Number Names in Japanese: A Head-Medial Construction in a Head-Final Language

Emily M. Bender
Stanford University
bender@csli.stanford.edu

Draft of April 30, 2002

1 Introduction

Japanese is well-known for being resolutely head-final: verbs come at the end of clauses, postpositions follow nouns, and nouns follow determiners and all modifiers. It is therefore surprising to find constructions in the language that don't fit this pattern. In this paper, I will argue that number names like (1) *ni hyaku juu* 'two hundred and ten' are head-medial.

(1) ni hyaku juu
two hundred ten
'two hundred and ten'

The HPSG analysis I will present highlights how the type hierarchy can be used to capture partial generalizations such as the head-finalness of Japanese, while still allowing for exceptions like head-medial number names. This analysis illustrates how the formal tool of type hierarchies is suited to capturing such partial generalizations.

2 Locating the head

2.1 A phrasal analysis

Considering only small numbers, one might be tempted to simply list them in the lexicon. However, as soon as we consider larger number names, the advantages of a compositional analysis become clear. The analysis presented below requires only 33 lexical entries to account for all of the number names from *ichi* to the very large number name in (3).

(2) ichi [=1]

(3)

kyuu sen kyuu hyaku kyuu juu kyuu gai kyuu sen kyuu hyaku kyuu juu kyuu kei
kyuu sen kyuu hyaku kyuu juu kyuu chou kyuu sen kyuu hyaku kyuu juu kyuu oku
kyuu sen kyuu hyaku kyuu juu kyuu man kyuu sen kyuu hyaku kyuu juu kyuu
[=999,999,999,999,999,999,999,999]

There are, of course, two kinds of compositional analyses: syntactic and morphological. Martin (1987) finds that some local combinations within number names (e.g., the names for 11 through 19, 20, 30, 200, 300, etc.) form single phonological words. However, longer combinations made up of these pieces (such as *sanbyaku juuichi* '311') show phrasal phonology. The analysis presented here was developed within the context of an application that takes text-based input. As such, it was most convenient to apply the phrasal analysis uniformly. A similar analysis could be developed that provides lexical entries for every combination that forms a phonological word.

Now, syntactic analyses can involve headed or non-headed phrases. However, an analysis in terms of headed phrases, if available, is preferable because it allows for maximal reuse of construction types already posited for other aspects of Japanese grammar.

2.2 The external representative test

As Smith (1999) notes in his analysis of English number names, most of Zwicky's tests for determining the head of a phrase are not applicable to number names. The exception is the external representative test:

> The head determines the distribution of the entire phrase.

The number names in (4) each share one element in common with the number name in (1). The frame in (5) provides a context where their distributions differ.

(4)

a.	go	**hyaku**	san
	five	*hundred*	*three*
b.	**ni**	sen	san
	two	*thousand*	*three*
c.	go	sen	**juu**
	five	*thousand*	*ten*

(5)

a.	roku	sen	ni	**hyaku**	juu
	six	*thousand*	*two*	*hundred*	*ten*
b.	roku	sen	go	**hyaku**	san
	six	*thousand*	*five*	*hundred*	*three*
c.	*roku	sen	**ni**	sen	san
	six	*thousand*	*two*	*thousand*	*three*
d.	*roku	sen	go	sen	**juu**
	six	*thousand*	*five*	*thousand*	*ten*

The examples in (5) show that the expressions with *hyaku* ((1) and (4a)) have the same combinatoric potential. Expressions without *hyaku* differ. The other elements of (1) (*ni* 'two' and *juu* 'ten') are not relevant. Thus, the external representative test indicates that *hyaku* is the head of example (1).

If we forget for the moment that Japanese is supposed to be head-final, this isn't very surprising. English number names work the same way (see Smith 1999). So do number names in another SVO language: Chinese, the source from which Japanese borrowed this system. (The native Japanese number name system, by the way, is preserved in the modern language only for the numbers one to ten, and a few other minor cases.) The problem is how to incorporate the head-medial number names into the same grammar that handles all the head-final constructions of the language.

2.3 Semantics and selectional restrictions

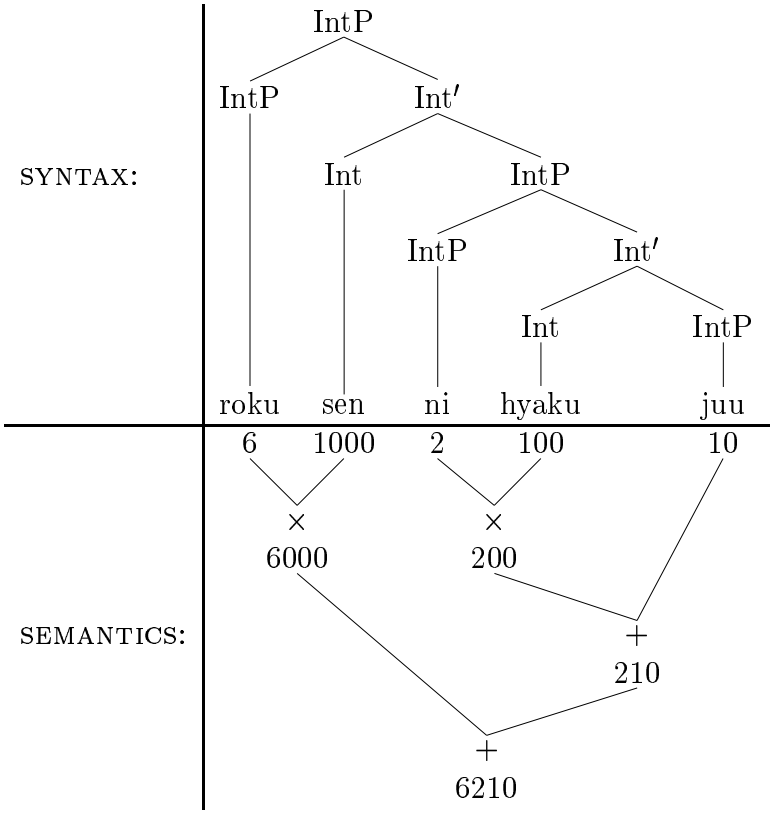
Before presenting the analysis, I would like to first lay out some further data concerning the semantics and selectional restrictions of number name heads.

If *hyaku* is the head of (1), the semantics of these expressions can be stated simply, if informally, as in (6).

(6) (specifier × head) + complement

The external representative test shows *sen* to be the head of (5a-b), and the semantic schema in (6) generalizes appropriately:

(7)



Note that there is a mismatch in the order of composition between the syntax and the semantics. This is not problematic in HPSG, because syntax and semantics are both independently head-driven.

With this semantic background, the selectional restrictions illustrated in (5) can be understood as follows:

a. The head directly gives a lower bound on the order of magnitude of the whole IntP.

b. Because of the way numbers are named in Japanese, the head indirectly gives an upper bound on the order of magnitude of the whole IntP.

c. The complement of a head (such as *sen*) must be of a lesser order of magnitude than the head itself.

d. The specifier of a head must be of a lesser order of magnitude than the head itself.

e. Specifiers express a value no greater than 9999, no matter how large the order of magnitude of the head.

This last point is illustrated in (8).

(8) a. ni sen man
two thousand ten thousand
20,000,000

b. *ni man oku
two ten thousand ten million
Intended: 2,000,000,000,000

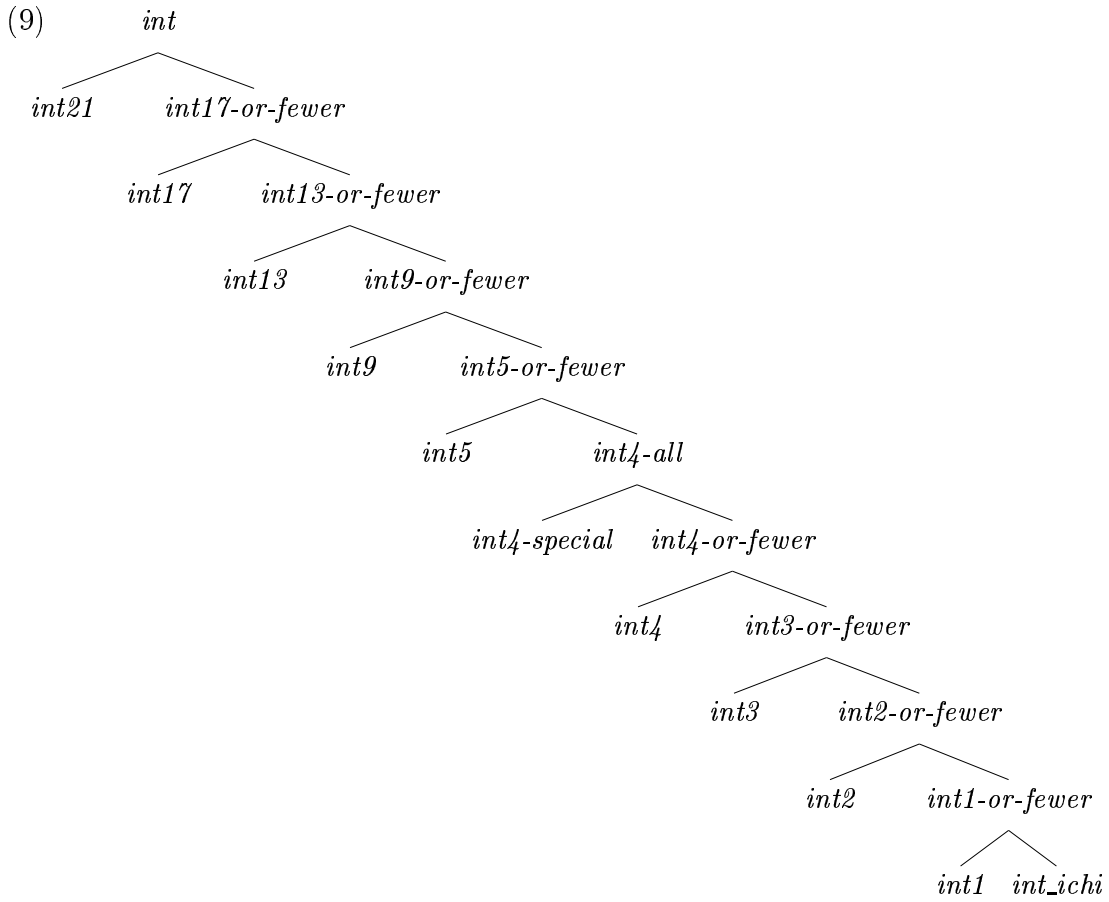
3 HPSG analysis

This analysis was developed and tested in the context of a broad-coverage implemented Japanese grammar developed collaboratively between YY Technologies and DFKI (Siegel and Bender 2002). It is based on an earlier grammar developed for the *Verbmobil* project (Siegel 2000).

3.1 Lexical entries

Following Smith, a hierarchy of head types can be used to capture the generalizations about order of magnitude and complement/specifier selection. As shown in (9), the head type *int* for 'integer' has subtypes representing different numbers of digits. Basically, for every number name head, there is a type corresponding to the number of digits in the number it denotes. These types are grouped together with the types for smaller number names in a binary branching structure. There is some further complexity around 4 digit and 1 digit numbers that is there to account for some idiosyncratic selectional restrictions outlined in section 4.

(9)



The way these head types are used is illustrated in the lexical entry for *sen* 'thousand' in (10). This is the entry for uses of *sen* like (5a-b). Syntactic information about this lexical entry is encoded in the feature CAT (short for category). Its HEAD type is *int4*. It selects for a specifier of HEAD type *int1*. In other words, a single digit integer specifier. It also selects for an object of HEAD type *int3-or-fewer*, that is, one that expresses at most a 3 digit number.[1]

[1] This style of valence features is an aspect of the analysis of scrambling in Japanese. See Siegel 2000.

(10)
$$\begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \textit{int4} \\
\text{SAT} & \begin{bmatrix}
\text{OBJ_SAT} & \textit{obl} \\
\text{OBJ2_SAT} & \textit{sat} \\
\text{SUBJ_SAT} & \textit{sat} \\
\text{SPR_SAT} & \textit{obl}
\end{bmatrix} \\
\text{VAL} & \begin{bmatrix}
\text{OBJ} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{int3-or-fewer} \\ \text{SAT} & \textit{saturated} \end{bmatrix} \\
\text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{1} \end{bmatrix}
\end{bmatrix} \\
\text{OBJ2} & \textit{none} \\
\text{SUBJ} & \textit{none} \\
\text{SPR} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{int1} \\ \text{SAT} & \textit{saturated} \end{bmatrix} \\
\text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{2} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} \\
\text{CONT} & \begin{bmatrix}
\text{TOP} & \boxed{3} \\
\text{LISZT} & \left\langle \begin{bmatrix} \textit{plus-rel} & \\ \text{HANDEL} & \boxed{3} \\ \text{TERM1} & \boxed{1} \\ \text{TERM2} & \boxed{4} \end{bmatrix}, \begin{bmatrix} \textit{times-rel} & \\ \text{HANDEL} & \boxed{4} \\ \text{FACTOR1} & \boxed{2} \\ \text{FACTOR2} & \boxed{5} \end{bmatrix}, \begin{bmatrix} \textit{const-rel} & \\ \text{HANDEL} & \boxed{5} \\ \text{CONST_VALUE} & \textit{'1000} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix}$$

Semantic information about the lexical entry is encoded in the feature CONT (short for content), using Minimal Recursion Semantics. (On Minimal Recursion Semantics, see the papers by Copestake et al.) There are two main aspects to this representation (other details have been suppressed). The first is a 'top handle' or pointer into the semantics that other elements which select this expression can use, tagged here as [3]. The second is a list of relations. This lexical entry has three relations: a constant value relation (with the value 1000), a times relation, relating the constant value to the 'top handle' of the specifier, and a plus relation, relating the result of the times relation to the 'top handle' of the object.

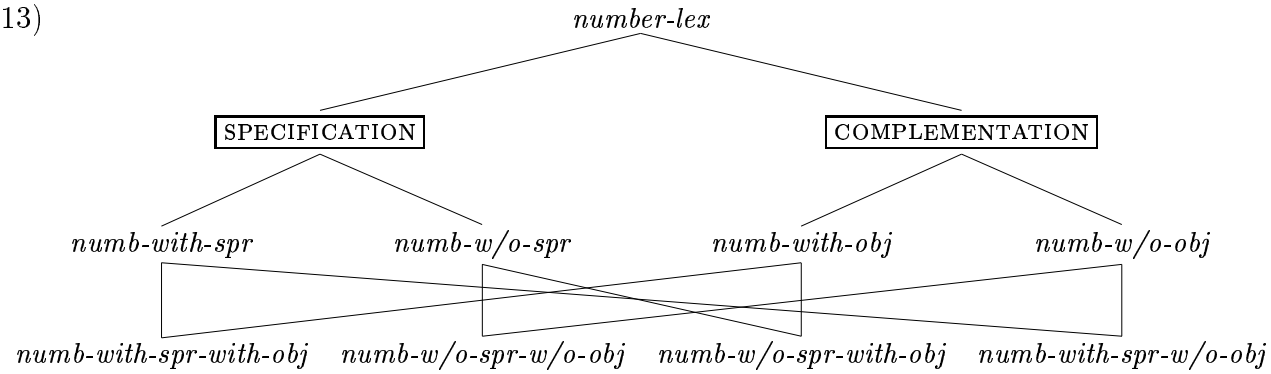
The word *sen* can also appear without a complement, as in (11). In this case, its lexical entry is as in (12). Note that in this case, *sen* does not select an object and there are only two relations on the liszt: a constant value relation and a times relation.

(11) roku sen
six *thousand*

(12)
$$\begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \textit{int4} \\
\text{SAT} & \begin{bmatrix}
\text{OBJ_SAT} & \textit{sat} \\
\text{OBJ2_SAT} & \textit{sat} \\
\text{SUBJ_SAT} & \textit{sat} \\
\text{SPR_SAT} & \textit{obl}
\end{bmatrix} \\
\text{VAL} & \begin{bmatrix}
\text{OBJ} & \textit{none} \\
\text{OBJ2} & \textit{none} \\
\text{SUBJ} & \textit{none} \\
\text{SPR} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{int1} \\ \text{SAT} & \textit{saturated} \end{bmatrix} \\
\text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{1} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} \\
\text{CONT} & \begin{bmatrix}
\text{TOP} & \boxed{2} \\
\text{LISZT} & \left\langle \begin{bmatrix}
\textit{times-rel} \\
\text{HANDEL} & \boxed{2} \\
\text{FACTOR1} & \boxed{1} \\
\text{FACTOR2} & \boxed{3}
\end{bmatrix}, \begin{bmatrix}
\textit{const-rel} \\
\text{HANDEL} & \boxed{3} \\
\text{CONST_VALUE} & \textit{'1000}
\end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix}$$

Many number names will show this dual patterning, and in addition, some can appear with or without a specifier. A multiple inheritance type hierarchy can be used to capture the similarities across lexical entries, as shown in (13).

(13)



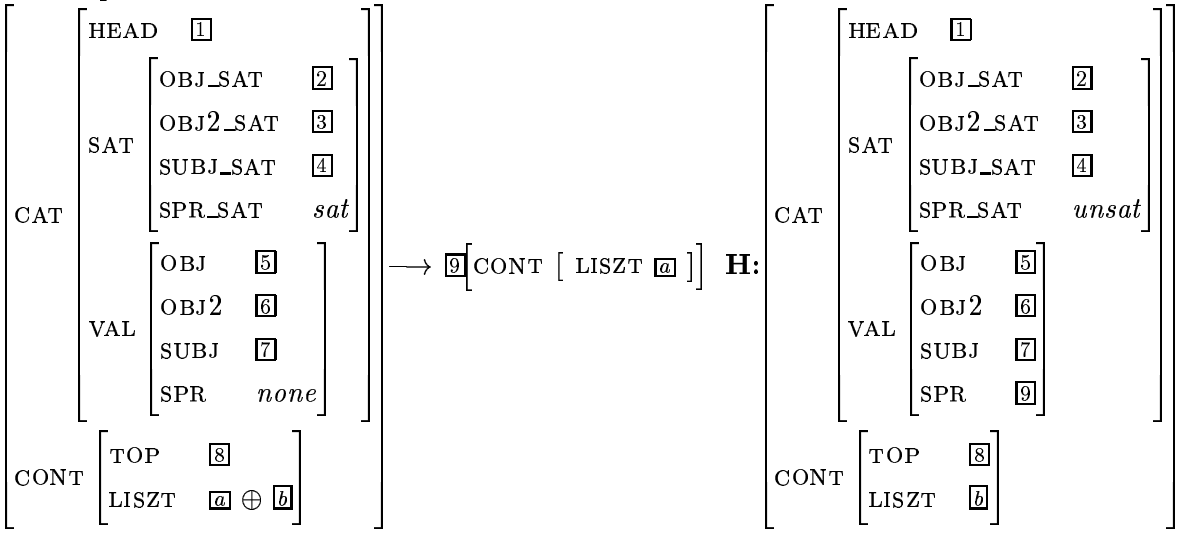
3.2 Phrase structure rules

These lexical entries only license integer phrases in combination with phrase structure rules. Smith's analysis of English number names leverages the ordinary head-complement and head-specifier phrase structure schemata proposed for that language. As Japanese is primarily head-final, the existing implemented Japanese grammar had no mechanism for realizing a selected argument to the right of the head.

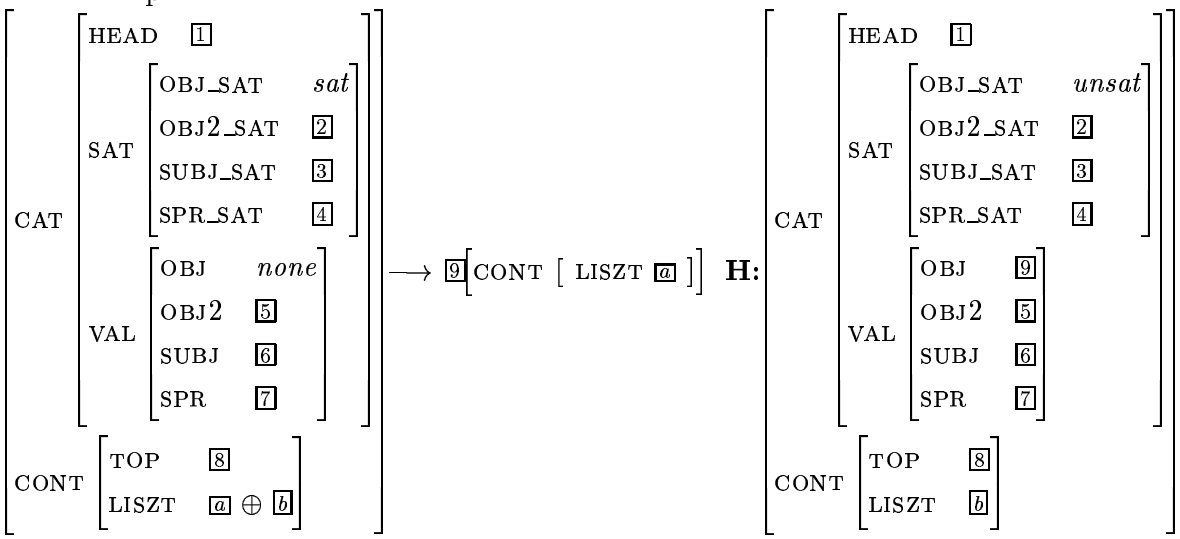
The existing head-specifier and head-complement rules from the JACY grammar are shown in (14) and (15). (These representations include constraints inherited from supertypes

that do not need to be stipulated in the description of these rules. They have also been simplified somewhat for purposes of presentation.)

(14) Head-Specifier Rule



(15) Head-Complement Rule



The head-specifier rule will work as is: It states that a phrase can be built of a specifier-seeking head preceded by an element that meets all the requirements of the selecting head. This is indicated by the tag 9 identifying the specifier requirement of the head with the whole feature structure for the non-head daughter. The resulting phrase is specifier-saturated.

On the other hand, the existing head-complement rule won't work for number names, as it too realizes the selected complement before the head. Given the existing JACY grammar, the best way to account for the head-medial number names is to add a head-initial head-complement rule. The two head-complement rules are shown in (16) and (17) (once again including inherited constraints and simplified). The primary difference between these rules is the order of the daughters.

(16) Head-Complement Rule (Head Final)

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SAT} & \begin{bmatrix} \text{OBJ\_SAT} & \textit{sat} \\ \text{OBJ2\_SAT} & \boxed{2} \\ \text{SUBJ\_SAT} & \boxed{3} \\ \text{SPR\_SAT} & \boxed{4} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{OBJ} & \textit{none} \\ \text{OBJ2} & \boxed{5} \\ \text{SUBJ} & \boxed{6} \\ \text{SPR} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{8} \\ \text{LISZT} & \boxed{a} \oplus \boxed{b} \end{bmatrix} \end{bmatrix} \longrightarrow \boxed{9}\left[\text{CONT}\ [\ \text{LISZT}\ \boxed{a}\ ]\right]\ \textbf{H:} \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1}\textit{final-head} \\ \text{SAT} & \begin{bmatrix} \text{OBJ\_SAT} & \textit{unsat} \\ \text{OBJ2\_SAT} & \boxed{2} \\ \text{SUBJ\_SAT} & \boxed{3} \\ \text{SPR\_SAT} & \boxed{4} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{OBJ} & \boxed{9} \\ \text{OBJ2} & \boxed{5} \\ \text{SUBJ} & \boxed{6} \\ \text{SPR} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{8} \\ \text{LISZT} & \boxed{b} \end{bmatrix} \end{bmatrix}$$

(17) Head-Complement Rule (Head Initial)

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SAT} & \begin{bmatrix} \text{OBJ\_SAT} & \textit{sat} \\ \text{OBJ2\_SAT} & \boxed{2} \\ \text{SUBJ\_SAT} & \boxed{3} \\ \text{SPR\_SAT} & \boxed{4} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{OBJ} & \textit{none} \\ \text{OBJ2} & \boxed{5} \\ \text{SUBJ} & \boxed{6} \\ \text{SPR} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{8} \\ \text{LISZT} & \boxed{a} \oplus \boxed{b} \end{bmatrix} \end{bmatrix} \longrightarrow \textbf{H:} \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1}\textit{init-head} \\ \text{SAT} & \begin{bmatrix} \text{OBJ\_SAT} & \textit{unsat} \\ \text{OBJ2\_SAT} & \boxed{2} \\ \text{SUBJ\_SAT} & \boxed{3} \\ \text{SPR\_SAT} & \boxed{4} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{OBJ} & \boxed{9} \\ \text{OBJ2} & \boxed{5} \\ \text{SUBJ} & \boxed{6} \\ \text{SPR} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{TOP} & \boxed{8} \\ \text{LISZT} & \boxed{a} \end{bmatrix} \end{bmatrix} \boxed{9}\left[\text{CONT}\ [\ \text{LISZT}\ \boxed{b}\ ]\right]$$

At the same time, the two head-complement rules must be constrained so that they don't overgenerate, licensing ungrammatical examples such as those in (18).

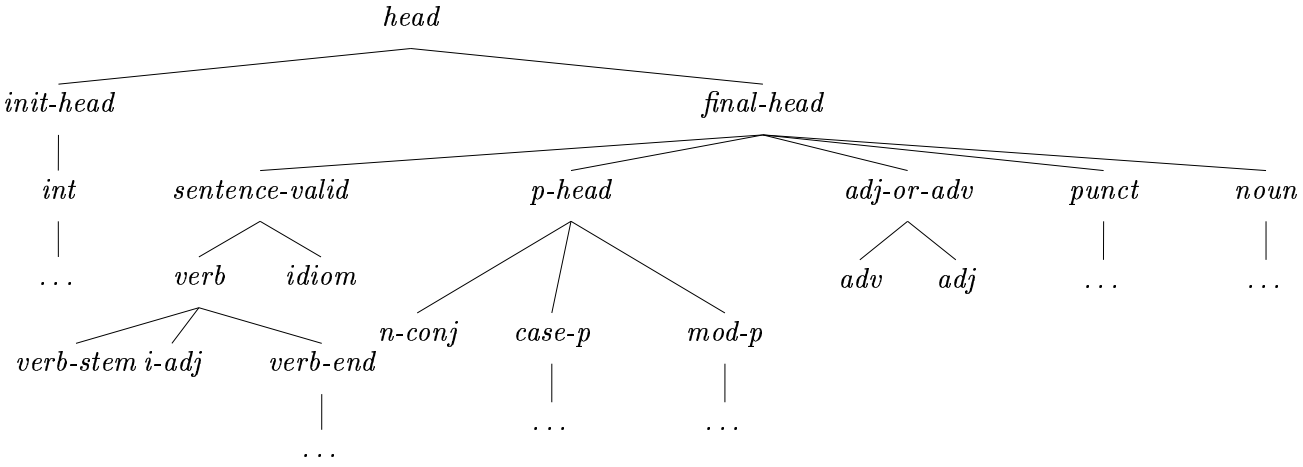
(18) a. *ni juu hyaku
two ten hundred
Intended: 'Two hundred and ten'

b. *Suzuki-ga yomu hon-wo
Suzuki-NOM read book-ACC
Intended: 'Suzuki reads a book.'

Such overgeneration can be prevented by constraining the type of the HEAD value of the head daughter in each rule. The type hierarchy is crucial in stating this constraint succinctly: The type *final-head* groups together all of the head types that are allowed in rule (16). This type contrasts with *init-head*, supertype to *int* and all of its subtypes. These types are the

ones allowed in rule (17). This hierarchy leaves open the possibility that other subtypes of *init-head* might turn up, a point I'll return to in section 5 below.

(19) Partial type hierarchy under *head*



3.3 Compositional semantics

Like most HPSG analyses, the heart of the semantics is in the lexical entries: They provide lists of semantic relations and specify how the semantics of the arguments are tied in to those relations. The role of the grammar rules is to gather up the relations from each of their daughters and to pass up the 'top handle' of the head daughter to the mother. Together, the grammar rules and lexical entries interact to ensure that the highest node for the interger phrase shown in (20) will contain the list of relations shown in (21). Although somewhat cumbersome, a walk through this list will show that it is a representation of the value 6210.

(20)

roku	sen	ni	hyaku	juu
six	*thousand*	*two*	*hundred*	*ten*

(21)

$$\left\langle \begin{bmatrix} \textit{const-rel} \\ \textsc{handel} & \boxed{1} \\ \textsc{const\_value} & \textit{'2} \end{bmatrix}, \begin{bmatrix} \textit{const-rel} \\ \textsc{handel} & \boxed{2} \\ \textsc{const\_value} & \textit{'100} \end{bmatrix}, \begin{bmatrix} \textit{times-rel} \\ \textsc{handel} & \boxed{3} \\ \textsc{factor1} & \boxed{1} \\ \textsc{factor2} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \textit{const-rel} \\ \textsc{handel} & \boxed{4} \\ \textsc{const\_value} & \textit{'10} \end{bmatrix}, \right.$$
$$\left. \begin{bmatrix} \textit{plus-rel} \\ \textsc{handel} & \boxed{5} \\ \textsc{term1} & \boxed{3} \\ \textsc{term2} & \boxed{4} \end{bmatrix}, \begin{bmatrix} \textit{const-rel} \\ \textsc{handel} & \boxed{6} \\ \textsc{const\_value} & \textit{'6} \end{bmatrix}, \begin{bmatrix} \textit{const-rel} \\ \textsc{handel} & \boxed{7} \\ \textsc{const\_value} & \textit{'1000} \end{bmatrix}, \begin{bmatrix} \textit{times-rel} \\ \textsc{handel} & \boxed{8} \\ \textsc{factor1} & \boxed{6} \\ \textsc{factor2} & \boxed{7} \end{bmatrix}, \begin{bmatrix} \textit{plus-rel} \\ \textsc{handel} & \boxed{9} \\ \textsc{term1} & \boxed{5} \\ \textsc{term2} & \boxed{8} \end{bmatrix} \right\rangle$$

4 Capturing further idiosyncrasies

This analysis is well-suited to capture idiosyncratic properties of specific number names.[2]

[2]Thanks to Atsuko Shimada for providing grammaticality judgments on these points.

The first case is certain number names (*juu* 'ten' and *hyaku* 'hundred') which can take any one-digit specifier, except *ichi* 'one':

(22) a. ni juu/hyaku
two *ten/hundred*
b. (*ichi) juu/hyaku
(*one*) *ten/hundred*

These facts can be handled by setting up contrast between the head types *int1* for *ni* 'two' through *kyuu* 'nine' and *int_ichi* for *it ichi* 'one', and group these together as *int1-or-fewer* (see (9)). *Juu* and *hyaku* will select for *int1* specifiers, while other number name heads that can take any single digit specifier will select for *int1-or-fewer* specifiers. Since all supertypes of *int1-or-fewer* are also supertypes of *int_ichi*, this analysis (correctly) predicts that any number name which can take a two digit or larger specifier can take *ichi* as a specifier as well.

The next wrinkle has to do with the behavior of *sen*, 'thousand'. It can occur with the specifier *ichi* 'one', but only when it is itself the specifier or complement of some larger number name. On the other hand, it can occur without any specifier, but not when it is the specifier for some larger number name.

(23) a. sen en
thousand yen
'one thousand yen'
b. *is- sen en
c. is- sen man en
one thousand ten thousand yen
'ten million yen'
d. *sen man en

These facts can be accommodated by positing two different lexical entries for *sen* 'thousand', with different head types. Sen_1 is [HEAD *int4-special*] and selects an *int1* specifier. Larger number names select for specifiers which are [HEAD *int4-or-fewer*], excluding this *sen*. Sen_2 selects for a *int_ichi* specifier, and is itself [HEAD *int4*]. Complements of number names larger than 1000 are [HEAD *int4-all*], or one of its supertypes, allowing either *sen* in this position. A further feature (e.g., [STANDALONE +/−]) to constrain the distribution of sen_2. Larger environments that select for number names (e.g., numeral classifiers) can place constraints on the value of this feature.

This section has been brief, and the data may well vary across dialects/idiolects, but I hope the analysis I have sketched show that the approach is able to account for idiosyncratic wrinkles in the distribution of number names without losing generalizations captured above.

5 Other uses of *init-head*

As mentioned above, the type hiearchy in (9) is designed to allow for the possibility that there are other subtypes of *init-head*. Indeed, since the implementation of the above analysis,

at least one further candidate has surfaced: (certain) numeral classifiers like elements. The elements in question are those that can occur with *han* 'half' as in (24a). *Nen* can occur by itself, as in (24b). Our analysis of numeral classifiers as well as formatives like *nen* is that they take number names as specifiers.

(24) a. ni nen han
two *year* *half*
'two and half years'
b. ni nen
two *year*
'two years'

When a phrase like *ni nen* combines with *han*, the result has to be that *han* combines semantically with the non-head part of the original phrase—that is, the 'two'. Furthermore, there has to be an additional plus relation to specify how the two connect. Given all this, the only word of the three that has access semantically to everything needed to put it all together is *nen*. The analysis that suggests itself is to have an entry for *nen* that provides the plus relation, selects a specifier as usual, and selects *han* as a complement—to its RIGHT.

Thus formatives like *nen* in combination with *han* provide another instance where the (marked) head-initial head-complement rule is useful. As further grammar development brings us to further minor constructions, I wouldn't be surprised if we found more such cases.[3]

6 Conclusion

The analysis presented here is just one of many possible analyses of number names in Japanese (and of marked word order in general). However, any such analysis would need a way of talking about number names (and numeral classifiers) on the one hand, and all other types of heads on the other. With such partial generalizations, type hierarchies provide an elegant means of capturing the generalizations that are there while allowing for the inevitable exceptions one finds in scaling a grammar up to real-world applications.

References

Copestake, Ann, Dan Flickinger, Robert Malouf, Susanne Riehemann, and Ivan A. Sag. 1995. Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, Leuven.*

Copestake, Ann, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001), Toulouse, France.*

[3]Indeed, we have found a few more cases in the more peripheral reaches of our grammar, including currency symbols like $ and ¥and the word *No.* 'number', all treated as numeral classifier-like formatives.

Martin, Samuel E. 1987. *A Reference Grammar of Japanese*. Tokyo: Charles E. Tuttle Company. 2nd edition.

Siegel, Melanie. 2000. HPSG analysis of Japanese. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*. Berlin: Springer.

Siegel, Melanie, and Emily M. Bender. 2002. Efficient deep processing of japanese. Unpublished ms., Stanford University and DFKI.

Smith, Jeffrey D. 1999. English number names in HPSG. In G. Webelhuth, J.-P. Koenig, and A. Kathol (Eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, 145–160. Stanford, CA: CSLI.

Zwicky, Arnold M. 1985. Heads. *Journal of Linguistics* 21:1–29.