Ambiguity avoidance as contrast preservation: Case and word order freezing in Japanese*

Kathryn Flack

University of Massachusetts, Amherst

Many sorts of ambiguity are tolerated in language. Pronouns may be ambiguous as to their referents, quantifiers may have ambiguous scopes, and structural ambiguity can arise, as in (1) from Japanese (see Inoue and Fodor (1995) for more examples of ambiguity in Japanese).

(1) Sumiko-to Jiroo-no okaasan = Sumiko-to [Jiroo-no okaasan]
Sumiko-CONJ Jiroo-GEN mother 'Sumiko and [Jiroo's mother]'

= [Sumiko-to Jiroo]-no okaasan
'[Sumiko and Jiroo]'s mother'

But ambiguity is not always tolerated. Occasionally a syntactic process appears to be blocked or triggered in order to prevent ambiguity. One such process is Japanese scrambling, which cannot occur in sentences where subjects and objects are morphologically identical (i.e. are not distinguished by case morphology). If scrambling were allowed in such sentences, the sentences would be ambiguous as to their subjects. This word order freezing occurs both in sentences like (2), where the subject and object both receive nominative morphology, and also in sentences like (3), where both case particles are dropped. In both (2) and (3), the (ungrammatical) scrambled structure with subject *Hanako* would sound exactly like the unscrambled structure with subject *Taroo*; when the scrambled sentences are blocked, subject-related ambiguity is prevented.

* Thanks to Michael Becker, Tim Beechey, Rajesh Bhatt, Paul de Lacy, Lyn Frazier, Makoto Kadowaki, John McCarthy, Naz Merchant, Anne-Michelle Tessier, and Adam Wayment, and the audience at HUMDRUM 2005. Special thanks to Ellen Woolford and Shigeto Kawahara, both tireless sources of very different kinds of information, judgements, and general support. A version of this paper appears in UMOP 32: Papers in Optimality Theory III.

(2)	*"Taroo-ga$_i$ Hanako-ga t$_i$ kowa-i." T-NOM H-NOM is.afraid-PRES *'Hanako is afraid of Taroo.'	=	"Taroo-ga Hanako-ga kowa-i." T-NOM H-NOM is.afraid-PRES 'Taroo is afraid of Hanako.'
(3)	*"Taroo-∅$_i$ Hanako-∅ t$_i$ osore-ru." T-(ACC) H-(NOM) fear-PRES *'Hanako fears Taroo.'	=	"Taroo-∅ Hanako-∅ osore-ru." T-(NOM) H-(ACC) fear-PRES 'Taroo fears Hanako.'

Word order freezing also prevents subject ambiguity in Hindi, German, Korean, and other languages (see Lee 2001 and references therein).

Ambiguity avoidance is difficult to capture as a grammatical phenomenon. Previous attempts to explain word order freezing as in Japanese and other such "anti-ambiguity" phenomena have generally taken one of two approaches. One style of explanation is extremely local, arguing that e.g. word order freezing results from some other inherent property of the double nominative construction, rather than as the result of a grammatical pressure to avoid ambiguity (as in Tonoike 1980a,b). The other common sort of explanation is a global claim that avoidance of some or all kinds of ambiguity is a fundamental property of grammar (as in bidirectional OT; see e.g. Kuhn 2001, Lee 2001). I will argue that the former sort of explanation cannot capture crucial generalizations about the variety of processes that give rise to word order freezing in Japanese, while the latter explanation makes it difficult to explain exceptions to general patterns of ambiguity avoidance.

This paper claims that an OT grammar can penalize subject-related ambiguity (Prince and Smolensky 1993) using an anti-ambiguity contrast preservation constraint, PRESERVECONTRAST(Subject). Contrast preservation phenomena have received a great deal of attention in recent work in phonology (e.g. Flemming 1995, 1996; Ito and Mester 2003; Lubowicz 2003; Padgett 2003, 2004; Tessier 2004); this paper will demonstrate that syntax is also concerned with evaluating patterns of contrast preservation and neutralization among possible sets of outputs.

When PRESERVECONTRAST(Subject) is highly ranked, subject-related ambiguity is prevented; as this ambiguity avoidance constraint, like all OT constraints, is violable, subject ambiguity is tolerated when it is low-ranked. One major difficulty in using constraints to detect and penalize ambiguity is that an OT evaluation traditionally maps a single input to a single output. Ambiguity, however, is not a property of a single input-output pair, but rather can only be identified by examining a pair of inputs and noting that they are paired with the same output correspondent, as in (4). A suitably unambiguous input-output mapping therefore cannot be chosen without reference to other input-output mappings.

(4) Input_A Input_B
↘ ↙
Output

In order to give a constraint-based analysis of ambiguity avoidance, I will argue for a shift from the single-input architecture of OT to an architecture which evaluates 'clusters' of related inputs and outputs; contrast preservation theories of phonology argue for a similar shift to evaluation of sets of related forms. While this is a significant modification of OT, I will demonstrate that it can be accomplished while allowing most of the fundamental properties of an OT grammar to hold unchanged.

The paper is structured as follows. The first section describes the ambiguity avoidance strategy used in Japanese, and provides a range of evidence indicating that this is a grammatical phenomenon rather than a processing bias. Section 2 then investigates the Japanese patterns in more detail, arguing that scrambling occurs in Japanese syntax and that word order freezing occurs at PF, when the tail of the scrambling chain is pronounced rather than its head, assuming that movement creates chains of copies in syntax and that chains are resolved at PF as in the copy theory of movement (Chomsky 1995; Richards 1997). This atypical pronunciation of the scrambling chain is motivated by the constraint PRESERVECONTRAST(Subject), which penalizes scrambling when it would introduce subject-related ambiguity. The fact that word order freezing is chosen as the repair for ambiguity follows from the constraint hierarchy of Japanese; that is, as in all OT analyses, while ambiguity is penalized by PRCONTR(Subj), the choice of repair is grammar-dependent.

Section 3 describes necessary modifications to the traditional OT model which allow constraints crucial access to multiple, potentially ambiguous inputs and outputs. Evaluation of clusters requires minor modifications to Gen and Eval, as well as a means of generating input clusters; I argue that while the crucial evaluation of clusters here occurs at PF, input clusters should in fact be initially generated at syntax. Section 4 returns to Japanese and demonstrates that predictions made by the interaction of PRESERVECONTRAST(Subject) and the rest of the Japanese constraint hierarchy are borne out: PRCONTR(Subj) is violable, allowing focus-driven scrambling to induce ambiguity, and a novel source of Japanese morphological ambiguity (stylistic case particle drop) also causes word order freezing. Section 5 finally contrasts this analysis with other proposals which have attempted to explain anti-ambiguity and word order freezing effects. These alternative proposals variously attribute the effects to inherent properties of double nominative constructions, to processing biases, to universal filters on outputs, and to bidirectional OT evaluation. The present proposal is found to cover a broader range of data with fewer modifications to an existing architecture than any other proposed analysis.

1. Case marking and word order in Japanese

Most Japanese verbs make a morphological distinction between subjects and objects: subjects generally receive the nominative suffix *–ga* while objects are assigned accusative *–o*. Subjects and objects can therefore typically be unambiguously identified even if an object has scrambled to the front of the sentence, disrupting the canonical SOV word order, as in (5).

(5) a. Taroo-ga Hanako-o osore-ru
Taroo-NOM Hanako-ACC fear-PRES
'Taroo fears Hanako'

b. Hanako-o_i Taroo-ga t_i osore-ru

This morphological distinction between subjects and objects is lost in Japanese stative predicates (e.g. *kowa-i* 'be afraid of', *kikoe-ru* 'hear', *waka-ru* 'understand', *itosii* 'think tenderly of', *zyoozu* 'be good at'; Kuno 1973: 81-82, Tsujimura 1996: 211), which assign nominative *–ga* to both subjects and objects. Scrambling, which occurs freely in nominative-accusative sentences, is blocked in double nominative sentences, as in (6).[1]

(6) a. Hanako-ga Taroo-ga kowa-i
Hanako-NOM Taroo-NOM afraid.of-PRES
'Hanako is afraid of Taroo'

b. *Taroo-ga_i Hanako-ga t_i kowa-i

Despite the absence of morphology which would distinguish subjects and objects in double nominative sentences, these arguments are still unambiguously recoverable due to the fact that word order is fixed in just this context. Word order freezing appears to compensate for the lack of distinctive morphology, in that the word order is suddenly fixed in order to prevent the subject-object ambiguity which would otherwise arise.

To see this more explicitly, consider the double nominative sentences (7a,c) expressing the meanings *Hanako is afraid of Taroo* and *Taroo is afraid of Hanako*. If scrambling were permitted in these contexts, the pairs of surface forms (7a,d) and (7b,c) would be homophonous and thus ambiguous; it would be impossible for a listener to uniquely identify the subject and object, and so the intended meaning, of any of the forms.

(7) a. Hanako-ga Taroo-ga kowa-i 'Hanako is afraid of Taroo'
b. *Taroo-ga_i Hanako-ga t_i kowa-i 'Hanako is afraid of Taroo'

c. Taroo-ga Hanako-ga kowa-i 'Taroo is afraid of Hanako'
d. *Hanako-ga_i Taroo-ga t_i kowa-i 'Taroo is afraid of Hanako'

[1] The scrambling discussed throughout the major portion of this paper is an optional process which moves a discourse topic to the front of a clause (Kuroda 1988; Saito and Fukui 1998); no clear meaning difference accompanies this movement. This sort of scrambling is different from focus-driven scrambling, which produces exhaustive meanings like 'Taroo helps <u>Hanako</u> (and no one else)', or contrastive meanings like 'Taroo helps <u>Hanako</u> (not Ziroo).' Focus-driven scrambling is further different from discourse topic scrambling in that the former can cross clause boundaries, can produce ambiguity, and is obligatorily accompanied by focus intonation. Evidence that focus-driven and non-focus-driven scrambling can differ is also found in Hindi (Dayal 2003), where only specific nominals can scramble in non-contrastive focus contexts, but specific and nonspecific nominals can both scramble if they are contrastively focused. Japanese focus-driven scrambling is discussed in more detail in section 3.

Word order freezing holds even in cases where independent properties of the sentence, e.g. animacy or selectional restrictions, should make available only one reading of the sentence. In (8), the most sensible interpretation of the sentence is the one where the object has scrambled, where the meaning is *Taroo is afraid of earthquakes*. This scrambled reading is, however, impossible, and the sentence can have only the unlikely nonsense reading *Earthquakes are afraid of Taroo* in (8b). We see from this that word order is frozen in SOV order even when the meanings of the words should inherently render the sentence unambiguous.[2]

(8) a. *jishin-ga$_i$ Taroo-ga t_i kowa-i
earthquakes-NOM Taroo-NOM afraid.of-PRES
'Taroo is afraid of earthquakes'

b. jishin-ga Taroo-ga kowa-i
'Earthquakes are afraid of Taroo'

Scrambling is also blocked in double nominative sentences even when the alternative reading is not only unlikely but truly impossible due to the selectional requirements of the verb in question. This is shown in the pair of sentences in (9), where the verb *zyoozu-da* 'is good at (a skill)' must have as its object a skill that one can excel at, and should have an animate, probably human subject as well. While (9a) is an appropriate comment in a context where tennis is under discussion (and thus discourse topic scrambling is felicitous), the very similar (9b) is not; again, scrambling cannot occur in a double nominative construction.

(9) a. tenisu-o$_i$ Taroo-ga t_i suki-da
tennis-ACC Taroo-NOM like-PRES
'Taroo likes tennis'

b. *tenisu-ga$_i$ Taroo-ga t_i zyoozu-da
tennis-NOM Taroo-NOM good.at-PRES
'Taroo is good at tennis'

Taroo is an ideal subject for *zyoozu-da*, but an impossible object; similarly, *tenisu* 'tennis' is an ideal object but an impossible subject. The scrambled version of the sentence in (9b) is therefore not ambiguous in any sense other than the strictly morphosyntactic. If scrambling were permitted, a listener could always recover the underlying subject based on the verb's selectional restrictions despite the morphological ambiguity. Word order freezing in Japanese therefore does not seem to be a processing bias, as such a bias would freeze word order only in contexts where ambiguity would result otherwise, but would permit scrambling in ultimately unambiguous contexts like

[2] There is a clear contrast between this double nominative sentence, in which scrambling is impossible, and the very similar nominative-accusative sentence in which scrambling can occur: *jishin-o$_i$ Taroo-ga t_i osore-ru* 'Taroo fears earthquakes' is grammatical, presumably due to the case markers which distinguish subject from object.

(8) and (9). Word order freezing appears instead to be the result of a strict grammatical prohibition on scrambling in double nominative sentences.[3]

Further evidence that avoidance of subject ambiguity in Japanese is grammatical rather than based in processing comes from cross-linguistic variation in tolerance for subject ambiguity. Processing biases tend to be consistent across languages, while grammatical properties can vary. Cross-linguistic variability is therefore further evidence that, while subject ambiguity avoidance is common, it is the result of a violable grammatical principle rather than a universal processing bias. Texistepec Popoluca (Reilly 2002) (as well as the closely related Sierra Popoluca) is a language in which morphological ambiguity between arguments results in truly ambiguous sentences. Texistepec Popoluca verbs agree with both subjects and objects; arguments themselves are not marked for case. When both subjects and objects are third person, there is no morphological indication of which argument is the subject and which is the object. Reilly (p.c.) reports, “speakers’ intuitions in a plausibly equi-biased [morphologically ambiguous] sentence…are split 50/50” between VSO and VOS word orders, as shown in (10); further, speakers readily produce both VSO and VOS sentences in identical contexts.

(10)	maʔ	dʲ-agaʔ	maʃ-dʲaʔa	kaŋ-daʔa
	PERF	3/3-kill	Tomás-BIG.MASC	jaguar-BIG.MASC

‘Tomás killed the jaguar’
‘The jaguar killed Tomás’

In sum, Japanese is very different from a language like Texistepec Popoluca in the extent to which it enforces a rigid word order in morphologically ambiguous contexts. The Japanese facts appear to result from a violable grammatical prohibition against scrambling in double nominative sentences. This is a within-language example of the common cross-linguistic observation that languages generally use either case marking or fixed word order as a mechanism for unambiguously identifying arguments.

2. Word order freezing as contrast preservation

In order to capture the fact that scrambling cannot occur in Japanese when a subject and object are morphologically ambiguous, I will propose an OT constraint PRESERVECONTRAST(Subject) which demands output preservation of input subject contrasts, as in *Taroo is afraid of Hanako* versus *Hanako is afraid of Taroo*. In this section, I will describe the particular contrasts which are relevant to word order freezing in Japanese, and the details of the scrambling operation which is blocked in double nominative contexts. I will then define PRESERVECONTRAST(Subject), and will show how

[3] There is evidence from processing that the second nominative argument in a sequence is likely assumed by listener to be the subject of an embedded clause (Inoue (1991), Uehara (1997); cf. Nagai (1995)); this assumption is corrected by the final portion of the sentence, as listeners realize that only a single verb, which takes a nominative object, is present. Once a listener has determined that both nominative arguments belong to a single clause, they are unambiguously interpreted in subject-object order.

it can cause word order to freeze in the appropriate contexts. I will also demonstrate that while PRESERVECONTRAST(Subject) penalizes subject ambiguity, the repair for this ambiguity – word order freezing – is not identified by the constraint but is determined by the rest of the constraint hierarchy of Japanese.

2.1. Contrast preservation and neutralization

In general, patterns of ambiguity and the avoidance thereof in a language can be thought of as patterns of contrast neutralization and realization, respectively. In OT terms, ambiguous sentences are those where two different inputs both map to a single output; that is, as in (11a), those output strings which can be derived from two different inputs. Ambiguity therefore causes the contrast between inputs to be neutralized on the surface. Unambiguous sentences are those which correspond to only single inputs (as in (11b)); such inputs are therefore unambiguously recoverable.

(11) a. Input_A ↘ ↙ Input_B
Output

b. Input_A → Output_A
Input_B → Output_B

Contrast preservation will be a major theme of the analysis of word order freezing offered in this paper; the patterns of contrast relevant to this phenomenon are as follows.

In Japanese nominative-accusative sentences, scrambling can occur freely without neutralizing contrasts between inputs with different subjects. Scrambling is an optional process (Kuroda 1988, Saito and Fukui 1998), and so a single input in which the object is a discourse topic may map to either a scrambled or an unscrambled output. The unambiguous case marking in nominative-accusative sentences allows the two inputs in (12) to map to four unambiguous outputs.[4]

(12) a. osore-PRES <Agent: Hanako, Theme: Taroo>
→ Hanako-ga Taroo-o osore-ru
→ Taroo-o$_i$ Hanako-ga t$_i$ osore-ru

b. osore-PRES <Agent: Taroo, Theme: Hanako>
→ Taroo-ga Hanako-o osore-ru
→ Hanako-o$_i$ Taroo-ga t$_i$ osore-ru

In double nominative sentences, however, scrambling is blocked, as shown in (13).

[4] Here we see mappings between numeration-like inputs with lexical items, features, and argument structures and surface forms. In the next section, I will look more closely at the derivation and separate the syntactic mapping from numeration-like inputs to syntactic structures (where scrambling occurs) from the PF mapping from syntactic structures to surface forms (where I argue word order freezing occurs).

(13) a. kowa-PRES <Agent: Hanako, Theme: Taroo>
→ Hanako-ga Taroo-ga kowa-i
→ *Taroo-ga$_i$ Hanako-ga t$_i$ kowa-i

b. kowa-PRES <Agent: Taroo, Theme: Hanako>
→ Taroo-ga Hanako-ga kowa-i
→ *Hanako-ga$_i$ Taroo-ga t$_i$ kowa-i

Double nominative sentences have no unambiguous case morphology providing an inherent surface contrast between subjects and objects. Therefore if nominative objects could scramble, the output contrast between pairs of inputs with different subjects (and objects) would be neutralized: *Hanako is afraid of Taroo* and *Taroo is afraid of Hanako* could be pronounced identically, as in (14). By blocking scrambling, however, the contrast is preserved between inputs whose subjects differ.

(14) *"Taroo-ga$_i$ Hanako-ga t$_i$ kowa-i." = "Taroo-ga Hanako-ga kowa-i."
T-NOM H-NOM fear-PRES T-NOM H-NOM fear-PRES
*'Hanako fears Taroo.' 'Taroo fears Hanako.'

2.2. Scrambling and contrast preservation

Scrambling in Japanese is a syntactic process in which discourse topics optionally move to sentence-initial position (Kuroda 1988, Saito and Fukui 1998). Under the copy theory of movement (Chomsky 1995; Richards 1997), scrambling creates a copy of a discourse topic and places this copy at the beginning of the sentence; the two copies of the object form a chain. After a scrambling chain is formed in syntax, its pronunciation is determined at PF, where all copies except one (generally the head) are deleted (see Fanselow 2001 for an overview of chain resolution effects). The effects of scrambling are visible in an output if the head of a scrambling chain surfaces; scrambling appears to be blocked in an output if PF considerations force the tail of the chain to surface instead.

I claim that scrambling occurs in syntax in both nominative-accusative and double nominative sentences; the different behavior of scrambling chains in these two types of sentences emerges at PF, where the scrambling chains are resolved differently. (15) shows PF mappings from syntactic structures to surface forms of nominative-accusative sentences in (15a) and double nominative sentences in (15b).

(15) a. O-ACC$_i$ S-NOM O-ACC$_i$ VERB → O-o$_i$ S-ga ~~O-o~~$_i$ VERB

b. O-NOM$_i$ S-NOM O-NOM$_i$ VERB → ~~O-ga~~$_i$ S-ga O-ga$_i$ VERB

The effects of scrambling surface in nominative-accusative sentences; this means that when an accusative object scrambles, the head of this chain is pronounced. In contrast, the tail of a scrambling chain is pronounced in a double nominative sentence. Pronunciation of the tail of the chain in double nominative sentences prevents ambiguity between sentences with different subjects, as was demonstrated above in (13) and (14).

In order to capture this pattern in an OT grammar, some constraint which forces preservation of the subject contrast must dominate a constraint which generally causes chain heads to be pronounced. I propose that the latter condition on chains is enforced by MAX(Head).

(16) MAX(Head) Do not delete the head of a chain.

The constraint which dominates MAX(Head) in order to preserve subject contrasts is somewhat more complex, and will be introduced in the next section.

First, though, it should be noted that the claim that there is a higher, unpronounced copy of a nominative object when scrambling has occurred in syntax predicts that this copy should be available for interpretation at LF. More specifically, if scrambling is blocked in a double-nominative sentence, an object quantifier or pronoun which is pronounced in situ should be interpretable with either wide or narrow scope due to the hypothesized higher copy which is present in syntax but not pronounced at PF. Unfortunately, it is extremely difficult to test whether this higher copy is available at LF. It is true that a nominative object quantifier, as in (17a), must be interpreted in situ, with narrow scope, while the object quantifier in the nominative-accusative sentence in (17b) can scramble at LF to take wide scope.

(17) a. Dareka-ga daremo-ga kowa-i
someone-NOM everyone-NOM afraid.of-PRES
'Someone is afraid of everyone.'
= There is someone who is afraid of everyone. (someone > everyone)
= *Everyone is feared by someone. (*everyone > someone)

b. Dareka-ga daremo-o osore-ru
someone-NOM everyone-ACC fear-PRES
'Someone fears everyone.'
= There is someone who is afraid of everyone. (someone > everyone)
= Everyone is feared by someone. (everyone > someone)

The absence of a wide-scope reading for these nominative object quantifiers does not, however, appear to be directly related to the absence of scrambling in double-nominative sentences. Evidence for widespread restrictions on nominative object quantifiers can be found in sentences with nominative objects but dative subjects, as in (18), where again the nominative object quantifier cannot scramble at LF to take wide scope. There is no need to block this LF scrambling in order to preserve a distinction between subject and object, as the two are distinguished by case in dative-nominative sentences; the absence of LF scrambling here – and likely in the double-nominative sentence above – therefore strongly appears to be a peculiarity of the nominative object quantifier.

(18) Dono-seito-ni-mo nanika-ga yak-er-u
every-student-DAT-MO something-NOM bake-POTENTIAL-PRES
'Every student can bake something.'
= For every student, there is something that they can bake. (every > some)
= *There is something that every student can bake. (*every > some)

The claim that nominative object quantifiers are inherently atypical in that they cannot give rise to wide-scope readings is, finally, further supported by the observation that an accusative object quantifier may scramble when it is focused (see section 4.1 for more discussion of focus-driven scrambling), while a nominative object quantifier may not, as shown in (19).[5]

(19) a. *Daremo-ga$_i$ dareka-ga t_i kowa-i

b. Daremo-o$_i$ dareka-ga t_i osore-ru

There is, therefore, no evidence against the claim made here that double-nominative sentences do in fact undergo scrambling in syntax, and that the higher copy is ultimately present though unpronounced due to PF considerations. In the absence of evidence against the presence of this unpronounced higher copy, I will proceed assuming that it is present and forced to delete by a contrast preservation constraint which outranks MAX(Head).

2.3. PRESERVECONTRAST(Subject)

A constraint which preserves contrasts between inputs whose subjects differ must be able to see more than one input at a time; that is, as shown in (19) (repeated from above), contrast neutralization is a property of multiple input-output mappings, and so it is impossible to know whether a given input-output pairing neutralizes a contrast without checking for other distinct inputs which map to an identical output.

(20) a. Input$_A$ Input$_B$
↘ ↙
Output

b. Input$_A$ Input$_B$
↓ ↓
Output$_A$ Output$_B$

In order to allow constraints to detect and penalize ambiguity, the traditional architecture of OT must therefore be modified such that inputs, as well as output candidates, are 'clusters' of related forms which can be examined for neutralization of subject contrasts. This section proceeds with this intuitive understanding of clusters; formal details of the

[5] Other possible LF effects pose more basic analytic problems. One might suspect that coreference between arguments and the reflexive pronoun *zibun* might be licensed by covert scrambling; *zibun*, however, must be coreferent with a subject, and so in a double nominative sentence would itself need to be the object, and so the surface subject > object scope in a sentence like *Taroo-ga zibun-ga suki-da* 'Taroo likes himself' would be sufficient to license the only possible coreference. Coreference between nonreflexive pronouns and arguments only holds across clause boundaries, and scrambling across clause boundaries cannot occur without focus (as discussed in section 4.1) and so is unlike the scrambling discussed here.

implementation and consequences of a cluster-based model of OT are discussed in section 3.

As described above, word order is frozen in double nominative sentences when some constraint which penalizes neutralization of a subject contrast dominates MAX(Head). In order to enforce this ban on subject contrast neutralization, I propose the constraint PRESERVECONTRAST(Subject), defined in (21). PRESERVECONTRAST(Subject) is similar in spirit to, though formally distinct from, phonological contrast preservation constraints in Lubowicz (2003) and the Dispersion Theory constraints in Flemming (1995, 1996) and Padgett (2003, 2004). PRESERVECONTRAST(Subject) assigns violations when inputs whose subjects contrast map to identical outputs, neutralizing a subject contrast. Inputs with a relevant subject contrast will be identified as those with lexically different material in subject position, i.e. spec-IP. Crucially, this constraint operates at PF, where inputs are syntactic structures and outputs are simply phonological strings with no remaining syntactic structure.

(21) PRESERVECONTRAST(Subject): Given two pairs of input-output correspondents *I*, *O* and *I'*, *O'* where *O* and *O'* are in cluster *C*, if Subject*(I)*[6] ≠ *Subject(I')* and *O* = *O'*, assign one violation to *C*.

"Inputs with different subjects must map to separate outputs."

PRESERVECONTRAST(Subject) penalizes only subject-related ambiguity. A fundamental motivation for introducing a contrast preservation constraint which specifies the contrast which should be preserved, rather than a very general anti-ambiguity constraint like *AMBIGUITY which would penalize ambiguity from all sources, is the observation that languages tolerate many kinds of ambiguity, as discussed at the beginning of this paper (exemplified in (1)). Cases of word order freezing in Korean, Hindi, and German, which have been the target of anti-ambiguity proposals in recent OT literature (Kuhn 2001; Lee 2001) behave like Japanese in that they block scrambling in order to avoid subject-related ambiguity. Therefore, by using a constraint which targets subject ambiguity while ignoring other sources of ambiguity, the present analysis can easily explain why subjects are a major trigger for anti-ambiguity phenomena. The fact that subjects are preferentially protected from ambiguity is likely related to the fact that subjects are often noted to be more prominent than are other arguments (see e.g. Aissen 1999 and references therein); it seems natural that grammars should protect such a prominent argument. (see fn. 8 for an additional argument against a general constraint *AMBIGUITY.)

As is typical of violable constraints, when PRESERVECONTRAST(Subject) is highly ranked, it can force other constraints (e.g. MAX(Head)) to be violated in order to preserve a contrast. The following section will demonstrate how this ranking accounts for the avoidance of subject ambiguity via word order freezing in Japanese.

---

[6] The function *Subject(X)* here returns the value of the lexical material in spec-IP of a given structure.

2.4. PreserveContrast(Subject) and word order freezing

The first part of this section will show how high-ranking PreserveContrast(Subject) causes word order to freeze in Japanese double-nominative sentences, while allowing scrambling in typical nominative-accusative sentences. The remainder of the section will demonstrate how word order freezing is chosen by the constraint hierarchy of Japanese as the repair for subject ambiguity.

2.4.1 Word order freezing in double nominative sentences

In double nominative sentences, scrambling is blocked in order to preserve a subject contrast. This is enforced by the constraint ranking PreserveContrast(Subject) » Max(Head). The effects of this ranking are shown in (22), where inputs and outputs consist of sets of forms. Each output cluster contains a possible output correspondent of each member of the input cluster; subscripts indicate corresponding input and output forms. As this evaluation occurs at PF, inputs are syntactic structures and outputs are simply phonological strings. Here and throughout this paper, input subjects are bolded in order to show input subject contrasts.

(22)

[T-nom$_i$ **H-nom** T-nom$_i$ kowa-pres]$_1$ [**T-nom** H-nom kowa-pres]$_2$	PrContr(Subj)	Max(Head)
➔ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_1$ a. [T-ga H-ga kowa-i]$_2$		*
[T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ b. [T-ga H-ga kowa-i]$_2$	*!	

The two members of the input cluster shown in (22)[7] differ in their subjects: *Hanako* occupies spec-IP in the input form with subscript *1*, while *Taroo* is in spec-IP of input form *2*. Any output candidate where this input contrast is neutralized – where the correspondents of the input forms *1* and *2* are identical – will incur a violation of PrContr(Subj). In this PF evaluation, outputs are phonological strings with no remaining syntactic structure; PrContr(Subj) therefore assigns violations to pairs of outputs whose phonological forms are identical, regardless of the syntactic structures from which these were derived. In candidate (22b), deletion of the tail of the chain in scrambled form *1* produces a phonological string which is identical to the unscrambled form *2*; (22b) thus violates PrContr(Subj). Candidate (22a) is therefore the winner despite its violation of lower-ranked Max(Head). This evaluation therefore determines the output forms of all members of the input cluster.[8]

[7] Input and output clusters, of course, contain more members than are shown here. However, just as only output candidates which are relevant to the analytical point under discussion are generally shown in tableaux, I will only show the cluster members relevant to the contrast whose preservation is being evaluated. For a detailed discussion of the composition and evaluation of input and output clusters, see section 3.

[8] A deeper problem with a general, non-contrast-specific constraint like *Ambiguity, rather than PrContr(Subj), can be seen here as follows. A more exhaustive input cluster would contain both (cont.)

Crucially, the ranking PRCONTR(Subj) » MAX(Head) only causes word order to freeze in sentences where a double nominative case pattern fails to distinguish between the subject and object. This is ensured by the fact that PRCONTR(Subj) is not violated even if scrambling occurs in nominative-accusative sentences; case marking will always distinguish between nominative-accusative inputs whose subjects differ.

(23)

[T-ACC$_i$ **H-NOM** T-ACC$_i$ kowa-PRES]$_1$ [**T-NOM** H-ACC kowa-PRES]$_2$	PRCONTR(Subj)	MAX(Head)
a. [~~T-o~~$_i$ H-ga T-o$_i$ kowa-i]$_1$ [T-ga H-o kowa-i]$_2$		*!
b. ➔ [T-o$_i$ H-ga ~~T-o~~$_i$ kowa-i]$_1$ [T-ga H-o kowa-i]$_2$		

The two members of the input cluster in (23) have different subjects in their respective spec-IP positions. Because the nominative and accusative suffixes are distinct, however, deletion of the head vs. tail of the scrambling chain will never cause scrambled form *1* and unscrambled form *2* to be homophonous. That is, forms *1* and *2* differ in both output candidate clusters, and so PRCONTR(Subj) is never violated. The choice between output clusters falls to MAX(Head), which protects chain heads from deletion. Deletion of the tail of the chain in (23a) incurs a violation of MAX(Head); (23b) is therefore the winner, as expected.

While PRCONTR(Subj) only causes word order freezing in sentences where subjects and objects bear identical case morphology, the constraint PRCONTR(Subj) is itself quite general, and does not refer specifically to case. This constraint also does not specify that ambiguity should be repaired by word order freezing; rather, as is generally true in OT analyses, the means by which some crucial violation is repaired is determined by the rest of the constraint hierarchy in Japanese, as described in the following section.

---

scrambled and unscrambled forms for each argument structure. If a constraint like *AMBIGUITY penalized any and all contrast neutralizations, neutralization of the subject contrast (as in candidate cluster *a* in the tableau below) and neutralization of the scrambling contrast (as in candidate cluster *b*; crucially not penalized by PRCONTR(Subj)) would incur identical violations of *AMBIGUITY. The decision would be left to MAX(Head), which would prefer deletion of chain tails and thus neutralization of the subject contrast, rather than neutralization of the scrambling contrast as actually occurs. Penalizing specifically subject-related ambiguity using PRCONTR(Subj) avoids this problem.

[T-NOM$_i$ **H-NOM** T-NOM$_i$ kowa-PRES]$_1$ [**H-NOM** T-NOM kowa-PRES]$_2$ [**T-NOM** H-NOM kowa-PRES]$_3$ [H-NOM$_i$ **T-NOM** H-NOM$_i$ kowa-PRES]$_4$	*AMBIGUITY	MAX(Head)
a. *➔ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ [H-ga T-ga kowa-i]$_2$ [T-ga H-ga kowa-i]$_3$ [H-ga$_i$ T-ga ~~H-ga~~$_i$ kowa-i]$_4$	** 1=3, 2=4 subject contrast neutralized	
b. [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_1$ [H-ga T-ga kowa-i]$_2$ [T-ga H-ga kowa-i]$_3$ [~~H-ga~~$_i$ T-ga H-ga$_i$ kowa-i]$_4$	** 1=2, 3=4 scrambling contrast neutralized	*!*

2.4.2. Choosing the optimal repair

In order to 'repair' a violation of PRCONTR(Subj), Japanese forces scrambled inputs to delete the head of a scrambling chain and thus incur a violation of MAX(Head). Both the choice of which member of the potentially ambiguous pair is repaired (the scrambled input, rather than the unscrambled input) and the nature of this repair (deleting the head of the chain) are determined by the constraint ranking in Japanese, as described below. The grammar-dependence of this repair is a classic feature of processes in OT grammars.

The fact that the scrambled input 'changes' its surface form (from typical chain head pronunciation to pronunciation of the tail of the chain instead) in order to preserve the subject contrast follows from the fact that a more optimal alternative output is available for the scrambled input than is available for the unscrambled input, given the Japanese constraint ranking. That is, we know that pronouncing the head of a scrambling chain is relatively unmarked, as it occurs freely in morphologically unambiguous contexts. Similarly, the faithful pronunciation of an unscrambled sentence is also unmarked, as it also occurs freely. Allowing both of these unmarked structures to surface would violate high-ranked PRCONTR(Subj), however, so one member of this pair of forms must surface in a more marked manner. The choice of which form must be repaired is based on which form has a better repair available.

To see this more concretely, we must imagine alternative surface forms for the scrambled and unscrambled inputs; that is, additional PF output candidates which are produced by Gen for each input. Here I will assume, conservatively, that PF operations are limited to decisions about spelling out elements of the syntactic structure. While ambiguity is repaired here by failing to spell out the head of the scrambling chain, it is also possible that some element of the unscrambled form could fail to be spelled out. The repair of the unscrambled form is ruled out as follows.

The most unmarked realization of the scrambled input allows the scrambled object to surface in its scrambled position. This input can also be realized with the object in its base position, thus incurring a violation of MAX(Head). Turning to the other input, the most unmarked realization of the unscrambled form has SOV order and all appropriate case morphology. A possible alternate realization of the unscrambled input might be missing e.g. its object case particle; such a form would no longer sound like the scrambled form, and would incur a violation of MAX(Case). Output clusters which contain these possible repairs of the PRCONTR(Subj) violation are shown in (24).

(24)

[T-NOM$_i$ **H-NOM** T-NOM$_i$ kowa-PRES]$_1$ [**T-NOM** H-NOM kowa-PRES]$_2$	PRCONTR (Subj)	MAX (Case)	MAX(Head)
➔ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_1$ a. [T-ga H-ga kowa-i]$_2$			*
[T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ b. [T-ga H-ga kowa-i]$_2$	*!		
[T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ c. [T-ga H-∅ kowa-i]$_2$		*!	

We know that the cluster (24a), which repairs the scrambled form *1* and violates MAX(Head), wins. This cluster is therefore more optimal than cluster (24c), which repairs the unscrambled form *2* instead and violates MAX(Case); this demonstrates that MAX(Case) must outrank MAX(Head).

In other words, the choice of the form which is repaired in order to avoid a PRCONTR(Subj) violation is made by determining which form can be changed at the least overall cost to the cluster. That is, if the input form *1* in (24) can be best repaired by adding a MAX(Head) violation to the cluster, and input form *2* can be best repaired by adding a MAX(Case) violation, and the ranking MAX(Case) » MAX(Head) holds, then the MAX(Head)-violating cluster is the most optimal, and the scrambled form is repaired.

The constraint hierarchy similarly determines the best repair for the scrambled form; that is, the fact that the scrambled form is repaired by pronouncing the tail of the chain, rather than by some other process, also follows from constraint ranking. The scrambled form could also be repaired by dropping its object case particle (which was ruled out as a repair for the unscrambled form above). Such a repair would, as above, produce a cluster which satisfies PRCONTR(Subj), but which incurs a violation of MAX(Case). These possible repairs to the scrambled form *1* are shown in (25); the ranking MAX(Case) » MAX(Head), established above, also rules out this alternative repair of the scrambled form.

(25)

[T-NOM$_i$ **H-NOM** T-NOM$_i$ kowa-PRES]$_1$ [**T-NOM** H-NOM kowa-PRES]$_2$	PRCONTR (Subj)	MAX (Case)	MAX (Head)
a. ➔ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_1$ [T-ga H-ga kowa-i]$_2$			*
b. [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ [T-ga H-ga kowa-i]$_2$	*!		
c. [T-∅$_i$ H-ga ~~T-∅~~$_i$ kowa-i]$_1$ [T-ga H-ga kowa-i]$_2$		*!	

The actual repair in (25a), in which the tail of the chain is pronounced, must be more harmonic according to the Japanese constraint hierarchy than any other possible repair, including deleting the object case particle as in (25c). Here, MAX(Case) » MAX(Head) determines that chain head deletion is a better repair than morphological disambiguation of the arguments. The fact that the repair is determined by the constraint ranking, rather than being a cross-linguistically fixed repair for ambiguity, is a major difference between this proposal and those of e.g. Hankamer (1973) and Woolford (1986), discussed in section 5.3.

This section has demonstrated that a transderivational contrast preservation constraint, PRESERVECONTRAST(Subject), can account for the word order freezing which occurs in morphologically ambiguous contexts in Japanese. The ranking of this constraint within the constraint hierarchy of Japanese determines the way violations of this constraint are repaired. It has been shown that case marking allows preservation of a subject contrast even when word order is free, while a lack of unambiguous case marking neutralizes this contrast; as this sort of ambiguity violates high-ranking PRCONTR(Subj), it is repaired in the most harmonic way possible according to the constraint hierarchy: by freezing word order.

3. Formalizing the proposal

The inclusion of a transderivational constraint like PRESERVECONTRAST(Subject) in the grammar requires that the OT architecture be modified such that inputs and output candidates are clusters of related forms, rather than individual forms as in most previous OT literature.[9] The following section explores the implementation and consequences of such a modification. I will first sketch the machinery necessary for the PF evaluation of cluster described above, namely how Gen forms output candidate clusters based on an input cluster and then how Eval calculates constraint violations over clusters. It will be shown that the necessary modifications to Gen and Eval are relatively minor. I will then address the question of where PF input clusters are generated; I will argue that syntax should also evaluate clusters of inputs and outputs, and that PF input clusters are simply 'inherited' syntax output clusters.[10] An algorithm will be provided for creating appropriate clusters of initial syntax inputs. Finally, I will show that the shift to a cluster-based model of OT allows previous successful OT syntax analyses developed in single-input models to still hold when clusters are evaluated; this observation is related to the claim that an optimal PF output cluster is composed entirely of actual winning output forms, and so a single evaluation in this cluster-based architecture determines a set of input-output mappings.

3.1. Output cluster generation

In the familiar single-input model of OT, all evaluations began with a single input; Gen created from this single input a set of possible output correspondents. In the cluster-based model discussed here, it is argued that PF evaluations must consider clusters of inputs and outputs. The source of PF inputs will be discussed in section 3.3 below; assuming that such an input cluster exists, however, we will first explore how Gen must be modified in order to produce a set of possible output cluster candidates from an input cluster.

Gen is traditionally considered to be a function which maps an input to a set of outputs. Importantly, there is no variation in Gen across (or within) languages. Each time Gen is applied to an input *I*, it produces the same set of outputs (see McCarthy 2002: 8-10 for discussion). In order to produce output clusters, there must be three relatively straightforward modifications to Gen.

[9] Exceptions to the single-input model are found in work concerned with contrast preservation, e.g. Flemming (1995, 1996), Ito and Mester (2003), Lubowicz (2003), Padgett (2003, 2004), and Tessier (2004).

[10] Most syntax evaluations produce a single winning output cluster, though in cases of syntactic optionality, there can be more than one winning cluster. As will be discussed in section 3.3.3, all syntax output clusters are merged into a single PF input cluster.

First, Gen must apply to the input cluster, and produce a set of output candidates for each member of an input cluster. In doing this, Gen will generate the same set of output candidates of a given form that it would in a single-input model; it will, however, generate multiple sets of output correspondents rather than simply a single one during a single evaluation.

Second, Gen must produce the Cartesian product of these sets of output candidates. That is, each output cluster should contain a single output correspondent of each member of the input cluster, and each combination of output forms should be present in some cluster. This will occur if Gen combines the sets of output candidates for each input into all possible unique output clusters which each contain one output correspondent of each member of the input cluster.

Finally, just as standard Gen specifies correspondence relations between input and output segments and features, the modified version of Gen will also indicate correspondences between members of the input cluster and members of output clusters. This will allow constraints to identify related pairs of inputs and outputs.

These steps, as applied to an input cluster with a scrambled form and an unscrambled form whose subjects are different, as has been considered in tableaux throughout this paper, are shown below in (26).

(26) PF input cluster:

{ [H-NOM T-NOM H-NOM kowa-PRES]$_1$,
[H-NOM T-NOM kowa-PRES]$_2$ }

Gen produces PF output correspondents of each (labelling correspondence relations):

[T-NOM$_i$ H-NOM T-NOM$_i$ kowa-PRES]$_1$	→ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$
	→ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$
[H-NOM T-NOM kowa-PRES]$_2$	→ [T-ga H-ga kowa-i]$_{2A}$,
	→ [T-ga H-∅[11] kowa-i]$_{2B}$

Gen builds unique PF output clusters:

{ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-ga kowa-i]$_{2A}$ };
{ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-∅ kowa-i]$_{2B}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-ga kowa-i]$_{2A}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-∅ kowa-i]$_{2B}$ }

[11] Recall from section 2.4.2 that dropping the object case particle is an alternative hypothetical output form of this PF input.

3.2. Evaluation of output cluster candidates

Just as in a single-input model of OT, in an evaluation of clusters, Eval produces a harmonic ordering of the output candidate clusters created by Gen. As Gen was modified to create output candidate clusters rather than single output forms, Eval must be modified to evaluate clusters rather than single forms. More specifically, in the traditional OT architecture, constraint violations are assessed for single outputs, or single pairs of inputs and outputs in the case of faithfulness constraints. In this cluster-based architecture, markedness and faithfulness constraints evaluate each member of an output cluster (along with any relevant input correspondents); within clusters, PRCONTR(Subj) evaluates pairs of outputs, along with their input correspondents. Eval must therefore be modified such that violations can be assessed over clusters. As was true of the modifications to Gen described above, these modifications prove to be fairly minimal.

Markedness and faithfulness constraints are defined over individual outputs and input-output pairs, respectively. In a cluster-based model of OT, the constraints must evaluate each output in each output cluster, along with corresponding inputs in the case of faithfulness. Eval can therefore be redefined such that constraints are applied to each member of an output cluster, referring to corresponding inputs as well when the definition of e.g. a faithfulness constraint necessitates this. Eval then sums the violations of a given constraint incurred by each cluster member to determine the total violations of that constraint incurred by the cluster as a whole.

Evaluation of markedness and faithfulness constraints over clusters is therefore fairly straightforward. Evaluation of PRCONTR(Subj) violations is somewhat more complex, as this constraint must compare each output in a given output cluster (along with its input correspondent) with each other output (and corresponding input) in the cluster. That is, PRCONTR(Subj) must consider all nC2 pairs of outputs from a cluster of *n* outputs, examining each of these pairs for subject ambiguity.

The complexity of this transderivational constraint does not necessitate any further modification to Eval, however. Just as the necessity of comparing an output form to an input form in evaluating a faithfulness constraint is encoded in the constraint, the comparison of pairs of inputs and outputs is also encoded in the definition of PRCONTR(Subj). While the general application of constraints to clusters (i.e. applying constraints to each member of an output cluster, and summing the violations across outputs) is defined in the new version of Eval described here, the further details of whether constraints examine only outputs (markedness), or input-output pairs (faithfulness), or pairs of input-output pairs (PRCONTR(Subj)) is determined by the constraints themselves.

The formal definition of PRESERVECONTRAST(Subject) is repeated in (27), below. This definition identifies the forms which are compared by the constraint: two outputs *O* and *O'* their input correspondents *I* and *I'*. If *I* and *I'* have different subjects, and *O* and *O'* are identical, then the cluster containing *O* and *O'* incurs a violation of PRCONTR(Subj).

(27) PRESERVECONTRAST(Subject): Given two pairs of input-output correspondents *I*, *O* and *I'*, *O'* where *O* and *O'* are in cluster *C*, if Subject*(I)* ≠ *Subject(I')* and *O* = *O'*, assign one violation to *C*.

"Inputs with different subjects must map to separate outputs."

Given the modification to Eval which allows it to sum violations incurred by each member of an output cluster, and the definition of PRCONTR(Subj) given in (27), evaluation of output cluster candidates proceeds as usual. Working from the top of the constraint hierarchy down, each candidate is assessed for violations of each constraint. Candidates which incur fatal violations are eliminated from the pool of possible winners at each step. The following tableau exemplifies the way in which Eval treats clusters.

(28)

[T-NOM$_i$ **H-NOM** T-NOM$_i$ kowa-PRES]$_1$ [**T-NOM** H-NOM kowa-PRES]$_2$	PRCONTR (Subj)	MAX (Case)	MAX (Head)
a. ➔ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$ [T-ga H-ga kowa-i]$_{2A}$			*
b. [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$ [T-ga H-∅ kowa-i]$_{2B}$		*!	*
c. [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$ [T-ga H-ga kowa-i]$_{2A}$	*!		
d. [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$ [T-ga H-∅ kowa-i]$_{2B}$		*!	

Evaluation proceeds as follows. When undominated PRCONTR(Subj) evaluates output candidate cluster (28a), it initially examines the first member of the cluster, the scrambled form with subscript *1A*. As PRCONTR(Subj) must compare (all) pairs of outputs in a cluster, the rest of the output cluster is searched for forms identical to *1A*. As none are found (i.e. as *2A* is not identical to *1A*), PRCONTR(Subj) is satisfied by this candidate. When PRCONTR(Subj) evaluates candidate cluster (28c), however, it again begins with the first member of this cluster, *1B*, and searches the rest of the cluster for an identical form. Here, such a form is found in *2A*. PRCONTR(Subj) then examines the input correspondents of this pair of outputs to see whether their subjects differ; as input form *1* has subject *H-NOM* and input form *2* does not (rather, its subject is *T-NOM*), this pair (and thus this cluster) incurs a fatal violation of PRCONTR(Subj).

Evaluation of the remaining clusters falls to the two faithfulness constraints. Candidate clusters (28b) and (28d) each contain one member – form *2B*, in which the object case marker is deleted – which incurs a violation of MAX(Case) and another which does not; MAX(Case) is thus fatally violated once by each of these clusters. As candidate cluster (28a) has incurred no violations of either of these undominated constraints (unlike the other candidates), it is therefore the winner. This is of course despite the fact that one of its members, the scrambled form *1A* in which the tail of the chain is pronounced, incurs a violation of low-ranked MAX(Head).

Eval therefore can be straightforwardly modified in order to allow PRCONTR(Subj), as well as traditional markedness and faithfulness constraints, to apply to clusters. This and the previous section have therefore shown that, given a PF input cluster, the necessary modifications to OT which allow PF to map an input cluster to an optimal output cluster can be readily accomplished. The following section will address the somewhat more complex question of where this PF input cluster arises.

3.3. Input cluster generation

Thus far in the discussion of the necessary modifications to OT which allow evaluation of clusters, we have simply assumed the existence of an appropriate PF input cluster. We must now consider the source of this input cluster. The answer to this question will build on the premise that, in a single-input evaluation, a PF input is simply the 'inherited' winning form from the preceding syntax evaluation (Woolford 2001, 2005). Here, I will extend this to the cluster-based model of OT, arguing that PF input clusters should similarly be inherited winning clusters from preceding syntax evaluations. In order for syntax to produce a winning output cluster, I will show that we can easily allow syntax inputs to be clusters as well, and will propose a mechanism for building these syntax input clusters. This section will first address the formation of syntax input clusters (and the motivation for doing so), and then will address issues of syntactic optionality as they contribute to the formation of the appropriate PF input cluster, as optionality is crucial to a discussion of scrambling in Japanese. When syntactic optionality results in two (or more) winning syntax clusters, all members of these possible winning syntax clusters are merged into a single PF input cluster; otherwise, when only a single cluster can win in syntax, this single cluster is directly inherited as the input cluster in the subsequent PF evaluation.

3.3.1. The source of PF input clusters: Syntax output clusters

In previous OT syntax work (Woolford 2001, 2005), inputs to PF evaluations are the winning outputs of syntax evaluations. In the cluster-based model of OT syntax proposed here, I argue that a PF input cluster should similarly be a winning syntax output cluster; in order to obtain this winning syntax cluster, I further argue that syntax inputs and outputs should be clusters, as PF inputs and outputs are. In order to motivate this claim, I will first set out desiderata for PF input clusters, and will then demonstrate that clusters with these desired properties can be best obtained by generating syntax input and output clusters. The following section will then formalize the mechanism which produces syntax input clusters.

An ideal PF input cluster, in which patterns of subject ambiguity can be evaluated, should contain forms which are lexically identical, which have different argument structures, and which are possible syntax outputs. The first condition guarantees that the forms in the cluster do in fact threaten to be ambiguous; the second guarantees that forms demonstrating the subject contrast relevant to PRCONTR(Subj) are included in the cluster; the third guarantees that irrelevant forms are not compared. These desiderata are motivated below.

First, in determining which forms threaten to merge, a crucial observation is that forms which are lexically identical but structurally distinct are often ambiguous, as in the pair (29a,b). Forms like (29b,c), which are structurally identical but lexically distinct, however, are under no threat of merger.

(29) a. Sumiko-to [Jiroo-no okaasan]
Sumiko-CONJ Jiroo-GEN mother
'Sumiko and [Jiroo's mother]'

b. [Sumiko-to Jiroo-no] okaasan
'[Sumiko and Jiroo's] mother'

c. [Taroo-to Hanako-no] otoosan
Taroo-CONJ Hanako-GEN father
'[Taroo and Hanako's] father'

In order to create a PF input cluster containing forms which are likely to merge, therefore, the members of this input cluster should be lexically identical to each other.

While the PF input cluster contains inputs formed of the same words, these inputs must differ in their argument structures. The impetus for this requirement follows directly from the evidence that ambiguity is prevented in Japanese between inputs with different subjects, i.e. different argument structures. In order to provide PRCONTR(Subj) with relevant forms among which problematic subject ambiguity can be detected and penalized, the cluster should contain lexically identical inputs whose argument structures vary.

Finally, PF input clusters should be composed of forms which are possible winners of the preceding syntax evaluation. This is because PF forms should be compared only with other forms with which they might actually merge. Therefore if a particular structure or movement is banned in Japanese syntax evaluations – for example, Japanese doesn't allow movement out of complex NPs – PF inputs in which that structure or movement occurs should not be considered. That is, if a hypothetical form that had movement out of a complex NP became ambiguous with another form and caused a fatal violation of PRCONTR(Subj), this violation would be meaningless in reality, as one of the forms which gave rise to it can in fact never surface in Japanese, and so the ambiguity would never actually exist in reality. Such meaningless comparisons should be avoided, and easily can be if all members of a PF input cluster can be guaranteed to be actual winners of syntax evaluations.

A straightforward way of producing a cluster of possible syntax winners which are lexically identical but have different argument structures is to create a syntax input cluster of lexically identical syntax inputs with different argument structures. I assume that syntax inputs are essentially numerations with argument structure; manipulation of syntax inputs therefore allows direct access to, and manipulation of, argument structures. A detailed proposal for generating sets of syntax inputs is offered in the next section; once this syntax input cluster is generated, Gen can produce syntax output candidate clusters (as it did in PF; see section 3.1), Eval can choose a winning syntax output cluster, and this output cluster can become a PF input cluster which consists, as desired, of lexically identical forms with different argument structures – crucially, all of which are necessarily actual possible syntax winners.

3.3.2. The source of syntax input clusters: CGen

It has now been established that a syntax input cluster composed of forms which are lexically identical but have different argument structures can be used to generate an appropriate PF input cluster, which will allow PRCONTR(Subj) to detect and penalize subject ambiguity. A crucial question at this point regards the source of this syntax input cluster. As this is the initial input to OT evaluation, it needs to be produced by some component of the grammar which does not have a correlate in a single-input model of OT. I will propose a mechanism which can generate an appropriate cluster of syntax inputs given a single form. As this mechanism is somewhat similar to Gen in that it finds forms related to a single form (though CGen finds related inputs, while regular Gen finds possible output candidates for an input), it will be called CGen as it generates clusters.

Input cluster generation begins with a 'base' syntax input: the input whose output is sought by the evaluation (e.g. because this is the output the speaker intends to produce). CGen begins with this syntax input and creates a cluster of inputs which threaten to merge with its output form; as established above, these related inputs are lexically identical but have different argument structures. In order to create this set, CGen must identify the lexical component of the base, then combine these words in all possible well-formed argument structures (that is, all argument structures in which theta rules are fully saturated). Placing the responsibility for producing well-formed input clusters in CGen renders CGen again similar to Gen, as Gen is generally considered to produce output candidates which adhere to cross-linguistic principles of well-formedness (phonological forms are structured according to the prosodic hierarchy; syntactic structures satisfy the case filter, etc.). This cluster will contain inputs which, due to their varying argument structures, demonstrate the subject contrast relevant to PRCONTR(Subj).

As an example, we can build an input cluster for a double nominative sentence whose base input is in (30).

(30) [kowa-PRES <Agent:Hanako, Theme:Taroo>]

The lexical items {*kowa*, *Hanako*, *Taroo*} in this base should be recombined into all possible argument relationships. In this case, since the verb *kowa* assigns two theta roles and there are two possible arguments present, the cluster contains the two forms in (31).

(31) { [kowa-PRES <Agent:Hanako, Theme:Taroo>],
[kowa-PRES <Agent:Taroo, Theme:Hanako>] }

I will assume, conservatively, that any functional morphology present in the base input (e.g. *PAST*, here) should remain associated with its lexical host in all cluster members. If evidence were found that ambiguity between syntax inputs with different functional morphology could trigger a violation of PRCONTR(Subj), CGen could be enhanced to create larger clusters with a variety of functional morphemes.

The resulting cluster in (31) for the simple input in (30) is quite small; while sentences with more words could create larger clusters, the clusters built by this argument-structure-rearranging algorithm will be much smaller than the set of all possible inputs. Crucially, clusters will always contain a finite number of forms, and so evaluation of clusters produced by CGen will necessarily be computable.

3.3.3. 'Inheritance' of PF input clusters from syntax

When CGen creates forms with all possible argument structures given a set of lexical items, as described in the previous section, the resulting syntax input cluster includes forms corresponding to *Taroo is afraid of Hanako* as well as *Hanako is afraid of Taroo*, capturing the crucial subject contrast. As shown in (32) (repeated from above), though, the potential source of subject ambiguity in Japanese is a pair of forms which differ in both their subjects and also in whether scrambling has occurred or not.

(32) *"Taroo-ga$_i$ Hanako-ga t$_i$ kowa-i." = "Taroo-ga Hanako-ga kowa-i."
T-NOM H-NOM fear-PRES T-NOM H-NOM fear-PRES
*'Hanako fears Taroo.' 'Taroo fears Hanako.'

The PF input cluster must therefore include both forms where scrambling has occurred and also forms where it has not. While the subject difference is one which follows from argument structure and is therefore encoded in the syntax input cluster, scrambling is an optional syntactic process, and forms which scramble do not have different inputs from those which do not.

Optionality is generally conceived of in OT as the possible pairing of a single input with more than one output; for various theories of optionality, see Anttila (1997), Boersma (1997, 1998), Boersma and Hayes (2001), Prince and Smolensky (1993). The table in (33) demonstrates the optional mapping of syntax inputs with discourse topic objects to either scrambled (33b,d) or unscrambled (33a,c) syntax outputs.

(33)

	Syntax input	Syntax output/PF input	PF output
a.	kowa-PRES <H, T>	→ H-NOM T-NOM kowa-PRES	→ H-ga T-ga kowa-i
b.		→ T-NOM$_i$ H-NOM T-NOM$_i$ kowa-PRES	→ ~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i
c.	kowa-PRES <T, H>	→ T-NOM H-NOM kowa-PRES	→ T-ga H-ga kowa-i
d.		→ H-NOM$_i$ T-NOM H-NOM$_i$ kowa-PRES	→ ~~H-ga~~$_i$ T-ga H-ga$_i$ kowa-i

As described above, each OT evaluation chooses a single winning output cluster, and each output cluster contains a single output correspondent of each member of the input cluster. In order for a single syntax input like *kowa-PRES <H, T>* to be optionally paired with either an unscrambled syntax output *H-NOM T-NOM kowa-i* or a scrambled syntax output *T-NOM$_i$ H-NOM T-NOM$_i$ kowa-i*, there must therefore be two possible winning syntactic output clusters. That is, there must be one cluster of unscrambled forms (34a) which wins in some syntax evaluations, and another of scrambled forms (34b) which wins in other syntax evaluations. In order to compare the scrambled and unscrambled forms, these two winning syntax clusters must be merged into a single PF input cluster – composed of all possible syntax winners – as in (34c).

(34) a. Winning syntax cluster of scrambled forms:

{ [T-NOM$_i$ H-NOM T-NOM$_i$ kowa-PRES],
[H-NOM$_i$ T-NOM H-NOM$_i$ kowa-PRES] }

b. Winning syntax cluster of unscrambled forms:

{ [H-NOM T-NOM kowa-PRES],
[T-NOM H-NOM kowa-PRES] }

c. PF input cluster, composed of the union of possible winning syntax clusters:

{ [T-NOM$_i$ H-NOM T-NOM$_i$ kowa-PRES],
[H-NOM$_i$ T-NOM H-NOM$_i$ kowa-PRES],
[H-NOM T-NOM kowa-PRES],
[T-NOM H-NOM kowa-PRES] }

The exhaustive inclusion of all possible winners in a PF input cluster is consistent with the goal stated above, that an input cluster contain all forms which might possibly merge with the base in any given evaluation.

3.3.4. Summary of the model

This section has determined that, in the cluster-based model of OT necessary for evaluating patterns of subject ambiguity, PF input clusters should consist of the union of all possible winning syntax output clusters. Syntax operates on clusters, beginning with an input cluster which is produced by CGen given some base input. In both syntax and PF evaluations, Gen takes an input cluster and produces a set of candidate output clusters, and Eval determines which cluster (or clusters) win given the constraint hierarchy. A detailed example of the model of cluster formation and evaluation developed here is given in (35).

(35) **SYNTAX: Input cluster generation**

Base syntax input: kowa-PRES <Agent: Hanako, Theme: Taroo>

CGen creates a cluster of syntax inputs with different argument structures:

{ [kowa-PRES <Agent: Hanako, Theme: Taroo>]$_1$,
[kowa-PRES <Agent: Taroo, Theme: Hanako>]$_2$ }

SYNTAX: Gen and Eval

Gen produces syntax output correspondents of each input:

[kowa-PRES <Agent: H, Theme: T>]$_1$ → [H-NOM T-NOM H-NOM kowa-PRES]$_{1A}$
→ [T-NOM H-NOM kowa-PRES]$_{1B}$

[kowa-PRES <Agent: T, Theme: H>]$_2$ → [T-NOM H-NOM T-NOM kowa-PRES]$_{2A}$
→ [H-NOM T-NOM kowa-PRES]$_{2B}$

Gen builds unique syntax output clusters:

{[H-NOM T-NOM H-NOM kowa-PRES]$_{1A}$, [T-NOM H-NOM T-NOM kowa-PRES]$_{2A}$};
{[H-NOM T-NOM H-NOM kowa-PRES]$_{1A}$, [H-NOM T-NOM kowa-PRES]$_{2B}$};
{[T-NOM H-NOM kowa-PRES]$_{1B}$, [T-NOM H-NOM T-NOM kowa-PRES]$_{2A}$};
{[T-NOM H-NOM kowa-PRES]$_{1B}$, [H-NOM T-NOM kowa-PRES]$_{2B}$}

Eval chooses (one or more) syntax winners:

➔ {[H-NOM T-NOM H-NOM kowa-PRES]$_{1A}$, [T-NOM H-NOM T-NOM kowa-PRES]$_{2A}$};
{[H-NOM T-NOM H-NOM kowa-PRES]$_{1A}$, [H-NOM T-NOM kowa-PRES]$_{2B}$};
{[T-NOM H-NOM kowa-PRES]$_{1B}$, [T-NOM H-NOM T-NOM kowa-PRES]$_{2A}$};
➔ {[T-NOM H-NOM kowa-PRES]$_{1B}$, [H-NOM T-NOM kowa-PRES]$_{2B}$}

PF: Input cluster generation

The PF input cluster is the union of winning syntax output clusters:

{ [H-NOM T-NOM H-NOM kowa-PRES]$_1$,
[T-NOM H-NOM T-NOM kowa-PRES]$_2$,
[T-NOM H-NOM kowa-PRES]$_3$,
[H-NOM T-NOM kowa-PRES]$_4$ }

(only the crucial contrast between forms 1 and 4 will be considered below)

PF: Gen and Eval

Gen produces PF output correspondents of each member of the input cluster:

[T-NOM$_i$ H-NOM T-NOM$_i$ kowa-PRES]$_1$ → [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$
→ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$

[H-NOM T-NOM kowa-PRES]$_4$ → [T-ga H-ga kowa-i]$_{4A}$,
→ [T-ga H-∅[12] kowa-i]$_{4B}$

[12] Recall from section 2.4.2 that dropping the object case particle is an alternative hypothetical output form of this PF input.

Gen builds unique PF output clusters:

{ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-ga kowa-i]$_{4A}$ };
{ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-∅ kowa-i]$_{4B}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-ga kowa-i]$_{4A}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-∅ kowa-i]$_{4B}$ }

Eval chooses the PF winner:

➔ { [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-ga kowa-i]$_{4A}$ };
{ [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_{1A}$, [T-ga H-∅ kowa-i]$_{4B}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-ga kowa-i]$_{4A}$ };
{ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_{1B}$, [T-ga H-∅ kowa-i]$_{4B}$ }

3.4. Consequences of this model: Winning output clusters are composed entirely of winning outputs

This section has demonstrated that clusters of inputs and outputs can be constructed and evaluated with relatively minimal changes to the OT architecture. The cluster-based architecture proposed here is still significantly different from the single-input model of OT in which many successful syntactic analyses (see e.g. Barbosa et al. 1998, Legendre, Grimshaw, and Vikner 2001) have been developed. This shift to a cluster-based architecture is therefore a relatively unappealing move unless analyses developed in a single-input architecture still hold in this cluster-based model. The following discussion will demonstrate that this is the case; this result will follow from the observation that a winning output cluster is composed of individual winning outputs. That is, every member of a winning cluster is an actual output which surfaces in the language; in order to prevent ambiguity, outputs cannot be chosen one at a time, but rather a set of (satisfactorily) unambiguous outputs must all be chosen at once. This means that a number of distinct base inputs can give rise to the same evaluation, and the same winning output cluster. That is, a single evaluation determines the winning output correspondents of a number of inputs.

From another perspective, the guarantee that a winning output cluster in a given evaluation is necessarily composed of actual winning output forms of each member of the input cluster is important in guaranteeing that appropriate patterns of contrast preservation and neutralization are taken into account in each evaluation. This will also be discussed below. Finally, the observation that various base inputs give rise to the same evaluation prompts a discussion of the formal status of bases, and of whether evaluations must necessarily begin with some individual base.

3.4.1. Previous results still hold in a cluster-based architecture

In order for syntactic analyses developed in a single-input model of OT to still hold in this cluster-based model of OT, a winning cluster must include those outputs which would appropriately win in single-input evaluations – except in evaluations which could not be successfully handled in a single-input model, like the Japanese case discussed here which crucially depends on evaluation of clusters by PRCONTR(Subj). That is, except in analyses where PRCONTR(Subj) is necessary, cluster-based models and single-input models should produce the same winners. In order to show this, I will first examine what it means to be a winning output in a single-input evaluation, and then will show that such outputs are members of the winning cluster in a cluster-based model.

An analysis in which PRCONTR(Subj) is necessary is one in which PRCONTR(Subj) rules out output candidates which would otherwise (inappropriately) win. As has been discussed through this paper, such an analysis (as of Japanese double nominatives) is one where the appropriate output cannot be determined without comparison to related forms, and therefore can only be evaluated in a cluster-based model. An analysis in which PRCONTR(Subj) is not necessary, then – one which would hold in a single-input model of OT – is one in which PRCONTR(Subj) is not needed to rule out a likely winner. In other words, analyses which hold in a single-input model are those in which markedness and faithfulness alone determine winning outputs. PRCONTR(Subj) may be trivially satisfied if the winning outputs do not generate ambiguity, or it may be violated if ambiguity does result (a situation in which PRCONTR(Subj) is shown to be violable is discussed in section 4.1); crucially, however, the markedness and faithfulness portion of the constraint hierarchy alone can determine the winning outputs in these cases.

It can therefore be shown that (appropriate) single-input winners still win in cluster-based evaluations. Take some input whose output correspondent can be chosen using only markedness and faithfulness constraints. In a cluster-based model of OT, an input cluster will be generated for this base. The goal is to show that the winning output cluster, which contains output correspondents of each member of the input cluster, will contain the same output for each input as would be chosen by individual single-input evaluations. This can be demonstrated as follows.

Each output cluster contains a correspondent of each member of the input cluster; the complete set of output clusters contains all combinations of all possible output correspondents of each member of the input cluster. In a single-input evaluation, each input in the input cluster had some optimal output correspondent, which could be chosen based on violations of markedness and faithfulness constraints. Based on the markedness and faithfulness constraints, then, each member of the input cluster has some most harmonic output correspondent. There is necessarily some output cluster which is composed entirely of these most harmonic (with respect to markedness and faithfulness) output correspondents of each input; this is the cluster that should win, to allow the cluster evaluation to mimic the single-input evaluation. This is the case, as follows.

As any additional PRCONTR(Subj) constraint violations incurred by this cluster will be irrelevant to determining the winning output (i.e. it is either trivially satisfied, or relatively low-ranked and thus violated; we know that it must be irrelevant if markedness and faithfulness alone can determine the winners), the cluster which is composed of individual winning outputs will be the winner of the cluster-based evaluation. Therefore, all outputs which could appropriately win in single-input evaluations will still win in cluster-based evaluations. The shift to a cluster architecture simply adds additional power which can be used when comparison of multiple input-output pairs is crucial in determining winners as well.

3.4.2. Meaningful patterns of contrast preservation and neutralization are evaluated

There is another reason why it is crucial that winning output clusters in this cluster-based model are composed entirely of actual winning outputs; this reason concerns the patterns of contrast preservation and neutralization determined by the ranking of PRCONTR(Subj). If an output cluster identified as optimal in one evaluation turned out to include output forms which did not surface in the language, this would be problematic, as it could be the case that the winning cluster wouldn't have any problematic ambiguity, but the actual winning output forms in the language might in fact have this ambiguity. It must therefore be true that a winning output cluster is composed entirely of winning outputs, in order to ensure that meaningful patterns of contrast preservation and neutralization are evaluated.

This can be illustrated using the schematic tableau in (36), where properties of inputs and outputs are summarized below the tableau. Here, the input base is /A/ and its possible output correspondents are [A_1] and [A_2]; /B/ is also member of the input cluster, and its possible output correspondents are [B_1] and [B_2]. This evaluation determines that the output correspondent of /A/ is [A_1], and also that /B/ surfaces as [B_1]. /A/ and /B/ have different subjects and their possible output correspondents [A_2] and [B_1] are identical; therefore, the cluster {[A_2], [B_1]} incurs a fatal violation of high-ranked PRCONTR(Subj). Crucially for the point being demonstrated here, /B/ is prevented from surfacing as [B_1] in order to avoid ambiguity. Given that [B_2] violates high-ranked FAITH(X) and [A_1] violates lower-ranked FAITH(Y), the winning output cluster is the unambiguous cluster of forms {[A_1], [B_1]}.

(36)

/A, B/	PRCONTR(Subj)	FAITH(X)	FAITH(Y)
a. ➔ [A_1], [B_1]			*
b. [A_1], [B_2]		*!	*
c. [A_2], [B_1]	*!		
d. [A_2], [B_2]		*!	

Base = /A/
/A/, /B/ have different subjects
[A_2], [B_1] are identical, but faithful and unmarked
[A_1] violates FAITH(Y)
[B_2] violates FAITH(X)

An evaluation with /A/ as the base therefore chose an output form for /A/ – [A_1] – and also an output form for /B/ – [B_1] – such that the output correspondents of /A/ and /B/ are not identical. The ambiguity avoidance patterns chosen in this evaluation would be meaningless, however, if a separate evaluation with /B/ as the base could choose [B_2] (which is ambiguous with the output form previously chosen for /A/) as the most harmonic output correspondent of /B/. If this were possible, the language would have chosen output [A_1] in an /A/-based evaluation believing that this output was unambiguous; another /B/-based evaluation would, however, have chosen an identical output [B_1] for a distinct input /B/, producing the ambiguity which the evaluation of /A/ strove to avoid. In order for evaluation of ambiguity in clusters to be meaningful – in order to guarantee that any evaluation of /A/ or /B/ will pair these inputs with the same unambiguous outputs [A_1] and [B_2] – we must prove that all evaluations in which /B/ is a member of the input cluster select [B_2] as the output correspondent of /B/.

It can be proven that input cluster {/A/, /B/} will always be paired with the unambiguous output cluster {[A_1], [B_2]} as follows. As was discussed in the preceding sections, Gen will always produce the same output candidate clusters for a given input cluster; similarly, the fixed constraint ranking in this language will always assign the same violations to these clusters and will always choose the same winning cluster. Therefore the only threat to the /B/ – [B_1] pairing would be an evaluation where /B/ is a member of an input cluster other than {/A/, /B/} and thus its patterns of merger with /A/ could be ignored, or its patterns of merger with other inputs could be considered. This will never be the case, as follows.

As established in section 3.3.1-2, input clusters are composed of all inputs which are lexically identical to the base. The base /A/ generated an input cluster composed of all inputs which are lexically identical to /A/, including /B/. Any member of this cluster could in fact be the base and the same cluster would be generated, as the single criterion for shared cluster membership is lexical identity. Therefore, any cluster which contains /A/ will contain /B/ as well, and vice versa, as /A/ and /B/ are lexically identical. /B/ can thus never be a member of an input cluster which does not include /A/, or which includes members other than those in the original {/A/, /B/} cluster. As this input cluster will always choose the same winning output cluster {[A_1], [B_2]}, we can trust that the patterns of contrast preservation and neutralization considered in a given evaluation are in fact meaningful in the language as a whole.

3.4.3. The formal status of bases

The preceding discussion shows that it is not ultimately necessary to identify a single base from which an input cluster is generated. Any member of a given input cluster could be considered the base; the same input cluster would still be produced, and the same output cluster would be chosen, and so each member of the input cluster would still be paired with the same output form. Evaluation of clusters crucially identifies a most harmonic output cluster given a particular input cluster. Designating varying members of the input cluster as the base does not change the choice of a winning output cluster.

While the mechanism of input cluster generation considered here (CGen) makes crucial reference to a base, input clusters could be generated without a base. For example, a list of all possible combinations of lexical items could be compiled from the lexicon. CGen could then construct an input cluster for each set of lexical items, and the most harmonic output correspondent of any possible input could thus be determined. Such a model of cluster generation and evaluation could therefore establish the same relationships between inputs and outputs without using a base.

Despite the fact that bases are formally not crucial, reference to a base is nevertheless a useful expository device. There is often a particular input with whose output correspondent we are particularly concerned; assigning this member of the input cluster special status as a base allows us to clearly discuss this form, as well as other inputs which threaten to merge with it and thus affect the choice of an output correspondent for the base. Assuming a base also allows us to discuss evaluations in this cluster-based architecture in a manner similar to previous discussions of evaluations with single input forms, which were concerned with finding the output correspondent of a single input in a given evaluation.

The shift from a single-input architecture to a cluster-based architecture of OT evaluation is a major one, and no major changes to the architecture of a grammar should be undertaken lightly. This section has demonstrated, however, that the consequences of this shift can be exhaustively described and as such are relatively minor elaborations on the known components of an OT grammar. Further, this change allows the theory to explain a greater number of phenomena than before; we can now account for anti-ambiguity effects, while allowing previous analyses of grammatical phenomena which were developed in a single-input model of OT to remain valid in a cluster-based architecture.

4. Predictions of the theory

The inclusion of the transderivational PRESERVECONTRAST(Subject) constraint in an OT grammar makes a number of predictions. Like other OT constraints, PRCONTR(Subj) should be violable; that is, it should be possible for a subject contrast to be neutralized (in some or all contexts in a language) if some other constraint outranks PRCONTR(Subj), preventing a repair of subject contrast neutralization. Further, as PRCONTR(Subj) penalizes subject ambiguity without making reference to the source of this ambiguity, potential subject ambiguity in Japanese from sources other than double nominative constructions should also cause word order freezing. This section will show that both of these predictions are borne out in Japanese. It will also discuss other predictions made by this theory which are left for future research.

4.1. Focus-driven scrambling: PRCONTR(Subj) is violable

Word order freezing in Japanese double nominative sentences is the consequence of the constraint ranking PRCONTR(Subj) » MAX(Head). Given that all constraints in an OT grammar are potentially violable, it is possible that some conflicting constraint could dominate PRCONTR(Subj), causing the grammar to tolerate subject ambiguity in particular contexts. This tolerance of ambiguity is found in Japanese focus scrambling contexts.

The scrambling discussed throughout this paper has involved discourse topic objects surfacing at the front of sentences. This sort of scrambling fails to occur in double nominative sentences, as doing so would cause a fatal violation of PRCONTR(Subj); when case fails to distinguish arguments in this construction, word order takes over. There is another type of scrambling in Japanese in which a focused object moves to the front of the sentence. When focused objects, as opposed to discourse topic objects, scramble in double nominative sentences, ambiguity can in fact arise. For example, if a double nominative sentence is a question in which the object *wh*-word is focused, or if it is the answer to such a question, the object can be fronted and receives focus intonation.[13] As shown in (37), where underlining indicates phonological focus, such a sentence can have either a scrambled or unscrambled reading.

(37) a. Dare-ga Hanako-ga kowa-i?
who-NOM Hanako-NOM afraid.of-PRES
'Who is Hanako afraid of?'
'Who is afraid of Hanako?'

b. Taroo-ga Hanako-ga kowa-i.
Taroo-NOM Hanako-NOM afraid.of-PRES
'Hanako is afraid of Taroo.'
'Taroo is afraid of Hanako.'

When scrambling is the result of focus, the head of the scrambling chain must be pronounced at PF despite the fact that this neutralizes a contrast between sentences with different subjects.[14] The input which means *Hanako is afraid of Taroo* is paired with the output *Taroo-ga Hanako-ga kowa-i*, despite the fact that this output is also the optimal correspondent of the input which means *Taroo is afraid of Hanako*. The winning output cluster which contains these two identical output correspondents of different inputs will therefore violate PRCONTR(Subj); the fact that it wins anyway means that some other conflicting constraint must be ranked above PRCONTR(Subj).

The way in which focus licenses ambiguity can be accounted for as follows. The head of a scrambling chain must surface, in a context where this would cause a violation of PRCONTR(Subj), specifically when the scrambling is motivated by a [Focus] feature. Such a pattern where an element (e.g. the head of a chain) is faithfully realized when it is in a particular prominent context (e.g. when it is focused) often follows from a high-

[13] While focus intonation distinguishes between e.g. focused and unfocused subjects, subject ambiguity still results despite this focus intonation, as the intonational patterns characteristic of focused subjects and focused scrambled objects are identical; therefore an initial nominative noun in a double nominative sentence can be either a focused subject or a scrambled focused object.

[14] It seems quite unusual that focus can cause ambiguity, as focus is often thought of as a disambiguator. This should not be too startling, though, for the following reason. A sentence where focus has licensed scrambling and thus caused ambiguity, as in *Taroo-ga*$_i$ *Hanako-ga* t_i *kowai* 'Hanako is afraid of Taroo' is most natural in response to a question like 'Who is Hanako afraid of?' In such a context, only one of the two possible interpretations of the string *Taroo-ga Hanako-ga kowai* is sensible. In most contexts where focus would drive scrambling, there should be a similar strong contextual preference for one argument structure over another; this preference is less strong in contexts where the discourse topic status of an object may – or may not – compel scrambling. So while focus licenses ambiguity, it does not generate a pair of contextually plausible alternatives and so really generates only grammatical, rather than pragmatic, ambiguity.

ranking positional faithfulness constraint (Beckman 1998); in this case, the pattern is the result of high-ranking MAX(Head)/[Focus].

(38) MAX(Head)/[Focus] Do not delete the head of a chain when it bears a [Focus] feature.

The constraint ranking MAX(Head)/[Focus] » PRCONTR(Subj) » MAX(Head) produces the desired output cluster, as shown in (39).

(39)

[T-NOM$_{[Foc]i}$ **H-NOM** T-NOM$_{[Foc]i}$ kowa-PRES]$_1$ [**T-NOM**$_{[Foc]}$ H-NOM kowa-PRES]$_2$	MAX (Head)/ [Focus]	PRCONTR (Subj)	MAX (Head)
a. [~~T-ga~~$_i$ H-ga T-ga$_i$ kowa-i]$_1$ [T-ga H-ga kowa-i]$_2$	*!		*
b. ➔ [T-ga$_i$ H-ga ~~T-ga~~$_i$ kowa-i]$_1$ [T-ga H-ga kowa-i]$_2$		*	

In form *1* in output candidate cluster (39a), the tail of the scrambling chain is pronounced, rendering the output forms *1* and *2* distinct. PRCONTR(Subj) is satisfied, and so this candidate would ordinarily win. Because the scrambled object bears a [Focus] feature, however, deletion of the head of the chain incurs a fatal violation of MAX(Head)/[Focus]. (39b) is therefore the winner, despite the fact that pronunciation of the head of the scrambling chain causes forms *1* and *2* to be ambiguous.

The Japanese ranking MAX(Head)/[Focus] » PRCONTR(Subj) also explains why long-distance scrambling can cause ambiguity. Hirotani (2000) and Kitagawa (1990) note that while focused phrases and discourse topics may both be scrambled within a clause, only focused phrases may undergo long-distance scrambling, which crosses clause boundaries. Given this restriction, the sentence in (40) is ambiguous as it can have either of the structures in (41a,b), where *Taroo-ga* is either in situ or scrambled out of the embedded clause due to focus. The structure in (41c) is ungrammatical, however, as the discourse topic *Taroo-ga* cannot scramble out of the embedded clause.

(40) Taroo-ga Hanako-ga kinoo warat-ta-to it-ta
Taroo-NOM Hanako-NOM yesterday laugh-PAST-COMP say-PAST

(41) a. Taroo-ga$_{[Foc]i}$ [Hanako-ga [t$_i$ kinoo warat-ta]-to it-ta]
'Hanako said that it was Taroo who laughed yesterday'

b. Taroo-ga$_{[Foc]}$ [Hanako-ga kinoo warat-ta]-to it-ta
'Taroo said that Hanako laughed yesterday'

c. *Taroo-ga$_i$ [Hanako-ga [t$_i$ kinoo warat-ta]-to it-ta]
'Speaking of Taroo, Hanako said that Taroo laughed yesterday'

As in the double nominative sentences discussed above, subject ambiguity can only result from focus-driven scrambling; the connection to long-distance scrambling is the result of the inherent connection between focus and long-distance scrambling.

Ambiguity in long-distance scrambling, like ambiguity in single-clause sentences where scrambling is driven by focus, is therefore permitted by the constraint ranking MAX(Head)/[Focus] » PRCONTR(Subj) » MAX(Head). This is shown in tableau (42).

(42)

[T-NOM$_{[Foc]i}$ [H-NOM [T-NOM$_{[Foc]I}$ kinoo warat-PAST]-COMP it-PAST]$_1$ [T-NOM [H-NOM kinoo warat-PAST]-COMP it-PAST]$_2$	MAX(Head)/ [Focus]	PRCONTR (Subj)	MAX (Head)
[~~T-ga~~$_i$ [H-ga [T-ga$_I$ kinoo warat-ta]-to it-ta]$_1$ a. [T-ga [H-ga kinoo warat-ta]-to it-ta]$_2$	*!		*
➔ [T-ga$_i$ [H-ga [~~T-ga~~$_I$ kinoo warat-ta]-to it-ta]$_1$ b. [T-ga [H-ga kinoo warat-ta]-to it-ta]$_2$		*	

The results of this section can be generalized to make a broader point about the violability of PRCONTR(Subj): like all other constraints in an OT grammar, this contrast preservation constraint is violable. Just as high-ranking PRCONTR(Subj) can force violation of lower-ranked constraints (resulting in more marked outputs), constraints which outrank PRCONTR(Subj) can force its violation, resulting in subject ambiguity.[15] This section has demonstrated that subject ambiguity occurs when MAX(Head)/[Focus] outranks PRCONTR(Subj). While the general correlation between morphological ambiguity and fixed word order is enforced by relatively high-ranking PRCONTR(Subj), just as some languages fail to disambiguate arguments, focus contexts within Japanese fail to prevent this sort of ambiguity as well.

4.2. Case particle drop: Subject contrast preservation beyond double nominative sentences

As has been shown above, when a double nominative verb fails to assign unique cases to its subject and object, the constraint ranking PRCONTR(Subj) » MAX(Head) causes the order of these arguments becomes fixed in order to prevent ambiguity (unless scrambling is driven by focus). In other situations in which case does not distinguish between subject and object, then, these constraints should also block scrambling in order to disambiguate.

[15] Another case of subject ambiguity is found in sentences with null anaphora, e.g. *Taroo-ga kowa-i*, which can mean either 'Taroo is afraid of him' (*Taroo-ga pro kowai*) or 'He is afraid of Taroo' (*pro Taroo-ga kowa-i*) – the subject of this sentence can be either the overt *Taroo* or the null anaphor. It seems likely that this is tolerated due to a lack of possible PF repairs. That is, Japanese generally resolves subject ambiguity by pronouncing the tail of a chain, but because ambiguity here results from the fact that one argument is entirely null, there is no chain-resolution repair available. As nothing can be added to a sentence at PF that was not present in the input, there is nothing that could be spelled out differently in one member of this ambiguous pair to disambiguate. PRCONTR(Subj) is therefore apparently violable both when a higher-ranked constraint forces its violation and also when there is simply no available repair for ambiguity.

Another source of morphologically ambiguous subjects and objects in Japanese is the stylistic process of case particle drop. In colloquial speech, it is possible to drop case particles from subjects, objects, or both, as in (43).[16]

(43) a. Hanako-ga Taroo-o osore-ru
Hanako -NOM Taroo-ACC fear-PRES
'Hanako fears Taroo.'

b. Hanako-ga Taroo-∅ osore-ru

c. Hanako-∅ Taroo-o osore-ru

d. Hanako-∅ Taroo-∅ osore-ru

There is no threat of ambiguity when a single particle is dropped from a nominative-accusative sentence, as the argument which retains its particle is still unambiguously identified as either subject or object. Scrambling is therefore possible, as shown in (44).

(44) a. Taroo-$\varnothing_i$ Hanako-ga t_i osore-ru

b. Taroo-o_i Hanako-∅ t_i osore-ru

When particles are dropped from both arguments, however, there is no overt morphological indication of which noun is the subject and which is the object. If scrambling were allowed in such a situation, the distinct structures in (45) would have identical surface forms. Just as scrambling is blocked in morphologically ambiguous double nominative sentences, scrambling is also blocked just in these morphologically ambiguous sentences where both subject and object case particles are dropped.[17]

(45) a. Hanako-ga Taroo-o osore-ru → Hanako-∅ Taroo-∅ osore-ru

b. Hanako-o_i Taroo-ga t_i osore-ru → *Hanako-$\varnothing_i$ Taroo-∅ t_i osore-ru

As in double nominative sentences, word order freezing in case particle drop contexts also follows from the constraint ranking PRCONTR(Subj) » MAX(Head).

---

[16] Case particle drop is subject to a number of discourse conditions. Accusative *–o* drops more freely than nominative *–ga*, and *–ga* drops most easily when the speaker is using her full name to refer to herself in subject position; *–ga* drop is reportedly most common among young female speakers of Japanese. (43c,d) are therefore most plausible as statements made by Hanako about herself. It is relatively difficult for objects without case particles to scramble, but they can do so in strongly colloquial contexts; an object without accusative *–o* scrambles more easily in a sentence like *Taroo$_i$ ore-ga t_i bokotta yo* 'Taroo, I fucked (him) up, man' than in (44), as the former uses the male-oriented first person pronoun *ore*, the slang verb *bokotta*, and the colloquial particle *yo*, which conspire to make the sentence inherently extremely colloquial. *–ga* and *–o* are the only case particles which can drop; ditransitive sentences therefore never have two case-less objects and thus are never potentially ambiguous.

[17] It should come as no surprise that when case particles drop from double nominative sentences, word order remains frozen. The sentences are morphologically ambiguous whether case particles are present or not, and so the ranking PRCONTR(Subj) » MAX(Head) causes word order to freeze in any case.

(46) [H-ACC$_i$ **T-NOM** H-ACC$_i$ osore-PRES]$_1$ [**H-NOM** T-ACC osore-PRES]$_2$	PRCONTR (Subj)	MAX(Head)
➔ [~~H-∅~~$_i$ T-∅ H-∅$_i$ osore-ru]$_1$ a. [H-∅ T-∅ osore-ru]$_2$		*
[H-∅$_i$ T-∅ ~~H-∅~~$_i$ osore-ru]$_1$ b. [H-∅ T-∅ osore-ru]$_2$	*!	

Note that this result provides evidence for the claim that scrambling is blocked at PF rather than in syntax, in the following way. Scrambling is blocked in nominative-accusative sentences when both case particles would drop (and thus scrambling would cause ambiguity). The decision as to whether to spell out case particles is a PF phenomenon. Therefore, in these sentences, the syntax evaluation cannot 'know' that the case particles will drop – this will only be determined in the subsequent PF evaluation. Scrambling must therefore be blocked (e.g. its effects must be prevented from surfacing) at PF, as that is the only level of evaluation where the relevant information is present. The assumption that scrambling also occurs in syntax in double-nomintive sentences allows a uniform treatment of the identical behavior of these two phenomena.

4.3. Predictions for future research

In this section, I have shown that PRCONTR(Subj) has crucial characteristics of OT constraints. PRCONTR(Subj) is ranked highly Japanese, triggering a general pattern of disambiguation (via word order freezing) in double nominative sentences. Like all constraints, however, PRCONTR(Subj) is violable. This is seen in focus-driven scrambling, where it is dominated by MAX(Head)/[Focus]; ambiguity is thus tolerated in these contexts. PRCONTR(Subj) further penalizes a general state of affairs (subject ambiguity) regardless of its source; the constraint itself does not specifically target double nominative sentences, or instances case particle drop, but rather its effects capture the generalization which holds both within Japanese and also cross-linguistically, that all sources of morphological ambiguity are undesirable and should be prevented if possible.

This analysis also makes other predictions which will be left for future research. It was mentioned in section 2.2 above that, in claiming that double nominative sentences (as well as sentences which have undergone case particle drop) can undergo scrambling in syntax but delete the head of the scrambling chain at PF, the higher copy of the scrambled object should be available for interpretation at LF. While a number of obstacles to investigating this were noted, this analysis does make the prediction that if the practical problems of investigating the properties of these LF chains could be overcome, one should find evidence for the syntactic scrambling which does not surface.

As has also been discussed above, when PRCONTR(Subj) triggers avoidance of subject ambiguity, the means by which this ambiguity is avoided – word order freezing, in Japanese – is determined by the grammar of the language. Word order freezing seems to be a common repair, occurring in Korean, German, and Hindi as well as Japanese (Lee 2001). This analysis nevertheless makes the strong prediction that other languages should repair violations of PRCONTR(Subj) by means other than word order freezing. Alternative repairs could take the form of omitted case particles, as suggested above; it would also be possible for a scrambled object to surface in scrambled position, but leave a disambiguating resumptive pronoun in its base position. It could even be possible that scrambling a morphologically ambiguous argument might require agreement to surface on a verb when it wouldn't otherwise.

Finally, while preservation of the subject contrast has been the focus of this paper as well as most current OT literature on syntactic contrast preservation (e.g. Kuhn 2001, Lee 2001), it is an open question whether there are other contrasts that also give rise to disambiguating phenomena. Section 2.3 offered functional justifications for why subject ambiguity seems likely to be prevented more often than other forms of ambiguity; there may, however, be languages in which e.g. direct and indirect objects are subject to similar anti-ambiguity processes. If this is the case, it would suggest that PRESERVECONTRAST constraints other than PRCONTR(Subject) should be included in the grammar.

5. Alternative analyses of word order freezing

Various theoretical proposals have been made regarding the general phenomenon of ambiguity avoidance processes, as well as the more specific one of word order freezing in Japanese. These proposals fall into four major categories. (1) It has been argued that "anti-ambiguity" processes like word order freezing have are epiphenomenal results of properties of particular constructions, e.g. double nominative case assignment. It has also been argued that the general anti-ambiguity mechanisms which result in word order freezing are encoded as (2) processing strategies, or (3) as universal anti-ambiguity filters on outputs. (4) There has also been a recent attempt to capture anti-ambiguity effects (and thus word order freezing) within OT by using bidirectional evaluation.

This section will contrast these strategies for explaining word order freezing with the contrast preservation analysis proposed in this paper. I will show that unlike these alternative proposals, the present account offers a full description of the effects seen in Japanese, as well as an explicit, fully characterizable description of the necessary modifications to a grammar which includes this anti-ambiguity mechanism.

5.1. Conditions on nominative objects

Tonoike (1980a,b) argues that word order freezing in Japanese follows directly from inherent structural properties of double nominative sentences. He claims that these constructions have a clause boundary between the subject and object; both arguments are therefore in subject positions. This complex structure explains both the assignment of nominative case to both arguments and also the lack of scrambling: scrambling can occur only within a clause (as opposed to topicalization, which is not bounded by a clause), and therefore the "object" cannot move out of its clause to a position beyond the "subject."

I will not discuss the particular structure that he proposes for this construction in any detail, as various substantive objections to this proposal have been raised, e.g. by Kuno (1980a,b). In general, however, Tonoike's proposal is an example of this kind of approach to anti-ambiguity processes, in that it claims that word order freezing in double nominative sentences is simply an accidental result of other properties of the double nominative construction. Proponents of this sort of approach could attempt to find properties of all constructions in which apparent anti-ambiguity processes occur which themselves accidentally prevent ambiguity.

While such an explanation of word order freezing is appealing in that it does not demand the addition of transderivational anti-ambiguity constraints to the grammar, data from Japanese show that such an analysis misses an important generalization. As observed above, colloquial case particle drop patterns show a striking similarity to double nominative constructions: when morphological ambiguity arises as the result of either a double nominative verb or case particle drop, word order freezes. Assignment of nominative case to the objects of stative verbs and stylistic omission of case particles follow from very different grammatical mechanisms; it would be an extraordinary coincidence if the inherent properties of these two unrelated processes both happened to prevent scrambling in the same potentially ambiguous contexts. An analysis in which a single disambiguating process, word order freezing, can be triggered by potential ambiguity from any source allows a unified explanation of these two very similar phenomena which an 'epiphenomenal' account like Tonoike's could not capture.

5.2. Anti-ambiguity parsing strategies

Kuno (1980b: 183) argues that the restrictions on scrambling in Japanese double nominative sentences follow from a "performance-level anti-ambiguity constraint." That is, Kuno claims that there is a universal tendency to avoid scrambling and other syntactic processes when they could result in ambiguity. A similar proposal in which avoidance of ambiguity follows from biases on processing is found in Ruwet (1972), which claims that listeners of French have access to a variety of parsing strategies which allow them to unambiguously interpret sentences which appear to be structurally ambiguous.

As both Kuno and Ruwet comment, effects following from grammatical processes are generally absolute, while those that follow from processing strategies are graded and apply as needed. If Japanese word order freezing truly resulted from a processing bias, scrambling should be possible in cases where e.g. verbal selectional restrictions mean that a double nominative sentence is under no real threat of ambiguity. As was discussed in section 1, however, scrambling in morphologically ambiguous contexts is blocked even when there is such an animacy difference. Data like (47) (repeated from (9), above) therefore indicate that the word order freezing in Japanese follows not from a processing bias but rather from an absolute grammatical restriction.

(47) *tenisu-ga$_i$ Taroo-ga t$_i$ zyoozu-da
tennis-NOM Taroo-NOM good.at-PRES
'Taroo is good at tennis'

5.3. Anti-ambiguity filters

It is often claimed that the grammar inherently prevents movement and deletion operations from creating structural ambiguity; these claims are generally best understood as filters which cause the derivations of ambiguous syntactic structures to crash. The best-known of these is Chomsky's (1981) UG principle of recoverability of deletion, which bans deletion of elements when they cannot be recovered from the remaining surface structure. Similar proposals have been made to account for particular patterns of ambiguity avoidance in both deletion and movement contexts.

Hankamer's (1973: 51) Peripheral Gap Principle is one instantiation of such an anti-ambiguity filter. The Peripheral Gap Principle claims that "[i]f any interpretation is possible for an unacceptably ambiguous structure, it will be that interpretation under which the location of the deletion site is peripheral rather than internal." Woolford's (1986, 1988) Syntactic Mapping principle is similar in spirit, and extends the basic principle of avoiding internal gaps in potentially ambiguous structures to include movement effects as well as deletion. The mapping filter checks output structures to ensure that, in the absence of disambiguating features or morphology, lexical items are associated to structural positions in a strictly unidirectional manner. Such an extension of Chomsky's and Hankamer's deletion-oriented principles would be crucial in explaining Japanese word order freezing in terms of a filter, as this anti-ambiguity process is one involving movement rather than deletion.

While, again, I will not address the details of these proposals (see e.g. section 5 of Hankamer (1973) for problems with the PGP; Woolford's Mapping proposal is not intended to extend to scrambling), these sorts of inviolable filters are an important class of analyses of anti-ambiguity effects. A major problem for such analyses is the entirely acceptable ambiguity found in Texistepec Popoluca, repeated in (48).

(48)	maʔ	d^j-agaʔ	maʃ-d^jaʔa	kaŋ-daʔa
	PERF	3/3-kill	Tomás-BIG.MASC	jaguar-BIG.MASC

'Tomás killed the jaguar'
'The jaguar killed Tomás'

The filters which have been proposed to account for recoverability-oriented phenomena have been considered inviolable cross-linguistic principles which categorically prevent violating structures from surfacing. (48), however, shows that languages vary in whether they ban or permit sentences with ambiguous argument structures. This data demonstrates that any sort of anti-ambiguity principle must be violable; capturing the prohibition on subject-oriented ambiguity in the violable OT constraint PRESERVECONTRAST(Subject) allows this crucial violability.

5.4. Bidirectional OT evaluation

A more recent approach to anti-ambiguity effects has involved a modification of the OT architecture very different from the modification proposed in this paper. There is a growing OT syntax literature (Blutner (2001), Jager (2000), Kuhn (2001); Lee (2001);

Wilson (2001), among others) concerning bidirectional OT evaluation. The bidirectional architecture has often been used to account for anti-ambiguity phenomena like word order freezing in a variety of languages; Lee (2001) offers a bidirectional analysis of word order freezing in Hindi and Korean, and Kuhn (2001) similarly analyzes German word order freezing.

The intuition behind a bidirectional account of ambiguity avoidance is this: ambiguity is problematic because a hearer cannot recover the speaker's intended meaning from an ambiguous sentence. In order to avoid ambiguity, the grammar should ensure that the intended meaning is recoverable from every surface form. This is established by testing every pair of forms and meanings to ensure that the appropriate meaning always generates, and is always recoverable from, the surface form.

More concretely, a bidirectional grammar evaluates possible meaning-form pairs from both production and comprehension perspectives. The production evaluation occurs first, taking a meaning *M* as an input and choosing a form *F* to pair with it. The grammar then uses the same constraint ranking to perform a comprehension evaluation, in which the form *F* is the input and the grammar determines which meaning *M'* is recovered from *F*. If the recovered meaning *M'* is the same as the initial meaning input *M* (i.e. if *M* is identical to *M'*), the meaning-form pair </M/, "F"> is bidirectionally optimal and thus grammatical. If the recovered meaning *M'* and the original meaning *M* are not identical, then the form *F* can never be used to express the meaning *M*, as *M* is not recoverable from *F*.

(49) Bidirectional optimization:

/M/ → PRODUCTION → "F" → COMPREHENSION → /M'/

</M/, "F"> is bidirectionally optimal iff
/M/ → PRODUCTION → "F", and
"F" → COMPREHENSION → /M/

This bidirectional architecture eliminates all ambiguity from languages.[18] As stated above, ambiguity arises when a single form may have either of two meanings; that is, when two meaning-form pairs which use the same form, </M_1/, "F"> and </M_2/, "F">, are both optimal. A unidirectional OT model of syntax evaluation can produce such a situation easily, if the underlying contrast between inputs M_1 and M_2 is one which is neutralized by a particular constraint ranking. As a traditional (non-cluster-based) unidirectional model generally takes into account only a single input-output pair at a

[18] 'Weak' bidirectional OT (Blutner 2001), a common modification of this 'strong' bidirectional model of OT, does not offer a solution to the problem of complete blockage of ambiguity; this is because weak bidirectional OT still demands unique pairings of forms and meanings. While a weak bidirectional model allows non-optimal forms to surface when they are paired with non-optimal meanings (the best form and best recoverable meaning are paired, then the next-best forms and next-best meanings are also paired in subsequent optimizations), this model still demands one-to-one pairings between forms and meanings, and so still prevents all ambiguity unless it is also modified in the manner described below. See Beaver and Lee (2003) for an overview of this and other proposed bidirectional models.

time, ambiguity often results. A bidirectional model is inherently incapable of pairing two meanings with a single form. Given the standard assumption that Eval chooses one and only one most harmonic output with each input, a form input F can be paired with either output M_1 or M_2, but not both. Such a grammar therefore is extremely successful at preventing ambiguity.

Lee (2001) provides a bidirectional analysis of word order freezing in morphologically ambiguous contexts in Korean; the ideas behind her analysis similarly describe word order freezing in Japanese. The crux of her argument is that, in a double nominative sentence, scrambling would pair two meanings with a single form as in (50).

(50) "Taroo-ga Hanako-ga kowa-i"

↙ ↘

/Hanako is afraid of Taroo/ /Taroo is afraid of Hanako/

In a bidirectional model, scrambling is blocked for a meaning like *Hanako likes Taroo* because this meaning is not recoverable from the scrambled surface form *Taroo-ga Hanako-ga kowa-i*. This string is preferentially interpreted as *Taroo is afraid of Hanako*, as this default SOV word order is assumed to hold in the absence of information indicating that scrambling has occurred. Because the initial meaning input and the recovered meaning are not identical, the meaning-form pairing </Hanako is afraid of Taroo/, "Taroo-ga Hanako-ga kowa-i"> isn't bidirectionally optimal; see Lee (2001) for details of the analysis.

(51) /Hanako is afraid of Taroo/

→ PRODUCTION → "Taroo-ga Hanako-ga kowa-i"

→ COMPREHENSION → /Taroo is afraid of Hanako/

NOT → COMPREHENSION → /Hanako is afraid of Taroo/

➔ </Hanako is afraid of Taroo/, "Taroo-ga Hanako-ga kowa-i">
isn't bidirectionally optimal.

Bidirectional OT is very good at preventing ambiguity; in the model described in (51), no ambiguous sentence will ever be grammatical. This is, in fact, a major problem for the model. While languages occasionally block processes (e.g. scrambling) when these would result in ambiguity, it is much more common for languages to allow ambiguous sentences to surface. Japanese usually blocks scrambling in order to prevent ambiguity; however, section 4.2 above showed that ambiguity is allowed to result from scrambling in focus contexts. Japanese also allows many cases of structural ambiguity like (52), repeated from above; see Inoue and Fodor (1995) for more examples of ambiguity in Japanese.

(52) Sumiko-to Jiroo-no okaasan
Sumiko-CONJ Jiroo-GEN mother
'Sumiko and [Jiroo's mother]'
'[Sumiko and Jiroo]'s mother'

The bidirectional system, designed to eliminate all ambiguity, must be modified in order to incorporate these instances of ambiguity. One often-cited proposal for such a modification is that of Asudeh (2001), which derives ambiguity from variable constraint ranking (following Boersma (1997, 1998) and Boersma and Hayes (2001)). In this framework, particular constraints may be reranked with respect to each other across evaluations. For example, two constraints A and B can be ranked either A » B or B » A on a given evaluation. Ambiguity arises when the ranking A » B pairs M_1 with *F* (in both production and comprehension optimizations), and the ranking B » A pairs M_2 with *F*. A single constraint hierarchy which allows this reranking can thus produce both bidirectionally optimal meaning-form pairs </M_1/, “F”> and </M_2/, “F”>; ambiguity can thus be introduced into a bidirectional model.

Variable constraint ranking seems promising as a means of incorporating ambiguity into a bidirectional system. A more thorough assessment of its capabilities awaits a careful investigation of its properties, as research into ambiguity in bidirectional systems is in very early stages. One important point about the results in this area to date is that Asudeh used variable ranking to account for a phenomenon which exhibited both ambiguity and also optionality, where optionality is the pairings of multiple forms with a single meaning, i.e. both </M/, “F_1”> and </M/, “F_2”>. Asudeh’s system therefore produced four possible form-meaning pairs: </M_1/, “F_1”>, </M_2/, “F_1”>, </M_1/, “F_2”>, </M_2/, “F_2”>. He used a pair of constraints whose reranking would produce ambiguity in a comprehension evaluation as well as optionality in a production evaluation; two forms and two meanings therefore could surface in all possible pairings. It remains to be seen whether variably-ranked constraints can also account for cases of ambiguity without optionality and vice versa, as cases of either ambiguity or optionality are much more common cross-linguistically than are cases of symmetrical ambiguity and optionality.

A more general point about the shift to a bidirectional OT architecture in order to account for anti-ambiguity phenomena is the following. Bidirectional OT is a near-complete rethinking of the OT architecture, and while e.g. Kuhn (2001) has demonstrated that bidirectional evaluations are guaranteed to be decidable, there have been few attempts at systematic investigation of the properties of forms and meanings, of inputs and outputs, or of the ways in which particular constraints (e.g. MAX, DEP) may assign violations differently in production and comprehension evaluations. As a result of this, it is difficult, if not impossible, to determine whether successful analyses which have been developed in unidirectional models of OT can still hold in a bidirectional model.

Until these properties of the bidirectional system are more thoroughly investigated, it is very difficult to understand a bidirectional model well enough to confidently allow it to replace the comparably well-understood unidirectional model. The unidirectional cluster-based OT architecture, on the other hand, can be shown to make relatively minor, exhaustively characterizable modifications to the traditional model of OT. It can further be shown that results proven in a single-input model of OT will also hold in a cluster-based model. Further, the ways in which languages choose to tolerate some sorts of ambiguity while blocking others are easily characterizable in the cluster-based model, while they are difficult to capture in bidirectional OT. The degree to which the properties of a cluster-based model of OT can be understood, as well as the ways in which observed patterns of ambiguity can be modeled, make it a more appealing approach to ambiguity resolution than a bidirectional model.

6. Conclusion

This paper has examined the general cross-linguistic correlation between case marking and free word order by investigating the roles that case marking and word order play in preventing ambiguity in Japanese. When ambiguity as to which argument is the subject could result from ambiguous or nonexistent case morphology, the normally free word order of Japanese generally becomes fixed. I have demonstrated that this and other anti-ambiguity processes pose a formal problem for OT, as they cannot be captured in a model of OT where constraints evaluate only single inputs and outputs. This is because ambiguity is a property of a pair of input-output mappings; therefore, in order to allow OT constraints to detect and penalize ambiguity, I have argued that constraints must evaluate sets of inputs and outputs. Further, the high-ranking new PF constraint PRESERVECONTRAST(Subject) penalizes subject-related ambiguity; in Japanese, this triggers the desired word order freezing in double nominative sentences.

This shift to a cluster-based model of OT provides a necessary increase in the explanatory power of OT grammars, while maintaining the fundamental properties of OT. While the new constraint PRCONTR(Subj) evaluates sets of inputs and outputs, it otherwise behaves like a normal constraint. While this constraint fatally penalizes subject ambiguity, the repair for ambiguity is determined by the ranking of other constraints, rather than being stipulated by PRCONTR(Subj) itself. Further, PRCONTR(Subj) also does not make specific reference to the source of subject ambiguity, and so triggers word order freezing whenever subjects and objects are morphologically ambiguous, regardless of the source of this ambiguity. This is crucial in Japanese as word order becomes frozen not only in double nominative sentences but also when stylistic factors cause both subject and object case particles to be dropped in nominative-accusative sentences. The use of a consistent repair for problems from a variety of sources is a hallmark of an OT analysis. Finally, while subject ambiguity is generally avoided in Japanese, it is tolerated when scrambling is driven by focus. This sort of exceptionality to a general pattern can be easily captured in an OT analysis, as constraints – including PRCONTR(Subj) – are violable.

Using a violable constraint to penalize ambiguity, rather than claiming either that processes like word order freezing are not in fact motivated by ambiguity avoidance or that languages rigidly prevent ambiguity from occurring has a number of advantages which allow this approach to more closely capture the facts of Japanese. As mentioned above, two very different sources of potential subject ambiguity both trigger word order freezing. This is difficult to explain without reference to the single shared property of double nominative sentences and stylistic case particle drop: the fact that both can cause morphological ambiguity between subjects and objects. The present approach captures this similarity, while an analysis claiming that word order is frozen because of independent properties of the constructions at hand would necessarily regard this as a mere coincidence. The present approach also easily handles the fact that word order freezing is a general, but not absolute, tendency. The focus-driven scrambling exceptions to word order freezing are difficult to admit into a theory which claims that subject ambiguity is universally avoided on principle. Finally, the use of a constraint which specifically penalizes subject ambiguity allows a ready explanation of how languages prevent this sort of ambiguity while allowing so many others.

References

Anttila, Arto (1997) *Variation in Finnish Phonology and Morphology*. Doctoral dissertation, Stanford University.

Asudeh, Ash (2001) Linking, optionality, and ambiguity in Marathi. In P. Sells (ed.), *Formal and Empirical Issues in Optimality Theoretic Syntax*, pp. 257-312. Stanford: CSLI.

Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis, and David Pesetsky (eds.) (1998) *Is the Best Good Enough? Optimality and Competition in Syntax*. Cambridge, MA: MIT Press.

Beaver, David I. and Hanjung Lee (2004) Input-output mismatches in OT. In R. Blutner and H. Zeevat (eds.) *Optimality Theory and Pragmatics*, pp. 112-153. Hampshire: Palgrave Macmillan.

Beckman, Jill N. *Positional Faithfulness*. Doctoral dissertation, University of Massachusetts, Amherst.

Blutner, Reinhard (2001) Some aspects of optimality in natural language interpretation. *Journal of Semantics* 17: 189-216.

Boersma, Paul (1997) How we learn variation, optionality, and probability. In *IFA Proceedings 21*, pp. 43-58. University of Amsterdam: Institute of Phonetic Sciences. [ROA 221-109]

Boersma, Paul (1998) *Functional Phonology: Formalizing the Interactions Between Articulatory and Perceptual Drives*. The Hague: Holland Academic Graphics.

Boersma, Paul, and Bruce Hayes (2001) Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32: 45-86.

Chomsky, Noam (1981) *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam (1995) *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Dayal, Veneeta (2003) Bare nominals: Non-specific and contrastive readings under scrambling. In S. Karimi (ed.) *Word Order and Scrambling*, pp. 67-90. Cambridge: Blackwell.

Fanselow, Gisbert (2001) When formal features need company. In C. Féry and W. Sternefeld (eds.), *Audiatur Vox Sapientiae*, 131-152. Berlin: Akademie-Verlag.

Flemming, Edward (1995) *Auditory Representations in Phonology*. Doctoral dissertation, UCLA.

Flemming, Edward (1996) Evidence for constraints on contrast: The dispersion theory of contrast. In C.-S. Hsu (ed.), *UCLA Working Papers in Phonology 1*, pp. 86-106.

Hankamer, Jorge (1973) Unacceptable ambiguity. *Linguistic Inquiry* 4: 17-68.

Hirotani, Masako (2000) Constraint on prosodic structures in grammar and parser: Scrambled and unscrambled sentences in Japanese. Ms., University of Massachusetts, Amherst.

Inoue, Atsu (1991) *A Comparative Study of Parsing in English and Japanese*. Doctoral dissertation, University of Connecticut.

Inoue, Atsu, and Janet Dean Fodor (1995) Information-paced parsing of Japanese. In R. Mazuka and N. Nagai (eds.), *Japanese Sentence Processing*, pp. 9-64. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Jager, Gerhard (2000) Some notes on the formal properties of bidirectional Optimality Theory. Ms., University of Potsdam/Stanford University. [ROA 414-0900]

Kitagawa, Yoshihisa (2000) Anti-scrambling. Ms., University of Rochester.

Kuhn, Jonas (2001) *Formal and Computational Aspects of Optimality-Theoretic Syntax*. Doctoral dissertation, Universitat Stuttgart.

Kuroda, S.-Y. (1988) Whether we agree or not: A comparative syntax of English and Japanese. In W. Poser (ed.), *Papers from the Second International Workshop on Japanese Syntax*, 103-143. Stanford, CA: CSLI.

Kuno, Susumu (1973) *The Structure of the Japanese Language*. Cambridge, MA: MIT Press.

Kuno, Susumu (1980a) A note on Tonioke's intra-subjectivization hypothesis. In Y. Otsu and A. Farmer (eds.) *MITWPL 2: Theoretical Issues in Japanese Linguistics*, pp. 149-156.

Kuno, Susumu (1980b) A further note on Tonioke's intra-subjectivization hypothesis. In Y. Otsu and A. Farmer (eds.) *MITWPL 2: Theoretical Issues in Japanese Linguistics*, pp. 171-184.

Lee, Hanjung (2001) Markedness and word order freezing. In P. Sells (ed.), *Formal and Empirical Issues in Optimality Theoretic Syntax*, pp. 63-128. Stanford: CSLI.

Legendre, Géraldine, Jane Grimshaw, and Sten Vikner (eds.) (2001) *Optimality-Theoretic Syntax*. Cambridge, MA: MIT Press.

Lubowicz, Anna (2003) *Contrast Preservation in Phonological Mappings*. Doctoral dissertation, University of Massachusetts, Amherst.

McCarthy, John J. (2002) *A Thematic Guide to Optimality Theory*. Cambridge: Cambridge University Press.

Nagai, Noriko (1995) Constraints on topics and their gaps: From a parsing perspective. In R. Mazuka and N. Nagai (eds.) *Japanese Sentence Processing*, 77-104. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Padgett, Jaye (2003) Systemic contrast and Catalan rhotics. Ms., UCSC. [ROA 574-0203]

Padgett, Jaye (2004) Russian vowel reduction and Dispersion Theory. In *Phonological Studies 7*, pp. 81-96. Kaitakusha, Tokyo.

Prince, Alan, and Paul Smolensky (1993) Optimality Theory: Constraint interaction in generative grammar. Tech report RUCCS TR-2, Rutgers University, New Brunswick, NJ. [ROA 537-0802]

Reilly, Ehren M. (2002) A survey of Texistepec Popoluca verbal morphology. B.A. thesis, Carleton College.

Richards, Norvin (1997) *What Moves Where When in Which Language?* Doctoral dissertation, MIT.

Ruwet, Nicolas (1972) How to deal with syntactic irregularities: Conditions on transformations or perceptual strategies? In F. Kiefer and N. Ruwet (eds.) *Generative Grammar in Europe*, pp. 419-444. Dordrecht: Reidel.

Saito, Mamoru, and Fukui, Naoki (1998) Order in phrase structure and movement. *Linguistic Inquiry* 29: 439-474.

Tonoike, Shigeo (1980a) Intra-subjectivization. In Y. Otsu and A. Farmer (eds.) *MITWPL 2: Theoretical Issues in Japanese Linguistics*, pp. 137-148.

Tonoike, Shigeo (1980b) More on intra-subjectivization. In Y. Otsu and A. Farmer (eds.) *MITWPL 2: Theoretical Issues in Japanese Linguistics*, pp. 157-170.

Tsujimura, Natsuko (1996) *An Introduction to Japanese Linguistics*. Oxford: Blackwell.

Uehara, Keiko (1997) Judgments of processing load in Japanese: The effect of NP-*ga* sequences. Journal of Psycholinguistic Research 26: 255-263.

Wilson, Colin (2001) Bidirectional optimization and the theory of anaphora. In G. Legendre, J. Grimshaw and S. Vikner (eds.), *Optimality-Theoretic Syntax*, pp. 465-507. Cambridge, Mass: MIT Press.

Woolford, Ellen (1986) The distribution of empty nodes in Navajo: A mapping approach. *Linguistic Inquiry* 17: 301-330.

Woolford, Ellen (1988) Syntactic mapping in VSO languages. In *Linguistics in the Morning Calm 2: Selected Papers from SICOL-1986*, pp. 605-613. Seoul: Hanshin Publishing Co.

Woolford, Ellen (2001) Case patterns. In G. Legendre, S. Vikner, and J. Grimshaw (eds.) *Optimality Theoretic Syntax,* pp. 509-543. Cambridge, MA: MIT Press.

Woolford, Ellen (2005) Differential subject marking at argument structure, syntax, and PF. Ms., University of Massachusetts, Amherst. To appear in H. de Hoop and P. de Swart (eds.), *Differential Subject Marking*. Studies in Natural Language and Linguistic Theory series, Kluwer Academic Publishers.

Department of Linguistics
South College
University of Massachusetts
Amherst MA 01003

flack@linguist.umass.edu