# Current Topics in JPSG*

HARADA Yasunari

What follows is an attempt to give some idea of what kind of a grammatical endeavor JPSG is like, what kinds of discussions are going on in the working group meetings held almost every week at ICOT, what obstacles we are facing during the process of formulating a phrase structure grammar for the Japanese language. First I will quite informally introduce some of the concepts that underlie the whole JPSG project. Then I will deal with some topics of the relatively recent discussions of our group. All materials presented here are tentative formulations of ideas that emerged out of discussions among members of the ICOT JPSG Working Group. None of the arguments and formulations presented here should be taken to be definitive.

## 1. an informal introduction to JPSG

What is JPSG? In a sense, it is quite easy to answer this question. My answer could be: It is another unification-based grammatical theory. On the other hand, it is quite difficult to answer a second question: What are the differences between JPSG and other unification-based grammatical theories? People today sometimes claim that there are not many differences among various unification-based theories other than their backgrounds, preferred notational systems and research inclinations.

When JPSG project started 3 years and 4 months ago, it was supposed to be an attempt to construct a linguistically well-motivated, mathematically well-defined, and in some sense 'practical' parser for the Japanese language. Linguistically well-motivated, in the sense that it would not be something made up of thousands of phrase structure rules. Mathematically well-defined, in the sense that we would have some real theory in which to define concepts

that are utilized in the grammatical description of Japanese. Practical, in the sense that the resulting parser would be able to cope with, hopefully, most of the sentences that we would find in textbooks for elementary school children.

Thus, we were supposed to produce a specific software product, namely a parser and related tools, in a specific programming language, namely an extension of Prolog[1] that researchers at ICOT were then developing, for a specific natural language, namely Japanese, to be utilized on a particular machine called PSI (personal sequential inference machine).[2]

After some preliminary discussions which lasted about a year, during which period we compared various existing frameworks, we decided to employ the basic ideas of GPSG and HPSG as grammatical description formalism[3], and as the central concept underlying our theoretical framework, we employed the notion of what we now call constraint-unification.[4] And as we started to develop semantic description for Japanese, we are heavily influenced by the ideas of situation theory and situation semantics[5], although because of my background and research interest, I will not refer to that aspect of JPSG in the remainder of this article.

What are the differences of JPSG and HPSG? Three years ago, it may have been possible to characterize the differences in the following way. Although the underlying ideas of HPSG did not depend on its procedural definition of various concepts utilized therein, they were still defined in bottom-up procedural terms, whereas JPSG was going to be defined purely in declarative terms. Also, the underlying notion of unification is extended in such a way that it can directly cope with disjunctive specification of feature values.[6]

Today, however, it is very difficult to find any differences between JPSG and HPSG. Most of the grammatical concepts in Pollard and Sag (1988) are written in declarative terms. The emphasis was more on declarative grammar than in earlier formulations of the same framework. As for extension of unification, it is the most up-to-date topic among people doing computational linguistics within unification-based paradigm. In fact, partly because we find less and less differences between our theory in Tokyo and the theory being developed at CSLI, and partly because we are rather bored of these alphabetic

variations, we started to call it simply "phrase structure grammar", without any preceding alphabet.[7]

In any event, I will try to formulate our understanding of how grammatical theory should be conceived of at the beginning of this informal presentation.

### 1.1 what constraint unification is all about

In our framework, a grammatical object such as grammatical categories and phrase structure rules will consist of what we call "constrained patterns". A constrained pattern is a pattern together with constraints that obtain among variables that appear in that pattern. Unification of constrained patterns are defined in such a way that all constraints regarding each variable within constrained patterns involved hold in the resulting constrained pattern.

Constraint unification is an extension of unification; it was conceptualized so as to facilitate description of grammar by incorporating such aspects as disjunction into unification mechanism itself. I will give you some simple examples that would show how it works. In what follows, constrained pattern will be designated by Prolog-like expressions. When there are no constraints involved, they are like Prolog terms. Very often, they will be names starting with a lowercase letter, followed by designation of constants and variables, as shown in (1.1.1). Just like in Prolog, we designate constants by names starting with a lowercase letter and variables with names starting with an uppercase letter. Constraints on the variables that appear on the pattern follow the symbol ":–". Although the 'meaning' of this symbol is quite different here from its ordinary use in standard Prolog, the similarity would help understand what the expressions in (1.1.2) and (1.1.3) are intended to signify. Thus, all expressions in (1.1.1) through (1.1.3) are legitimate designations of constrained patterns with their names specified as "object".

(1.1.1)

a. object(X, a).
b. object(b, Y).
c. object(b, a).

(1.1.2)

a. object(X, a):–member(X, [b, c]).
b. object(c, Y):–member(Y, [a, b]).
c. object(c, a).

(1.1.3)

a. object(X, Y):–member(X, [a, c]),member(Y, [b, d]).
b. object(X, Y):–member(X, [a, c]),member(Y, [b, c]).
c. object(X, b):–member(X, [a, c]).

In each of (1.1.1), (1.1.2) and (1.1.3), if we try to unify the constrained pattern in (a) with that in (b), unification succeeds with the resulting constrained pattern in (c), given ordinary list-membership relation between List and Member in member(Member, List).

A grammatical category is traditionally thought of as an object consisting of feature specifications, where a feature specification consists of a feature name and a value assigned to it. In the framework we are considering here, a grammatical category is also an object consisting of feature specifications, but we also allow variables to be present in its description along with designations of constraints that obtain among these variables. So a grammatical category might look something like this.

(1.1.4) a grammatical category

```
category( ... , X1, ... , X2, ... ):–
    cond_1_1(X1),
    cond_1_2(X1),
    ..... ,
    cond_2_1(X2),
    cond_2_2(X2),
    ..... ,
    cond_3_1(X1, X2),
    cond_3_2(X1, X2),
    ..... ,
```

..... .

Among the conditions involved, here we have in mind, for instance, conditions on what values a given feature can take, or conditions on what relations hold among feature specifications, such as those researchers in GPSG called Feature Cooccurrence Restrictions. [Gazdar et. al. 1985: p. 33]

Likewise, a phrase structure rule is a relation that hold of several grammatical categories. For instance, a binary branching phrase structure rule is a relation among three grammatical categories, the mother and its two daughters. In our framework, variables are allowed in the description of a given phrase structure rule along with constraints that must hold of these variables.

(1.1.5) a binary branching phrase structure rule

```
psr( ... , Mother, Left, Right, ... ):-
    principle_1(Mother, Left, Right),
    principle_2(Mother, Left, Right),
    ..... ,
    ..... .
```

A description of a fragment of a natural language such as English is determined when we (1) define what objects constitute grammatical categories in that language, (2) state what phrase structure rules are to be utilized in that language and (3) specify what lexical items belong to what syntactic category or categories in that language.

Although we conceive of grammatical categories and phrase structure rules as constrained patterns, the foregoing notation is slightly too messy to follow. Thus, we utilize another set of notational conventions, in which these grammatical objects are given a little more familiar outlook. Thus, in (1.1.6), some examples of grammatical categories in this notation are given.[8]

(1.1.6) some examples of grammatical categories

```
NP    [pos n, subcat <>, spec <>, sem m']
      n[]:m'
```

PP [pos p, subcat <>, spec <>, form for]
p[for]
VP [pos v, spec <[pos n, subcat <>, sem X] >
form fin, sem love'(X, j')]
v[fin, spec <n[]:X >]:love'(X, j')
P [pos p, subcat <[pos n, subcat <>,
case acc, sem j'] >, form on]
p[subcat <n[acc]:j' >, on]
V [pos v, subcat <[pos n, subcat <>] >,
spec <[pos n, subcat <>, pn 3s] >]
v[subcat <n[] >, spec <n[3s] >]

In (1.1.7) and (1.1.8) feature names are given along with indications of what each feature is intended to signify and possible ranges of values for these features.

(1.1.7) features and their values for English

| | | |
|---|---|---|
| pos | part of speech | { n, v, p, a, det } |
| pn | person and number | { nil, 1s, 2s, 3s, 1p, 2p, 3p } |
| case | case | { nil, nom, poss, acc } |
| spec | specifier | list of categories |
| subcat | complement | list of categories |
| slash | syntactic gap | list of categories |
| form | verb form | { nil, base, fin, presp, pastp } |
| sem | semantics | semantic representations |

(1.1.8) features and their values for Japanese

| | | |
|---|---|---|
| pos | part of speech | { n, v, p, adv, adn, ... } |
| gr | grammatical relation | { sbj, obj, ... } |
| subcat | complement | set of categories |
| slash | syntactic gap | set of categories |
| morph | morphology | morphological representations |
| infl | inflection | { vc, vv, ... } |

| | | |
|---|---|---|
| form | postposition form | { ga, wo, ni, ... } |
| sem | semantics | semantic representations |

Some examples of phrase structure rules are given below. At this point, we cannot go into the discussions of whether they are linguistically proper or not. Our concern here is how they would look like in our notation and formalism.

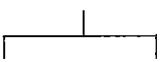
(1.1.9) phrase structure rules

a. specification

Mr ⟶ Sp Hd

where spec(Hd) = <Sp|spec(Mr) >

a'. P[spec <>, ... ]:S



X P[spec <X >, ... ]:S

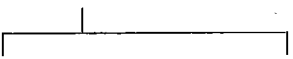
a". psr(Mr, Sp, Hd):– cu(spec(Hd), <Sp|spec(Mr) >).

b. complementation

Mr ⟶ Hd Ct

where subcat(Hd) = <Ct|subcat(Mr) >

b'. P[subcat <>, ... ]:S



P[subcat <X >, ...]:S X

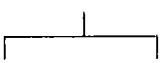
b". psr(Mr, Hd, Ct):– cu(subcat(Hd), <Ct|subcat(Mr) >).

c. topicalization

Mr ⟶ Bdr Hd

where slash(Hd) = <Bdr|slash(Mr) >, pos(Hd) = v

c'. P[slash <>, ... ]:



X P[slash <X >, ... ]:S

c". psr(Mr, Bdr, Hd):- cu(slash(Hd), <Bdr|slash(Mr) >),
cu(pos(Hd),v).

In these examples, (a), (b), and (c) are the phrase structure rules in our 'readable' notation, (a'), (b'), and (c') are the corresponding local structures that these rules will sanction, and (a"), (b") and (c") are their internal representations in the underlying Prolog-like notation. The expressions that follow "where" in (a), (b) and (c) are the clauses in which constraints or conditions to be satisfied are stated. Here, an expression of the form Feature_name(Category_designator) refers to the value of the feature designated by Feature_name with respect to the grammatical category referred to by Category_designator. The symbol "=" represents constraint-unification.

In our theory, phrase structure rules are like any other object; a constrained pattern. Thus we can talk about unification of phrase structure rules. For instance, phrase structure rules as given above are of course too unrestricted, and must be unified with syntactic principles such as HFP, FFP and SFP that are discussed in GPSG literature.[9] If these three are in fact the only syntactic principles that should be incorporated in an adequate description of English or Japanese grammar, something like (1.1.10) has to be constraint-unified with each and every phrase structure rule in the language in question. In a sense, syntax of Japanese or English is all these phrase structure rules constraint-unified.

(1.1.10)

a. psr(Mr, D1, D2) :- hfp(Mr, D1, D2).
b. psr(Mr, D1, D2) :- ffp(Mr, D1, D2).
c. psr(Mr, D1, D2) :- sfp(Mr, D1, D2).

Actually, we have a slight conflict between linguistic tradition of syntactic descriptions and computational tradition of logic programming executives designs. Thus, if a given syntactic principle realizes differently with respect to local phrase structures or phrase structure rules involved, we would have to have some sort of tags in order to differentiate applications of various

subconditions of these principles. Thus, each phrase structure rule will invoke conjunction of relevant syntactic principles with a particular tag as shown in (1.1.11), while each syntactic principle will be a disjunction of conditions with these tags as shown in (1.1.12).

(1.1.11)

a. specification

```
psr(specification, Mr, Sp, Hd):-
  hfp(specification, Mr, Sp, Hd),
  ffp(specification, Mr, Sp, Hd),
  sfp(specification, Mr, Sp, Hd),
  ..... .
```

b. complementation

```
psr(complementation, Mr, Hd, Ct):-
  hfp(complementation, Mr, Hd, Ct),
  ffp(complementation, Mr, Hd, Ct),
  sfp(complementation, Mr, Hd, Ct),
  ..... .
```

c. topicalization

```
psr(topicalization, Mr, Bdr, Hd):-
  hfp(topicalization, Mr, Bdr, Hd),
  ffp(topicalization, Mr, Bdr, Hd),
  sfp(topicalization, Mr, Bdr, Hd),
  ..... .
```

(1.1.12)

a. head feature principle

```
hpf(specification, MH, LH, RH):-
  condhfp1(MH, LH, RH).
hfp(complementation, MH, LH, RH):-
  condhfp2(MH, LH, RH).
hfp(topicalization, MH, LH, RH):-
  condhfp1(MH, LH, RH).
```

```
.....
b.  foot feature principle
      ffp(specification, MF, LF, RF):-
        condffp1(MF, LF, RF).
      ffp(complementation, MF, LF, RF):-
        condffp2(MF, LF, RF).
      ffp(topicalization, MF, LF, RF):-
        condffp3(MF, LF, RF).
      .....
c.  subcat feature principle
      sfp(specification, MS, LS, RS):-
        condsfp1(MS, LS, RS).
      sfp(complementation, MS, LS, RS):-
        condsfp2(MS, LS, RS).
      sfp(topicalization, MS, LS, RS):-
        condsfp1(MS, LS, RS).
      .....
```

Within current logic programming environments, expressing linguistic generalizations in their smplest forms is not something attained easily.[10]

The foregoing story about everything in our theory being just constrained patterns, or patterns with conditions, also applies to the component of grammar usually referred to by the term "lexicon". That is, lexicon specifies relations between linguistic representations and the grammatical categories they correspond to. In that sense, it is no different from other 'component' of grammar. In fact, it is also patterns together with constraints as shown below.

(1.1.13)

```
lex(Spelling, Category):-
  condition_s_1(Spelling),
  .....
  condition_c_1(Category),
  .....
```

condition_sc_1(Spelling, Category),

..... .

Below, lexical entries for *John* and *loves* are given in our 'easy-to-read' notation as (a) and (b) respectively. Actually, however, we conceive of them more as those constrained patterns shown in (a') and (b').

(1.1.14) lexical entries

a. John |= n[3s, case X]:j'
where X = acc or X = nom

a'. lex(John, n[3s, case x]:j'):- member (X, [acc, nom]).

b. loves |= v[fin, spec <n[nom, 3s]:X >,
subcat <n[acc]:Y >]:love'(X, Y)

b'. lex(love, v[fin, spec <n[nom, 3s]:X >,
subcat <n[acc]:Y >]:love'(X, Y)).

What we call lexical rules are in fact the same kind of objects, constrained patterns, specifying relations that hold between linguistic representations and grammatical categories or feature specifications. Thus, third person singular lexical rule, which states the regular correspondence between infinitival and third-person-singular form of a verb would look something like this.

(1.1.15) third person singular lexical rule

a. Vs |= [fin, spec <n[nom, 3s]:X >]
if V |= [inf, spec <n[]:S >]
where Vs tpsf V

a'. lex(Vs, v[fin, spec <n[nom, 3s]:X >], R):-
lex(V, v[inf, spec <n[]:X >, R],
tpsf(Vs, V).

Of course, (1.1.14.a', b') and (1.1.15.a') are not really the internal representations for these lexical entries and lexical rule, but let's not complicate our explication too much.

Thus what we call lexical rules are not really procedural rules that are 'applied' to lexical entries to produce derivative lexical entries. We can talk about unifying a lexical entry with a lexical rule, thereby obtaining virtually the source and the resultative lexical entry. Lexicon in this sense is all the lexical entries and all the lexical rules constraint-unified.[11]

### 1.2 a fragment of Japanese grammar

Although we do not want to commit ourselves to premature or unwarranted universal claims, we are certain that a great deal could be treated uniformly between English and Japanese grammar. Thus we believe that the same overall framework is capable of dealing with these two languages. On the other hand, we have to postulate different features or specify different value ranges for the same features in some cases, and state different clauses for the same syntactic principles in order to adequately describe their syntax. Although we do not have time or space to present any comprehensive grammar for Japanese here, I will briefly sketch a very tiny fragment of it to show what kind of differences there could be if one tried to describe Japanese in our framework.

There are probably many contrasts between English and Japanese syntax, but let us for the time being concentrate on the following two distinctions. (1) Japanese is a verb-final or head-final language. (2) Japanese is a so-called 'inflectional' language. With these differences in mind, I will present a small fragment of Japanese. Among the things we will be concerned with here, the following will be important: (1) how to deal with the semantics of PP, (2) how to describe scrambling of grammatical elements, and (3) how to treat verb final elements.

Some of the features that are indispensable in a proper description of Japanese syntax is given below, with possible ranges for their values.

(1.2.1)

| | |
|---|---|
| pos | { n, v, p, adv, adn, ... } |
| gr | { sbj, obj, ... } |
| subcat | set of categories |

| | |
|---|---|
| adjacent | null or singleton set of categories |
| morph | morphological representations |
| infl | { vc, vv, ... } |
| form | { ga, wo, ... } |
| sem | semantic representations |

Given these features, our first phrase structure rule will be this.

(1.2.2) complementation 1 (free order)

$$M \longrightarrow C \quad H$$

where

$$\text{sem}(M) = \text{sem}(H),$$
$$\text{subcat}(M) \cup \{C\} = \text{subcat}(H),$$
$$\text{head}(M) = \text{head}(H)$$

This first rule will take care of the cases that would be dealt with in traditional phrase structure rules such as these.

(1.2.3)

a. $S \longrightarrow VP$

b. $VP \longrightarrow PP\ V$

c. $PP \longrightarrow N\ P$

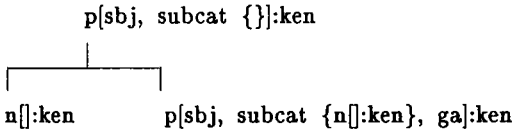
A question might be raised as to how we would treat PPs. Postpositions in Japanese sometimes appear to make intrinsic semantic contributions, but in cases like *ga, wo, ni,* it seems that they have little semantic content. This would correspond to the case of English *to* in dative constructions. So it might be argued that these are just some terminal symbols or case markers. In our analysis that I am presenting here, we assume, for the sake of uniform treatment of PPs, that the case marking postpositions are assigned the following feature specifications.

(1.2.4)

a. p[morph ga, gr sbj, subcat {n[]:S}, form ga]:S
b. p[morph wo, gr obj, subcat {n[]:S}, form wo]:S
c. p[morph ni, gr obj, subcat {n[]:S}, form ni]:S

This feature specification and the phrase structure rule above will sanction local phrase structures of the following form, among others.[12]

(1.2.5)




Although the postposition is the head, information regarding the semantics of the mother is determined entirely on the basis of information regarding the semantics of the noun rather than the postposition of the postposition phrase.

The fact that Japanese is a head-final language is reflected in our formulation of the phrase structure grammar given so far. In (1.2.2) the head follows the complement, rather than the other way around. Next, let us see how scrambling, or free ordering of postposition phrases are to be effected in our formulation. The fact to be described is simply this; if we have more than two complements to a verb, they can appear in whatever order.

In the case of intransitive verbs like *hasiru*, no reordering is possible because there is only one complement involved.

(1.2.6)

a. taroo-ga hasiru

In the case of transitive verbs like *kizutukeru*, two alternative orderings are allowed.

(1.2.7)

a. taroo-ga naomi-wo kizutukeru
b. naomi-wo taroo-ga kizutukeru

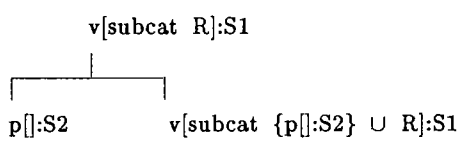
In the case of ditransitive verbs like *okuru*, six orderings are in principle conceivable.

(1.2.8)

a. taroo-ga naomi-ni hana-wo okuru
b. taroo-ga hana-wo naomi-ni okuru
c. naomi-ni taroo-ga hana-wo okuru
d. naomi-ni hana-wo taroo-ga okuru
e. hana-wo taroo-ga naomi-ni okuru
f. hana-wo naomi-ni taroo-ga okuru

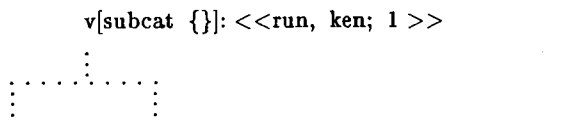
How do we sanction these variations? There are two conceivable solutions to this question. One is to have subcat that would take lists of categories as its value and postulate lexical rules that would 'scramble' or rearrange the subcat value list of verbs. The other is what we have here. Subcat will take sets rather than lists of categories as its value, so that phrase structure rule as given in (1.2.2) will sanction the following canonical local structure.

(1.2.9)

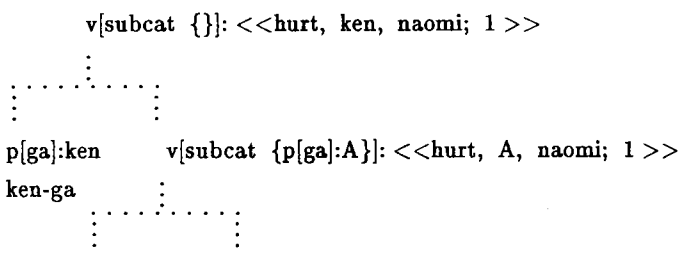



It would be quite easy to see that in the case of intransitive verbs, our grammar will sanction structures like (1.2.10), whereas in the case of transitive verbs, both (1.2.11) and (1.2.12) will be allowed.

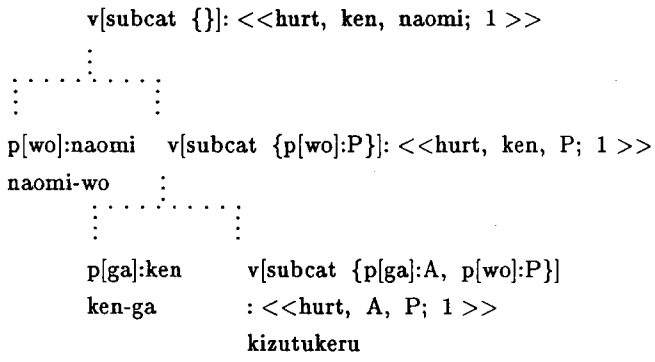
(1.2.10)




(1.2.11)




(1.2.12)




It would not be necessary to give tree diagrams to show how this is done for triadic verbs. Thus what the feature subcat shows about the category

involved is the set of those complements that can come in whatever order they like.

However, there are cases in which a given complement must be adjacent to the head in question. One such case is the causative construction. In order to show how a causative sentence is treated in our framework, we have to refer to the features infl, and adjacent, and also think about how control is treated.

The sentence we will discuss here is something like this.

(1.2.13)

ken-ga naomi-ni hon-wo yom-aseru

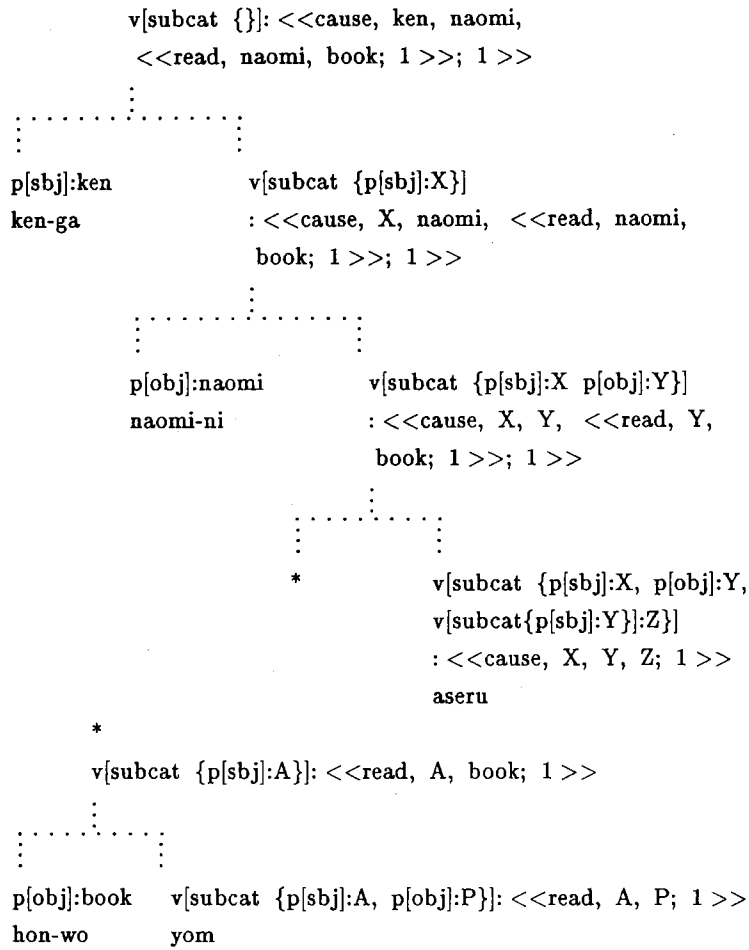
First of our assumption is that we have here a verbial element which subcategorizes for a subject, an object, and an intransitive verb phrase. The intended subject of the embedded verb phrase is of course the object of the sentence. That's what causative constructions are all about. So lexical entry for causative *aseru* would look something this;

(1.2.14)

v[morph aseru, subcat {p[sbj]:X, p[obj]:Y,
v[subcat {p[sbj]:Y}:Z}]: <<cause, X, Z; 1 >>

Given this lexical entry, structures like (1.2.15) will be sanctioned.

(1.2.15)




Now the problem here is that our complementation rule as formulated above would not dictate that *yom* must immediately precede *aseru*. In fact, however, *yom* must immediately precede *aseru*, forming what might be called a phonological phrase.[13]

Our problem here is that subject PPs or object PPs cannot intervene with

the verb phrase and causative *aseru*. Thus we have to have a different feature called adjacent, specifying that the category designated as the value for this feature must immediately precede the category involved. Two alternative formulations of this idea is conceivable. One is to have redundant information in subcat and adjacent. The other is to have subcat and adjacent in such a way that their union designates the set of complements involved. Here we employ the second strategy. Thus, another phrase structure rule is required to cancel adjacent feature value.

(1.2.18) complementation 2 (adjacent)

M $\longrightarrow$ C H

where

sem(M) = sem(H),
adjacent(M) = {}, adjacent(H) = {C},
head(M) = head(H)

In order for this idea to work, we have to add another condition on the complementation 1.[14]

(1.2.19) complementation 1 (non adjacent)

M $\longrightarrow$ C H

where

sem(M) = sem(H),
subcat(M) $\cup$ {C} = subcat(H),
head(M) = head(H),
adjacent(M) = adjacent(H) = {}

Lexical specification for the verbial *aseru* would have to be changed accordingly.

(1.2.20)

v[subcat {p[sbj]:X, p[obj]:Y}, adjacent{v[subcat{p[sbj]:Y}]:Z}]
: <<cause, X, Z; 1 >>

Given this adjacent feature, we have to reformulate our analysis of PPs. That is, since no grammatical element can intervene between nouns and postpositions, nouns are the adjacent complements of postpositions. Thus the following lexical entries.

(1.2.21)

a. p[morph ga, gr sbj, adjacent {n[]:S}, form ga]:S
b. p[morph wo, gr obj, adjacent {n[]:S}, form wo]:S
c. p[morph ni, gr obj, adjacent {n[]:S}, form ni]:S

Given these phrase structure rules and lexical entries, we can get the desired phrase structures for causative constructions. However, there are several remaining complications. The first is the possibility of scrambling out of the complement verb phrase. Thus the following sentences seems acceptable.

(1.2.22)

a. ken-ga naomi-ni sono hon-wo yom-aseru
b. ken-ga sono hon-wo naomi-ni yom-aseru
c. sono hon-wo ken-ga naomi-ni yom-aseru

Although it is possible to treat these constructions with slash features, we could modify our lexicon in the following way to treat these constructions.

(1.2.23)

lex(M, v[adjacent <[subcat S1 ∪ {P}, R1] >, subcat S2 ∪ {P}, R2]):-
lex(M, v[adjacent <[subcat S1, R1] >, subcat S2, R2]).

One remaining problem is the choice of particular morphological form according to the kind of particular verb. Thus a given verb would combine with causative *saseru* rather than *aseru* and also with passive *rareru* rather than *areru*, whereas for another verb things would be quite the opposite. We postulate a feature called infl which distinguishes between these two types

of inflectional combinations, regarding the use of causative *saseru/aseru* and passive *rareru/areru*.

(1.2.24)

| | | | |
|---|---|---|---|
| a. | iw-u | iw-aseru | iw-areru |
| b. | kak-u | kak-aseru | kak-areru |
| c. | tat-u | tat-aseru | tat-areru |
| d. | yom-u | yom-aseru | yom-areru |
| e. | hur-u | hur-aseru | hur-areru |
| f. | ki-ru | ki-saseru | ki-rareru |
| g. | mi-ru | mi-saseru | mi-rareru |
| h. | yase-ru | yase-saseru | yase-rareru |
| i. | e-ru | e-saseru | e-rareru |

So we would have virtually two lexical entries for *aseru* and *saseru*.

(1.2.26)

a. v[morph saseru, subcat {p[sbj]:X, p[obj]:Y},
adjacent{v[infl vv, subcat{p[sbj]:Y}]:Z}]: <<cause, X, Y, Z; 1 >>

b. v[morph aseru, subcat {p[sbj]:X, p[obj]:Y},
adjacent{v[infl vc subcat{p[sbj]:Y}]:Z}]: <<cause, X, Y, Z; 1 >>

Of course these two would not be specified as such; lexical rules can take care of this kind of redundancy, the details of which I would not go into here. Roughly it would look something like this.[15]

(1.2.27)

```
MORPH1 |= v[adjacent{v[infl vc, R1]:Z}, R2]
if
MORPH2 |= v[adjacent{v[infl vv, R1]:Z}, R2]
          where concat(C, MORPH1, MORPH2),
                member(C, [s, r]).
```

Or we might have it in the following form.

(1.2.32)

```
lex(MORPH1, v[adjacent{v[infl vc, R1]:Z}, R2]):-
    lex(MORPH2, v[adjacent{v[infl vv, R1]:Z}, R2]),
    concat(C, MORPH1, MORPH2),
    member(C, [s, r]).
```

## 2. current topics in JPSG

There are various reasons why JPSG is so poorly documented. One reason for this is that we do not have well established starting points of our discussion when we try to formulate a properly defined theory of Japanese grammar. We have not even reached the point where we could factor out various elements that interact when we make grammaticality judgments of sentences. And very often, choice of a particular verb or a particular noun drastically changes the whole picture. And the worst part about it is that we do not know beforehand what we must be beware of. I believe this should be true of any language we would be dealing with, but in the case of English, we have some 30 years of history of formal research research. In the case of Japanese, things are quite different, as I will show you shortly.

### 2.1 troubles with relatives

In order to properly formulate the conditions regarding how information concerning syntactic gaps are to be propagated in parse trees, we first have to deal with the following question: What elements in a sentence can be relativized? For example, what sort of postposition phrases can be relativized? And what about topicalization?

Here are some examples. Let's see first what happens to dislocation from simple sentences.[16]

(2.1.1) simple sentences

a. <sbj, ga > yoku ureta hon

a'. sono hon-wa <sbj, ga > yoku ureta

b. ken-ga <obj, wo > yoku yonda hon
b'. sono hon-wa ken-ga <obj, wo > yoku yonda
c. ken-ga takai hyooka-wo <io, ni > ataeta hon
c'. sono hon-wa ken-ga takai hyooka-wo <io, ni > ataeta
c". sono hon-ni-wa ken-ga takai hyooka-wo <io, ni > ataeta
d. ken-ga <location, ni > zaisekisiteiru daigaku
d'. sono daigaku-wa ken-ga <location, ni > zaisekisiteiru
d". sono daigaku-ni-wa ken-ga <location, ni > zaisekisiteiru
e. ken-ga <location, wo > yoku sanposuru kooen
e'. sono kooen-wa ken-ga <location, wo > yoku sanposuru
f. ken-ga <destination, ni > huninsita daigaku
f'. sono daigaku-wa ken-ga <destination, ni > huninsita
f". sono daigaku-ni-wa ken-ga <destination, ni > huninsita
g. ken-ga <destination, he > dairi-wo okutta iinkai
g'. sono iinkai-wa ken-ga <destination, he > dairi-wo okutta
g". sono iinkai-ni-wa ken-ga <destination, he > dairi-wo okutta
h. ken-ga <source, kara > syarei-wo moratta kaisya
h'. sono kaisya-wa ken-ga < source, kara > syarei-wo moratta
h". sono kaisya-kara-wa ken-ga <source, kara > syarei-wo moratta
i. ken-ga <location, de > benkyoosita daigaku
i'. sono daigaku-wa ken-ga <location, de > benkyoosita
i". sono daigaku-de-wa ken-ga <location, de > benkyoosita
j. ken-ga <accompaniment, to > sooguusita UFO
j'. sono UFO-wa ken-ga <accompaniment, to > sooguusita
j". sono UFO-ni-wa ken-ga <accompaniment, to > sooguusita
k. ken-ga <possessive, no > hihyoo-wo sita hon
k'. sono hon-wa ken-ga <possessive, no > hihyoo-wo sita
l. ken-ga <instrument, de > daigaku-ni kayou zitensya
l'. sono zitensya-wa ken-ga <instrument, de > daigaku-ni kayou
l". sono zitensya-de-wa ken-ga <instrument, de > daigaku-ni kayou
m. ken-ga <reason, de > daigaku-wo taigakusita ziken
m'. sono ziken-wa ken-ga <reason, de > daigaku-wo taigakusita
m". sono ziken-de-wa ken-ga <reason, de > daigaku-wo taigakusita

m"'. ken-ga <reason, de > daigaku-wo taigakusuru koto-ni natta ziken
n. ken-ga <comparison, yori > itigo-ga sukina kudamono
n'. sono kudamono-wa ken-ga <comparison, yori > itigo-ga sukida
n". sono kudamono-yori-wa ken-ga <comparison, yori > itigo-ga sukida

In previous literature on generative analyses of Japanese syntax, claims have been made to the effect that dislocation from PP other than subject or object is not possible. In fact, however, we find that all examples other than those that involve dislocation from PPs that function as <instrument >, <reason > and <comparison > are to some extent acceptable.

In the case of topicalization, if we had postposition along, all sentences seem quite all right. There is an interesting difference of judgment regarding the sentences in (1), (1') and (1"), which involved <instrument, de >. Most members of the group find both relativization and topicalization without postposition were out, whereas some members, including the present author, insisted that relativization is OK, and other members agreed that such judgment might be conceivable, although they do not share that judgment. In this case, it was suggested that subcategorization for the verb in question may be different for these two groups of people. And a slight change in wording in the sentence involving <reason > as shown in (m"') makes it easier to understand, although we have no explanation for this phenomena right now.

Anyway, for the most part, our grammar or syntax was actually simplified than when we first started to think of these matters.

Dislocation from embedded sentences seems to pose no more problem than from simple sentences. Although we have not reached definite conclusions as to which of these PPs are complements and which of these should be treated as adjuncts, very roughly we can say that dislocation from PPs other than <instrument >, <reason > and <comparison > are acceptable.

(2.1.2) embedded sentences
a. [naomi-ga [ <sbj > yoku ureta] to omotteiru] hon
a'. sono hon-wa [naomi-ga [ <sbj, ga > yoku ureta] to omotteiru]
b. [naomi-ga [ken-ga <instrument, de > daigaku-ni kayou]

to omotteiru] zitensya

b'. sono zitensya-wa [naomi-ga [ken-ga <instrument, de > daigaku-ni kayou] to omotteiru]

b". sono zitensya-de-wa [naomi-ga [ken-ga <instrument, de > daigaku-ni kayou] to omotteiru]

c. [naomi-ga [ken-ga <reason, de > daigaku-wo taigakusita] to omotteiru] ziken

c'. sono ziken-wa [naomi-ga [ken-ga <reason, de > daigaku-wo taigakusita] to omotteiru

c". sono ziken-de-wa [naomi-ga [ken-ga <reason, de > daigaku-wo taigakusita] to omotteiru

d. [naomi-ga [ken-ga <comparison, yori > itigo-ga sukida] to omotteiru] kudamono

d'. sono kudamono-wa [naomi-ga [ken-ga <comparison, yori > itigo-ga sukida] to omotteiru

d". sono kudamono-yori-wa [naomi-ga [ken-ga <comparison, yori > itigo-ga sukida] to omotteiru]

The examples above show whether dislocation is possible out of complements to verbs. We have to consider next whether information regarding the existence of syntactic gaps can be inherited from adjuncts. First, it seemed that we cannot get this information propagated from adjuncts on the basis of sentences involving *atta* in the following examples. However, it was pointed out during the course of our discussion that sentences involving *sitteiru* are markedly better than sentences involving *atta*.

(1.2.3) dislocation from relative clause

a. [naomi-ga [ <sbj_1 > yoku ureta] koto-wo siranai] hon_1

b. [naomi-ga [ <sbj_2 ><obj_1 > yonda] otoko_2-ni atta] hon_1

b'. [naomi-ga [ <sbj_2 ><obj_1 > yonda] otoko_2-wo oozei sitteriru] hon_1

c. [naomi-ga [ <sbj_2 ><source_1 > syarei-wo moratta] otoko_2-ni atta] kaisya_1

c'. [naomi-ga [ <sbj_2 ><source_1 > syarei-wo moratta]
otoko_2-wo sitteiru] kaisya_1

d. [naomi-ga [ <sbj_2 ><inst_2 > gakkoo-ni kayou]
otoko_2-ni atta] zitensya_1

d'. [naomi-ga [ <sbj_2 ><inst_1 > gakkoo-ni kayou]
otoko_2-wo sitteiru] zitensya_1

e. [naomi-ga [ <sbj_2 ><reason_1 > taigakusita]
otoko_2-ni atta] ziken_1

e'. [naomi-ga [ <sbj_2 ><reason_1 > taigakusita]
otoko_2-wo sitteiru] ziken_1

f. [naomi-ga [ <sbj_2 > itigo-ga <comp_1 > sukina]
otoko_1-ni atta] kudamono_1

f'. [naomi-ga [ <sbj_2 > itigo-ga <comp_1 > sukina]
otoko_2-wo sitteiru] kudamono_1

At this point we have no satisfactory explanations for how this is so. Anyway, we now assume that dislocation from inside a relative clause are syntactically possible, while in some cases sentences are not allowed for some other reasons.[17]

Thus one important generalization about Japanese relative constructions is that elements inside a relative clause can be bound to an element outside that relative construction.[18]

**2.2** some cases with quantifier float

One of the topics of our recent working group discussions was how we should treat quantifier float in Japanese.[19]

In Japanese we find numeral-classifier combinations and inherent quantifiers in complement position of prenominal p[no].[20]

(2.2.1)

a. 3-nin-no gakusei-ga kita

b. subete-no gakusei-ga kita

Numeral-classifier combinations and some inherent quantifiers can precede noun phrases without an intervening occurrence of postposition *no*.

(2.2.2)

a. 3-nin gakusei-ga kita

b. ?subete gakusei-ga kita

Both numeral-classifier combinations and inherent quantifiers could occur in preverbal position.

(2.2.3)

a. gakusei-ga 3-nin kita

b. gakusei-ga subete kita

So we have here three basic positions in which quantifying expressions can occur, although at this point we cannot be sure of the fine-grained syntactic structures involved.[21]

The third pattern as shown in (2.2.3) is that of floated quantification. It looks as if they are in preverbal modifier position. Their distribution seems like that of verb modifiers.

(2.2.4)

a. gakusei-ga 3-kai kita

b. gakusei-ga tokidoki kita

Ueda (1986) gives the following examples as evidence against the idea that quantifiers in these preverbal positions are to be treated as verb modifiers.

(2.2.5)

a. *taroo-ga gakusei-ni 3-nin nagurareta

b. taroo-ga gakusei-ni 3-kai nagurareta

The question here is, if quantifiers are simply verb modifiers, why is it

that preverbal numeral-classifier combination *3-nin* cannot quantify agentive postposition phrase *gakusei-ni*. A more important point to be considered here, however, is that this preverbal numeral-classifier combination can quantify a preceding subject postposition phrase, as can be seen from the following example.

(2.2.6)

sensei-ga gakusei-ni 3-nin nagurareta

We can't simply say, however, that quantifier float is disallowed from p[ni]. Inoue (1978: p.173) gives the following examples.

(2.2.7)

a. gakusei-ni suu-nin denwasita
b. yadoya-ni 2,3-ken atattemita

We would like to clarify first of all how quantification and scrambling interact, *i.e.* what quantifier can quantify what PP in what position. Of course, we have to consider the interaction of various adverbial elements and quantifiers. What follows are some of the test sentences for grammaticality judgment, on which our group at ICOT are discussing various syntactic and semantic aspects of Japanese grammar for quantification.

(2.2.8)

a. kinoo-wa gakusei-ga 3-nin kita
b. kinoo kita-no-wa gakusei-ga 3-nin-da
c. gakusei-ga kinoo-wa 3-nin kita
d. *kinoo gakusei-ga kita-no-wa 3-nin-da

(2.2.9)

a. sensei-ga 3-nin sake-wo mottekita
b. sensei-ga kinoo 3-nin sake-wo mottekita
c. *sensei-ga sake-wo 3-nin mottekita

d. ?sensei-ga sake-wo kinoo 3-nin mottekita
e. *sensei-ga kinoo sake-wo 3-nin mottekita

If we concentrate our attention to cases where only subject, object and quantifiers are involved, we have to consider the following 12 patterns, because classifier-noun combination affects the grammaticality of a given sentence.

(2.2.10)

a. 3-nin gakusei-ga sake-wo nonda
b. 3-hon gakusei-ga sake-wo nonda
c. ?3-nin sake-wo gakusei-ga nonda
d. 3-hon sake-wo gakusei-ga nonda
e. gakusei-ga 3-nin sake-wo nonda
f. gakusei-ga 3-hon sake-wo nonda
g. *gakusei-ga sake-wo 3-nin nonda
h. gakusei-ga sake-wo 3-hon nonda
i. sake-wo 3-nin gakusei-ga nonda
j. sake-wo 3-hon gakusei-ga nonda
k. sake-wo gakusei-ga 3-nin nonda
l. sake-wo gakusei-ga 3-hon nonda

The important point to note here is the subject/object asymmetry as shown in the different grammaticality assigned to (g) and (l). Although *3-hon* can quantify the object NP across the subject NP in (l), *3-nin* cannot quantify the subject NP across the object NP in (g).

Since the simplest assumption to treat quantifiers as adjuncts to verbs or verb phrases is denied, there remain two possibilities to consider. One is to treat quantifiers as a kind of pseudo-complements. The other is to consider that quantifiers and postposition phrases form constituents. In other words, postpositions would optionally subcategorize for quantifiers, and quantifiers or classifiers would optionally subcategorize for postposition phrases. The difficulty in these approaches might be that it is difficult to obtain the desired quantifying expression, without invoking impossible combinations.

In the remaining pages, I will explain how our idea of treating quantifiers as pseudo-complements could be formulated in such a way that it would incorporate the facts about the interaction of quantifier float and scrambling just discussed.

In our formulation which I briefly sketched in the previous chapter, we had a feature called subcat which correspond to the complements such as subject, object, indirect object and so on, as illustrated below.

(2.2.11)

| complement | subcat { |
|---|---|
| subject | p[ga, sbj], |
| object | p[wo, obj], |
| 2nd object | p[ni, obj] |
| | } |

Our idea is to treat quantifiers as nouns and along with ordinary complements, these additional pseudo-complements would appear as elements in subcat value sets for verbs. In order to retain syntactic and semantic relations between quantifiers and nouns, we would have lists consisting of maximally two categories, which in turn are to be elements in our subcat value. Thus if we have quantifiers for the subject, our subcat would be something like this.

(2.2.12)

a. 3-nin gakusei-ga sake-wo noda

| complement | subcat { |
|---|---|
| subject | <p[ga, sbj], n[sbj] >, |
| object | <p[wo, obj] > |
| | } |

If we have a quantifying expression for the object, the corresponding subcat value would be something like this.

(2.2.12)

b. 3-hon gakusei-ga sake-wo nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj] >, |
| object | <p[wo, obj], n[obj] > |
| | } |

And for the rest of the examples in (2.2.10) we would have the following subcat values. Note that the correspondence between subcat list sets and the resulting sentences is not one-to-one. There are several sentences that would correspond to a given underlying subcat value.

(2.2.12)

c. ?3-nin sake-wo gakusei-ga nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj], n[sbj] >, |
| object | <p[wo, obj] > |
| | } |

d. 3-hon sake-wo gakusei-ga nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj] >, |
| object | <p[wo, obj], n[obj] > |
| | } |

e. gakusei-ga 3-nin sake-wo nonda

| | |
|---|---|
| complement | subcat { |
| subject | <n[sbj], p[ga, sbj] >, |

| | |
|---|---|
| object | <p[wo, obj] > |
| | } |

f. gakusei-ga 3-hon sake-wo nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj] >, |
| object | <p[wo, obj], n[obj] > |
| | } |

g. *gakusei-ga sake-wo 3-nin nonda

| | |
|---|---|
| complement | subcat { |
| subject | <n[sbj], p[ga, sbj] >, |
| object | <p[wo, obj] > |
| | } |

h. gakusei-ga sake-wo 3-hon nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj] >, |
| object | <n[obj], p[wo, obj] > |
| | } |

i. sake-wo 3-nin gaksusei-ga nonda

| | |
|---|---|
| complement | subcat { |
| subject | <p[ga, sbj], n[sbj] >, |
| object | <p[wo, obj] > |
| | } |

j. sake-wo gakusei-ga 3-nin nonda

| complement | subcat { |
|---|---|
| subject | <n[sbj], p[ga, sbj] >, |
| object | <p[wo, obj] > |
| | } |

l. sake-wo gakusei-ga 3-hon nonda

| complement | subcat { |
|---|---|
| subject | <p[ga, sbj] >, |
| object | <n[obj], p[wo, obj] > |
| | } |

There are several important assumptions that we have not made explicit enough. The first of these is that we would treat quantifiers as nouns.

(2.2.13)

Assumption 1: quantifiers are nouns with its semantics incorporating the semantics of the nouns they quantify.

The second assumption is that we have 'basic' lexical entries for verbs in which no designation for quantifiers are made. Instead of subcat values being sets of grammatical categories, we have here subcat values sets of lists of categories, and in these basic lexical entries, subcat values are sets of singleton lists of categories.

(2.2.14)

Assumption 2: basic lexical entries for verbs subcategorize for PPs, with subcat value sets consisting of singleton lists.

e.g. subcat {<p[ga, sbj, IND1] >, <p[wo, obj, IND2] >}

We assume the existence of lexical rules that would augment the basic lexical entries for verbs in such a way that floated quantifiers would be sanctioned.

(2.2.15)

Assumption 3: lexical rules will augment the basic lexical entries for verbs in such a way that quantifiers are put on top or at the bottom of a list which is a subcat value set element. The value for gr and ind unify between the added category and inherent category.

e.g. subcat {<n[sbj, I1], p[ga, sbj, I1] >, <p[wo, obj, I2] >}
subcat {<p[ga, sbj, I1], n[sbj, I1] >, <p[wo, obj, I2] >}
subcat {<p[ga, sbj, I2] >, <n[obj, I2], p[wo, obj, I2] >}
subcat {<p[ga, sbj, I2] >, <p[wo, obj, I2], n[obj, I2] >}

We have to assume that the phrase structure rule for complementation is modified accordingly. In order for the subject/object asymmetry to be guaranteed, we have to assume some additional restriction on the application of complementation.

(2.2.16)

wo > ga

This means that if we pop an element from the list that contains [ga], then if anything remains in that list, that element must be pushed on top of the list that contains [wo], if such exists in the subcat list. This will correctly predict that sentences (2.2.10.c) and (2.2.10.g) will be ruled out.

## NOTES

* The present article is based on the memo I prepared for the talk I gave at the mini-workshop on JPSG, held on Aug. 2nd and 4th, 1988 at CSLI (Center for the Study of Language and Information), Stanford University. I am grateful to Stanley Peters, director at CSLI, for giving me this opportunity and also for providing me a shared office and a computer account in order that I might make my preparations for the talk in a comfortable and efficient environment. I am indebted to SUZUKI Hiroyuki (Matsushita Electric Industrial

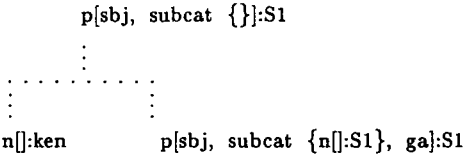
Co. Ltd. Tokyo Laboratory, currently at CSLI) and HANDA Kenichi (Electro-Technical Laboratory) for making necessary arrangements through computerized electronic mailing network systems usually referred to as "e-mails". I would like to express my gratitude to all the participants of the workshop, especially TUTIYA Shun, Ivan SAG, John NERBONNE, IIDA Masayo and Peter SELLS, among many others, for their stimulating comments and warm welcome.

Part of the material presented here about the overall conception of grammar is based on a talk I gave this March (Harada 1988b). The fragment of Japanese that I presented here is partly based on Gunji, Hasida, Sirai and Harada (1987) and partly on Sirai (1987).

1. Clocksin and Mellish (1981) offer a quite easy to read introduction to the programming language called Prolog.
2. See Yokota, Yamamoto, Taki, Nishikawa and Uchida (1984) for details about this machine that now exists. When the JPSG project started, there was no PSI existing in the real world as a physical object, however.
3. Gazdar *et. al.* (1985), Pollard and Sag (1988), and Gunji (1987) give overall expositions of GPSG, HPSG, and JPSG, respectively.
4. Hasida and Sirai (1986) is the only document widely available in which the concept of constraint-unification is extensively explicated, although as the title suggests a slightly different terminology and formalization are employed.
5. Barwise and Perry (1983) is admittedly outdated today, but it is the only widely available source of information about situation semantics.
6. See Pollard (1984) for earlier formulations of HPSG.
7. See, for example, Harada (1988a).
8. In what follows, categories are designated by a left square bracket ("[") followed by an indefinite number of feature specifications separated by commas(",") followed by a right square bracket ("]"). A feature specification is maximally a feature name followed by its value. However, when the value uniquely determines the name, the name can be omitted. Also, when the value is nil or a null-list ( <>) the entire feature specification can be omitted. Finally a category of the form [pos P, ... , sem S] is sometimes designated as P[ ... ]:S.
9. HFP is for Head Feature Principle in Pollard and Sag (1988: p.58) or Head Feature Convention in Gazdar et al. (1985: p.97). FFP is for Foot Feature Principle in Gazdar et. al. (1985: p.82). SFP is for Subcat Feature Principle in Gunji et. al. (1987).
10. See Harada (1988b) for a more detailed discussion of this point.
11. There are several problems in this notion of lexicon and lexical rules. We have not yet settled how we should control the 'application' of lexical rules. For instance, can a lexical rule 'apply' to the 'same' lexical item repeatedly? How should we control the 'order' of application of lexical rules, if they have to be specified?

12. This is what our grammar sanctions, because once unification succeeds, all things that are to be unified are unified. However, since it is a little difficult to see what's going on, we will usually write in the following way, thus showing what each daughter might have looked like before unification.

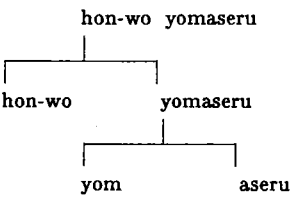
(i)



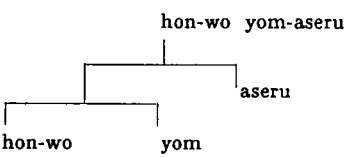

Once unification succeeds, no structure of this form is produced, but sometimes we'd like to see what the lexical entries looked like, so we may be writing trees in this form.

13. Traditional Japanese school grammar would say that *yom* and *aseru* form a single phrase as shown in (i) below. We also admit that *yomaseru* is a single phonological phrase, although we assign totally different phrase structure to this string of formatives as shown in (ii).

(i)




(ii)




14. Or we could state the following ordering relation on category valued feature cancellations.

(i)

adjacent < subcat

Here feature cancellation is something like this.

(ii)

$$M \longrightarrow C \quad H$$
$$CVF1(M) \cup \{C\} = CVF1(H)$$

If this local structure is possible only when the following condition is met, we say that CVF2 is more locally bound than CVF1; or CVF2 $<$ CVF1.

(iii)

$$CVF2(M) = CVF2(H) = \{\}$$

See Harada (1988a) and Harada (1988b) for detail.

15. There are basically two approaches to deal with morphological inflections in Japanese within the general framework of unification grammar that I am describing so far. One is to treat inflectional endings as head of the phrase, in the same way that we treated postposition of postposition phrases. Most of the feature specifications can be specified to coincide with the 'root' or 'adjacent complement'. The second idea is a little complicated.

In our phrase structure rules given so far, there was no mention of morph features, and this was obviously insufficient. The simplest idea would be that the mother's morph is the concatenation of those of its daughters.

(i)

$$M \longrightarrow C \quad H$$
$$morph(M) = morph(C) + morph(H)$$

We know that this is not true, especially, if we start to think of real phonological matters, such as stress patterns, intonation, and junctures. But as long as we take spellings as linguistic representations, this simplification works to a certain degree, especially for languages like English. Thus far, we have assumed something like this, but we can think of some other way of getting the linguistic representation of the mother from those of its daughters. Thus something like (ii) would be a quite conceivable constraint for the morphological representations of the constituents involved in a local phrase structure.

(ii)

$$M \longrightarrow C \quad H$$
$$stem(C) + suffix(C) + stem(H) = stem(M)$$

16. In this section, overt markings for grammaticality are omitted so that the readers can check for themselves before seeing what I have to say about these constructions. Designations like <source, kara > are intended to signify what elements are missing in the relevant construction. These are not to be taken as specifying the underlying linear orderings of elements, however.
17. In order to properly state the conditions that govern distribution of syntactic gaps, we have to see whether crossing bindings of gaps are allowed in Japanese. Because of our treatment of scrambling, we have to consider sentences of the following form.

(i)

a. [[[ <sbj_1 ><obj_2 > kaita] e_2-ga [ <sbj_3 > mukasi <obj_4 > totta] syasin_4-ni niteita] syasinka_3-ga iru] gaka_1

b. [[[ <sbj_1 ><obj_2 > kaita] e_2-ga [ <sbj_3 > mukasi <obj_4 > totta] syasin_4-ni niteita] gaka_1-ga iru] syasinka_3

It is difficult to comprehend sentences like these; it is difficult to say whether they mean anything at all. And yet some of these seem to have some meaning in a way.

(ii)

a. [[[ <sbj_1 ><obj_2 ><obj_3 >kak-aseta] gakusya2-ga sinda] hon_3-ga yuumeini natta] hensyuusya_1

b. [[[ <sbj_1 ><obj_2 ><obj_3 >kak-aseta] hon_3-ga yuumeini natta] gakusya2-ga sinda]] hensyuusya_1

According to some of the members in our working group, the latter seemed better than the former, but in any case, they are hard to understand.

(iii)

[ <sbj_1 > [ <sbj_2 > syuuwaisita to omotta] seizika_2-ga zisyokusita] kisya_1

Some of the members claim that this sentence is simply incomprehensible but at least we agreed that this was by far the best among the constructions we considered here.

At this point, we have no conclusive way of neatly stating these facts, but if we are to formulate a theory on how traces are introduced and how that information is propagated in parse trees, we must incorporate these into our theory.

18. Participants of the mini-workshop whose native language is English suggested

that a similar difference can be found in English also. Incidentally, dislocation from adjuncts seems quite natural and acceptable in the following sentence, which a member of the family of a computational linguist spontaneously uttered during the course of a natural everyday conversation.

(i) These are the things I will be very unhappy if you break.

Japanese sentences corresponding to this would be quite acceptable.

19. There are several reasons why our analysis of quantifier float is not yet properly formulated. First of all we do not know as yet how quantifiers are syntactically combined with the expressions they quantify. Second, we have not settled on how we should deal with the semantics of quantification. Third, we have not yet formulated fully the syntax and semantics of adjuncts. These are interrelated questions, each depending on the solution of the other. These and related problems, such as whether singular pronouns can be bound or co-referential with plural expressions, to name one, are the topics of our discussions during these years.

20. In Japanese, we have two very commonly used universal quantifiers, *mina* and *subete*. There are subtle difference in their usage and distribution. Although both *mina* and *subete* can quantify subject PP *gakusei-ga* from preverbal position, only *subete* can occur as complement to prenominal PP with postposition *no*.

(i)
  a. gakusei-ga mina kita
  b. gakusei-ga subete kita

(ii)
  a. *mina-no gakusei-ga kita
  b. subete-no gakusei-ga kita

21. Another possibility is that numeral-classifier combinations and quantifiers can occur postnominally, preceding postpositions.

(i)
  a. gakusei-3-nin-ga kita
  b. gakusei-subete-ga kita

We will not deal with this construction here.

## REFERENCES

Barwise, J. and Perry, J. (1983) *Situations and Attitudes*, MIT Press.

Clocksin, W.F. and Mellish, C.S. (1981) *Programming in Prolog*, Springer-Verlag.

Gazdar, G., Klein, E., Pullum, G.K. and Sag, I.A. (1985) *Generalized Phrase Structure Grammar*, Basil Blackwell.

Gunji, T. (1987) *Japanese Phrase Structure Grammar: A Unification-based Approach*, D. Reide.

Gunji, T., Sirai, H., Hasida, K. and Harada, Y. (1987) "Grammatical Description with Local Constraints", US-Japan AI Symposium.

Harada, Y. (1988 a) "A Phrase Structure Grammar Account of 'Island Constraints'" (in Japanese), *Humanitas*, No. 26, pp. 23–52.

Harada, Y. (1988 b) *On Describing Natural Languages in Unification Environments*, L.A.U.D. Symposium on Linguistic Approaches to Artificial Intelligence.

Hasida, K. and Sirai, H. (1986) "Conditioned Unification" (in Japanese), *Computer Software*, Vol. 3, No. 4, pp. 28–38.

Inoue, K. (1987) *Nihongo-no Bunpoo Kisoku*, Taishuukan Shoten.

Pollard, C.J. and Sag, I.A. (1988) *Information-Based Syntax and Semantics, Vol. 1: Fundamentals*, CSLI Lecture Notes, No. 12, Center for the Study of Language and Information, Stanford University.

Sirai, H. (1987) "Nihongo Koobunkaiseki Nyuumon" (in Japanese), *Professional Sigma*, Vol. 1, No. 1, pp. 10–16, No. 2, pp. 26–35, No. 3, pp. 124–138, Gijutsu-Hyuooron-Sha.

Ueda, M. (1986) "Quantifier Float in Japanese", University of Massachusetts Occasional Papers.

Yokota, M., Yamamoto, A., Taki, K., Nishikawa, H. and Uchida, S. (1984) *The Design and Implementation of a Personal Sequential Inference Machine (PSI)*, ICOT TR-045, Institute for New Generation Computer Technology.