Lund University, Dept. of Linguistics
Working Papers 42 (1994), 85-102

Outline of a Japanese AXG Parser

Birgitta Lastow

Introduction

The aim of this paper is to describe how a special type of parser applied to Japanese can be implemented and to discuss its characteristic features, capability and limitations. The parser uses a particular type of grammar called 'Appending X-bar Grammar' or 'AXG' devised by Prof. Bengt Sigurd (cf. Sigurd & Lastow 1993, Sigurd 1994). AXG is based on X-bar syntax, the system incorporated in most current grammatical theories (cf. Jackendoff 1977, Gazdar et al. 1985, Chomsky 1992, and the survey by Sells 1985). The main purpose of AXG was to test if it is possible to implement X-bar grammar in a simple way and to test its empirical potential. Parsers for Swedish and English in the same grammar representation can be used when performing machine translation (MT). The translation possibilities are at present very limited. This paper, however, will deal almost exclusively with the Japanese AXG module and the problems related to its implementation.

The Japanese parser can analyse core sentences containing subject, object, predicate (verbal or adjectival) and some different types of adverbs. The noun phrases may contain a demonstrative adjective, adjective ('*i*-type' or '*na*-type') and a noun or a relative clause preceding a noun. The verbal predicate contains the main verb and a number of auxiliaries showing durative aspect, perfective, politeness, negation and past tense. The adjectival predicate only contains the last three auxiliaries. Only one type of subclause and one type of special construction have been included. The most complete section of the parser is the morphology. The lexicon is limited and is a test-tool, rather than a well-developed part of the parser.

The Japanese parser has been developed as part of the Swetra project (cf. Sigurd et al. 1990) at the Department of Linguistics at Lund University, Lund, Sweden. The main goal has been to develop a parser for Japanese and find out to what extent it is possible to make a module with the same scope as the existing Swedish and English modules. The Swedish and English modules have similar grammar rules and use the same semantic notation and it is therefore comparatively easy to perform machine translation (MT) between these two languages. The Japanese module is very different and the MT task is therefore more complicated. The line of thought has however been eventually to extend the parser to a grammar module for machine translation. At present the Swedish and English modules cover all basic constructions, but the Japanese module still lacks essential syntactic patterns. We are also developing the Japanese module for use as a pedagogical tool in teaching and learning Japanese.

The AXG rule system

AXG consists of rules for joining single words into complex constituents. Each constituent has a number, a bar value, that indicates the level of complexity. The numbering in AXG does not completely follow the usual X-bar convention, since for mnemonic reasons the lowest number is 1 in AXG, representing a single word in the lexicon. The highest number is different for different languages. The Japanese module uses 10 and the Swedish and English modules developed by Bengt Sigurd, use 8 as the number for indicating a full sentence.

In Prolog, a logic programming language (cf. Rogers 1986), the predicate *append* is used for joining elements and this is what inspired the development of AXG. However, it is not practical to use *append* in the grammar because of the amount of backtracking it causes. To increase the speed of analysis and in order to shorten the code the pattern of Definite Clause Grammar (DCG) is used instead (cf. Gazdar and Mellish 1989). The idea of joining elements into larger complexes is maintained, however, and the name AXG is kept.

A typical AXG rule would be the following, where a single noun bar 1 is prefixed by an adjective bar 1, resulting in a complex noun bar 2:

```
jlexg(n,[B,B1],_,_,_,2,_,_,_) -->
        jlexg(a,B,_,_,_,1,_,_,_),
        jlexg(n,B1,_,_,_,1,_,_,_).
```

The word order in the complex noun is determined by the order of the individual elements in the rule. The first argument of the predicate *jlexg* (generalized lexicon) shows the word class. The letter *n* is used for nouns – complex or single – and *a* is used for adjectives. The second argument is used for building structures of word meaning representations, written in English 'Machinese'. The format of the lexical items is largely the same as in Swetra. As you can see in the rule above, the meaning representation of the noun bar 2, *[B, B1]*, is the sum of the meaning representations of its parts, *B* and *B1*, respectively. The meaning representation can serve as an interlingua between the AXG grammar modules of different languages. However, for translation into and from Japanese a functional representation is used. The functional representation represents the content of a sentence as a number of labelled chunks in an extra argument, where each chunk is represented by its meaning representation. The chunk labels are functional labels such as 'subj', 'pred', 'obj' and 'advl' (see *The functional representation* below).

AXG presupposes a language with no long distance dependencies, since the basic concept is the joining of elements, made with consideration only to the closest elements. Thus, only local constraints can be handled. The problem with the long distance dependencies found in Japanese is solved by introducing a special sentence level. Such a sentence level (see *The sentence level* below), is not implemented in the English or Swedish modules, where only AXG grammar rules have been used. However, a full treatment of passive sentences in Swedish and English would be simplified by using a special sentence level with different clause structures for active and passive clauses.

In AXG, agreement is handled by one of the arguments of the predicate and where necessary, a variable that guarantees the compatibility of the elements is inserted as this argument. This variable percolates the agreement value of the item with bar value 1, (single) word, to the level where agreement is crucial when adding a new word.

In the Japanese parser the agreement variable is used for other purposes than agreement between subject and predicate, for example to handle the compatibility of verb stems and the negation endings (co-occurrence restriction).

The Japanese parser – an overview

The Japanese AXG parser has been developed on a Macintosh IIsi using LPA MacProlog 4.5. The Japanese characters have been obtained by using the software 'The Japanese Language Kit' within the Swedish system version 7.1. The reference book for Japanese grammar was mainly Martin 1987.

The parser consists of three distinguishable parts, the grammar, the lexicon and the morphology module, which will be described below.

The grammar

The main part of the parser is the grammar, which contains rules for building sentences and their parts. The grammar has two subsections, the sentence constituent grammar rules and the sentence level, with sentence and clause structure rules. Some help predicates are also included, but they are not really part of the grammar.

The sentence constituent rules, but not the rules at the sentence level, are AXG rules. The clause structure rules for building full sentences are of the DCG format, but their righthand side consists of ordinary Prolog predicates. These predicates, however, call AXG rules.

The sentence level of the Japanese module was introduced mainly because of the co-occurrence restrictions between the verb and the particles in adverbials, but also because of the existence of rather large and well-defined segments of Japanese sentences. Japanese also has a large number of different sentence types, which are easily implemented by using clause structures.

Since Japanese is a SOV language, the verb always comes last and it is therefore not possible to decide whether a particle in the middle of a sentence is correct or not, before the parsing of the verb is completed. The choice of particle after a place expression would for example be *de* if the verb is a verb of action, but *ni* if it is a motion verb. In the case of clause structures, the choice of particle leads to a demand on the subcategory of the verb, which limits the choice of verbs. In the case with only AXG rules, a verb incompatible with the particle might be chosen, and then it will cause backtracking to the level of the particle, and a lot of work has to be repeated.

This fact suggests different rules for sentences with action and motion verbs respectively. Thus, the parsing of the preceding time adverbial and the subject may have to be redone, if the wrong clause structure were entered, but since most noun phrases and adverbial phrases are less complex this is a better alternative. The sentence level is therefore a convenient way to limit the amount of backtracking and it also shows the strong binding between the noun complex and its following particle and the somewhat weaker binding between the different noun phrases and verb phrases.

The morphology module

The morphology module consists of rules that conjugate all Japanese verbs and a table of endings that are used for conjugation. So far, all verb forms have not been used in the grammar module. The morphological rules are bidirectional, i.e. it is possible to either conjugate an infinitive form or to obtain the infinitive form from a conjugated form. There are 14 verb groups in Japanese.

The morphological rules include rules for building non-finite items (see *Non-finite items* below), i.e. complexes to which it is possible to add a finite item and thus obtain a finite complex. The so-called 'non-finite items' are abstract forms, used only in combination with a finite item. The morphology module also includes morphology help predicates which conjugate the verbs and find the correct ending.

When conjugating an infinitive form, the infinitival ending is cut off from the verb. The infinitival ending is given in the lexicon as an argument of each verb (see the description of the lexicon below). The last step is to add the correct ending to the stem of the verb. The ending is found in the conjugation table on the basis of the infinitival ending which shows which verb group the verb belongs to. Going in the opposite direction means looking up the infinitival form in the table on the basis of the inflectional ending.

The lexicon

When developing the lexicon the aim has not been to make a large and representative lexicon, but rather to make the smallest possible lexicon suitable for testing the grammar rules. The lexicon entries are instances of the fact *jlex* (Japanese lexicon). These facts have nine arguments (see examples below). The first argument shows the category and the second the meaning in 'Machinese'. The sixth is used for the bar value. These three arguments are always used. The remaining slots are used for information like subcategory, finiteness, co-occurrence restriction, co-occurrence value, intransitivity or transitivity and infinitival ending.

Example of adjective entries

```
/* na-type */
jlex(na,m(beautiful,_),_,fin,_,1,_,_,_) --> ['きれい'].
      %kirei
/* i-type */
jlex(a,m(young,_),_,fin,agr(dict, _, _),1,_,_,_) --> ['若い'].
      %wakai
```

Example of adverb entries

```
jlex(manneradv,m(slowly,_),_,_,_,1,_,_,_) --> ['ゆっくり'].
      %yukkuri
jlex(timeadv,m(today,_),_,_,_,2,_,_,_) --> ['今日'].
      %kyoo
```

Example of noun entries

```
jlex(n,m(pron1,pl),_,_,_,3,_,_,_) --> ['私達'].
      %watashitachi
jlex(n,m(tokyo,_),place,_,_,1,_,_,_) --> ['東京'].
      %Tokyo
jlex(n,m(book,_),ord,_,_,1,_,_,_) --> ['本'].
      %hon
```

Example of main verb entries

```
jlex(v,m(write,pres),action,fin,agr(dict,pl,aff),1,none,'く',['を']) -->
      ['書く'].            %kaku
jlex(v,m(live,pres),existence,fin,agr(dict,pl,aff),1,none,'む',['が']) -->
      ['住む'].            %sumu
```

The main verb entries are the entries containing the most information. The main verbs *kaku* 'write' and *sumu* 'live' are given below. The first argument gives the category *v* for verbs. The second argument gives the meaning and tense represented as two arguments to the fact *m*. The third argument gives the subcategory, and the fourth states that this is a finite form of the verb. The fifth argument shows that the verb is in the dictionary form, i.e. infinitival form, the style level is *pl*(ain) form and that it is an affirmative form of the verb. This argument is the co-occurrence value, i.e. the representation of the co-occurrence features of the verb. This information is used for example when building a predicate complex, since certain auxiliaries have specific co-occurrence restrictions on the form of the main verb. The sixth argument is the bar value. The number *1* indicates that this is a single verb. The seventh argument states the demand the verb has on other words or constituents, i.e. the co-occurrence restriction. If the co-occurrence value of another word or constituent is the same as the co-occurrence restriction of the verb, that verb may be appended to the word or constituent. For main verbs this demand concerns the choice of particle after the nominal phrase being subject in an active clause. *None* means that the verb has no particular demand, i.e. the particle could be either *wa* or *ga* (see also *Limitations of the parser*). For auxiliaries, this argument is used for stating the demand on the form of the preceding verb or verb complex. The eighth argument contains the infinitival ending and the ninth argument gives the object particle – を *o* for transitive verbs and が *ga* for intransitive verbs.

The Japanese parser in detail

The verbal predicate

The longest possible verbal predicate is:

Verb(-TE)	shimatte	imasen	deshita
Stem	PERF	DUR + POL + NEG	PAST

The order of the auxiliaries conveying durative aspect, politeness, negation and past tense is always the same as above (cf. Nakau 1973). This particular predicate is however unlikely to occur in ordinary speech, although it is possible to generate it. A more likely predicate would exclude either the perfective or the durative auxiliary.

The words and markers above showing perfective aspect, durative aspect, politeness and past tense are treated as auxiliaries (cf. Sigurd 1992). Each auxiliary has a specific demand on the preceding verb complex and this demand is expressed by co-occurrence features. For more information on these demands, see *Examples of predicate rules* below.

The functional representation

The functional representation is the second argument of those instances of the predicate 'jlexg' that represent verbs bar 4 or higher, adjectival nouns bar 5 or higher, adjectives bar 6 or higher and the predicates 'sent', 'main_cl' and 'sub_cl'. Main and subordinate clauses have the following standardised functional representations:

Main:	[advl(A1), subj(S), obj(O), advl ([A2,A3]), pred(Pr), part(Sp)]
Sub:	[advl(A1), subj(S), obj(O), advl ([A2,A3]), pred(Pr), conj(C)]

The only difference is that the main clause may have a sentence particle at the end, and the subordinate clause ends with a subordinate conjunction. In fact, the predicate for subclauses calls the predicate for main clauses, so that a subclause will be a main clause without sentence particle, followed by a conjunction. A full sentence may be only a main clause or a subclause followed by a main clause. If the sentence consists of coordinated clauses, the meaning representation and the functional representation of each clause are joined.

All parts of the functional representation, except the predicate, have the meaning representation as their argument, i.e. the argument variables are instantiated to the meaning representation for each part. The predicate part, however, is complex. The argument of the predicate, *Pr*, has the following inner structure:

verbal predicate:	[mainV(Mv), aux(Auxes), neg(Negation), past(Past)]
adjectival predicate:	[adj([B, Pol]), neg(Negation), past(Past)]

The argument of these predicates, however, is the meaning representation for the corresponding part of the predicate. The argument of 'aux' will include the word meaning representations of the auxiliaries showing perfective and durative aspect, if any. The argument of 'adj' consists of two parts, the word meaning representation of the adjective, *B*, and the politeness marker, *Pol*, if any. See *Examples of output* for examples of functional and semantic representations of analysed sentences. See also 'Problems with AXG and MT'.

The sentence level

The sentence level is part of the grammar and consists of several different clause structures and sentence structures. Each type of sentence structure is represented by a rule.

There are two types of main clauses, where the predicate may either be a verbal predicate, 'pred_v', that may take an object, or an adjectival predicate, 'pred_adj', which does not take an object. They have the same clause parts, except for the object and some adverbials. The structures containing a verbal predicate may have a 'place of existence' adverbial or a 'place of action adverbial'. The type of adverbial requires a verb of the same type (subcategory, represented by the variable *V*), i.e. a verb of existence or an action verb.

```
main_cl(v,[A1,S,O,[A2,A3],P,Sp],
[advl(A1),subj(S),obj(O),advl([A2,A3]),pred(Pr),part(Sp)],
V,fin,Agr,9,Dem,_,[Part]) -->
                timeadvl(A1),
                subj(S,Dem),
                place_of_existence_advl(A2,V),
                obj(O,[Part]),
                manneradvl(A3),
                pred_v(P,[pred(Pr)],V,fin,Dem,[Part]),
                sent_part(Sp).
```

Clause parts

The parts of the clauses at the sentence level are called 'clause parts', which call AXG rules. The clause parts 'translate' between functional roles and categories. Most of the constituents are optional, i.e. some may be left out. If they are non-existent they are represented by an empty list, '[]'. Below are examples of clause parts occurring in the parser.

```
timeadvl(A) -->
        jlexg(timeadv,A,_,_,_,2,_,_,_).
timeadvl([]) --> []. % no time adverbial

subj(S,Dem) -->
        jlexg(n,S,_,_,Dem,4,_,_,_).
subj([],_) --> []. % no subject

obj(O,[Part]) -->
        jlexg(n,O,_,_,_,3,_,_,_),
        [Part].
obj([],[Part]) --> []. % no object

pred_v(P,[pred(Pr)],V,fin,Dem,[Part]) -->
        jlexg(v,P,[pred(Pr)],V,fin,Agr,8,Dem,_,[Part]).

sub_conj(C, agr(_,pl,_)) -->
        jlex(subconj,C,_,_,_,_,1,agr(_,pl,_),_,_).
```

Examples of noun rules

The following rule shows how a *na*-type adjective is added to a noun, resulting in a noun bar 2. This requires the insertion of な *na* between the adjective and the noun.

```
jlexg(n,[B,B1],_,_,_,2,_,_,_) -->
        jlexg(na,B,_,_,_,1,_,_,_),
        ['な'],
        jlexg(n,B1,_,_,_,1,_,_,_).
```

It is not necessary to have an adjective, so a noun bar 2 may also just be single noun. This optionality is expressed in the following rule.

```
jlexg(n,B,_,_,_,2,_,_,_) --> jlexg(n,B,_,_,_,1,_,_,_).
```

Nouns with higher bar values also include a demonstrative adjective, a relative clause and a particle. The possible particles are *ga*, *wa* and *ni* showing subject, topic and agent respectively. The following rule shows the case of the agent particle, which is used in sentences with a causative verb.

```
jlexg(n,B,_,_,agr(causdict,_,_),4,_,_,_) -->
        jlexg(n,B,_,_,_,3,_,_,_),
        ['に'].
```

Examples of adjectival predicate rules

In Japanese, it is possible to have adjectives as predicates. In the case of *na*-type adjectives the copula is mandatory because of their noun-like qualities. Thus the adjectival predicate with lowest bar value would look like this:

```
jlexg(na,[m(be,pres),B],_,fin,agr(dict,pl,aff),5,_,_,_) -->
        jlexg(na,B,_,_,_,1,_,_,_),
        jlexg(v,m(be,pres),cop,fin,agr(dict,pl,aff),1,_,_,_).
```

The bar value 5 is used because the copula is found neither in causative nor passive form, nor with the auxiliaries *shimau* and *iru*. The politeness marker is attached to verbal predicates with bar value 5, producing a verb complex with bar value 6, hence the adjectival predicate is given bar value 5. The corresponding rule for *i*-type adjectival predicates is the following, in which case the copula is not mandatory and the predicate is a simple adjective:

```
jlexg(a,B,_,fin,Agr,5,_,_,_) --> jlexg(a,B,_,fin,Agr,1,_,_,_).
```

The adjectival predicates are similar to verbal predicates in that they can take a politeness marker, negation marker and past tense marker, which leads to an increase in bar values from 5 to 8. The markers are all optional.

Examples of verbal predicate rules

The bar value 2 is reserved for verbal predicates in causative form and bar value 3 for passive form. However, at present the grammar does not allow passive causatives and hence the bar value 3 is used for both causatives and passives. The following rule shows how causative forms are obtained. The predicate 'causativize' is part of the morphology module.

```
jlexg(v,B,V,fin,agr(dict,pl,aff),3,agr(causdict,_,_),eru_gr2,['を']) -->
        [X_caus], % X_caus correspond to a causative main Verb
        {jlexg(v,B,V,fin,agr(dict,_,_),1,_,Ending,['を'],[X], []),
        V\=cop,
        V\=aux,
        causativize(Ending, X, X_caus)}.
```

The co-occurrence feature is changed into 'agr(causdict,_,_)', which guarantees that the correct particle is attached to the agent.

Higher bar values are obtained by adding auxiliaries to the main verb. The auxiliaries show perfective aspect, durative aspect, politeness and past tense. They have different demands on the preceding verb complex, which are expressed as co-occurrence features. The auxiliary showing perfective aspect is *shimau* and it demands that the preceding verb is in *te*-fórm. This demand is expressed by the variable *Dem_verb*. The variable *Dem_subj* is expressing a demand of a particular particle that is to be attached to the subject as in the case with causatives discussed above.

```
jlexg(v,[B,m(_,perf)],[pred([mainV(B),aux([m(_,perf)])])],V,fin,Agr,4,Dem_
subj,Ending,[P]) -->
        jlexg(v,B,V,nonf,Dem_verb,3,Dem_subj,_,[P]),
        jlexg(v,m(_, perf),aux,fin,Agr,1,Dem_verb,Ending,_),
        {V\=aux,
        V\= cop}.
```

The verb complex is changed to *te*-form by the morphological rules creating non-finite complexes (see *Non-finite items* below).

The adding of the other auxiliaries is governed by similar rules. All auxiliaries are optional and the rules expressing optionality just percolate all features into a verbal predicate with a higher bar value.

Non-finite items

The morphology module includes rules for changing the finite items in the lexicon into non-finite items, i.e. verbs or adjectives, which could not be used as the last word of a sentence. The sentence constituent grammar rules use non-finite items for example in front of the auxiliaries. The auxiliaries demand that the verb complex preceding them is in a specific form. The non-finite rules make the appropriate forms of the constituents needed by altering their ending, which is done by using the help predicates in the morphology module. The non-finite items are needed because they facilitate the assumption of additive morphology in the sentence constituent grammar rules. Japanese morphology is, however, not truly additive, e.g. when adding the past tense marker to a verb in polite form, the past tense marker *-ta* cannot simply be added to the verb, because that would give the form *-masuta*, but the correct form is *-mashita*. Thus, it is necessary to remove the ending *-su*, then add *-shi* and then *-ta*.

The morphology module has four parts, which contain rules for *na*-type adjectives, *i*-type adjectives, copula and verbs. The rules for verb morphology include, for example, rules for changing the ending of main verbs like the following, which changes a verb in dictionary form into *ta*-form (past tense):

```
jlexg(v,B,V,nonf,agr(dict,pl,aff),3,Dem,Ending,[P]) -->
        [X_ta], % X_ta correspond to main Verb
        {jlexg(v,B,V,fin,agr(dict,_,_),3,Dem,Ending,[P],[X], []),
        V\=cop,
        V\=aux,
        make_ta(Ending, X, X_ta)}.
```

The actual cutting and pasting of endings are done with morphology help predicates like 'make_ta' above. The new ending is found in a conjugation table, where the row is given by the infinitival ending of the verb and the column by the form that is to be obtained. The table includes one column for each of the eight verb stems or verb forms possible in Japanese. The morphology help predicates also include predicates for obtaining the passive and causative forms of verbs. The row, i.e. the predicate, for verbs having the infinitival ending *bu* would be 'bu(ba, bi, bu, be, boo, nde, nda, be)'.

Rules for verb complexes including auxiliaries are also included in the morphology. For example, the following rule constructs a verb complex of the form *V(-te) shimatte*, where 'V' is a main verb. The verb complex is made by adding the auxiliary *shimau* in *te*-form, *shimatte*, to a verb in *te*-form.

```
jlexg(v,[B,m(_,perf)],[pred([mainV(B),aux([m(_,perf)])])],
V,nonf,Agr,4,Dem_subj,Ending,[P]) -->
        jlexg(v,B,V,nonf,Dem,3,Dem_subj,_,[P]),
        jlexg(v,m(_,perf),aux,nonf,Agr,1,Dem,Ending,_),
        {V\=cop}.
```

The Japanese parser and MT

MT using the functional representation as interlingua

The functional representation is used as an interlingua when performing machine translation, as is usual in Swetra. The structure of the functional representation has been described above and we will now see how it is used for the purpose of translation.

The analysis of a sentence returns the functional representation and the meaning representation. The functional representation is then manipulated and the new functional representation obtained is used when generating the translated sentence.

The rules for manipulating the functional representation are to be thought of as transfer rules. The necessary actions which are taken when translating from Swedish into Japanese are, for example, to delete the word meaning representation of all articles, to check if the Swedish sentence contains a negation and to check whether the sentence is in present or past tense. This means stepping through the lists of word meaning representations in search of this information. The resultant meaning representation chunks are included in the Japanese functional representation, and maybe in other parts.

When translating from Japanese, the contrary actions have to be taken. The representation of the article that is inserted in front of all nouns in the Japanese functional representation is at present the indefinite article, because the nouns in the Swedish or English lexicon are indefinite. It would be possible to add some more rules that introduce the definite article and change the co-occurrence features of the noun. That would, however, cause some minor problems in Swedish, since definiteness is also expressed by endings in Swedish, not only by separate articles. At present the same rules may be used both when translating into English and into Swedish.

The use of articles, which is dependent on the meaning and aspect of the sentence, and problems with inserting the correct article when translating from a language which does not have articles into a language which has, is a field of research in itself (see Gawrońska 1993).

Problems

Most of the problems with Japanese-English/Swedish MT has its origin in the differences between Japanese and Swedish or English sentence structures. The information on, for example, tense and negation is encoded in different ways in these languages. The tense marker is attached to the main verb in Swedish, but in Japanese it is attached to the last verb in the predicate complex, which could be a main verb but in most cases is an auxiliary. Negation is a separate word in Swedish, but an ending in Japanese.

Adverbials are also a major problem, since they can be inserted in many places in a Swedish sentence, but have more restricted places in a Japanese sentence. The type of adverbial is also crucial in Japanese, since manner adverbials are placed just in front of the predicate, but other adverbials are often placed at the beginning of the sentence.

These problems should, however, be solvable by using fairly intricate transfer rules. Problems that are more difficult to solve are, for example, when the passive voice is to be used in Japanese, and how Swedish and English auxiliaries are to be expressed in Japanese and vice versa. The use of subordinate clauses also causes problems.

The order and complexity of the meaning representation of nouns bar 3 also vary in the different modules at present, making translation impossible. It would not, however, be difficult to rearrange the word meaning representations so that they would fit, but one has to decide which language module should be altered.

Limitations of the parser

The grammar of the Japanese parser is by no means adequate to cover all common Japanese sentences. The coverage of subordinate clauses is particularly small. For the time being only a subordinate clause ending with the particle *to* is included. This type of subordinate clause corresponds to a conditional clause, where the condition brings about a non-controllable event or state. There are, however, other ways of forming a conditional clause in Japanese. One is to use the *ba*-form of the verb at the end of the subordinate clause. This construction only indicates that the subordinate clause expresses a condition. How to make the correct choice of construction is a difficult problem.

No connective words that usually occur at the beginning of a sentence are included, like *tokoro ga* 'but, still, however' and *shikashi* 'but, however', *shitagatte* 'therefore, consequently'. These words refer to what has been said in the sentence(s) preceding the one under consideration and are therefore difficult to give an exact semantic interpretation for. This is also true for the conjunctions or connective forms of verbs. The interpretation depends on both parts of the sentence and their ordering, and not only on the meaning of the conjunction. The same connective verb form can have different interpretations in different contexts.

Another area that has not been included at all is comparison. Many particles are not included either and no distinction between the topic marker *wa* and the subject marker *ga* has been made. The rules for when to use *wa* and when to use *ga* are very complicated (cf. Kuno 1973). The particle *no* which functions as a nominaliser or a connector between nouns has not been included.

Derivations of words from other words are not taken care of. The construction of adverbials from adjectives would be easy to implement, but would yield semantic problems. The construction of adverbs with a noun expression larger than a single noun and a particle is not handled at present.

The choice of plain or polite form is left to the user as a global parameter. The most polite forms like *gozaru* 'be' are not included. The potential form and the honorific form of verbs are not included either.

Passive and causative of intransitive verbs are not handled. They give rise to semantic problems when translating and also problems with particles.

Words or expressions for onomatopoeia, phenomimes (phonetic representations of phenomena perceptible by non-auditory senses) and psychomimes (phonetic representations of human psychological states) are not included either.

Examples of output

The following examples illustrate the output of different types of Japanese sentences. The examples consist of three parts – the example sentence, the meaning representation and the functional representation. The last section contains some simple translated sentences.

Analysis of Japanese sentences

1. Relative clause (adjoined to the subject) and question particle.

```
[書いた, 本, を, 読みました, か]
[[], [], [[m(write, pres), m(_1404, past)], m(book, _1663)], [[], []],
[m(read, pres), m(_1688, past)], m(questionmark, _2069)]

[advl([]), subj([]), obj([[m(write, pres), m(_1404, past)],
m(book, _1663)]), advl([[], []]), pred([mainV(m(read, pres)), aux([]),
neg([]), past(m(_1688, past))]), part(m(questionmark, _2069))]
```

2. Affirmative i-type adjectival predicate, polite form, present tense.

```
[その, きれい, な, 本, は, 軽い, です]
[[], _969, [m(that, _1044), [m(beautiful, _1065), m(book, _1076)]],
[], _975, [m(light, _1190), m(_1162, pol)], _979]

[advl([]), subj([m(that, _1044), [m(beautiful, _1065), m(book, _1076)]]),
obj([]), advl([[], []]), pred([adj([m(light, _1190), m(_1170, pol)]), neg([]),
past([])]), part(_979)]
```

3. Negative i-type adjectival predicate, polite form, past tense.

```
[男の子, は, 若く, ありません, でした]
[[], _1236, m(boy, _1312), [], _1242, [[m(young, _1482), m(not, _1423)],
m(_1340, past)], _1246]

[advl([]), subj(m(boy, _1312)), obj([]), advl([[], []]),
pred([adj([m(young, _1482), []]), neg(m(not, _1383)),
past(m(_1340, past))]), part(_1246)]
```

4. Subject particle 'が'. Time adverb and manner adverb.

```
[今日, 男の子, が, 本, を, ゆっくり, 読みます]
[m(today, _1313), m(boy, _1342), m(book, _1369), [[],
m(slowly, _1380)], [m(read, pres), m(_1474, pol)], _1267]
```

[advl(m(today, _1313)), subj(m(boy, _1342)), obj(m(book, _1369)),
advl([[], m(slowly, _1380)]), pred([mainV(m(read, pres)), aux([]),
neg([]), past([])]), part(_1267)]

5. Place of existence adverbial and verb of existence.
   Topic particle 'は'. Negative verbal predicate, polite form, past tense.
   [私達, は, 東京, に, 住んで, いません,
   でした]
   [[], m(pron1, pl), [], [m(tokyo, _3378), []], [[[m(live, pres), m(be, dur)],
   m(not, _5006)], m(_4286, past)], _2735]

   [advl([]), subj(m(pron1, pl)), obj([]), advl([m(tokyo, _3378), []]),
   pred([mainV(m(live, pres)), aux([m(be, dur)]), neg(m(not, pol)),
   past(m(_4286, past))]), part(_2735)]

6. Subordinate clause (ending with 'と') .Affirmative verbal predicate,
   plain form, present tense.
   [本, を, 読む, と, わかる]
   [[[], [], m(book, _9789), [[], []], m(read, pres)], m(if, _9939), [[], [], [],
   [[], []], m(understand, pres), _9578]]

   [advl([]), subj([]), obj(m(book, _9789)), advl([[], []]),
   pred([mainV(m(read, pres)), aux([]), neg([]), past([])]),
   conj(m(if, _9939)), advl([]), subj([]), obj([]), advl([[], []]),
   pred([mainV(m(understand, pres)), aux([]), neg([]), past([])]),
   part(_9578)]

Translation examples

1. 私達 は 本 を 書く
   [m(pron1, pl), [m(write, pres), [m(indef, _1549), m(book, sg)]]]
   We write a book

2. Vi skriver inte en bok
   [m(pron1, pl), [m(write, pres), m(not, _1331), [m(indef, _1350),
   m(book, sg)]]]
   私達 は 本 を 書きません

3. Vi skriver en ny bok
   [m(pron1, pl), [m(write, pres), [m(indef, _1624), [m(new, _1651),
   m(book, sg)]]]]
   私達 は 新しい 本 を 書きます

Conclusions

The present version of the Japanese parser can analyse the basic uncoordinated sentences of Japanese. The number of sentences that it is possible to translate is smaller.

The AXG format of the grammar is easy to understand. It makes it possible to divide the parsing problem into separate smaller problems, which also makes it easy to obtain an overall view of the grammar. Furthermore, it is easy to expand an AXG-module, since the rules are not connected to each other in any way, except for the bar value. If the language which is to be implemented had a purely additive morphology and no interaction between constituents it would be very easy to implement. In the case of Japanese the morphology is not additive, and the attempt to make it look as if it were additive from the grammar's point of view has made the morphological rules complicated and difficult to understand. However, the morphology is bi-directional, which is an advantage.

Using special clause structures is convenient, but makes the program somewhat slower. The more layers that are added, the greater the complexity of the grammar and the number of possible sentences increases rapidly, which slows down the processing. Most of the problems mentioned are, however, shared with all formal grammars.

Machine translation (MT) with functional representation works well for the sentence types included at present and there is no reason why it should not work equally well with more complex sentences. MT without the functional representation is more or less impossible, since translation requires some manipulation of the meaning representation (due to differences in the parsing tree) before performing the actual translation. This manipulation requires knowledge of where to find particular constituents, which cannot be located in the meaning representation of the whole sentence (at least not without great effort). Thus, the functional representation provides a segmentation of the word meaning representations, thereby enabling searches for particular constituents, for example when adding an article in front of a noun.

When performing MT one would also like to be able to translate some parts even if the whole sentence cannot be translated. This possibility of partial translation is missing in the Japanese parser. However, one could write a predicate that is called only when the translation predicate fails, which gives the translation of each clause part in the clause that is translated successfully. By backtracking this could result in several possible translations if the clauses may be divided into parts in different ways.

In the future, the Japanese parser will be used in a program for practising and learning Japanese. The pedagogical purpose will limit the need for including very complex sentences, in the same way that the sentences in text books for teaching Japanese are of limited complexity. A user-interface which hides the Prolog clauses and makes the environment user friendly will be added. This interface will also be written in Prolog, as far as it is possible, and the remainder will probably be written in Pascal.

References

Chomsky, Noam. 1992. *A minimalist program for linguistic theory*. Occasional Papers in Linguistics Number 1. Cambridge (Mass): MIT Press.

Gawrońska, Barbara. 1993. *An MT oriented model of aspect and article semantics*. Lund: Lund University Press.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan Sag. 1985. *Generalized phrase structure grammar*. Oxford: Basil Blackwell.

Gazdar, Gerald & Christopher Mellish. 1989. *Natural language processing in PROLOG*. Reading (Mass.): Addison-Wesley.

Jackendoff, Ray. 1977. *X-bar syntax: A study of phrase structure*. Cambridge (Mass): MIT Press.

Kuno, Susumu. 1973. *The structure of the Japanese language*. Cambridge (Mass): MIT Press.

Martin, Samuel E. 1987. *A reference grammar of Japanese*. Rutland, Vermont: Tuttle.

Nakau, Minoru. 1973. *Sentential complementation in Japanese*. Tokyo: Kaitakusha.

Rogers, Jean B. 1986. *A Prolog primer*. Reading (Mass.): Addison-Wesley.

Sells, Peter. 1985. *Lectures on contemporary syntactic theories*. Publications of the CSLI. Stanford.

Sigurd, Bengt. 1992. 'A new analysis for machine translation of the auxiliary and main verb complex'. *Studia Linguistica* 46:1, 30-48.

Sigurd, Bengt. 1994. 'Fragments of a Swedish appending X-bar grammar (AXG)'. In Nils Jörgensen, Christer Platzack & Jan Svensson (eds.), *Språkbruk, grammatik och språkförändring. Festskrift till Ulf Teleman*, 219-228. Lund: Dept. of Scandinavian Languages.

Sigurd, Bengt, Mats Eeg-Olofsson, Barbara Gawrońska & Per Warter. 1990. *SWETRA – A multilanguage translation system*. Praktisk Lingvistik 14. Dept. of Linguistics, Lund University

Sigurd, Bengt & Birgitta Lastow. 1993. *Fragments of a Swedish appending X-bar grammar (AXG)*. Proceedings NLPRS '93, Fukuoka, Japan, 6-7 Dec., 353-357. Fukuoka: Organizing Committee of Natural Language Processing Pacific Rim Symposium.

Lund University, Dept. of Linguistics
Working Papers 42 (1994), 103-115

Aspect and Species – A Comparison Between Polish and Swedish

Ann Lindvall

1. Introduction

This article presents a comparison between the two linguistic categories verbal aspect and nominal species in two languages, Polish and Swedish, and intends to prove a correlation between these categories. This might seem surprising, as both categories on the surface seem totally independent from each other.

Aspect, with its values imperfective/perfective, is a property of the verb, while species, with its values indefiniteness/definiteness, is a property of the noun. Aspect is a basic feature in the Slavic languages, while species is characteristic for the Germanic and Romance languages. However, there are strong reasons to believe that both properties, aspect and species, exist in all languages, albeit sometimes in a form not explicitly recognisable as such. The theoretical point of departure is a functional typological approach.

Aspect as a universal phenomenon has been studied by Givón 1984, who stresses its importance for the sentence as a whole, with its propositional-semantic features. The existence of species in the languages of the world has been described in a large typological study by Krámský 1972. Especially interesting are studies of the whole sentence according to the theme/rheme-principle, as well as studies of the text in general. In all of these studies, the context plays an important role. Comparisons between Czech and English have been done by Mathesius 1961, whilst an extensive analysis of aspect and article in Russian/Polish and English/Swedish has been done by Gawrońska 1993.

It is the hypothesis of this article that there is a deeper relation between verbal aspect and nominal species, especially within the direct object. It is probably not a strong direct correlation, as many other factors also play a