A Bare Grammar for Japanese

Jessica WebsterLove
Senior Thesis, Fall 2007
Bryn Mawr College

Abstract

In this thesis we apply the work of Edward Keenan and Edward Stabler in Bare Grammar to "Jpn", a bare grammar for a small fragment of Japanese, to show that the framework constructed in this monograph, while in need of adjustment and augmentation to handle the linguistic features of the language, is overall an appropriate model. A "bare grammar" is a generative framework for syntax, motivated by a desire for the greatest generality possible in this type of theory. To achieve this generality the authors use linguistic universals, assuming as little as possible about common cross-linguistic structure. As Keenan and Stabler point out in their introduction, many alternative theories proposed prior to Bare Grammar rely upon properties that are assumed by an individual theory to be general, but are in fact only descriptive of the language(s) that theory was founded on. (An example is the assumption that grammatical, or syntactic, categories are uniform across all languages, or that the rules for generating complex expressions are uniform across all languages.)

In extending the Bare Grammar framework, we first outline the model and the concept of linguistic universals as given by Keenan and Stabler. Following this, we discuss the features of Japanese to be modeled in the representative fragment Jpn, define this fragment, and analyze how well a bare grammar deals with its peculiarities. We conclude that Bare Grammar requires some change, pointing out areas for improvement by discovering features not wholly accounted for.[1]

[1] I would like to thank Jason Kandybowicz, my thesis advisor, for his continual optimism and copious amounts of helpful feedback. I am also grateful to Suzanne Landi, Kit La Touche, and Ted Fernald for reading and giving comments on the various drafts of this thesis. In addition, Helen Grundman was willing beyond all reasonable expectation to help me in interpreting the math in my source material, and to provide emotional support throughout the semester. Ayako Fukui was also extremely helpful and patient in acting as a native speaker resource for me.

I. Introduction

A primary goal of generative grammar is to identify structural features common to all natural languages. To achieve this, the standard methodology has been to provide an in-depth analysis of a small, carefully controlled set of languages and from there to develop a theory of natural language grammar based on them. This theory is then expanded and revised to account for other languages with the hope of eventual convergence on a unified theory of human grammar. Unfortunately, many such theories prove to be too influenced by the features of the specific language or languages used for evidence, assuming that the features peculiar to the set of languages under investigation are in fact common to all languages.

For instance, most theories of grammar assume that the range of grammatical categories (e.g. NPs) is invariant across all languages, though this is not necessarily true. Compare English and Spanish: in Spanish, nouns have gender (masculine or feminine), whereas this feature is absent with English nouns. Moreover, Spanish adjectives must have gender agreeing with that of the noun they modify. It seems reasonable, then, that in Spanish the categories of masculine and feminine nouns may well be distinguished. English, however, has no need for such separation within its noun category, lacking such agreement requirements.

In contrast to such theories, the Bare Grammar (BG) model developed by Keenan and Stabler (2003) explicitly avoids making such generalizations about the structure of every human language. These other theories, under this assumption of generality, view each individual language as a special case of this common human grammar, having its own peculiar arrangement of parameters—head-initial or -final, for example. Instead, BG views each language independently, allowing that different languages can have different grammars. This includes allowing individualized generation rules and grammatical categories: any two languages may have the same categories, for instance, but then this must be described explicitly.

Bare Grammar, which is a generative model in the Minimalist framework (see Chomsky 1995), focuses on identifying more basic properties as linguistic universals, features that are truly common to all languages. These linguistic universals are what Bare Grammar asserts comprise a true universal grammar. Moreover, Keenan and Stabler's work uses mathematical proof techniques to establish that these objects are indeed consistent in all considered languages rather than simply assuming so.

In this thesis we will apply BG to a fragment of Japanese to show that the model accounts for some salient linguistic features of this language. This thesis is extending the Bare Grammar model: in Keenan and Stabler's 2003 monograph, a number of languages are modeled using BG, but Japanese is not addressed. The analysis of Japanese developed in this thesis reveals a need for some augmentation and adjustment of BG, but ultimately supports it as a model.

II. Bare Grammar

At a basic level, a Bare Grammar is a formal description of a language. This description allows for the construction of the phrases and sentences of that language. It is given in terms of the words of the language, labels indicating roles the words can fill, and the rules used for constructing complex expressions from these words. To model a natural language, a fragment of that language (i.e. a subset its of basic components) is constructed to represent its salient features, and BG is then applied to determine whether these features can be accounted for.

Informally, a bare grammar is stated in terms of:

- vocabulary items—the uninflected words of a given language;
- categories—the labels describing words' argument structures;
- lexical items—the inflected vocabulary items together with categories; and
- rules—specifications of how various lexical items combine.

A bare grammar G is formally defined as a 4-tuple $<V_G, Cat_G, Lex_G, Rule_G>$, where:

- V_G is the set of vocabulary items *s*;
- Cat_G is the set of categories C;
- Lex_G is the set of lexical items (*s*, C), a subset of V×Cat; and
- $Rule_G$ is the set of generating or structure-building functions, which are partial functions $(V^*\times Cat)^+ \rightarrow V^*\times Cat$ (Keenan & Stabler 2003).

When no confusion is possible, subscripts are generally omitted.

As they are not transparent, we will define here terms and notation used above; short definitions of notation are also given in the appendix for reference. When we write V×Cat, read "the cross product of V and Cat", we mean the set of all pairs (*s*, C) consisting of a vocabulary item *s* and the category C of that vocabulary item. A function, represented by an arrow → (and often assigned a name such as *f* or *h*), from a set A (the function's domain) to a set B (its codomain, or range) is a rule that assigns to each element of A exactly one element of B. We say that a function *f* maps an element *x* of A to an element *f(x)* ("*f* of *x*") of B. (So for *x* and *y* in A, and a function *f* that goes from A to B (written f: A→B), if $x=y$ then $f(x)=f(y)$; alternatively, this can be stated as $f(x)\neq f(y)$ implies that $x\neq y$.) The star of V* is the Kleene star: by V* we mean the set of finite sequences of elements of V. Then V*×Cat is the set of possible expressions that can be built up from the vocabulary elements, together with their appropriate categories; these expressions are of the form (*s*, C). As a term, $(V^*\times Cat)^+$ indicates sets $\{(s_1, C_1), (s_2, C_2), \ldots, (s_n, C_n)\}$ of any finite number n of elements (s_i, C_i) of V*×Cat. Thus the superscript + is simply an instruction to take a finite set of elements of V*×Cat. So then a rule F maps a finite set of expressions (with their accompanying categories) to a single expression (with category).

Regarding the elements of Rule, note also that partial functions are those that apply to some, but not all, elements of a given domain (in this case, lexical items and the expressions built up from them) (Keenan & Stabler 2007). The language L_G generated by the grammar G is then the closure of Lex_G under $Rule_G$, i.e. "the set of possible expressions we can build starting from those in Lex and applying the functions F∈Rule finitely many times" (Keenan & Stabler 2003).

We will give an abbreviated example to illustrate how a bare grammar works: a fragment of English and a bare grammar Eng for it, adapted from Keenan & Stabler (2003). This will be immediately followed by an explanation of the terms and notation used, as well as definitions of the elements of $Rule_{Eng}$. $Eng=<V_{Eng}, Cat_{Eng}, Lex_{Eng}, Rule_{Eng}>$ is defined as follows:

(1) V_{Eng} = {laugh, praise, John, Bill, himself, and, both}
Cat_{Eng} = {P0, P1, P2, P01/P12, P1/P2, CONJ}
Lex_{Eng} = P1: laughed

P2: praised
P01/P12: John, Bill
P1/P2: himself
CONJ: and

note: 'both' has no category: it is introduced into a sentence by the rule Coord (see below).

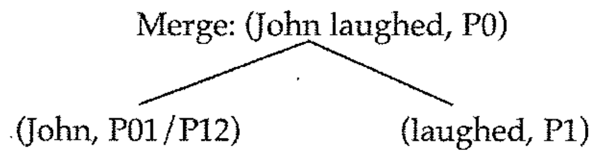
$Rule_{Eng}$= {Merge, Coord}.

Lexical items (as elements of V×Cat) are actually pairs, such as (laughed, P1). (The table format used above is for convenience of presentation only.) The categories, excluding CONJ, are predicate symbols describing the argument structure of the lexical items they are associated with. So "laughed" has category P1 because it takes one argument, the subject of a sentence. (In other words, it is a "one-place predicate".) "John" has category P01/P12 because it can combine with either a P1 (one-place predicate) or a P2 (two-place predicate), to yield a P0 or P1 respectively. So, when a P01/P12 combines with a one-place predicate, it produces a P0, otherwise known as a sentence S (e.g. "John laughed."), and when it combines with a two-place predicate it produces a P1 (e.g. "John praised", which has a place in its argument structure for a direct object). Similarly, "himself" is in category P1/P2 because when it combines with a P2 it produces a P1.

We now turn to describing the elements of $Rule_{Eng}$. We will begin with informal explanations of these rules, then proceed to more rigorous definitions.

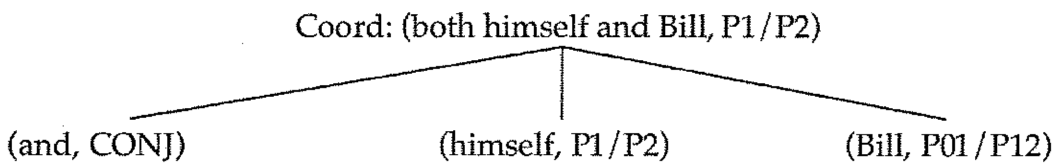
The combinations of lexical items described above are in fact uses of the rule Merge: Merge joins two expressions. We can represent the operation of this function Merge with Function-Argument (FA) trees (which closely resemble standard trees, but indicate the rules used and list lexical items rather than vocabulary items and categories separately, and in fact are more accommodating of some language features).

(2)



The other rule of Eng, Coord, creates conjunctive expressions of the form "both x and y". It can operate on any two (distinct) expressions with categories in the set of coordinable categories, together with one element with category CONJ. The coordinable categories, which are denoted cC_{Eng}, are all the categories that can be coordinated—in our example, all categories except CONJ, i.e. in the set {P1, P2, P1/P2, P01/P12, P0} (that is, cC = Cat – {CONJ}). Additionally, Coord does not produce the category of the resultant expression in the same way as Merge. If the two elements to be acted on by Coord have the same category (e.g. two P1s), the category of the expression produced by Coord will also have that category; if these two elements have different categories, the resultant expression will be of category P1/P2. An example FA tree (here coordinating a P1/P2 and a P01/P12) is:

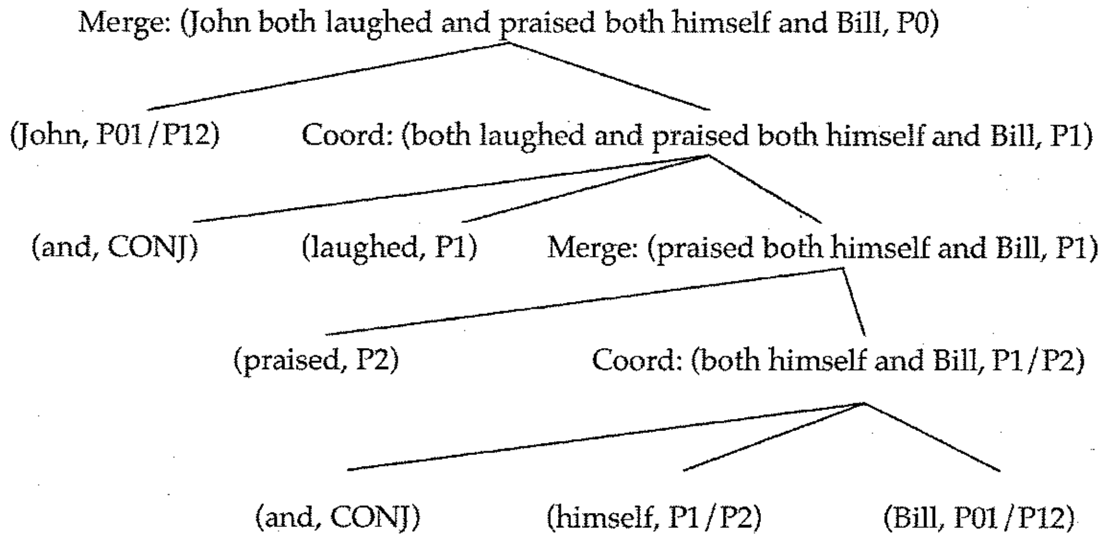
(3)



Note that "both" is not operated on by Coord, but rather inserted by it. This is part of the definition of the rule.

These two rules, Merge and Coord, can be applied to appropriate expressions in Eng in any order and any number of times, until a P0 is produced. For example, the sentence "John both laughed and praised both himself and Bill" can be derived from applying Merge and Coord to increasingly complex strings as shown in (4) below.

(4)



The definitions of Merge and Coord can be more formally presented as (5a,b), below. We use the following notation: s and t (as well as "and" and "or") are strings; A, B, C, and C′ are each the category of the string listed right above them; cC_{Eng} is the set of coordinable categories of Eng, as above; and nC_{Eng} is the set of nominal categories of Eng, {P1/P2, P01/P12}. Additionally, the elements listed under the column heading "Domain" are elements in the set that the function maps from; the "Value" column shows the result of the application of the function to the strings from the domain; and the "Conditions required" are restrictions on the categories or strings being acted on.

(5) a. Merge:

Domain	Merge	Value	Conditions required
s t A B	→	$s^\frown t$ P0	A = P01/P12, B = P1
s t A B	→	$s^\frown t$ P1	A = P1/P2 or P01/P12, B = P2

b. Coord:

Domain	Coord	Value	Conditions required

and CONJ	s C	t C	→	both$\frown$s$\frown$and$\frown$t C	$C \in cC_{Eng}$
or CONJ	s C	t C	→	either$\frown$s$\frown$or$\frown$t C	$C \in cC_{Eng}$
and CONJ	s C	t C′	→	both$\frown$s$\frown$and$\frown$t P1/P2	$C \neq C' \in nC_{Eng}$
or CONJ	s C	t C′	→	either$\frown$s$\frown$or$\frown$t P1/P2	$C \neq C' \in nC_{Eng}$

As noted above, the grammars describing different languages may have different sets Cat and Rule. In this way BG accounts for language variation.

III. Linguistic Invariants

Linguistic invariants are a key concept in the Bare Grammar model. They are the tool that enables cross-linguistic comparison and contrast. Though it is not within the scope of this thesis to perform such a cross-linguistic analysis, we will give here an introduction to invariants, so that we may prove later in this thesis the invariance of certain elements of the grammar we will construct. We begin, as before, with a basic understanding: a linguistic invariant is a linguistic object (e.g. an expression, property, or relation) that is unchanged when acted on by a structure-preserving transformation. To see what a "structure-preserving transformation", or structure map, is, it is simplest to use an example.

(6) a. <u>Bob</u> went to the store.
b. <u>Joe</u> went to the store.
c. <u>He</u> went to the store.

Based on our knowledge as English speakers, we can state that these three sentences have the same structure. Thus, there is a structure-preserving transformation h that maps (6a) to (6b), operating by substitution. In this case, the underlined word in both the first sentence (the domain) and the second sentence (the codomain) is a proper noun; if every structure-preserving transformation h maps a proper noun to another proper noun (preserving grammaticality), then proper nouns are said to be invariant in English. However, we can easily prove that this is not the case: (6c) is also a grammatical sentence of English, with the same structure as (6a, b), but "he" is not a proper noun. Thus, structure-preserving transformations can map proper nouns to linguistic objects that are not proper nouns, and so "is a proper noun" is not an invariant property in English.

In contrast, "function words" (as opposed to "content words") do turn out to be invariant properties of certain expressions in English.

(7) a. Bob went to the store.
b. *Bob went drink the store.
c. *Bob went is the store.

Though a true proof would require more work, we can state here simply that we are not able to identify a non-prepositional element or expression to replace "to" in (7a) that preserves the structure of the sentence: anything but a preposition in place of "to" removes the prepositional phrase structure. Even other "function words" from different classes are not acceptable—in (7c) we see an auxiliary verb replacing the preposition. We can say then that prepositions are linguistic invariants in Eng.

Toward a formal definition of structure-preserving transformations, we specify that these maps must work as above, that is as substitutions. This in fact means that they must be bijections, and that they must not change the way expressions are built up. A bijection is a function that is both onto (or "surjective") and one-to-one (or "injective"). Being onto says that when a structure map h acts on a language L, the result is again the entire language. (Note that h acts on the entire language L: unlike elements of Rule, structure maps are total functions rather than partial ones.) This way, a structure-preserving transformation does not omit any "structurally significant" sets of expressions (Keenan & Stabler 2003:19). Having h be one-to-one prevents sentences such as "Bob either biked or biked", which could be derived if for instance both "walked" and "ran" were mapped to "biked". The statement that structure maps must not change how expressions are built up tells us that for some expressions s and t, if s can be derived by applying elements of Rule to t in a certain order, then $h(s)$ can be derived by applying to $h(t)$ the same elements of Rule in the same order.

Using these properties, we now define the structure maps, called the automorphisms, of a given grammar G as follows:

> **Definition:** A map $h{:}L_G \rightarrow L_G$ is an automorphism of G if and only if h is a bijection and $h(F)=F$ for all $F \in Rule_G$.

The set of all automorphisms of G is denoted Aut_G.

We then say that:

> **Definition:** A linguistic object A in language L_G is invariant if and only if for every $h \in Aut_G$, $h(A)=A$.

(Note that any $F \in Rule_G$ is then trivially invariant.) In other words, if A is a linguistic object that is invariant, then every automorphism h maps every expression s of A to another expression with the same structure, and if $s \in A$ then $h(s) \in A$ as well. It is important to realize that A can be any object, such as a property of expressions, a relation, or a class of function words. In the case where A is a property, for instance, this definition gives us that for any expression s that has this property, say P, every expression $h(s)$ with the same structure as s has property P as well, for every $h \in Aut_G$, i.e. $h(P)=P$.

There are a few important definitions and properties of automorphisms that it is important to note as well. For a set of expressions K, we define $h(K)$ to be the set $\{h(s)\}$ where s is an element of K. So, if K is the set $\{a, b, c\}$ then $h(K)$ is the set $\{h(a), h(b), h(c)\}$. In addition, for any $F \in Rule_G$, any automorphism h commutes with F: $h(F(s, t)) = F(h(s), h(t))$. We will use these definitions and facts in section VI below.

IV. Some Salient Properties of Japanese Grammar

Japanese is a head-final case-marking language that uses suffixes, generally called "particles" in the literature, to indicate grammatical functions (Nominative, Accusative, Topic, Dative, Locative, and quite a few others). As a preliminary to developing our fragment of Japanese and its bare grammar, we will here examine some key relevant features of the language.

IV.1 Word Order

The least marked word order in Japanese sentences is SOV. This is flexible, though: the subject and object may be interchanged with only a minor added emphasis on the object, shown in (8a,b) below. That is, Japanese exhibits "scrambling" of its nominative- and accusative-marked elements. That sentences are verb-final, however, is a fairly strict generalization. (There is one exceptional construction where the topic is moved behind the verb (cf. (8c) below), but this is rather informal, and we will exclude it from the scope of our examination.)

(8) a. Hanako ga Tarou o hometa.[2]

Hanako-NOM Tarou-ACC praised

'Hanako praised Tarou.'

b. Tarou o Hanako ga hometa.

Tarou-ACC Hanako-NOM praised

'Hanako praised Tarou.' (Slight emphasis on "Tarou")

c. ?Hanako ga hometa Tarou o.

Hanako-NOM praised Tarou-ACC

'Hanako praised Tarou.'

[2] *Tarou* and *Hanako* are very common names in Japanese—the stereotypically common names, as it were, like *John* and *Mary* in English.

IV.2 Case Marking (Particles) and Tense Marking

As mentioned above, particles are suffixes used in Japanese to indicate expressions' grammatical functions. The sentences in (8) are simple examples of particles in this role: *ga* marks a word as nominative, while *o* indicates the accusative case. These same particles may still be applied when a word is modified:

(9) Tarou ga omoshiroi eiga o mita.

Tarou-NOM interesting movie-ACC saw

'Tarou saw an interesting movie.'

The class of elements encompassed by the term "particle" is quite broad, and includes items that express such a wide range of ideas that a good generalization of how to translate them is quite difficult, if even possible. It is not the goal this thesis, however, to explore this field, and so we will restrict ourselves to those particles already introduced, used to mark nominative and accusative case.

Japanese is a morphologically impoverished language. Verbs have only two basic tenses, past and non-past, to conjugate for, and do not inflect for person or number. An example of this is (10):

(10) Wara -t -ta

laugh PAST

Note the absence of any pronominal or agreement morpheme: the verb consists only of the root *wara-* and the past-tense marker *-ta*. (The additional *-t-* is an epinthetic consonant.) Since the subject is not specified, this could potentially mean any one of "I laughed", "You laughed", "She laughed", "They laughed", etc. The non-past construction is similar:

(11) Wara -u

laugh NON-PAST

Again with the caveat that the subject could be anything, this could have such translations as "He laughs" or "He will laugh".

IV.3 Topic-drop

Note that the example sentences above—"Waratta" and "Warau"—are in fact grammatical sentences of Japanese, despite having no indication of subject. It is extremely common in Japanese to delete elements that are known from the context of a conversation, in a phenomenon called "topic-drop". So, for instance, once a topic such as a specific person is established, the name of that person will drop out of the conversation almost entirely. Sentences that lack such overt subjects and objects are by far the most typical type of sentences in Japanese: it sounds very odd to keep repeating this information throughout a conversation. (If a referent is not established by context, it is still grammatically possible to drop the argument that specifies the referent; such a sentence would be likely to elicit a question asking for clarification.)

As mentioned above, the element dropped can be a sentence's subject or its object, as in a and b of (12) below, respectively, and in some cases both can be dropped at once, as in (12c).

(12)a. Tarou o hometa.

Tarou-ACC praised.

'(She) praised Tarou.'

b. Hanako ga hometa.

Hanako-NOM praised

'Hanako praised (him).'

c. Hometa.

praised

'(She) praised (him).'

Although this phenomenon is called "topic-drop", actually supposing that an argument of the verb disappears completely creates significant problems for a bare grammar of Japanese. The hypothesis that will be used in this thesis to overcome this difficulty is that the place of a "missing" argument is in fact held by *pro*, the null DP that takes case. (This hypothesis is based on that proposed by Huang for Chinese.) Thus, we would rewrite sentences such as (12a,b,c) above as follows:

(12)a′. *pro* (ga) Tarou o hometa.

pro-NOM Tarou-ACC praised

b′. Hanako ga *pro* (o) hometa.

Hanako-NOM *pro*-ACC praised

c′. *pro* (ga) *pro* (o) hometa.

pro-NOM *pro*-ACC praised

When *pro* takes the place of an element in a sentence, the case-marking particle that accompanied that element also becomes null (represented above by placing the particles in parentheses), so that (12a′,b′,c′) in fact are the same sentences as (12a,b,c). In the grammar we will develop in the next section, we will assume that when *pro* is used in place of an overt noun phrase it follows the least marked word order of Japanese sentences, SOV. Thus, for example, in (12c′) above, we assume the nominative-marked *pro* comes first, followed by the accusative-marked *pro*, and finally the verb.

IV.4 Coordination

In the grammar for our fragment, we will also address coordination of various elements. In Japanese, the elements used to create conjunctions (and disjunctions) are also grouped under the name of "particles"; we will disregard this, however, and focus simply on their coordinating features.

The creation of expressions of the forms "*x* and *y*" and "*x* or *y*" in Japanese closely resembles that in English: the equivalents of these two expressions are "*x to y*" and "*x ka y*", respectively. One important difference, however, is that while in English the word "and" can be used to coordinate any types of expressions, in Japanese the domain of *to* is more restricted than this. Most often the coordination of verbs and adjectives is accomplished with inflection of the words themselves, a construction that we will not concern ourselves with here; this construction is the one most often translated into English expressions such as "looked and listened". The parallel disjunction of verbs with *ka* is also not a common construction; expressions such as "I walk or ride my bike" are generally stated differently in Japanese, often to allow use of the "either *x* or *y*" construction (see below).

The creation of expressions such as "both *x* and *y*" also merits discussion[3]. Since Japanese is a head-final language, the coordinating elements follow the expressions they govern, rather than preceding them as in English. In addition, Japanese repeats a single element to create this effect: *mo*. In the terms of the rule Coord from Eng above, we may say that Japanese inserts a second instance of this element, rather than inserting an entirely separate word as English does. An example of a coordinated expression of this form in Japanese is (13):

(13)Hanako mo Tarou mo

'both Hanako and Tarou'

As above, such coordination is most straightforward when applied to nouns, and we will restrict ourselves to coordinating nouns when constructing our grammar for Jpn.

[3] The equivalent of English "neither *x* nor *y*" is achieved in Japanese not with a different set of words but by simply negating the verb at the end of the sentence. For example, "Hanako mo Tarou mo eiga o mita" is "Both Hanako and Tarou saw the movie", while "Hanako mo Tarou mo eiga o minakatta" is "Neither Hanako nor Tarou saw the movie".

(In the case of verbs here, the stem is used instead of the entire inflected verb, and additional elements are required.) Expressions of the form "either *x* or *y*" are created identically to "both *x* and *y*", using *ka* in place of *mo*:

(14)Hanako ka Tarou ka

'either Hanako or Tarou'

In the cases of both *mo* and *ka,* we will posit in the development of our grammar below that one of these repeated elements is inserted by the rule, while the other is an element with category CONJ and is thus acted on by the rule. (In English the first is inserted, but the head-final structure of Japanese may indicate that it is the second instance of the element that is inserted in this language.)

We may now develop our representative fragment of Japanese and our bare grammar for it (Jpn), incorporating these features we have just examined.

V. Jpn

We will now develop Jpn, a grammar for a fragment of Japanese (given as V of Jpn) that represents the features discussed in the previous section. We first provide a definition of it, followed immediately by explanation of the terms used and the rules, in the format used above for our example Eng. Define Jpn = <V, Cat, Lex, Rule> as follows:

(15) V = {*wara-* 'laugh', *ne-* 'sleep', *home-* 'praise', *mi-* 'see', *ga* '-nom', *wo* '-acc', *Hanako, Tarou, to* 'and', *mo* 'both'/'and', *ka* 'either'/'or', *pro, -(r)u* 'NONPAST', *-ta* 'PAST'}

Cat = {NP, Ja, Jn, JPa, JPn, sP1n, sP2, Tense, P0, P1a, P1n, P2, CONJ}

Lex =

NP:	*Hanako, Tarou, pro*
rP1n:	*wara-, ne-*
rP2:	*home-, mi-*
CONJ:	*to, ka, mo*
Ja:	*wo*
Jn:	*ga*[4]
Tense:	*-(r)u*[5], *-ta*

Rule = {Merge, Coord, Mark, Junct}

As above with the presentation of Eng, the table format in which Lex_{Jpn} is given is simply for convenience of presentation: elements of Lex_{Jpn} are pairs, such as (*Hanako*, NP).

---

[4] The reader will note that there are two separate categories for case-marking particles, one for each case, but only a single category for tense-marking elements. In the following section we will address this. We will give a proof that each of the case-marking particles is invariant, and thus cannot be interchanged and preserve grammaticality, which would be possible were they in fact in the same category. From the description in the preceding section of Japanese tense markers, however, the reader may see that if *-(r)u* and *-ta* are interchanged in a verb grammaticality is indeed preserved, thus justifying the decision to group these elements in one category.

[5] Two types of Japanese verbs are typically identified in the literature: *-u* verbs and *-ru* verbs. These names refer to the nonpast ending that the verb takes. In the given roots here, *home-, mi-,* and *ne-* are *-ru* verbs, and so form *homeru, miru,* and *neru* when marked with the nonpast element, while *wara-* is an *-u* verb, and thus forms *warau*. Thus the nonpast tense marker has two forms, represented with the parenthesized *r* here.

First we will discuss the categories of Jpn. The elements and interactions present in Jpn are largely parallel to those of Eng presented above: nouns that combine with n-place predicates to yield (n-1)-place predicates. The case marking present in Japanese, however, complicates the situation and necessitates additional categories to fully explain the construction of grammatical sentences. This process begins with the two categories Ja and Jn, which are accusative-marking and nominative-marking respectively. These combine with elements with category NP, noun phrases that are not yet marked for case, to produce elements with category JPa and JPn (respectively), case-marked noun phrases. (See the description of the rule Mark below.) To correspond with this "splitting" of the general noun phrase category, the verb categories of Jpn specify not only how many arguments they take (as did the P0, P1 and P2 of Eng above) but also what case those arguments must be marked with. For example, elements with the category P1a take one argument, which must be an accusative-marked noun (category JPa). Elements with category P2 take both an accusative and a nominative argument, in either order as discussed in the grammar analysis above. (Note that since there are no verbs that take an accusative argument but no nominative argument, the category P1a is not present in the enumeration of Lex.) Next, to account for tense-marking in Japanese, the categories rP1n, rP2, and Tense have been introduced. Verb roots, which have category rP1n or rP2 (the "r" standing for "root"), combine with tense markers, category Tense, to produce elements with category P1n or P2 respectively. (Again see Mark below.) Elements with category CONJ are, as in Eng, conjunctive or disjunctive elements.

We now turn to our definition of the elements of Rule_{Jpn}. (See section II for description of the column labels.) The first, Mark, combines case marking of nouns and tense marking of verb roots.

(16) Mark:

Domain	Mark	Value
s t Jn NP	→	t⌢-nom JPn
s t Ja NP	→	t⌢-acc JPa
s t Tense rP1n	→	t⌢s P1n
s t Tense rP2	→	t⌢s P2

As explained above, this rule creates case-marked nouns and tense-marked verbs. In the first two lines of this definition, we have replaced "s" with "-nom" or "-acc" after the application of Mark because these case markers are the only elements in their respective categories. Otherwise this rule is quite straightforward. One example of its application is Mark:((-nom, Jn), (*Tarou*, NP)) = (*Tarou*-nom, JPn); another would be Mark:((-*ta*, Tense), (*wara-*, rP1n)) = (*wara-tta*, P1n).

The second rule we consider, Merge, is also fairly easy to understand, as it closely resembles the rule of the same name in Eng. The changes necessary for Japanese consist simply of adjusting Merge to be sensitive to case.

(17) Merge:

Domain	Merge	Value
s t JPn P1n	→	s⌢t P0
s t JPa P1a	→	s⌢t P0
s t JPn P2	→	s⌢t P1a
s t JPa P2	→	s⌢t P1n

Notice that Merge is used to derive expressions with category P1a (a category that was absent from elements of Lex_{Jpn}). An example of such a derivation is Merge((*Tarou*-nom, JPn), (*hometa*, P2)) = (*Tarou*-nom *hometa*, P1a). This is an expression that has a position for one more argument, which must be accusative-marked—for example, it could combine, again by Merge, with (*Hanako*-acc, JPa) to form the expression (*Hanako*-acc *Tarou*-nom *hometa*, P0).

The rule Coord acts largely as described in Eng above, creating expressions with form equivalent to English expressions such as "both *x* and *y*". As discussed in the previous section, we will deal here only with the coordination of nouns, as the rules applying to verbs and other objects are more complex.

(18) Coord:

Domain			Coord	Value	Conditions required
mo CONJ	s NP	t NP	→	s⁀mo⁀t⁀mo NP	s, t ≠ *pro*
ka CONJ	s NP	t NP	→	s⁀ka⁀t⁀ka NP	s, t ≠ *pro*

We exclude *pro* because allowing one of the strings being coordinated to be *pro* would create difficulties: the presence of the null DP would create sentences equivalent to the English "Both Bob and walked" and "I saw both and John". To create the sentence in (13) in the above section, we would apply this rule thus: Coord((*mo*, CONJ), (*Hanako*, NP), (*Tarou*, NP)) = (*Hanako mo Tarou mo*, NP)[6]. The rule Junct acts much like Coord, but lacks the automatic insertion of the extra (repeated) element.

---

[6] In future work on this bare grammar, one issue that should be addressed is that no case-marking particle follows phrases created with the rule Coord. For example, the sentence "Hanako mo Tarou mo eiga o mita" from a footnote in the above section is correct, while "Hanako mo Tarou mo ga eigo o mita" is not. This same fact applies

(19) Junct:

Domain	Junct	Value	Conditions required
to s t CONJ NP NP	→	$s \frown to \frown t$ NP	s, $t \neq$ *pro*
ka s t CONJ NP NP	→	$s \frown ka \frown t$ NP	s, $t \neq$ *pro*

Thus, for instance, we can use this rule in the derivation Junct((*to*, CONJ), (*Hanako*, NP), (*Tarou*, NP)) = (*Hanako to Tarou*, NP).

Once we have established a working fragment and grammar, we may proceed to investigating their linguistic invariants.

---

when the coordinated expression is in the position that would normally take the accusative case, but according to our rule a coordinated expression is not yet assigned case. Nor can we define this rule to assign case, as coordinated expressions may go in nominative or accusative positions.

VI. Some Invariants of Jpn

Invariants play a key role in the Bare Grammar theory, since they allow comparisons cross-linguistically, leading toward the goal of a generalized theory of grammar. We will here concern ourselves with three objects that we wish to show are invariant: L_{Jpn}, and the nominative and accusative case-marking particles (*ga* and *o*).[7]

The technical proof that L_{Jpn} is invariant is quite straightforward. Recall that L_{Jpn} is the language generated by the grammar Jpn.

<u>Proof</u> By definition (given in Section III), given an arbitrary automorphism $h \in Aut_{Jpn}$, $h(L_{Jpn}) = \{h(s) \mid s \in L_{Jpn}\}$. By definition of being an automorphism,

---

[7] We are, in writing this section, operating under the assumption that Jpn is unambiguous, under Keenan and Stabler's definition of ambiguity:

Definition. A grammar G is <u>ambiguous</u> if and only if at least one of the following holds:

1. One of the generating functions fails to be one-to-one
2. The ranges of two generating functions overlap
3. The range of a generating function overlaps with Lex_G.

If a grammar is ambiguous, its automorphisms cannot be properly defined on the elements of Rule, and thus these automorphisms may be missing properties necessary for proving the invariance of certain objects, including L_G.

In Keenan and Stabler's 2003 monograph, they point out in their development of Eng that changing the Coord rule so that it does not insert "both" (or "either", as their version of Eng included disjunction unlike our adapted version in this thesis) allows for multiple possible derivations of expressions such as (Either John or Bill and Sam laughed, P0). In this example, G is ambiguous because Coord fails to be one-to-one. Sentences like this one are ambiguous in Japanese just as they are in English, so it is possible and even likely that our rule Junct does in fact make Jpn ambiguous, which in turn throws into question the validity of any of these proofs since they rely on automorphisms and their properties.

Another possible source of ambiguity in our grammar Jpn is its reliance on *pro* in topic-drop situations. As stated above, we are assuming in this thesis that the insertion of *pro* follows the word order SOV. However, the orders SOV and OSV are equally acceptable in Japanese, and there is in fact no indication in sentences that have undergone topic-drop which place the null element is actually occupying. This again gives us sentences that have multiple potential derivations.

As we said, we will disregard these difficulties for the purpose of the proofs for this section. Still, this would seem to be an aspect of BG that calls for closer examination. There do in fact exist ambiguous sentences in languages, as shown just above, and thus in order for BG to truly represent languages it must find a way to account for this. Ignoring it certainly makes for a cleaner model, and serves as a useful starting point, but eventually this issue needs to be addressed.

h:L_{Jpn}→L_{Jpn} is surjective. Thus h acts on every element s of L_{Jpn} and maps to all of L_{Jpn}—so $h(L_{Jpn}) = L_{Jpn}$, i.e. L_{Jpn} is invariant.

In plain English, this simply means that being grammatical is an invariant property: if an expression that is grammatical undergoes a structure-preserving transformation, the resulting expression will also be grammatical. To rephrase in somewhat more technical terms, if an expression that is in the language, i.e. derivable from applying elements of $Rule_{Jpn}$ to elements of Lex_{Jpn} finitely many times, is acted on by an automorphism of the language then the result will also be in the language.

We will next show that the nominative case marker, -nom, is invariant.

Proof Let h be an element of Aut_{Jpn}, n be an element of category NP, t be an element of category Tense, and v be an element of either category rP1n or rP2. Suppose that h(-nom) ≠ -nom; we will show this leads to a contradiction.

Consider Merge(Mark(-nom, n), Mark(t, v)). By the definition of Mark in the previous section, Mark(-nom, n) produces an expression with category JPn. So, in order for this to be a valid application of the rule Merge—i.e. for (Mark(-nom, n), Mark(t, v)) to be in the domain of Merge—v must have the category rP1n.

Now we apply h, and obtain

$$h(\text{Merge}(\text{Mark}(\text{-nom}, n), \text{Mark}(t, v \in rP1n))) = \text{Merge}(\text{Mark}(h(\text{-nom}), h(n)), \text{Mark}(h(t), h(v))).$$

We appeal again to the restrictions on the domains of the rules in Jpn. For the expression (Mark(h(-nom), h(n)), Mark(h(t), h(v))) to be in the domain of Merge, the first element, Mark(h(-nom), h(n)), must have category either JPa or JPn, and the second element must have category P1n, P1a, or P2. This implies that h(n) has category NP, while h(-nom) is either -nom or -acc. Since by supposition h(-nom) does not equal -nom, it must be -acc. So the result of Mark(h(-nom), h(n)) has category JPa. By argument similar to above, this means that the category of h(v) must be rP2, so that the result of Mark(h(t), h(v)) must have category P2.

Then the entire expression Merge(Mark(h(-nom), h(n)), Mark(h(t), h(v))), by definition of Merge, must have category P1n. We can thus apply Merge once more to add an element, which we'll call n′, of category JPn and create a P0.

Now consider h^{-1}, the inverse function of h, acting on this new application of Merge. (Differing brackets are used only for ease of reading, not to indicate any difference in meaning.)

$$h^{-1}(\text{Merge}[(n', \text{JPn}), \text{Merge}\{\text{Mark}(\text{-acc}, h(n)), \text{Mark}(h(t), h(v))\}])$$

$$= \text{Merge}[(h^{-1}(n'), \text{JPn}), h^{-1}(\text{Merge}\{\text{Mark}(\text{-acc}, h(n)), \text{Mark}(h(t), h(v))\})]$$

$$= \text{Merge}[(h^{-1}(n'), \text{JPn}), \text{Merge}\{\text{Mark}(h^{-1}(h(\text{-nom})), h^{-1}(h(n))), \text{Mark}(h^{-1}(h(t)), h^{-1}(h(v)))\}]$$

$$= \text{Merge}[(h^{-1}(n'), \text{JPn}), \text{Merge}\{\text{Mark}(\text{-nom}, n), \text{Mark}(t, v)\}]$$

$$= \text{Merge}[(h^{-1}(n'), \text{JPn}), \text{Merge}\{(n\frown\text{-nom}, \text{JPn}), (v\frown t, \text{P1n})\}]$$

$$= \text{Merge}[(h^{-1}(n'), \text{JPn}), \text{P0}].$$

But nowhere in the domain of Merge is there a P0, so this is a contradiction. Thus for our arbitrary $h \in \text{Aut}_{\text{Jpn}}$, h(-nom) = -nom, implying that -nom is invariant.

The proof that -acc is invariant is parallel. The invariance of these two case markers tells us that they cannot be interchanged (thus justifying our assigning them separate categories—see previous section), either with each other or with elements that have any other category.

VII. Conclusion

In this thesis we have presented an effort to extend the Bare Grammar model of Keenan and Stabler to Japanese. After summarizing the necessary concepts, we developed a fragment of Japanese and applied BG to it. The results of this highlighted a few areas not sufficiently addressed in Keenan and Stabler's 2003 work, but for the most part BG provided a good framework for describing the features of the Japanese language studied in this thesis.

By obtaining such results, we have made a step not only toward expanding the range of languages to which BG has been applied and thus giving it increased credibility, but also toward improving BG as a model. Clearly, as more issues are addressed by BG it will grow into a model even more capable of accounting for the many diverse features found in human languages. Each of the languages whose grammars are developed in the 2003 work, as well as that of Japanese developed here, helps push BG toward this ultimate goal.

As this is an undergraduate thesis, written in the space of only a single semester, we were not able to include every issue that could have been addressed. Future work from this point would first need to address the difficulties with the fragment Jpn discussed above, particularly those regarding ambiguity. A related point to keep in mind is that in actual human languages there do exist ambiguous sentences, and a model will not be complete without accounting for these. The more recent work of both Keenan and Stabler should be checked to determine whether any of these issues with Bare Grammar discovered here have been found and addressed. In addition, one primary future goal would be completion of proofs of more invariants in Jpn. Given the limited timeframe, our main object in writing this thesis has been to establish the basis for any future work on applying Bare Grammar to Japanese, i.e. determining a working fragment of the language on which to base proofs. However, we have made as much progress as possible toward these proofs of invariance, and hope it will at the least provide other interested parties with a starting point.

Appendix: Mathematical Notation

$a \in A$	"(object) a is an element of (set) A"
A^*	the set of finite sequences of elements of A
$V \times Cat$	pairs (s, C) consisting of a vocabulary item s and category C of that vocabulary item
$(V^* \times Cat)^+$	sets $\{(s_1, C_1), (s_2, C_2), \ldots, (s_n, C_n)\}$ of any finite number n of elements (s_i, C_i) of $V^* \times Cat$
$f: A \rightarrow B$	a function f mapping from A to B
$f(a)$	function f applied to element a
$f(A)$	function f applied to all elements $a \in A$
$\frown$	concatenation

s, t, n, n', t, v	variable names; used for strings (elements of V)
C, C'	variable names; used for categories (elements of Cat)
h	used as a name for automorphisms
F	used as a name for a function in Rule

Works Cited

1. Carnie, Andrew. Syntax: A Generative Introduction. 2nd ed. Malden, Mass.: Blackwell Publishing, 2007.
2. Chomsky, Noam. The Minimalist Program. Cambridge, Mass.: MIT Press, 1995.
3. Huang, C.T. James. "Pro-Drop in Chinese." The Null Subject Parameter. Ed.s Osvaldo Jaeggli, Kenneth J. Safir. Boston, Mass.: Kluwer Academic Publishers, 1989. 185-214.
4. Keenan, Edward L. and Edward P. Stabler. Bare Grammar: Lectures on Linguistic Invariants. Stanford, Calif.: CSLI Publications, 2003.
5. ---. "Universals Across Languages." Paper presented at the Workshop on Model Theoretic Syntax, European Summer School on Logic, Language and Information, August 2007. < http://www.linguistics.ucla.edu/people/stabler/epspub.htm>