation for every pair of C_1 and C_2 violation in a designated domain δ. However, Self "Conjunction" is peculiar in that it pairs violations among those of a single constraint. Local Conjunction pairs violations across those of different constraints.

4.3.3 Iterative Evaluation

Besides the Counting Operation, the required effect can also be achieved by a new operation of Iterative Evaluation for the case of OCP, so that a subset of candidates with a second violation could be created among candidates with any number of *[voice] violations. Let us assume that the operation is something like this:

(33) Iterative Evaluation ($C^n \times C^{n+1}$)

Let C^1 be the first evaluation of candidates as to constraint C, C^2 the second. C^2 evaluates the candidates as to whether they add a further violation to C on top of a C^1 violation.

In our case, $\{*[voice]^1 \times *[voice]^2\}_{Stem}$ would thus be violated by a candidate with further violations of *[voice] in addition to another violation of the same constraint.

(34)

	$*[\text{voice}]^1$	$*[\text{voice}]^2$	$\{*[\text{voice}]^1\times*[\text{voice}]^2\}_{\text{Stem}}$
yakizoba	*z/*b	*b/*z	*
yakisoba	*b		
yakizopa	*z		

The directionality of scanning does not matter here: whichever of [z] and [b] in the first candidate is found in the first scan, the other will be available in the second. It is still necessary to have a locality specification as part of the operation, as the OCP effect in Rendaku only applies inside the second element (which is regarded as the stem).

Whichever violation is found, multiple evaluation (scanning) of a candidate entails that scanning is incrementally carried out. Still, although it has not been argued explicitly in the literature, the evaluation of an OT constraint is assumed to be "global" in the sense that it scans a candidate as a whole and assigns violation marks to as many marked structures that are found in the candidate. That is, the scanning is considered to be non-incremental. We thus have to consider how the notion of "local" iterative evaluation can be adapted to the "global" evaluation of OT.

One possible solution is to assume that the violations of a candidate form an ordered set according to their linear order within a candidate string. $C^n \times C^{n+1}$ evaluates the candidate based on the ordering relation between violations:

(35) a. Suppose that the set of violations of a constraint C incurred by a candidate is an ordered set: violations incurred by a candidate are ordered based on the location where C is violated in the phonological string of the candidate:

The set of violations of C: $\{*_1, *_2, \ldots, *_n\}$, where $*_1 < *_2 < \ldots < *_n$ in the phonological string of a candidate. ('*a* < *b*' means that '*a* precedes *b* stringwise.')

b. For $*_n$ (a violation of C) in the ordered violation set, $C^n \times C^{n+1}$ marks one violation mark iff there is a $*_{n-1}$ in the same set.

In set-theoretic terms, the evaluation process of (35b) is as follows:

(36) Among the ordered pairs of the binary Cartesian power (C × C) of the set of violations of C, assign one violation mark to the pair where the order of elements

is consistent with that of the violation of C.

In the case of *yakizoba* and *yakisoba*, *yakizoba* has the ordered set of *[voice] violation {z_1, b_2}, while *yakisoba* has that of {b_1}. According to (35b), *yakizoba* violates *[voice]^1×*[voice]^2, because for {b_2}, there is a prior violation {z_1}. In *yakisoba*, on the other hand, there is no prior violation to {b_1} in the set. Thus, *yakisoba* does not violate *[voice]^1×*[voice]^2. According to (36), the violations of these candidates are computed as follows:

(37) a. *yakizoba*

i. The ordered set of *[voice] violations: {z_1, b_2} where $z_1 < b_2$

ii. The ordered pairs of *[voice] × *[voice]: (z_1, z_1), (z_1, b_2), (b_2, z_1), (b_2, b_2)

iii. Among the pairs in (ii), the only pair whose order of the elements is consistent with that of (i) is (z_1, b_2).

iv. Thus, *yakizoba* incurs one violation of {*[voice]^1×*[voice]^2}$_{\text{Stem}}$.

b. *yakisoba*

i. The ordered set of *[voice] violations: {b_1}

ii. The ordered pairs of *[voice] × *[voice]: (b_1, b_1)

iii. There is no pair in (ii) whose order of the elements is consistent with that of (i).

iv. Thus, *yakisoba* incurs no violation of {*[voice]^1×*[voice]^2}$_{\text{Stem}}$.

If we adopt the assumption of the locus of violation, in which the constraints specify the constituent in the candidate where violation takes place (Łubowicz 2005; see 3.2.3), it is possible to assume that the violations are ordered based on their loci within a candidate string, and furthermore, to assume the existence of a constraint that evaluates the candidate according to this ordering relation within the violation set.

4.3.4 Cancellation

Recall that the problem of Self “Conjunction” lies in the fact that the sets of violations for each conjunct are identical because they are identical constraints, as we observed in (28). A possible solution is thus to make these sets distinct by canceling violations of one of the conjunct constraints.

Although it is not discussed in detail, Smolensky (2006) mentions this possibility in his discussion of distinct violation:

> Suppose A ≡ $\mathbb{C}$ and B ≡ $\mathbb{C}$ refer to the same constraint. It follows that A$\&_{\mathcal{D}}$B = $\mathbb{C}\&_{\mathcal{D}}\mathbb{C} \equiv \mathbb{C}^{(2)}$ is violated when, in a single domain of type $\mathcal{D}$, there is a violation of A ≡ $\mathbb{C}$ distinct from a violation of B ≡ $\mathbb{C}$; that is, when a violation of A ≡ $\mathbb{C}$ is ignored, there remains a violation of B ≡ $\mathbb{C}$.
>
> (Smolensky 2006: 123)

When we apply this idea to Rendaku, the cancellation process generates the following sets of violations for *yakizoba* and *yakisoba*:

(38) a. Let $C_1 = C_2 \equiv$ *[voice]. It follows that $C_1\&_{\text{Stem}}C_2 = C^2_{\text{Stem}} \equiv$ *[voice]$^2_{\text{Stem}}$.

b. *yakizoba*

i. Sets of violations: $C_1 \equiv$ *[voice] = {z, b}; $C_2 \equiv$ *[voice] = {z, b}

ii. Take one violation {b} from the set of C_1, and cancel its corresponding violation in the set of C_2: C_1 = {z, b}; C_2 = {z, ~~b~~}.

iii. As a result, *yakizoba* has the sets of violations: C_1 = {z, b}; C_2 = {z}

c. *yakisoba*

i. Sets of violations: C_1 = {b}; C_2 = {b}

ii. Take one violation {b} from the set of C_1, and cancel its corresponding violation in the set of C_2: C_1= {b}; C_2 = {~~b~~}.

iii. As a result, *yakisoba* has the sets of violations: C_1 = {b}; C_2 = {ø}.

Based on the sets of violations obtained from the cancellation process in (38), *[voice]$^2_{\text{Stem}}$ is evaluated as in (39):

(39)

	C_1 = *[voice]	C_2 = *[voice]	$C_1\&_{\text{Stem}}C_2$ = *[voice]$^2_{\text{Stem}}$
yakizoba	*z, *b	*z	*
yakisoba	*b		

Note that it is the candidate with a single violation (i.e. *yakisoba*) that requires the cancellation process. Otherwise, the candidate would unexpectedly violate *[voice]$^2_{\text{Stem}}$ (see (26)). The evaluation for a candidate with two violations (i.e. *yakizoba*) is practically unchanged with or without the cancellation process, because what matters is whether the Self "Conjoined" constraint is violated or not.

As in previous approaches, this approach can also capture the situation depicted in diagram (29): through the process of cancellation, candidates with multiple violations are properly delimited as a subset. Moreover, Self Conjunction can be regarded as a special case of constraint conjunction under this approach, as assumed in previous studies.

This approach has a potential problem, however. If the cancellation process is applied iteratively, the process would end up producing an empty set for every candidate. Take (38a) for example. As we have seen, canceling one violation {b} from C_1 results in C_1 = {z, b} and C_2 = {z}. At this stage, there remains a violation {z} that is shared by both sets, and this violation *can* be canceled. However, if {z} were canceled from the set of C_2, C_2 would be empty and *yakizoba* would unexpectedly satisfy *[voice]$^2_{\text{Stem}}$. Thus, we have to posit additional assumptions to eliminate the possibility of excessive cancellation.

One possible solution is to assume that the cancellation process is "bidirectional": once a violation of C_1 is cancelled from the set of C_2, one of the remaining violations of C_2 will be cancelled from the set of C_1. Thus, from the sets in (36a. iii) (i.e. C_1 = {z, b} and C_2 = {z}), the violation {z} of C_2 will be cancelled from the set of C_1, resulting in the sets: C_1 = {b} and C_2 = {z}. It should be noted that the result of this bidirectional cancellation is consistent with the normal interpretation of Self Conjunction: two violations incurred by a candidate are distributed between C_1 and C_2 as if they were violations of two distinct constraints.

Another possible solution is to consider cancellation as part of the evaluation process: a violation of C^2_D is assigned if there is any remaining violation in the set of C_2 after a violation corresponding to that of C_1 has been canceled. In the case of *yakizoba* vs. *yakisoba*, *yakizoba* incurs one violation of *[voice]$^2_{\text{Stem}}$ because there remains a violation of {z} in the set of C_2 after the cancellation of {b}. Contrastively, *yakisoba* does not violate *[voice]$^2_{\text{Stem}}$ because, after the cancellation of {b}, there is no violation left in the set of C_2. This "cancellation as evaluation" mechanism works well for Self "Conjunction". However, the "self-conjoined" constraint on this assumption is quite unusual as an OT constraint because its evaluation is no longer based on the linguistic representation of the candidate but the number of violations included in the sets.

4.3.5 Summary

It is possible to solve the problem of Self Conjunction in various theoretical ways, each of which has its own problems. Moreover, it is not yet known whether these different approaches would make any empirical difference. We have to wait for further studies to determine which of these approaches is the most appropriate.

5. Conclusion

In this paper, we have discussed two issues concerning constraint conjunction: (i) how the locality effect can theoretically be exercised; and (ii) how Self Conjunction can most appropriately be dealt with. Although we could not arrive at a definitive answer to either of these issues, we now know that (i) the locality function of Local Conjunction can best be exercised by some means of loci of constraints; and that (ii) it is impossible to analyze the case of Self Conjunction solely through the mechanisms of regular Local Conjunction.

It is obvious that we need further research to determine (and refine) the most appropriate analysis for these issues. The present discussion, however, contributes to phonology, if only because the issues involved have rarely been discussed. Moreover, the current discussion will help raise the awareness of assuming new conjoined constraints in future studies.

Acknowledgments

The basic premises of this paper go back to a discussion held between the authors and Haruka Fukazawa on the PHonEM mailing list in 2000, and which recently recurred in a discussion of Rendaku with Shigeto Kawahara. The authors express their deepest gratitude to these and other phonologists who participated in the discussion for providing great amount of input in writing this paper. The authors also want to thank Mark Campana for suggesting stylistic improvements. Financial support was given to the first author by Grant-in-Aid for Scientific Research (B) (No. 26284059) and (C) (No. 21520513).

References

Alderete, John. 1997. Dissimilation as local conjunction. *Proceedings of North East Linguistic Society 27*, ed. by Kiyomi Kusumoto, 17–32. Amherst, MA: GLSA.

Baertsch, Karen. 2002. *An optimality theoretic approach to syllable structure: The split margin hierarchy*. Indiana: Indiana University dissertation.

Bradley, Travis G. 2007. Constraints on the metathesis of sonorant consonants in Judeo-Spanish. *Probus* 19.171–207.

Crowhurst, Megan and Mark Hewitt. 1997. Boolean operations and constraint interactions in Optimality Theory. Unpublished manuscript. Chapel Hill, NC: University of North Carolina at Chapel Hill; Waltham, MA: Brandeis University. Online: Rutgers Optimality Archive, ROA-229.

Gnanadesikan, Amalia. 2004. Markedness and faithfulness constraints in child phonology. *Constraints in phonological acquisition*, ed. by Rene Kagér, Joe Pater, and Wim Zonneveld, 73–108. Cambridge: Cambridge University Press.

Gouskova, Maria. 2002. Exceptions to sonority distance generalizations. ed. by Mary Andronis, Erin Debenport, Anne Pycha, and Keiko Yoshimura, *CLS 38: The main session*, 253–268. Chicago: Chicago Linguistic Society.

Hayes, Bruce. 1989. Compensatory lengthening in moraic phonology. *Linguistic Inquiry* 20.253–306.

Hayes, Bruce. 1995. *Metrical stress theory: Principles and case studies*. Chicago: University of Chicago Press.

Hualde, José Ignacio. 2005. *The sounds of Spanish*. Cambridge: Cambridge University Press.

Ito, Junko. 1986. *Syllable theory in prosodic phonology*. Amherst, MA: University of Massachusetts dissertation.

Ito, Junko. 1989. A prosodic theory of epenthesis. *Natural Language and Linguistic Theory* 7.217–259.

Ito, Junko and Armin Mester. 1996. Rendaku I: Constraint conjunction and the OCP. Handout of talk at the Kobe Phonology Forum.

Ito, Junko and Armin Mester. 2003. *Japanese morphophonemics: Markedness and word structure*. Cambridge, MA: MIT Press.

Łubowicz, Anna. 2002. Derived environment effects in Optimality Theory. *Lingua* 112.243–280.

Łubowicz, Anna. 2005. Locality of conjunction. *Proceedings of the 24th West Coast Conference on Formal Linguistics*, ed. by John Alderete, Chung-hye Han, and Alexei Kochetov, 254–262. Somerville, MA: Cascadilla Proceeding Project.

McCarthy, John J. 1986. OCP effects: Gemination and antigemination. *Linguistic Inquiry* 17.207–263.

McCarthy, John J. 2002. *A thematic guide to Optimality Theory*. Cambridge: Cambridge University Press.

McCarthy, John J. 2003. Comparative markedness. *Theoretical Linguistics* 29.1–51.

Parker, Steve. 2001. Non-optimal onsets in Chamicuro: An inventory maximised in coda position. *Phonology* 18.361–386.

Pons-Moll, Clàudia. 2011. It is all downhill from here: A typological study of the role of Syllable Contact in Romance languages. *Probus* 23.105–173.

Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, MA: Blackwell Publishing.

Rubach, Jerzy. 1990. Final devoicing and cyclic syllabification in German. *Linguistic Inquiry* 21. 79-94.

Smith, Jennifer. 2002. *Phonological augmentation in prominent positions*. Amherst, MA: University of Massachusetts dissertation.

Smith, Jennifer. 2004. Making constraints positional: Toward a compositional model of CON. *Lingua* 114.1433–1464.

Smolensky, Paul. 1993. Harmony, markedness, and phonological activity. Handout from Rutgers Optimality Workshop I, New Brunswick, NJ.

Smolensky, Paul. 1995. On the internal structure of the constraint component Con of UG. Handout from talk at University California, Los Angeles.

Smolensky, Paul. 1997. Constraint interaction in generative grammar II: Local conjunction, or random rules in Universal Grammar. Handout from Hopkins Optimality Theory Workshop/Maryland Mayfest ’97, Baltimore.

Smolensky, Paul. 2006. Optimality in phonology II: Harmonic completeness, local constraint conjunction, and feature domain markedness. *The harmonic mind: From neural computation to optimality-theoretic grammar, vol. 2: Linguistic and philosophical implications*, ed. by Paul Smolensky and Géraldine Legendre, 27–160. Cambridge, MA: MIT Press.

Vennemann, Theo. 1988. *Preference laws for syllable structure and the explanation of sound change: With special reference to German, Germanic, Italian, and Latin*. Berlin: Mouton de Gruyter.

Zec, Draga. 1988. *Sonority constraints on prosodic structure*. Stanford, CA: Stanford University dissertation.

Zec, Draga. 1995. Sonority constraints on syllable structure. *Phonology* 12.85–129.