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.

(2)  *“Taroo-ga; Hanako-ga ti kowa-i.” = “Taroo-ga Hanako-ga kowa-i.”
     T-NOM   H-NOM    is.afraid-PRES   T-NOM   H-NOM    is.afraid-PRES
     *‘Hanako is afraid of Taroo.’   ‘Taroo is afraid of Hanako.’

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, judgments, and support. A longer version of this paper, with more detailed discussion of the formal properties of the proposal, is available on the Rutgers Optimality Archive.
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

(3)  *“Taroo-∅; Hanako-∅ t_I osore-ru.”  = “Taroo-∅ Hanako-∅ osore-ru.”
    T-(ACC)  H-(NOM)    fear-PRES    T-(NOM)  H-(ACC)    fear-PRES
    *‘Hanako fears Taroo.’                        ‘Taroo fears Hanako.’
Ambiguity avoidance as contrast preservation

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 \(-\text{ga}\) while objects are assigned accusative \(-\text{o}\). 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-\text{ga} Hanako-\text{o} osore-\text{ru}
   \hspace{1cm} Taroo-NOM Hanako-ACC fear-PRES
   \hspace{1cm} ‘Taroo fears Hanako’
Kathryn Flack

b. Hanako-o, Taroo-ga ti 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 ti 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, Hanako-ga ti kowa-i  ‘Hanako is afraid of Taroo’

c. Taroo-ga Hanako-ga kowa-i      ‘Taroo is afraid of Hanako’

d. *Hanako-ga, Taroo-ga ti kowa-i  ‘Taroo is afraid of Hanako’

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

---
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 Hanako (and no one else)’, or contrastive meanings like ‘Taroo helps Hanako (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. Japanese focus-driven scrambling is discussed in section 3.
order is frozen in SOV order even when the meanings of the words should inherently render the sentence unambiguous.

(8) a. *jishin-ga; Taroo-ga t; 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-0i  Taroo-ga t; suki-da  
    tennis-ACC Taroo-NOM like-PRES  
    ‘Taroo likes tennis’

     b. *tenisu-gai  Taroo-ga t; 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 (8) and (9). Word order freezing appears instead to be the result of a strict grammatical prohibition on scrambling in double nominative sentences.

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 dʔ-agʔ maʃ-dʔaʔa kanʔ-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 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.
Ambiguity avoidance as contrast preservation

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.²

(12) a. osore-PRES <Agent: Hanako, Theme: Taroo>
    \[ \rightarrow \text{Hanako-ga Taroo-o osore-ru} \]
    \[ \rightarrow \text{Taroo-o_i Hanako-ga t_i osore-ru} \]

b. osore-PRES <Agent: Taroo, Theme: Hanako>
    \[ \rightarrow \text{Taroo-ga Hanako-o osore-ru} \]
    \[ \rightarrow \text{Hanako-o_i Taroo-ga t_i osore-ru} \]

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

(13) a. kowa-PRES <Agent: Hanako, Theme: Taroo>
    \[ \rightarrow \text{Hanako-ga Taroo-ga kowa-i} \]
    \[ *\text{Taroo-ga_i Hanako-ga t_i kowa-i} \]

b. kowa-PRES <Agent: Taroo, Theme: Hanako>
    \[ \rightarrow \text{Taroo-ga Hanako-ga kowa-i} \]
    \[ *\text{Hanako-ga_i Taroo-ga t_i kowa-i} \]

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 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).
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 schematic PF mappings from syntactic structures to surface forms of nominative-accusative sentences in (15a) and double nominative sentences in (15b).

(15) a. O-ACC\textsubscript{i} S-NOM O-ACC\textsubscript{i} VERB \rightarrow O-o\textsubscript{i} S-ga O-o\textsubscript{i} VERB
b. O-NOM\textsubscript{i} S-NOM O-NOM\textsubscript{i} VERB \rightarrow O-ga\textsubscript{i} S-ga O-ga\textsubscript{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).\textsuperscript{3}

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) \textup{MAX(Head)} \quad \text{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.

2.3. \textup{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.

\textsuperscript{3} Evidence for this higher LF copy is discussed in the version of this paper available on ROA.
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 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 (18). 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.

(18) PRESERVECONTRAST(Subject): Given two pairs of input-output correspondents $I, O$ and $I', O'$ where $O$ and $O'$ are in cluster $C$, if $\text{Subject}(I) \neq \text{Subject}(I')$ 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

---

4 The function $\text{Subject}(X)$ here identifies the lexical material in spec-IP of a given structure.
1999 and references therein); it seems natural that grammars should protect such a prominent argument. (see fn. 6 for an additional argument against a general constraint *AMBIGUITY.)

2.4. PRESERVECONTRAST(Subject) and word order freezing

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 first part of this section will show how high-ranking PRESERVECONTRAST(Subject) can account for the avoidance of subject ambiguity by causing 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 (19), 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.

(19) 

<table>
<thead>
<tr>
<th>Input Cluster</th>
<th>PRCONTR(Subj)</th>
<th>MAX(Head)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[T-NOM_i, H-NOM H-NOM kowa-PRES]_1, [T-NOM H-NOM kowa-PRES]_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>➔ [T-ga_i, H-ga T-ga kowa-i]_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. [T-ga H-ga kowa-i]_2</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>b. [T-ga H-ga kowa-i]_2</td>
<td>*!</td>
<td></td>
</tr>
</tbody>
</table>

The two members of the input cluster shown in (19) 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 (19b), deletion of the tail of the chain in scrambled form 1 produces a phonological string which is identical to the unscrambled form 2; (19b) thus violates PRCONTR(Subj). Candidate (19a) is therefore the winner.

5 Input and output clusters, of course, contain more members than those crucial forms shown here. For a detailed discussion of the composition and evaluation of input and output clusters, see section 3.
Ambiguity avoidance as contrast preservation

despite its violation of lower-ranked MAX(Head). This evaluation therefore determines the output forms of all members of the input cluster.⁶

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.

The two members of the input cluster in (20) 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 (20a) incurs a violation of MAX(Head); (20b) is therefore the winner, as expected.

⁶ 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 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.
Kathryn Flack

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.

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, pronouncing the head of a scrambling chain is relatively unmarked, as it occurs freely in morphologically unambiguous contexts. Similarly, 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 – i.e. which form must surface in a more marked manner – 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 (e.g. a case particle) 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 (21).
Ambiguity avoidance as contrast preservation

We know that the cluster (21a), which repairs the scrambled form \(1\) and violates MAX(Head), wins. This cluster is therefore more optimal than cluster (21c), 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 which form 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. If the input form \(1\) in (21) 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) \(\gg\) MAX(Head) holds, then the MAX(Head)-violating cluster is the most optimal, and the scrambled form is repaired.

Similarly, the constraint hierarchy also determines that the scrambled form is best repaired by pronouncing the tail of the chain, rather than by some other process. For example, this form could also be repaired by dropping a case particle; as above, the ranking MAX(Case) \(\gg\) MAX(Head) rules out this alternative repair for the scrambled form. Constraint ranking thus determines both which member of the ambiguous pair is repaired and also how this form is repaired.

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 source of clusters

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. The following section explores the implementation 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; these modifications will prove relatively minor. I will then address the question of where PF input clusters are generated. I argue that syntax should also evaluate clusters of inputs and outputs, and that PF input clusters are simply ‘inherited’ syntax output clusters. An algorithm will be provided for creating appropriate clusters of initial syntax inputs.

3.1. Generating and evaluating output clusters

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, and Eval calculated constraint violations incurred by each of these output forms. In the cluster-based model discussed here, PF evaluations must consider clusters of inputs and outputs. The source of PF inputs will be discussed in section 3.2 below; assuming that such an input cluster exists, however, we will first explore how Gen and Eval must be modified in order to produce and then evaluate a set of possible output cluster candidates which are based on an input cluster.

In order for Gen to produce output clusters based on a given input cluster, it must be modified in three ways. 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. See (28) below for a diagram of Gen’s treatment of clusters.

Moving on to evaluation of these output candidate clusters, Eval must be modified such that it assesses constraint violations incurred by output clusters rather than by single output forms. In order to evaluate violations of markedness and faithfulness constraints incurred by clusters of forms, the constraints must evaluate each output in each output cluster. The violations of a given constraint incurred by each cluster member

---

7 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).
are then summed to determine the total violations of that constraint incurred by the cluster as a whole. 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 \( \binom{n}{2} \) pairs of outputs from a cluster of \( n \) outputs, examining each of these pairs for subject ambiguity. Again, the violations are summed – each pair of forms which violates PRCONTR(Subj) adds one violation to the cluster’s total.

### 3.2. 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 assumption in previous OT work that, in a single-input evaluation, a PF input is the ‘inherited’ winning form from the preceding syntax evaluation (Woolford 2001, 2005). I will extend this to the cluster-based model of OT, arguing that PF input clusters are similarly inherited winning clusters from preceding syntax evaluations. In order for syntax to produce a winning output cluster, I will show that syntax inputs should 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 multiple 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.

#### 3.2.1. The source of PF input clusters: Syntax output clusters

In order to motivate the claim that syntax should operate on clusters so as to produce winning output clusters which can become PF input clusters, this section will set out desiderata for PF input clusters, and will then demonstrate that clusters with these desired properties can be best obtained when syntax evaluates clusters. The following section will formalize the mechanism which produces syntax input clusters.

I propose that a 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 (22a,b). Forms like (22a,c), which are structurally identical but lexically distinct, however, are under no threat of merger. 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.

(22) a. Sumiko-to [Jiroo-no okaasan]  
    Sumiko-CONJ Jiroo-GEN mother  
    ‘Sumiko and [Jiroo’s mother]’

b. [Sumiko-to Jiroo-no] okaasan  
   Sumiko-CONJ Jiroo-GEN mother  
   ‘[Sumiko and Jiroo’s] mother’

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

Next, while the PF input cluster contains inputs formed of the same set of words, these inputs must differ in their argument structures. The impetus for this requirement follows directly from the evidence that Japanese avoids ambiguity between inputs with different subjects, i.e. different argument structures. In order to provide PRCONTR(Subj) with relevant forms among which 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, a hypothetical form which had movement out of a complex NP could be ambiguous with another form and cause a fatal violation of PRCONTR(Subj). This violation would be meaningless in reality, however, as one of the forms which gave rise to it will never actually surface in Japanese, and so the ambiguity would never actually occur. Such meaningless comparisons can easily be avoided if all members of a PF input cluster are actual winners of syntax evaluations.

A straightforward way of producing a cluster of 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, and use the syntax grammar to choose the optimal syntax outputs for each of these forms. 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 and Eval can choose a winning syntax output cluster (both as in PF; see section 3.1), and this output cluster of syntax winners can become a PF input cluster which consists, as desired, of lexically identical forms with different argument structures.
3.2.2. The source of syntax input clusters: CGen

The crucial question at this point regards the source of the syntax input clusters which will ultimately give rise to PF input clusters. As a syntax input cluster 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, 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). 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 (23).

(23)  [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 (24).

(24)  { [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 (24) for the simple input in (23) 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.
3.2.3. ‘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 (25) (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.

(25) *“Taroo-ga Hanako-ga i kowa-i.” = “Taroo-ga Hanako-ga kowa-i.”
    T-NOM  H-NOM  afraid.of-PRES  T-NOM  H-NOM  afraid.of-PRES
    *‘Hanako is afraid of Taroo.’  ‘Taroo is afraid of 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 (26) demonstrates the optional mapping of syntax inputs with discourse topic objects to either scrambled (26b,d) or unscrambled (26a,c) syntax outputs.

(26) 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
c. kowa-PRES <T, H> | T-NOM H-NOM kowa-PRES | T-ga H-ga 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, H-NOM T-NOM, kowa-i*, there must be two possible winning syntactic output clusters. That is, there must be one cluster of unscrambled forms (27a) which wins in some syntax evaluations, and another of scrambled forms (27b) 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 (27c).

(27) a. Winning syntax cluster of scrambled forms:
    { [T-NOM, H-NOM T-NOM, kowa-PRES], [H-NOM, T-NOM H-NOM, kowa-PRES] }
b. Winning syntax cluster of unscrambled forms:

\[
\{ \begin{array}{l}
[H-\text{NOM} \ T-\text{NOM} \ kowa-\text{PRES}], \\
[T-\text{NOM} \ H-\text{NOM} \ kowa-\text{PRES}] 
\end{array} \}
\]

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

\[
\{ \begin{array}{l}
[T-\text{NOM}_i \ H-\text{NOM} \ T-\text{NOM}_i \ kowa-\text{PRES}], \\
[H-\text{NOM}_i \ T-\text{NOM} \ H-\text{NOM}_i \ kowa-\text{PRES}], \\
[H-\text{NOM} \ T-\text{NOM} \ kowa-\text{PRES}], \\
[T-\text{NOM} \ H-\text{NOM} \ kowa-\text{PRES}] 
\end{array} \}
\]

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. 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 evaluations begin with an input cluster, 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 (28).

(28) Syntax: Input cluster generation

Base syntax input: \( \text{kowa-PRES} <\text{Agent: Hanako, Theme: Taroo}> \)

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

\[
\{ \begin{array}{l}
[\text{kowa-PRES} <\text{Agent: Hanako, Theme: Taroo}>]_1, \\
[\text{kowa-PRES} <\text{Agent: Taroo, Theme: Hanako}>]_2 
\end{array} \}
\]

Syntax: Gen and Eval

Gen produces syntax output correspondents of each input:

\[
\begin{align*}
[\text{kowa-PRES} <\text{Agent: H, Theme: T}>]_1 \rightarrow & \ [H-\text{NOM} \ T-\text{NOM} \ H-\text{NOM} \ kowa-\text{PRES}]_{1A} \\
& \rightarrow \ [T-\text{NOM} \ H-\text{NOM} \ kowa-\text{PRES}]_{1B} \\
[\text{kowa-PRES} <\text{Agent: T, Theme: H}>]_2 \rightarrow & \ [T-\text{NOM} \ H-\text{NOM} \ T-\text{NOM} \ kowa-\text{PRES}]_{2A} \\
& \rightarrow \ [H-\text{NOM} \ T-\text{NOM} \ kowa-\text{PRES}]_{2B}
\end{align*}
\]
Kathryn Flack

Gen builds unique syntax output clusters:

\[
\begin{align*}
&\{[\text{H-NOM} \ \text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1A}, [\text{T-NOM} \ \text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2A}\}; \\
&\{[\text{H-NOM} \ \text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1A}, [\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2B}\}; \\
&\{[\text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1B}, [\text{T-NOM} \ \text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2A}\}; \\
&\{[\text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1B}, [\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2B}\}
\end{align*}
\]

Eval chooses (one or more) winning syntax clusters:

\[
\begin{align*}
&\{[\text{H-NOM} \ \text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1A}, [\text{T-NOM} \ \text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2A}\}; \\
&\{[\text{H-NOM} \ \text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1A}, [\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2B}\}; \\
&\{[\text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1B}, [\text{T-NOM} \ \text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2A}\}; \\
&\{[\text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_{1B}, [\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_{2B}\}
\end{align*}
\]

**PF: Input cluster generation**

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

\[
\begin{align*}
&\{[\text{H-NOM} \ \text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_1, \\
&[\text{T-NOM} \ \text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_2, \\
&[\text{T-NOM} \ \text{H-NOM} \ \text{kowa-PRES}]_3, \\
&[\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_4 \}
\end{align*}
\]

**PF: Gen and Eval**

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

\[
\begin{align*}
&[\text{T-NOM}, \ \text{H-NOM} \ \text{T-NOM}, \ \text{kowa-PRES}]_1 \rightarrow [\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1A} \\
&\quad \rightarrow [\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1B} \\
&[\text{H-NOM} \ \text{T-NOM} \ \text{kowa-PRES}]_4 \rightarrow [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4A}, \\
&\quad \rightarrow [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4B}
\end{align*}
\]

Gen builds unique PF output clusters:

\[
\begin{align*}
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1A}, [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4A} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1A}, [\text{T-ga}_i \ \text{H-∅} \ \text{kowa-i}]_{4B} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1B}, [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4A} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1B}, [\text{T-ga}_i \ \text{H-∅} \ \text{kowa-i}]_{4B} \}
\end{align*}
\]

Eval chooses the PF winner:

\[
\begin{align*}
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1A}, [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4A} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1A}, [\text{T-ga}_i \ \text{H-∅} \ \text{kowa-i}]_{4B} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1B}, [\text{T-ga}_i \ \text{H-ga} \ \text{kowa-i}]_{4A} \}; \\
&\{[\text{T-ga}_i \ \text{H-ga} \ \text{T-ga}_i \ \text{kowa-i}]_{1B}, [\text{T-ga}_i \ \text{H-∅} \ \text{kowa-i}]_{4B} \}
\end{align*}
\]

---

8 Recall from section 2.4.2 that dropping the object case particle is an alternative hypothetical output form of this PF input which repairs the PrCONTR(Subj) violation.
Ambiguity avoidance as contrast preservation

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, PRCNTR(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 PRCNTR(Subj), preventing a repair of subject contrast neutralization. Further, as PRCNTR(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.

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

Word order freezing in Japanese double nominative sentences is the consequence of the constraint ranking PRCNTR(Subj) » MAX(Head). Given that all constraints in an OT grammar are violable, it is possible that a conflicting constraint could dominate PRCNTR(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 PRCNTR(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 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.9 As shown in (29), where underlining indicates phonological focus, such a sentence can have either a scrambled or unscrambled reading.

9 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.
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. The input which means \textit{Hanako is afraid of Taroo} is paired with the output \textit{Taroo-ga Hanako-ga kowa-i}, despite the fact that this output is also the optimal correspondent of the input which means \textit{Taroo is afraid of Hanako}. The winning output cluster which contains these two identical output correspondents of different inputs will therefore violate \textsc{PrContr(Subj)}; the fact that it wins anyway means that some other conflicting constraint must be ranked above \textsc{PrContr(Subj)}.

Focus can license ambiguity as follows. The head of a scrambling chain must surface, in a context where this would cause a violation of \textsc{PrContr(Subj)}, specifically when scrambling is driven by a [Focus] feature. Such patterns where something (e.g. the head of a chain) is faithfully realized in a particular prominent context (e.g. when it is focused) often follow from a high-ranking positional faithfulness constraints (Beckman 1998); in this case, the pattern is the result of high-ranking \textsc{Max(Head)/[Focus]}.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{(30)} & \textbf{Max(Head)/[Focus]} & Do not delete the head of a chain when it bears a [Focus] feature. \\
\hline
\textbf{The constraint ranking Max(Head)/[Focus] > PrContr(Subj) > Max(Head)} & produces the desired output cluster, as shown in (31). \\
\hline
\end{tabular}
\end{center}

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{(31)} & \textbf{[T-NOM[FOC] H-NOM T-NOM[FOC] kowa-PRES]$_1$} & \textbf{Max (Head)/[Focus]} & \textbf{PrContr (Subj)} & \textbf{Max (Head)} \\
& \textbf{[T-NOM[FOC] H-NOM kowa-PRES]$_2$} & & & \\
\hline
a. & \textbf{[T-ga, H-ga T-ga, kowa-i]$_1$} & *! & & * \\
& \textbf{[T-ga, H-ga kowa-i]$_2$} & & & \\
\hline
\textbf{⇒ [T-ga, H-ga T-ga, kowa-i]$_1$} & \textbf{[T-ga, H-ga kowa-i]$_2$} & * & & \\
\hline
b. & & & & \\
\hline
\end{tabular}
\end{center}

In form \textbf{l} in output candidate cluster (31a), the tail of the scrambling chain is pronounced, rendering the output forms \textbf{l} and 2 distinct. \textsc{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 undominated \textsc{Max(Head)/[Focus]}. (31b) is therefore the winner, despite the fact that pronunciation of the head of the scrambling chain causes forms \textbf{l} and 2 to be ambiguous.
Ambiguity avoidance as contrast preservation

The results of this section can be generalized to make a broader point about the violability of \textsc{PrContr(Subj)}: like all other constraints in an OT grammar, this contrast preservation constraint is violable. Just as high-ranking \textsc{PrContr(Subj)} can force violation of lower-ranked constraints (resulting in more marked outputs), constraints which outrank \textsc{PrContr(Subj)} can force its violation, resulting in subject ambiguity. This section has demonstrated that subject ambiguity occurs when \textsc{Max(Head)/[Focus]} outranks \textsc{PrContr(Subj)}. While the general correlation between morphological ambiguity and fixed word order is enforced by relatively high-ranking \textsc{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: Other occasions of subject contrast preservation

As discussed above, when a double nominative verb fails to assign unique cases to its subject and object, the constraint ranking \textsc{PrContr(Subj)} \textasciitilde \textsc{Max(Head)} causes the order of these arguments becomes fixed in order to prevent ambiguity (unless scrambling is driven by focus). This predicts that these constraints should also block scrambling in other constructions where case does not distinguish between subject and object.

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 (32).\footnote{Case particle drop is subject to a number of discourse conditions. Accusative \textasciitilde\textit{o} drops more freely than nominative \textasciitilde\textit{ga}, and \textasciitilde\textit{ga} drops most easily when the speaker is using her full name to refer to herself in subject position; \textasciitilde\textit{ga} drop is reportedly most common among young female speakers of Japanese. (32c,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 \textasciitilde\textit{o} scrambles more easily in a sentence like \textit{Taroo, ore\textasciitilde ga t\textasciitilde bokotta yo} ‘Taroo, I fucked (him) up, man’ than in (33), as the former uses the male-oriented first person pronoun \textit{ore}, the slang verb \textit{bokotta}, and the colloquial particle \textit{yo}, which conspire to make the sentence inherently extremely colloquial. \textasciitilde\textit{ga} and \textasciitilde\textit{o} are the only case particles which can drop; ditransitive sentences therefore never have two case-less objects and thus are never potentially ambiguous.}

(32) a. Hanako\textasciitilde ga Taroo\textasciitilde o osore\textasciitilde ru
Hanako-NOM Taroo-ACC fear-PRES
‘Hanako fears Taroo.’

b. Hanako\textasciitilde ga Taroo\textasciitilde \emptyset osore\textasciitilde ru
c. Hanako\textasciitilde \emptyset Taroo\textasciitilde o osore\textasciitilde ru
d. Hanako\textasciitilde \emptyset Taroo\textasciitilde \emptyset osore\textasciitilde 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 (33).
Kathryn Flack

(33) a. Taroo-∅ Hanako-ga ti osore-ru
    b. Taroo-o, Hanako-∅ ti 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 (34) 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.11

(34) a. Hanako-ga Taroo-o osore-ru  →  Hanako-∅ Taroo-∅ osore-ru
    b. Hanako-o, Taroo-ga ti osore-ru  →  *Hanako-∅, Taroo-∅ ti 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).

(35)  

<table>
<thead>
<tr>
<th>[H-ACC1 T-NOM H-ACC1 osore-PRES]1</th>
<th>PrContr (Subj)</th>
<th>MAX(Head)</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ [H-∅, T-∅ H-∅, osore-ru]1</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>a. [H-∅ T-∅ osore-ru]2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. [H-∅ T-∅ osore-ru]2</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

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 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 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-nominative sentences allows a uniform treatment of the identical behavior of these two phenomena.

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 subject ambiguity regardless of its source; the constraint does not specifically target double nominative sentences or case

11 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.
particle drop. Instead, its effects capture the generalization which holds both within Japanese and also cross-linguistically, that all sources of morphological ambiguity are undesirable and prone to disambiguation processes, including word order freezing.

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. This section will first briefly summarize both local and global non-OT explanations of word order freezing and ambiguity avoidance; it will then explore in some detail an alternative OT theory of ambiguity avoidance, bidirectional OT (see e.g. Jager 2000; Kuhn 2001). 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. Non-OT approaches to ambiguity avoidance

There are two major approaches to anti-ambiguity phenomena outside of OT. Local explanations (e.g. Tonoike 1980a,b) claim that e.g. word order freezing follows from inherent structural properties of double nominative sentences, and that the resulting ‘avoidance’ of ambiguity is an accidental result of other properties of this construction. 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.

There have also been extremely global proposals of ambiguity avoidance. Various proposals (e.g. Kuno 1980b, Ruwet 1972) have been made claiming that ambiguity avoidance phenomena including Japanese word-order freezing results from parsing strategies requiring that apparently structurally ambiguous sentences have only a single, unambiguous interpretation. As described in section 1, however, parsing explanations of word order freezing do not account for the Japanese data described here. A parsing explanation would predict that world knowledge could be sufficient to disambiguate, and therefore that a sentence like Tenisu-ga, Taroo-ga t, zyoozu-da (‘Taroo is good at tennis’) should be grammatical, as this verb must take an animate subject (Taroo) and a skill-type object (tenisu). Scrambling is blocked even in this case, however; therefore, a parsing explanation fails to predict all cases in which word order freezes.

Finally, another type of global approach to ambiguity avoidance proposes universal grammatical filters which prevent ambiguity. 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 (e.g. Hankamer 1973; Woolford 1986, 1988) have been made to account for particular patterns of ambiguity avoidance in both deletion and movement contexts. Such universal restrictions are problematic in that they do not allow exceptions, both within and across languages. Recall from section 4.1 that scrambling is allowed to cause ambiguity in double nominative sentences when the object is focused; a different sort of ambiguity tolerance is found in Texistepec Popoluca (see (10), above), in which non-focus-driven scrambling is permitted to cause subject ambiguity. The filters which have been proposed to account for anti-ambiguity phenomena have been considered inviolable cross-linguistic principles which categorically ban violating structures; anti-ambiguity principles appear instead to be violable. Capturing the prohibition on subject-oriented ambiguity in the violable OT constraint \textsc{PrContr(Subject)} allows this crucial violability.

5.2. Bidirectional OT as a means of ambiguity avoidance

A recent approach to anti-ambiguity phenomena has involved a modification of the OT architecture 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 been used to account for anti-ambiguity phenomena 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, using an input meaning $M$ and choosing a form $F$ to pair with $M$. 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 $M = M'$), 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 $F$. 

(36) Bidirectional optimization:

\[
\begin{align*}
/M/ & \rightarrow \text{PRODUCTION} \rightarrow \text{“F”} \rightarrow \text{COMPREHENSION} \rightarrow /M'/ \\
</M/, \text{“F”}> & \text{ is bidirectionally optimal iff } \\
/M/ & \rightarrow \text{PRODUCTION} \rightarrow \text{“F”}, \text{ and } \\
\text{“F”} & \rightarrow \text{COMPREHENSION} \rightarrow /M/
\end{align*}
\]

This bidirectional architecture eliminates all ambiguity from languages.\(^{12}\) 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, \(</M_1/, \text{“F”}>\) and \(</M_2/, \text{“F”}>\), are both optimal. A unidirectional OT model of syntax evaluation easily produces such a situation if the underlying contrast between inputs \(M_1\) and \(M_2\) is neutralized by the constraint ranking. As a traditional (non-cluster-based) unidirectional model takes into account only a single input-output pair at a time, ambiguity often results. A bidirectional model is inherently incapable of pairing two meanings with a single form. Given the standard assumption that Eval pairs one and only one most harmonic output with each input, a form input \(F\) in the comprehension evaluation can map to either output \(M_1\) or \(M_2\), but not both. Such a grammar therefore is extremely successful at banning 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 (37).

(37) “Taroo-ga Hanako-ga kowa-i”

\[
\begin{align*}
\text{Hanako is afraid of Taroo} & \quad \downarrow \\
/\text{Taroo is afraid of Hanako}/ & \quad \uparrow
\end{align*}
\]

In a bidirectional model, scrambling is blocked for a meaning like \textit{Hanako likes Taroo} because this meaning is not recoverable from the scrambled surface form \textit{Taroo-ga Hanako-ga kowa-i}. This string is preferentially interpreted as \textit{Taroo is afraid of Hanako}, and so because the initial meaning input and the recovered meaning are not identical, the meaning-form pairing \(</\text{Hanako is afraid of Taroo}/, \text{“Taroo-ga Hanako-ga kowa-i”}>\) isn’t bidirectionally optimal; see Lee (2001) for details of the analysis.

\(^{12}\) ‘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. In a weak bidirectional model, the best form and its 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, though, and so still prevents all ambiguity. See Beaver and Lee (2003) for an overview of this and other proposed bidirectional models.
In the model of bidirectional OT described in (38), no ambiguous sentence will ever be grammatical; this is a major problem for the model. While languages occasionally block processes (e.g. scrambling) which 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 (39), repeated from above; see Inoue and Fodor (1995) for more examples of ambiguity in Japanese.

(39) 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 to this end is found in Asudeh (2001), which derives ambiguity from variable constraint ranking (following Boersma (1997, 1998); 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 any 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 ambiguous bidirectionally optimal meaning-form pairs \( \langle M_1, “F” \rangle \) and \( \langle M_2, “F” \rangle \).

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 exhibiting both ambiguity and also optionality, where optionality is the pairings of multiple forms with a single meaning, i.e. both \( \langle M, “F_1” \rangle \) and \( \langle M, “F_2” \rangle \). Asudeh’s system therefore produced four possible form-meaning pairs: \( \langle M_1, “F_1” \rangle, \langle M_2, “F_1” \rangle, \langle M_1, “F_2” \rangle, \langle M_2, “F_2” \rangle \), using a pair of constraints whose reranking produces 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. Until tolerance of ambiguity within a bidirectional system is more thoroughly investigated, it is difficult see how this model can account for the full range of data found in Japanese and other languages.

6. Conclusion

This paper has addressed the problem of word order freezing in Japanese, which I argue is a case of ambiguity avoidance. When ambiguity as to the subject of a sentence 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. The new constraint PRCONTR(Subj) evaluates sets of inputs and outputs, while otherwise behaving like other constraints. While PRCONTR(Subj) 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) does not make specific reference to the source of subject ambiguity, and so triggers word order freezing whenever subjects and objects are morphologically ambiguous. 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.

This approach crucially uses a violable constraint to penalize ambiguity. Alternative proposals have claimed either that processes like word order freezing are not motivated by ambiguity avoidance, or that languages rigidly prevent ambiguity from occurring. The proposal presented here has a number of advantages which allow it to more closely capture the facts of Japanese than do the alternative proposals. 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: 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
Kathryn Flack

deconstructions at hand would necessarily regard this as a mere coincidence. The present approach also easily explains the observation 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 many others.

References

Ambiguity avoidance as contrast preservation


Kathryn Flack


Department of Linguistics
South College
University of Massachusetts
Amherst MA 01003

flack@linguist.umass.edu