**Abstract**

This thesis describes the implementation of the open-source Japanese resource grammar as a part of the GF Resource Grammar Library (RGL). GF (Grammatical Framework) is a grammar formalism for multilingual grammars and their applications intended at performing various natural language processing tasks. The RGL is a set of natural language grammars with a shared abstract syntax and different concrete parts implemented in GF.

The Japanese resource grammar covers all the categories and rules of the GF abstract syntax, thus providing the full correlation with the resource grammars of other languages in the RGL. Due to some peculiarities of the Japanese language, the process of grammar development was complicated by a number of challenges that had not been observed before in the implementation of other GF grammars. One of the most important peculiarities is stylistic stratification in Japanese, which deeply affects its morphology and syntax. Moreover, a number of syntactic constructions predetermined in the abstract syntax are hardly possible in the natural Japanese language (e.g. some types of complex subordinate clauses). Being typologically distant from the European languages, Japanese brings up new issues in the discussion on the universal properties of languages and disputes the generality of some rules in the GF abstract syntax.

**Keywords:** Japanese, Grammatical Framework, resource grammar, natural language processing, computational linguistics, machine translation.
Contents

1 Introduction .......................................................... 5
   1.1 The GF system and Resource Grammar Library .............. 5
   1.2 Example of a simple GF grammar ............................. 6
   1.3 Thesis structure .............................................. 7

2 Implementation of the Japanese resource grammar in GF 8
   2.1 Structure of a GF resource grammar .......................... 8
   2.2 General characteristics of Japanese morphology. Stylistic
differentiation ..................................................... 9
   2.3 Noun Phrases .................................................. 10
      2.3.1 Nouns .................................................... 10
         2.3.1.1 Counting and counter words ..................... 11
         2.3.1.2 Relational nouns ................................. 12
         2.3.1.3 Proper nouns .................................... 12
      2.3.2 Quantifiers .............................................. 12
      2.3.3 Number determining elements .......................... 13
      2.3.4 Determiners ............................................. 13
      2.3.5 Personal pronouns .................................... 14
      2.3.6 Means of NP modification ............................. 14
      2.3.7 Conjunction of noun phrases .......................... 16
   2.4 Adjective Phrases ............................................. 16
      2.4.1 Types of adjectives .................................... 16
      2.4.2 Comparative adjectives ............................... 18
      2.4.3 Means of AP modification ............................. 19
      2.4.4 Conjunction of adjective phrases ..................... 19
   2.5 Verb Phrases .................................................. 20
      2.5.1 Verbs .................................................... 20
         2.5.1.1 Verb groups ...................................... 20
         2.5.1.2 Verb conjugation ................................... 21
         2.5.1.3 2- and 3-place verbs. Passive voice .......... 24
      2.5.2 Means of complementation of verbs and verb phrases .. 25
         2.5.2.1 Verb-phrase-complement verbs ................... 25
         2.5.2.2 Sentences and questions ........................ 26
2.5.2.3 Adjectives ........................................... 28
2.5.2.4 Adverbs ............................................. 28
2.5.3 Verb phrases missing complements .......................... 29
  2.5.3.1 Verbs with NP and VP/S/QS/AP complements ............. 30
  2.5.3.2 Complementation of VPSlash with VV and V2V .......... 31
  2.5.3.3 Complementation of VPSlash with the reflexive pronoun 32
2.5.4 Copula, complements to copula ............................. 32
2.6 Adverbs .................................................. 34
  2.6.1 Forming of adverbs ..................................... 34
  2.6.2 Comparative adverbs ................................... 35
  2.6.3 Conjunction of adverbs .................................. 35
2.7 Numerals .................................................. 36
  2.7.1 Cardinals ............................................... 36
  2.7.2 Ordinals ............................................... 37
2.8 Tense ..................................................... 38
2.9 Sentences .................................................. 38
  2.9.1 Declarative sentences .................................... 38
  2.9.2 Interrogative sentences .................................. 40
  2.9.3 Conjunction of sentences ................................ 42
  2.9.4 Subordinate clauses ...................................... 43
    2.9.4.1 Relative clauses .................................... 43
    2.9.4.2 Conjunction of relative clauses ....................... 44
    2.9.4.3 Other subordinate clauses ............................ 45

3 Evaluation of the Japanese Resource Grammar 46

4 Conclusion .................................................. 50
  4.1 Related work on building Japanese computational grammars 50
  4.2 Prospects for future development ............................ 51
  4.3 Results .................................................. 51

Bibliography .................................................. 53
Chapter 1
Introduction

1.1 The GF system and Resource Grammar Library

GF (Grammatical Framework, Ranta 2011) is a grammar formalism for multilingual grammars and their applications. Together with implementation it forms a framework for performing various Natural Language Processing (NLP) tasks.

A GF grammar consists of an abstract syntax, which defines a set of abstract syntax trees basing on the semantically relevant language structure, and one or more concrete syntaxes, which define how abstract syntax trees are mapped to (and from) strings.

The process of producing a string from an abstract syntax tree is called linearization. The opposite, producing an abstract syntax tree (or several, if the grammar is ambiguous) from a string is called parsing.

The important feature of the GF grammar formalism is a type system on both abstract and concrete syntax levels. This makes it possible to verify the well-formedness of an input as well as resolve ambiguities using semantic information contained in the type of the input.

The GF Resource Grammar Library (RGL) is a set of natural language grammars implemented in GF. These grammars are in a strong sense parallel: they are built upon a common abstract syntax, i.e. a common tree structure. The RGL aims at fairly complete descriptions of different natural languages, starting from the perspective of the linguistics structure.

The GF Resource Grammar Library is roughly divided into morphological and syntactic components. The morphological component is different for various languages, since it regards the inflection mechanisms. The syntax component displays a stronger parallelism: all languages in the library have a common representation of syntactic structures and structural words.

The GF resource grammar library currently covers 27 languages: Amharic (partial), Arabic (partial), Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Hindi (fragments), Interlingua, Italian, Latin (fragments), Latvian, Nepali, Norwegian (bokmål), Persian, Polish, Punjabi, Romanian, Russian, Spanish, Swedish, Thai, Turkish (fragments), Urdu.
The current work is aimed at building the Japanese resource grammar, which has not been included in the GF Resource Grammar Library yet, though Japanese is spoken by 125 mln people, hence, it is the 9th most common language in the world [Languages Spoken by More Than 10 Million People]. To compare, the Library already comprises such languages as Amharic and Nepali, that are spoken by not more than 17.5 mln people.

The library can be used as a resource for language processing tasks, such as translation, multilingual generation, software localization, natural language interfaces, and spoken dialogue systems. The library may also have some language-typological interest as an experiment showing how much grammatical structure can be shared between languages.

1.2 Example of a simple GF grammar

Let us consider a simple GF grammar producing only three phrases: “99 bottles of beer on the wall”, “98 bottles of beer on the wall” and “97 bottles of beer on the wall”.

The abstract syntax defines what meanings can be expressed in the grammar. We have three Numbers (97, 98, 99), and each of them can be a part of the whole Phrase. The GF code for the abstract syntax would be:

```plaintext
abstract BeerBottles = {
    flags startcat = Phrase ;
    cat Number ; Phrase ;
    fun
        Bottles : Number -> Phrase ;
        Num99, Num98, Num97 : Number
    }
}
```

The code has the following parts:
- a module header indicating that it is an abstract syntax module named BeerBottles;
- a module body in braces, consisting of:
  - a startcat flag declaration stating that Phrase is the default start category for parsing and generation;
  - category declarations introducing two categories, i.e. types of meanings;
  - function declarations introducing three meaning-building functions (numbers).

Language specific components are placed in the concrete part of the grammar. Here is a fragment from the English concrete syntax:

```plaintext
concrete BeerBottlesEng of BeerBottles = {
    lincat Number, Phrase = {s : Str} ;
    lincat
        Bottles num = {s = num.s ++ " bottles of beer on the wall"} ;
        Num99 = {s = "99"} ;
        Num98 = {s = "98"} ;
        Num97 = {s = "97"
    }
}
```
The major parts of this code are:

- a module header indicating that it is a concrete syntax of the abstract syntax *BeerBottles*, itself named *BeerBottlesEng*;
- a module body in curly brackets, consisting of
  - linearization type definitions stating that *Number* and *Phrase* are records with strings;
  - linearization definitions telling what records are assigned to each of the meanings defined in the abstract syntax.

Each category introduced in the abstract part has a linearization in the concrete part after the reserved word *lincat*, each function – after the reserved word *lin*. Category linearizations describe the type of the category declared in the abstract part for a concrete language. All categories have record types with one or more record fields.

We can produce analogous concrete syntaxes for any language. For example, a Japanese concrete syntax would look like:

```javascript
concrete BeerBottlesJap of BeerBottles = {
  lincat Number, Phrase = {s : Str} ;
  lin
    Bottles num = {s = "壁にはビールの" ++ num.s ++ "ボトル"} ;
    Num99 = {s = "九十九"} ;
    Num98 = {s = "九十八"} ;
    Num97 = {s = "九十七"} ;
} ;

As we can see, the GF grammar does not just replace words with their equivalents, but arranges them according to the grammatical structure peculiar for a concrete language. Thus, in the Japanese phrases the numbers are correctly inserted in the middle of the phrase.

1.3 Thesis structure

Chapter 2 is devoted to the implementation of the Japanese resource grammar. Its first section describes the structure of the grammar in general. The second one gives short information on the Japanese morphology and stylistic differentiation as a foundational principle of the Japanese grammar. Other sections roughly correspond to the parts of the resource grammar itself. The morphological principle was chosen to distribute the grammar rules, e.g. rules relating to noun phrases, verb phrases, adverbs, etc are assembled in separate files. In our opinion, it is reasonable to describe the resource grammar in the same way, section by section, although some rules can be covered in the related sections for the sake of convenience.

Chapter 3 describes the results of testing the Japanese resource grammar and explains the reasons for some difficulties in its application.

The concluding chapter 4 contains a discussion on the related works, results and prospects for future development.
Chapter 2

Implementation of the Japanese resource grammar in GF

2.1 Structure of a GF resource grammar

The main modules of a GF resource grammar and their dependencies are given in Figure 1.

![Diagram of GF resource grammar](image)

*Figure 1. The module structure of a GF resource grammar [Ranta 2011]*
Modules that are already given or derived mechanically:

- all abstract modules except Extra, Irreg;
- concrete of Common, Grammar, Lang, All;
- Constructors, Syntax.

Modules that have to be written by the resource grammarian:

- concrete of the row from Adjective to Structural;
- concrete of Cat and Lexicon;
- Paradigms;
- abstract and concrete of Extra, Irreg.

Moreover, language specific parameter types, morphology, etc are written in the module Res (Resource).

Here is a summary of the roles of the modules:

- Syntax: syntactic combinations and structural words;
- Paradigms: morphological paradigms;
- Constructors: syntactic combinations only;
- Irreg: irregularly inflected words (often mostly verbs);
- Extra: language-specific extra syntactic constructs;
- Lang: common syntax and lexicon;
- All: common grammar plus language-dependent extensions;
- Grammar: abstract syntax functions common to languages;
- Idiom: idiomatic expressions;
- Structural: lexicon of structural words;
- Cat: the type system common to languages;
- Common: concrete syntax common to (most) languages.

The most substantial part is the box containing the ten modules from Adjective to Verb. These modules are called phrase category modules. Each of them defines the constructors for one or a few related parts of speech.

### 2.2 General characteristics of Japanese morphology. Stylistic differentiation

The Japanese language combines features of the agglutinative and inflectional language types. The agglutinative system, typical of Turkic and Mongolian languages, is characterized by joining stems of content words and grammatical elements that are invariable in their structure (formants, clitics). The Japanese name system is almost entirely agglutinative; grammatical relations are expressed by means of postpositive elements (postpositions, particles). Inflectional features are realized in joining word stems and affixes varying in their structure. The systems of Japanese verb and predicative adjective are considerably inflectional, though there are also a lot of formants (conjunctions, particles, etc) there [Alpatov 2008].

According to morphological features one can allocate a large class of inflexible words and three classes of variously inflected words – verbs (including auxiliary verbs), predicative adjectives and copulas.

The Japanese language employs an extensive system of politeness and honorific markers. It is often the case that in order to utter any kind of expression, the speaker must keep in mind his/her social standing to the person
addressed, and the person being talked about. These markers appear on verbs, adjectives, and even nouns. For example, the informal form of the verb to go, 行く “iku”, is used when speaking with someone close to the speaker, but if the person addressed is a stranger or is older than the speaker, the politeness marker ~ます “masu” appears: 行きます “ikimasu”.

Practically all categories in our concrete grammar have the two style forms: the polite style (also known as respect language, honorifics, or 敬語 “keigo”) and plain style (普通形 “futsuukei”).

Only the ending of a sentence needs to be in the polite form to give the whole sentence the tone of the polite style. Any verbs or adjectives used within a complex sentence are in plain forms, regardless of the context and choice of style at the end. In the following sentence there are four verbs, and although the overall style of speech is polite, it is only the final verb which has a polite ~ます “masu” ending:

夏休みに毎年ヨーロッパへ行く人が年々増えているそうですが、オーストラリアへ行く人の方が多いと旅行会社の情報で分かりました。
Natsu-yasumi ni maitoshi Yooroppa e iku hito ga nennen fuete iru sou desu ga Ousutoraria e iku hito no hou ga ooi to ryokougaisha no jouhou de wakarimashita.
We know from information from travel firms that the number of people who go to Europe every year for their summer holidays is increasing year by year, although greater numbers are still going to Australia [Bunt 2003].

The choice of polite or plain style depends on the situation. The plain style is used primarily in informal spoken language, in most books, magazines and newspaper articles. The polite style is found in spoken language, and it also carries over into writing where the writer is “speaking” to the reader (e.g. emails, letters, postcards, lectures, radio and TV news, etc) or quoting someone’s words.

In fact, the necessity to consider both styles in the concrete GF grammar makes the paradigms of all content parts of speech twice larger, but the stylistic differentiation is an essential feature of the Japanese language, which cannot be ignored.

2.3 Noun Phrases

2.3.1 Nouns

Unlike many other languages, Japanese nouns do not have grammatical gender, do not change their forms to express grammatical relationships and take no articles. Moreover, they do not generally have a plural form. Sometimes only the context can help determine which is appropriate:

本はどこですか?
[book] WA [where] [is] KA
Hon wa doko desu ka
Where is the book? / Where are the books? [Bunt 2003]
However, the N category in our concrete syntax has the *Number* parameter, because a few nouns, all of which refer to people, can be shown to be plural by adding the suffix たち “tachi” (学生たち – *students*).

Additionally, in respectful speech, the prefix お “o” is often used with native nouns, as is the prefix ご “go” with Sino-Japanese nouns.

Therefore, some Japanese nouns may vary depending on the number or speech style. So we have three types of nouns in *ResJap.gf*: 1) those that vary according to the number (*numberNoun*); 2) those that vary according to the speech style (*styleNoun*); 3) those that are independent from these categories (*regNoun*). We also need to introduce a general type of noun (*mkNoun*) depending on both number and style (*Noun : Type = {s : Number => Style => Str; ...})*, though in *Lexicon.gf* there are no nouns varying both in number and style simultaneously.

All nouns also have the inherited feature “animateness” (*anim*) in order to choose the correct predicate at the Clause level.

### 2.3.1.1 Counting and counter words

Counting objects, people, animals, etc in Japanese involves the system of suffixes or counters added to the numbers. For example, 本 “hon” for long, thin objects (rivers, roads, train tracks, ties, pencils, bottles, etc), 人 “nin” for people, 台 “dai” for cars, bicycles, machines, mechanical devices, etc. One of the most complete lists of counters includes 508 items [Trussel].

Grammatically, counter words can appear either before or after the noun they count; they generally occur after the noun (following particles), and if used before the noun, they emphasize the quantity:

1. **Drank two bottles of beer (neutral intonation)**

   (1) ビールを二本飲んだ
   [beer] O [two] COUNTER [drank]
   *Biiru o nihon nonda*
   Drank two bottles of beer

2. **Drank two bottles of beer (emphasis on the number, as in response to a question “How many beers did you drink?”)**

   (2) 二本のビールを飲んだ
   [two] COUNTER [of] [beer] O [drank]
   *Nihon no biiru o nonda*
   Drank two bottles of beer

The paradigms for all types of nouns and CNs have the *counter* label. If not specified, the common counter つ “tsu” is used. It is added to all nouns, even if they seem uncountable. Virtually, Japanese only contains mass nouns since it lacks sufficient capability to singularize and pluralize nouns [Mazack 2007].

When counting inanimate objects, cardinals more than 9 are used without つ. That is why all categories connected with numerals (including *Det* that inherits *Num*’s labels) have the Boolean label *tenPlus*. At the same time, *CN* has the Boolean label *counterTsu* (inherited from *N*, *N2* or *N3*), which checks whether the nuclear noun has a special counter or not. If not, we do not place つ after the cardinal that is more than 9.
Moreover, some counters semantically coincide with the nouns (like “tree”, “house”, etc). We do not need to say “[tree] no [two] [tree]”, we say “[two] [tree]”, even if the word tree as a counter differs from the common word tree. So we add the Boolean label counterReplace to check if the counter is used instead of the main noun.

The Noun category finally gets the following record type:

\[
\text{Noun} : \text{Type} = \{s : \text{Number} \Rightarrow \text{Style} \Rightarrow \text{Str} ; \anim : \text{Animateness} ;
\text{counter} : \text{Str} ; \text{counterReplace} : \text{Bool} ; \text{counterTsu} : \text{Bool}\} ;
\]

Our task in counting is slightly simplified, because we do not deal with transcriptions, we take only the written form, i.e. kanji. For example, although the counter 人 in 1人, 2人, 3人 is pronounced differently, the graphical form remains the same. We also ignore different readings of cardinals themselves used alone or with the counter つ: 一 “ichi” – 一つ “hitotsu”, 二 “ni” – 二つ “futatsu”, 三 “san” – 三つ “mittsu”, etc.

### 2.3.1.2 Relational nouns

2-place and 3-place relational nouns include those having objects with prepositions (or particles in Japanese), e.g. “distance from ... to ...”. N2 and N3 made us add the object label to CN. For example, mother of the king in Japanese has the structure [king] no [mother] (“no” is a particle for possessives). But if we want to add an AP to this CN, we should insert it between the main noun and the object (beautiful mother of the king – [king] no [beautiful] [mother]). The main noun (with all adjectives) and the object are glued together at the NP level (through the DetCN rule).

For the same reason, N2 also gets the object label. It can be formed through the combination of N3 and NP, so the “nuclear” noun should be kept separately:

\[
\text{N2} = \text{Noun} ** \{\text{prep} : \text{Str} ; \text{object} : \text{Style} \Rightarrow \text{Str}\} ;
\]
\[
\text{N3} = \text{Noun} ** \{\text{prep1} : \text{Str} ; \text{prep2} : \text{Str}\} ;
\]

### 2.3.1.3 Proper nouns

There are two proper nouns in the GF lexicon – Paris and John, – but this is enough for us to make the stylistic distinction in the paradigm of proper nouns. As opposed to other names, names of people obtain the postfix さん “san” in the respectful style (ジョンさん “John-san”). PNs also have the label anim (animateness) for the further agreement with predicates (see 2.5.1).

### 2.3.2 Quantifiers

Quantifiers in the GF grammar include demonstrative pronouns this and that, possessive pronouns, definite and indefinite articles and the determiner no (note the difference between the morphological part of speech “determiner” and the GF category of the same name).

There are three most common demonstrative pronouns in Japanese:
- この “kono” means this thing very near the speaker;
- その “sono” means that thing somewhat near the speaker or something just mentioned;
・あの“ano” means that thing way over there or something everybody knows about.

Both “sono” and “ano” correspond to English that, we chose “sono” as a bit more common.

Besides the traditional s, Quant also has the sp label used in the DetNP function (forming noun phrases directly from determiners). “Kono” functioning as a subject turns into “kore” and “sono” turns into “sore”. The sp label is also essential for indefinite and definite articles. They are blank in pre-nominal position (i.e. there are no articles in Japanese), but in DetNP the indefinite article is replaced by何か“nanika”(something) and the definite article turns intoこれ“kore”(this).

Possessive pronouns are formed by placing the particleの“no” after the respective personal pronoun. Since our pronouns vary according to style (see 2.3.5), the s label of Quant obtains the Style parameter.

The determiner no behaves differently than all the other quantifiers. In Japanese it has no lexical realization, but the polarity of the predicate is turned into negative:

シャワーからお湯が出ません。
[shower][from][hot water][not come out]
Shawaa kara oyu ga demasen
There is no hot water in the shower.

To check if we deal with the word “no” we had to add the Boolean label no to Quant and Det, which inherits it from Quant. However, in nominal sentences (those without predicates) no will not be translated correctly, e.g. in prohibition sentences (“No smoking!”). This type of sentences has not been covered in the concrete grammar yet.

The record type for the Quant category is:

Quant = {s : Style => Str ; sp : Style => Str ; no : Bool} ;

2.3.3 Number determining elements

The Num category covers the markers of singular and plural numbers (which have no lexical realization) and cardinals. It inherits all labels of Card and has one Boolean label more –inclCard – to check if it is just a grammatical number marker or a real cardinal number with an optional postpositive (see 2.3.1.1, 2.7.1).

2.3.4 Determiners

As it was mentioned before, the GF category Det does not always coincide with the morphological category determinant. Det is formed by means of the rules

DetQuant : Quant -> Num -> Det ; -- these five
DetQuantOrd : Quant -> Num -> Ord -> Det ; -- these five best
*Det* inherits all the labels of *Quant* and *Num* and keeps their lexical realizations separately (*quant*: *Style* => *Str*; *num*: *Str*), because of the special word order: in the function *DetCN*: *Det* => *CN* => *NP*, *CN* is placed between *quant* and *num*, if *Det* contains a cardinal number (see 2.3.1.1).

Other *Dets* include the words *every*, *few*, *many*, *much*, *some* (in Singular and Plural senses).

Determiners are assumed to form noun phrases directly, without *CN* (e.g. *these five*). For this case, *Det* has the *sp* label, which excludes the possibility of an empty determiner (for example, if we combine the indefinite article and the marker of the grammatical singular number) (see 2.3.2).

### 2.3.5 Personal pronouns

Personal pronouns in Japanese refer to people, not things or concepts. Japanese has no equivalent to the English *it* in this category [Akiyama 2002]. We took the determiner *それは* “sore” (*this*) as a relatively close equivalent of *it*. Accordingly, the pronoun *they* also causes a problem if it replaces some inanimate objects. In this case the same word, “sore”, should be used. Unfortunately, we have no opportunity to check whether animate or inanimate objects are implied.

There are numerous Japanese pronoun forms varying by region, dialect, gender and, most important, level of speech. The incomplete list of pronoun forms includes 23 variants of *I*, 17 variants of *you*, 10 variants of *he*, etc [sci.lang.japan FAQ: Japanese pronouns]. We considerably cut this diversity and left only 1–2 most common variants for each pronoun used in honorific or plain styles.

Pronouns have a special Boolean label *Pron1Sg* used to ascertain whether we deal with the word *I* or not. It is essential in the choice of the correct equivalent for the verb *give* and the correct form to translate the VP complement *want* (see 2.5.2.1). For the same reasons *NP* has the parameter *meaning*: if it is a pronoun *I*, the NP’s meaning is *Me*, all other NPs get the value *SomeoneElse*.

When the meaning can be understood from the context, the Japanese prefer not to use personal pronouns [Akiyama 2002]. However, the context analysis is out of our task scope, so we did not make NPs formed by personal pronouns optional in order not to make some sentences even more ambiguous.

### 2.3.6 Means of NP modification

- **Adjectives**

  In Japanese adjectives are placed before the noun they are attributed to. Multiple adjectives are combined together by means of the *te*-form (see 2.4.1). To check whether a *CN* already contains an *AP* or not, we added the Boolean *hasAttr* label to *CN*. If *True*, all adjectives are taken in the *te*-form except the last one taken in its dictionary form.
AdjCN ap cn = {
    s = 
        st => case cn.hasAttr of {
            False => ap.attr ! st ++ cn.s ! n ! st ;
            True => ap.te ! st ++ cn.s ! n ! st
        } ; ... } ;

- Predeterminers
    The words all, only, most and not are defined as predeterminers in the GF grammar.
    The word not is of particular grammatical interest. Like the quantifier no, it has no lexical representation, but changes the predicate’s polarity into negative. To check if we deal with the word not, we introduced the Boolean label not for Predet.
    Predet = (s : Str ; not : Bool) ;

- Participles
    According to the comment in the abstract syntax, we consider only past participles in this section.
    Unlike English and other European languages, participial constructions in Japanese are placed before the noun phrase they are attributed to. The same is about most adverbs and relative clauses – they are not postmodifiers, as stated in the abstract syntax, but premodifiers.
    To form a past participle in Japanese we take a verb in the plain past passive form and place it before an NP:
    ヤフー・アンサーで聞かれた質問に、お答えします。
    Yahoo-ansaazu de kikareta shitsumon ni okotaewhimasu.
    I would like to answer the question asked in Yahoo! answers.

PPartNP np v2 = {
    s = 
        st => v2.pass ! Plain ! TPast ! Pos ++ np.s ! st ; ... } ;

- Adverbs
    Because of a rather strict word order in Japanese, we should check what type of adverb we are attaching to an NP. If the adverb’s prepositive field is True (see 2.6.1), that is, if the adverb is a subordinate clause, we keep it in the NP’s field prepositive. At the Clause level we will put it at the beginning of the clause. The NP and VP of the main clause are rarely separated by adverbial subordinate clauses, though there are exceptions depending on the meaning of certain words.
    Simple adverbs, such as today, now, are placed before an NP.

AdvNP np adv = {
    s = 
        st => case adv.prepositive of {
            True => np.s ! st ;
            False => adv.s ! st ++ np.s ! st
        } ;
    prepositive = 
        st => case adv.prepositive of {
            True => adv.s ! st ;
            False => []
        } ; ... } ;
Relative clauses, embedded sentences and questions

Relative clauses, embedded sentences and questions are placed immediately before an NP. We assume that the particle が “ga” (a new subject marker) is more probable in them, though this is not always true (the problem of particles は/が – see 2.9.1).

RelNP np rs = {s = \st => rs.s ! np.anim ! st ++ np.s ! st ; ... } ;

2.3.7 Conjunction of noun phrases

The way of joining two and more noun phrases in Japanese is different from that in English, as coordinating or alternative conjunctions (or, to be more exact, particles) are placed among all homogeneous sentence parts, not only before the last one. That is, N1 [and] N2 [and] N3 [and] N4, etc. Moreover, the particle も “mo”, which means both, but can also join more than two nouns, is placed after the last noun, and neither particle は “wa” nor が “ga” is needed after it.

Since we attach the conjoining particles at each new step of creating a noun phrase chain, we have to keep special fields for each particle – and, or and both:

BaseNP x y = {
    and = \st => x.s ! st ++ "と" ++ y.s ! st ;
    or = \st => x.s ! st ++ "か" ++ y.s ! st ;
    both = \st => x.s ! st ++ "も" ++ y.s ! st ++ "も" ;
    ... } ;

At the final step, when creating a normal NP, we check what type of conjunction we want to render by means of the type field of the Conj category. We do not add any words at this step; conj.null is an empty string created just not to lose the conjunction in a parse tree:

ConjNP conj np = {
    s = \st => case conj.type of {
        And => conj.null ++ np.and ! st ;
        Or => conj.null ++ np.or ! st ;
        Both => conj.null ++ np.both ! st
    } ; ... } ;

2.4 Adjective Phrases

2.4.1 Types of adjectives

Japanese adjectives belong to one of two groups: い “i”-adjectives and な “na”-adjectives. They have a complex system of conjugation, which is involved when adjectives are used in the predicative function.

• い adjectives

The dictionary form of adjectives of this group ends with い “i”. The rest of the word is a stem, which remains unchanged in all forms. When used in front of a noun, い adjectives do not change their form:
This is a new car.

Not all conjugation forms of adjectives (as well as verbs) are used in our concrete syntax. These are the forms of い adjectives that we used:

<table>
<thead>
<tr>
<th>Form</th>
<th>い adjective</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain nonpast</td>
<td>おいしい oishii</td>
<td>delicious</td>
</tr>
<tr>
<td>Polite nonpast</td>
<td>おいしいです oishii desu</td>
<td>delicious</td>
</tr>
<tr>
<td>Plain past</td>
<td>おいしかった oishikatta</td>
<td>was delicious</td>
</tr>
<tr>
<td>Polite past</td>
<td>おいしかったです oishikatta desu</td>
<td>was delicious</td>
</tr>
<tr>
<td>Negative plain nonpast</td>
<td>おいしくない oishiku nai</td>
<td>not delicious</td>
</tr>
<tr>
<td>Negative polite nonpast</td>
<td>おいしくありません oishiku arimasen</td>
<td>not delicious</td>
</tr>
<tr>
<td>Negative plain past</td>
<td>おいしくなかった oishiku nakatta</td>
<td>was not delicious</td>
</tr>
<tr>
<td>Negative polite past</td>
<td>おいしくありませんでした oishiku arimasen deshita</td>
<td>was not delicious</td>
</tr>
<tr>
<td>Te-form</td>
<td>おいしくて oishikute</td>
<td>delicious and...</td>
</tr>
<tr>
<td>Negative te-form</td>
<td>おいしくなくて oishikunakute</td>
<td>not delicious and...</td>
</tr>
<tr>
<td>Conditional (ba-form)</td>
<td>おいしければ oishikereba</td>
<td>if it is delicious</td>
</tr>
<tr>
<td>Negative ba-form</td>
<td>おいしくなければ oishikunakereba</td>
<td>if it is not delicious</td>
</tr>
<tr>
<td>Adverbial</td>
<td>おいしく oishiku</td>
<td>deliciously</td>
</tr>
</tbody>
</table>

Table 1. Conjugation forms of い adjectives used in the GF Japanese resource grammar

To avoid variants, we omitted two other possible forms: おいしくないです “oishiku nai desu” for the negative polite nonpast form and おいしくなかったです “oishiku nakatta desu” for the negative polite past form.

Te- and ba-forms are of special interest, since they express grammatical relations that are normally conveyed by separate functional words in European languages. Thus, the te-form usually replaces a word in a row of homogenous parts of speech connected by the conjunction and. The ba-form expresses the subjunctive mood and replaces the word if. Te- and ba-forms can also be negative.

We had to create the field for the adverbial form of adjective, which is then inherited by AP, because this form of AP is used in some VP constructions (e.g. become red, paint (it) red).

- な adjectives
  - な adjectives need the addition of a final な when used in front of the nouns they describe:

奇麗な花
Kireina hana
Pretty flowers

As for い adjectives, we used the following conjugation forms of な adjectives:

奇麗な花
Kireina hana
Pretty flowers
<table>
<thead>
<tr>
<th>Form</th>
<th>い adjective</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain nonpast</td>
<td>静かだ shizuka da</td>
<td>quiet</td>
</tr>
<tr>
<td>Polite nonpast</td>
<td>静かです shizuka desu</td>
<td>quiet</td>
</tr>
<tr>
<td>Plain past</td>
<td>静かだった shizuka datta</td>
<td>was quiet</td>
</tr>
<tr>
<td>Polite past</td>
<td>静かでした shizuka deshita</td>
<td>was quiet</td>
</tr>
<tr>
<td>Negative plain nonpast</td>
<td>静かではない shizuka de wa nai</td>
<td>not quiet</td>
</tr>
<tr>
<td>Negative polite nonpast</td>
<td>静かではありません shizuka de wa arimasen</td>
<td>not quiet</td>
</tr>
<tr>
<td>Negative plain past</td>
<td>静かでなかった shizuka de wa nakatta</td>
<td>was not quiet</td>
</tr>
<tr>
<td>Negative polite past</td>
<td>静かではありませんでした shizuka de wa arimasen deshita</td>
<td>was not quiet</td>
</tr>
<tr>
<td>Te-form</td>
<td>静かで shizuka de</td>
<td>quiet and...</td>
</tr>
<tr>
<td>Negative te-form</td>
<td>静かでなくて shizuka de wa nakute</td>
<td>not quiet and...</td>
</tr>
<tr>
<td>Conditional (ba-form)</td>
<td>静かであれば shizuka de areba</td>
<td>if it is quiet</td>
</tr>
<tr>
<td>Negative ba-form</td>
<td>静かでなければ shizuka de nakereba</td>
<td>if it is not quiet</td>
</tr>
<tr>
<td>Adverbial</td>
<td>静かに shizuka ni</td>
<td>quietly</td>
</tr>
</tbody>
</table>

Table 2. Conjugation forms of な adjectives used in the GF Japanese resource grammar

- Other parts of speech in the adjective function
  Some adjectives in the abstract lexicon have no equivalents among Japanese adjectives. Most commonly they are translated by means of nouns with particleの “no” in the attributive function and verbs in the nonpast continuous form acting as predicates. E.g. married: 既婚の“kikon no” and 結婚している“kekkonshite iru”. These “adjectives” conjugate according to the paradigm of the verb いる.

2.4.2 Comparative adjectives

Japanese adjectives do not have special forms for comparative, unlike English adjectives (hotter, deeper); instead they add extra words.

To say that something is, for example, “bigger” or “more expensive” than something else, the item to which it is compared is marked with the particleより “yori” and the adjective itself does not change:

东京はロンドンより大きいです。
[Tokyo] WA [London] [than] [big] [is] 

**Tokyoo wa Rondon yori ookii desu.**
Tokyo is bigger than London.

To say that something is “less” than something, we use the negative form of adjective:
寿司はピザよりおいしくありません。
Sushi wa piza yori oishiku arimasen.
Sushi is less tasty than pizza.

To check if we deal with the adverb more or less, we added the Boolean field less to the CAdv category.

CAdvAP cadv ap np = {
  pred = \st,t => case cadv.less of {
    True => table {
      Pos => np.s ! st ++ cadv.s ++ ap.pred ! st ! t ! Neg ;
      Neg => np.s ! st ++ cadv.s ++ ap.pred ! st ! t ! Pos
    } ;
    False => \p => np.s ! st ++ cadv.s ++ ap.pred ! st ! t ! p
  } ;
  attr = \st => case cadv.less of {
    True => np.s ! st ++ cadv.s ++ ap.attr ! Plain ! TPres ! Neg ;
    False => np.s ! st ++ cadv.s ++ ap.pred ! ap.attr ! st
  } ; ...
} ;

2.4.3 Means of AP modification

The way of AP modification by means of adverbs is similar to that of NP modification (see 2.3.6).

Words of the special category in the GF syntax – adjective-modifying adverbs (AdA) – include such words as almost, so, very, etc. They are normally placed before the AP.

AdAP ada ap = {pred = \st,t,p => ada.s ++ ap.pred ! st ! t ! p ; ... } ;

AP can be also modified by embedded sentences or questions, although the semantics is only clear for some adjectives (e.g. good that she is here). Japanese translations of sentences of this type have a structure that differs from the English one: an embedded sentence acts as a subject (its predicate is substantivized, e.g. by means of the word こと “koto”) and an adjective serves as a predicate of the whole sentence:

It is good that he is here.
彼がここにいることが良いです。
Kare ga koko ni iru koto ga ii desu.

SentAP ap sc = {pred = \st,t,p => sc.s ! Wa ! st ++ ことであった ++ ap.pred ! st ! t ! p ; ... } ;

2.4.4 Conjunction of adjective phrases

If we want to join adjectives with and in Japanese, we have to turn all of them except the last one into the te-form: [te-adj][te-adj][te-adj][adj]. If it is
conjunction or, we insert the particle か “ka” among all adjectives. The pecu-
liarity here is that if it is a “na”-adjective, we take its stem without “na”. For this reason we had to introduce the field dropNaEnging for all adjec-
tives and APs.

BaseAP x y = {  
s1and = x.te ;  
s1or = \st => x.dropNaEnging ! st ++ “か” ;  
s2pred = y.pred ;  
s2attr = y.attr ;  
s2te = y.te ; ... } ;

ConsAP x xs = {  
sland = \st => xs.sland ! st ++ xs.s2te ! st ;  
slor = \st => xs.slor ! st ++ xs.s2dropNaEnging ! st ++ “か” ;  
s2pred = x.pred ;  
s2attr = x.attr ;  
s2te = x.te ; ... } ;

ConjAP conj ap = case conj.type of {  
(And | Both) =>  
pred = \st,t,p => conj.null ++ ap.s1and ! st ++ ap.s2pred ! st ! t ! p ;  
attr = \st => conj.null ++ ap.s1and ! st ++ ap.s2attr ! st ; ... } ;  
Or => {  
pred = \st,t,p => conj.null ++ ap.slor ! st ++ ap.s2pred ! st ! t ! p ;  
attr = \st => conj.null ++ ap.slor ! st ++ ap.s2attr ! st ; ... } ;  
... } ;

2.5 Verb Phrases

2.5.1 Verbs

2.5.1.1 Verb groups

Japanese verbs do not have forms to indicate person, number, or gender. The same verb from is used no matter what the subject; this is true for all tenses:

<table>
<thead>
<tr>
<th>Person</th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>First person</td>
<td>行きます “ikimasu” / I go</td>
<td>行きます / we go</td>
</tr>
<tr>
<td>Second person</td>
<td>行きます / you go</td>
<td>行きます / you go</td>
</tr>
<tr>
<td>Third person</td>
<td>行きます / he/she/it goes</td>
<td>行きます / they go</td>
</tr>
</tbody>
</table>

Table 3. Independence of Japanese verbs from person, number and gender

In Japanese linguistics the three groups of verbs are traditionally distinguished: 1) -u ending verbs; 2) -iru/-eru ending verbs; 3) irregular verbs group [Akiyama 2002].

The stems of the -u ending verbs are formed by dropping the final -u. In fact, any verb whose dictionary form ends in anything but -eru or -iru belongs to Group 1. However, there is a number of exceptions: the verbs 入る “hairu” (to enter), 走る “hashiru” (to run), いる “iru” (to need), 帰る “kaeru” (to return),
To form the stem of an -iru/-eru ending verb, one should drop the final -ru.

There are only two irregular verbs in Japanese: 来る “kuru” (to come) and する “suru” (to do). The verb する is often combined with many nouns of Chinese or Western origin to make then into verbs: 勉強する “benkyousuru” (to study), 旅行する “ryokousuru” (to travel), ダンスする “dansusuru” (to dance), etc.

2.5.1.2 Verb conjugation

The following tables illustrate the main verb forms used in the GF concrete syntax.

Nonpast Tense

The plain form of the nonpast tense of Japanese verbs is the same as the dictionary form. The -ます “masu” form is used in formal situations. A so-called i-stem of a verb is used to get the -masu form.

| Group 1 | Take off the final -u, and add -imasu: kak – kakimasu, nomu – nomimimasu |
| Group 2 | Take off the final -ru, and add -masu: miru – mimasu, taberu – tabemasu |
| Group 3 | kuru – kimasu, suru – shimasu |

Table 4. The -ます form of Japanese verbs

The i-stem of verbs is obtained through the following rule:

```
mk_i_stem : Str -> VerbGroup -> Str = \neru,gr ->
  case gr of {
    Gr1 => case last neru of {
      “る” => init neru + “り” ;
      “す” => init neru + “し” ;
      “く” => init neru + “き” ;
      “ぐ” => init neru + “ぎ” ;
      “む” => init neru + “み” ;
      “ぬ” => init neru + “に” ;
      “ぶ” => init neru + “び” ;
      “つ” => init neru + “ち” ;
      _ => init neru + “い”
    } ;
    (Gr2 | Kuru) => init neru ;
    Suru => Predef.tk 2 neru + “し”
  } ;
```

Similar formulas are used to produce the verb’s a-stem, plain past form and ba-form described below.
### Past Tense

<table>
<thead>
<tr>
<th>Group 1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Polite</td>
<td>Replace -u with -imashita</td>
<td>kaku – kakimashita</td>
</tr>
<tr>
<td></td>
<td>nomu – nomimashita</td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>(1) Verb ending with -ku: replace -ku with -ita</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>kaku – kaita</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kiku – kiita</td>
</tr>
<tr>
<td></td>
<td>(2) Verb ending with -gu: replace -gu with -ida</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>isogu – isoida</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oyogu – oyoida</td>
</tr>
<tr>
<td></td>
<td>(3) Verb ending with -u, -tsu and -ru: replace them with -tta</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>utau – utatta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matsu – matta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kaeru – kaetta</td>
</tr>
<tr>
<td></td>
<td>(4) Verb ending with -nu, -bu and -mu: replace them with -nda</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>shinu – shinda</td>
</tr>
<tr>
<td></td>
<td></td>
<td>asobu – asonda</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nomu – nonda</td>
</tr>
<tr>
<td></td>
<td>(5) Verb ending with -su: replace -su with -shita</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hanasu – hanashita</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dasu – dashita</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Polite</td>
<td>Take off -ru, and add -mashita</td>
<td>miru – mimashita</td>
</tr>
<tr>
<td></td>
<td>taberu – tabemashita</td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>Take off -ru, and add -ta</td>
<td></td>
</tr>
<tr>
<td></td>
<td>miru – mita</td>
<td></td>
</tr>
<tr>
<td></td>
<td>taberu – tabeta</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Polite</td>
<td>kuru – kimashita, suru – shimashita</td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>kuru – kita, suru – shita</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Past tense form of Japanese verbs

### Present Negative

To make a sentence negative, verb endings are changed into negative forms: -masu is replaced with -masen for polite style verbs, and plain verbs are turned into the -nai form, in which their a-stem is used.

<table>
<thead>
<tr>
<th>Polite</th>
<th>All Verbs (Groups 1, 2, 3)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Replace -masu with -masen</td>
<td>nomimasu – nomimasen</td>
</tr>
<tr>
<td></td>
<td>tabemasu – tabemasan</td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>Group 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Replace the final -u with -anai (If verb ending is a vowel + -u, replace with -wanai)</td>
<td>kiku – kikanai</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nomu – nomanai</td>
</tr>
<tr>
<td></td>
<td></td>
<td>au – awanai</td>
</tr>
<tr>
<td></td>
<td>Group 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Replace -ru with -nai</td>
<td>miru – minai</td>
</tr>
<tr>
<td></td>
<td>taberu – tabenai</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>kuru – konai, suru – shinai</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Present negative form of Japanese verbs
Past Negative

<table>
<thead>
<tr>
<th>Polite</th>
<th>All Verbs (Groups 1, 2, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add -deshita to the formal present negative form</td>
<td>nomimasen – nomimasen deshita tabemasen – tabemasen deshita kimasen – kimasen deshita shimasen – shimasen deshita</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plain</th>
<th>All Verbs (Groups 1, 2, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace -nai with -nakatta</td>
<td>nomanai – nomanakatta tabenai – tabenakatta konai – konakatta shinai – shinakatta</td>
</tr>
</tbody>
</table>

Table 7. Past negative form of Japanese verbs

The conjugation tables presented above show the complexity of choosing the correct ending for a verb. In our concrete grammar we followed the traditional division of verbs into the three groups. Special functions were written to derive the verb’s a-stem, i-stem and plain past form according to its dictionary form and group. Therefore, each record for verbs in the concrete Lexicon consists of a string denoting the dictionary form (or more strings representing prepositions, if it is a 2- or 3-place verb) and the verb group. Knowing a verb’s group is also necessary to produce its passive form (see 2.5.1.3).

Alongside with the above mentioned forms, the records for the verbs in our grammar also contain the te- and ba-forms. The te-form is widely used to create tenses and moods, such as the continuous tenses, the imperative mood or for conjunction of clauses within one sentence. The ba-form is also known as “the conditional form”, which expresses the meaning of the English word if.

In our syntax, the te-form is derived from the verb’s plain past form: if it ends with -ta it is replaced with -te, or -da turns into -de. This is not the usual way to get the te-form; in grammar books it is more common to derive it from the dictionary form with consideration of the group that the verb belongs to. However, since we already have a function that produces the plain past form, it is easier to use it instead, because the mechanisms of forming the plain past form and the te-form are the same:

<table>
<thead>
<tr>
<th>Plain past form</th>
<th>Meaning</th>
<th>Change in final kana</th>
<th>Te-form</th>
</tr>
</thead>
<tbody>
<tr>
<td>会った “atta”</td>
<td>met</td>
<td>た → て</td>
<td>会って “atte”</td>
</tr>
<tr>
<td>読んだ “yonda”</td>
<td>read</td>
<td>だ → で</td>
<td>読んで “yonde”</td>
</tr>
<tr>
<td>書いた “kaita”</td>
<td>wrote</td>
<td>た → て</td>
<td>書いて “kaito”</td>
</tr>
<tr>
<td>した “shita”</td>
<td>did</td>
<td>た → て</td>
<td>して “shite”</td>
</tr>
<tr>
<td>来た “kita”</td>
<td>came</td>
<td>た → て</td>
<td>来て “kite”</td>
</tr>
</tbody>
</table>

Table 8. The te-form of Japanese verbs

To get the negative te-form, one should add -naide to the verb’s a-stem.

The ba-form is obtained through a function which gets the verb’s so called e-stem and adds -ba. To form the e-stem, we change the final kana of the dictionary form to the え “e” line of the kana chart:
Table 9. The ba-form of Japanese verbs

<table>
<thead>
<tr>
<th>Dictionary form</th>
<th>Meaning</th>
<th>Change in final kana</th>
<th>+ - ば</th>
</tr>
</thead>
<tbody>
<tr>
<td>起こる “okoru”</td>
<td>to occur</td>
<td>る → れ</td>
<td>起これば “okoreba”</td>
</tr>
<tr>
<td>読む “yomu”</td>
<td>to read</td>
<td>む → め</td>
<td>読めば “yomeba”</td>
</tr>
<tr>
<td>待つ “matsu”</td>
<td>to wait</td>
<td>つ → て</td>
<td>待てば “mateba”</td>
</tr>
<tr>
<td>会う “au”</td>
<td>to meet</td>
<td>う → え</td>
<td>会えば “aeba”</td>
</tr>
<tr>
<td>食べる “taberu”</td>
<td>to eat</td>
<td>る → れ</td>
<td>食べれば “tabereba”</td>
</tr>
<tr>
<td>する “suru”</td>
<td>to do</td>
<td>る → れ</td>
<td>食べれば “tabereba”</td>
</tr>
<tr>
<td>来る “kuru”</td>
<td>to come</td>
<td>る → れ</td>
<td>来れば “aeba”</td>
</tr>
</tbody>
</table>

The negative ba-form is obtained through the addition of -nakereba to the verb’s a-stem.

2.5.1.3 2- and 3-place verbs. Passive voice

2-place verbs are assumed to be used with an NP as a complement attached by means of some preposition. Japanese transitive verbs are usually connected with their complements by means of the particle を “o”. We should note that in the sense of morphology を is a particle, but for convenience we keep it under the field name prep (“preposition”) in a V2 record. It was made on the analogy of the way of determining relations among words by means of prepositions, which is common for the European languages. Moreover, 2-place adjectives and relational nouns also have “prepositions” in our grammar, e.g. の “no” or と “to”, though grammatically they are particles.

3-place verbs involve two complements, therefore they have two “prepositions”.

It is logical that 2- and 3-place verbs can be used in passive. The Japanese passive voice is formed in the following way: the ending -areru is attached to stems of Group 1 verbs, and -rareru is added to stems of Group 2 verbs. Relating to our paradigm, we take the verb’s a-stem, add -reru for Group 1 verbs or -rareru for Group 2 verbs. As for exceptions, する “suru” in Group 3 verbs turns into される “sareru”; 来る “kuru” becomes 来られる “korareru” (however, it is not a 2-place verb and its passive form is not used in our syntax). After the plain nonpast passive form is obtained, it can be conjugated as a normal verb belonging to Group 2.

It is interesting that there is a special sense of using the passive in Japanese, for which English has no corresponding use – so called adversity passive. This means that something happened to someone and it had an adverse effect on that person. The verb may be transitive or intransitive. For example:

田中さんは雨に降られました。
[Mr. Tanaka] WA [rain] [by] [be fallen]
Tanaka-san wa ame ni furaremashita.
Mr. Tanaka was caught in the rain. [Akiyama 2002]

The possibility of intransitive (1-place) verbs to become passive is not predetermined in the GF abstract syntax, but this can be provided in the Extra section in future.
2.5.2 Means of complementation of verbs and verb phrases

2.5.2.1 Verb-phrase-complement verbs

This category ($VV$) includes the verbs *want*, *can* and *must*. In Japanese, there are special ways of using each of these verbs, so in order to distinguish them, we introduced the parameter *ModSense* with options *Abil*, *Oblig* and *Wish*.

Japanese verbs have a special potential form with the meaning of *can, be able to do something*. For Group 1 verbs, their e-stem is taken and -*ru* is added (行く “iku” → 行ける “ikeru”, 飲む “nomu” → 飲める “nomeru”). The ending -*ru* of Group 2 verbs is changed into -*rareru* (食べる “taberu” → 食べられる “taberareru”, 見る “miru” → 見られる “mirareru”). する “suru” turns into できる “dekiru” and 来る “kuru” – into 来られる “korareru”.

For our grammar we chose the second way to get the verb’s potential form: the verb is taken in its plain nonpast form, substantivized with こと “koto” and then followed by the particle が “ga” and verb できる “dekiru” (*to be able*):

Japanese O [read] NOMINALIZER GA [can]

*Nihongo o yomu koto ga dekimasu.*

I can read Japanese.

Among several ways to translate the verb *must* into Japanese, we took the construction なければならない “nakerebanaranai”. The conditional -なければ is added to the a-stem of the verb (this is why all verb categories have the label *a_stem* in their records) and followed by the negative plain or polite form ofなる “naru” (*to become*):

年後日出張で東京まで行かなければなりません。

[day after tomorrow] [business trip] [on] [Tokyo] [to] [if not go then no good]

*Asatte shutcho de Toukyou made ikanakereba narimasen.*

I have to go to Tokyo on a business trip the day after tomorrow.

[Bunt 2003]

The verb *want* also has a number of equivalents in Japanese. We had to include at least two of them in our grammar, as there is no universal variant suitable for all subjects. Thus, to express one’s own wish to do something, one should add -たい “-tai” to the verb’s i-stem. -たい does not need です “desu” in the plain style, although it is followed by です in the polite style:

来年日本に行きたいです。

[next year] [Japan] [to] [want to go]

*Rainen Nihon ni ikitai desu.*

I want to go to Japan next year. [Bunt 2003]

The たい form conjugates in the same way as い “i”-adjectives:
Table 10. The たい form of Japanese verbs

<table>
<thead>
<tr>
<th>Form</th>
<th>Meaning</th>
<th>Ending</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain nonpast</td>
<td>want to</td>
<td>～たい</td>
<td>行きたい</td>
</tr>
<tr>
<td>Plain nonpast negative</td>
<td>do not want to</td>
<td>～たくない</td>
<td>行きたくない</td>
</tr>
<tr>
<td>Plain past</td>
<td>wanted to</td>
<td>～たかった</td>
<td>行きたかった</td>
</tr>
<tr>
<td>Plain past negative</td>
<td>did not want to</td>
<td>～たくなかった</td>
<td>行きたくなかった</td>
</tr>
</tbody>
</table>

However, if we are talking about someone else wanting to do something, then we use the ending 〜たがる “tagaru” (or, more precisely, its continuous form 〜たがっている “tagatte iru”) instead of たい (tai):

先生はコーヒーを飲みたがっています。
The teacher wants to drink coffee.

To correlate with the subject (i.e. to check if it is the pronoun I or not), VV and VP got the parameter Speaker with values Me and SomeoneElse, just like NP. At the Clause level VP agrees with NP’s meaning value (see 2.3.5). The verbs can and must also formally conjugate according to the Speaker parameter, though they have the same forms for Me and SomeoneElse:

\[
\begin{align*}
\text{mkWant} : VV &= \{ \\
  s &= \text{table} \{ \\
  \text{Me} &= \{ \st, t, p \Rightarrow (i\_mkAdj "たい").\text{pred} ! \st ! t ! p ; \\
  \text{SomeoneElse} &= \{ \st, t, p \Rightarrow "たがって" ++ \\
  &\quad (mkVerb "いる" Gr2).\text{s} ! \st ! t ! p \\
  \} ; \ldots \} ; \\
\text{mkCan} : VV &= \{ s = \{ \sp, \text{st}, t, p \Rightarrow (mkVerb "できる" Gr2).\text{s} ! \st ! t ! p ; \\
  \ldots \} ; \\
\text{mkMust} : VV &= \{ s = \{ \sp, \text{st}, t, p \Rightarrow (mkVerb "なる" Gr1).\text{s} ! \st ! t ! p ; \\
  \ldots \} ; \\
\end{align*}
\]

The general rule of complementing VP with VV gets rather complex (we omit supplementary VP fields, such as te, ba, i\_stem, a\_stem, etc):

\[
\text{ComplVV v vp = case v.sense of} \{ \\
  \text{Abil} \Rightarrow \{ \\
  \text{verb} &= \{ \sp, \text{a, st, t, p} \Rightarrow \text{vp.verb} ! \sp ! \text{a} ! \text{Plain} ! \text{TPres} ! \text{ResJap.Pos} \\
  \quad + "ことが" ++ v.s ! \sp ! \text{a} ! \text{Pres} ! \text{Neg} ; \ldots \} ; \\
  \text{Oblig} \Rightarrow \{ \\
  \text{verb} &= \{ \sp, \text{a, st, t, p} \Rightarrow \text{vp.a\_stem} ! \sp ! \text{a} ! \text{st} ++ "なければ" ++ \\
  \quad v.s ! \sp ! \text{st} ! \text{Neg} ; \ldots \} ; \\
  \text{Wish} \Rightarrow \{ \\
  \text{verb} &= \{ \sp, \text{a, st, t, p} \Rightarrow \text{vp.i\_stem} ! \sp ! \text{a} ! \text{st} ++ \\
  \quad v.s ! \sp ! \text{st} ! \text{t} ! \text{p} ; \ldots \} ; \ldots \} ; \\
\}
\]

2.5.2.2 Sentences and questions

To realize the complementation of verbs by means of declarative or interrogative clauses the GF grammar involves special categories of verbs:
sentence-complement verb (VS) and question-complement verb (VQ). The category VS includes such verbs as *fear, hope, know* and *say*; VQ includes *know* and *wonder*.

The mechanism of complementation of the verb *say* by means of sentences is the same as that of forming reported speech structures in Japanese:

```
[subord. sent.] to VS
plain predicate quotation particle [Kaiser 2004]
```

There is even an ongoing debate [Maier 2009] about the direct-indirect distinction in Japanese, where a sentence can be translated both through direct and indirect speech:

太郎は僕が東京へ行くと言った。
[Taro] WA [I] GA [Tokyo] [to] [go] QUOT. PART. [said]
```
Tarou wa boku ga Toukyou e iku to itta.
```
(1) Taro said that I would go to Tokyo.
(2) Taro said: “I will go to Tokyo”.

Nevertheless, we initially deal with indirect speech. Therefore, only the predicate in the subordinate sentence is taken in the plain style, and the subject should stylistically correlate with the main clause – this is the difference from quoting the direct speech, where the subject can be of any style. This is the reason for keeping separate fields for the subject and predicate in the S record type.

The verbs *fear, hope* and *know* are complemented with sentences in the same way as *say*, except that the particle と “to” is replaced with the substantivizing word こと “koto” and particle を “o”. Both と and こと + お are put into the prep (preposition) field of the obtained VPs, though it is not morphologically correct.

```
ComplVS vs sent = {
  verb = \sp,a,st,t,p => vs.s ! st ! t ! p ;
  prep = vs.prep ;
  obj = \st => sent.subj ! Ga ! st ++ sent.pred ! Plain ;
  ... }
```

Questions are used as complements of verbs in the similar way: regardless of the main clause’s style, the predicate of the interrogative clause should be plain. That is where the s_plain_pred field of the QS category is used. The interrogative clause is followed by the verb without any other particles [Morikawa 2003]:

彼は彼女がどこに住んでいるか知っています。
[he] WA [she] GA [where] [is living] KA [is knowing]
```
Kare wa kanojo ga doko ni sunde iru ka shitte imasu.
```
He knows where she lives.

```
ComplVQ vq qs = {
  verb = \sp,a,st,t,p => vq.s ! st ! t ! p ;
  prep = "" ;
  obj = \st => qs.s_plain_pred ! Ga ! st ;
  ... }
```
2.5.2.3 Adjectives

The GF lexicon contains one verb of the VA (adjective-complement verb) category – to become (Jap: なる “naru”). Adjectives acting as its complements are placed before it in the adverbial form:

タクシーがきれいになった。
[taxi] GA [beautiful] [became]
Takushii ga kirei ni natta.
Taxis have become smart.

ComplVA va ap = {
  verb = \sp,a,st,t,p => va.s ! st ! t ! p ;
  prep = “” ;
  obj = \st => ap.adv ! st ;
  ... } ;

2.5.2.4 Adverbs

The GF syntax provides two functions for adding adverbs to VP (and VPSlash described below): when they follow the verb and when they precede it. Japanese makes no distinction between those cases – the verb in a clause is always in the end, after all the complements.

If the adverb is simple (i.e. not formed by means of a clause), it goes directly before the VP (or VPSlash):

ジョンさんがここに来ました。
Jon-san ga koko ni kimashita.
John came here.

However, if an adverb is represented by a subordinate clause (e.g. when she sleeps), it cannot be attached directly to the verb because of the strict word order. Instead, we have to store this clause separately in the field prepositive and place it before the main clause when we assemble it (at the Clause level).

AdvVP vp adv = {
  verb = vp.verb ;
  obj = \st => case adv.prepositive of {
    True => vp.obj ! st ;
    False => adv.s ! st ++ vp.obj ! st
  } ;
  prepositive = \st => case adv.prepositive of {
    True => vp.prepositive ! st ++ adv.s ! st ;
    False => vp.prepositive ! st
  } ;
}

This results in ambiguous structures, since the same mechanism is used in complementation of other categories (e.g. NP or AP) with adverbs. For example, the translation of the sentence I am happy when she sleeps would have at least three readings since our concrete grammar can attribute the subordinate clause when she sleeps to the NP I, AP happy and VP am happy.
2.5.3 Verb phrases missing complements

This category (VPSlash) looks mostly like VP, but lacks an NP complement. For example, a 2-place verb (e.g. *love*) at first turns into a VPSlash (*love (it)*), which is then complemented with an NP (or a reflexive pronoun, which is not an NP in our grammar) and becomes a full-fledged VP (*love it*). Or, to become a VPSlash, a 3-place verb should be complemented with one NP (e.g. *give it (to her)*), and then through the same rule it becomes a VP (*give it to her*). The GF grammar has two rules for the two orders of complementing 3-place verbs (i.e. *give it (to her)* or *give (it) to her*), but in Japanese those NP complements have the fixed order ([to her] [it] [give]), so those two rules are linearized in the same way.

The V3 category in the GF syntax includes the verb *to give*, which is a part of the complex “giving-and-receiving” grammar in Japanese schematically represented as follows:

![Figure 2. Japanese Verbs of Giving](Wikibooks: Japanese/Lessons/Giving and Receiving)

We simplified this grammar and took more or less neutral verbs くれる “kureru” and 上げる “ageru” for the case when the receiver is the speaker (that is, *somebody gives me*) and all other cases.

太郎は私にマンガをくれた。
[Taro] WA [I] [to] [manga] O [gave]
*Tarou wa watashi ni manga o kureta.*
Taro gave me a manga.

私は妹にお菓子を上げる。
[I] WA [my sister] [to] [candy] O [give]
*Watashi wa imouto ni okashi o ageru.*
I give my sister a candy.

We did not take the verbs 差し上げる “sashiageru” (give to superiors), 下さる “kudasaru” (give to a speaker from a superior) and 造る “yaru” (give to inferiors; rather disrespectful) since we cannot decide whether the giver is superior or not. Thus, 先生 “sensei” (teacher) is likely to be a superior, but we are unable to analyse the context and decide, for example, who John is.
Because of this differentiation, the V3 category obtained the Speaker parameter in all verbal forms:

\[ V3 = \{ s : \text{Speaker} \rightarrow \text{Style} \rightarrow \text{TTense} \rightarrow \text{Polarity} \rightarrow \text{Str} ; \]
\[ \text{te, a\_stem, i\_stem, ba : \text{Speaker} \rightarrow \text{Str} ; \text{prepl, prep2 : Str} \} ; \]

The verb \textit{give} gets a special record in the Resource file:

\[ \text{mkGive} : \text{Verb3} = \{
    s = \text{table}\{\}
    \text{Me} \Rightarrow \{\text{st, t, p} \Rightarrow (\text{mkVerb } \"くれる\" \text{Gr2}).s ! \text{st} ! \text{t} ! \text{p} ;
    \text{SomeoneElse} \Rightarrow \{\text{st, t, p} \Rightarrow (\text{mkVerb } \"上げる\" \text{Gr2}).s ! \text{st} ! \text{t} ! \text{p}
    \} ; \ldots \} ; \}
\]

All other 3-place verbs have the same verbal forms for both \textit{Speaker} types:

\[ \text{mkVerb3} : \text{Str} \rightarrow \text{Str} \rightarrow \text{VerbGroup} \rightarrow \text{Verb3} = \]
\[ \{\text{uru, p1, p2, gr} \rightarrow \{
    \text{s} = \{\text{sp} \Rightarrow (\text{mkVerb uru gr}).s ; \}
    \text{ba} = \{\text{sp} \Rightarrow \text{mkBaForm uru} ; \ldots \} ; \}
\]

When we complement a V3 with an NP, the correct verb for \textit{give} is chosen depending on the NP’s field \textit{meaning} − \textit{Me} or \textit{SomeoneElse} (see 2.3.5):

\[ \text{Slash2V3 v3 np} = \{
    \text{s} = \{\text{sp, st, t, p} \Rightarrow v3.s ! \text{np.\text{meaning}} ! \text{st} ! \text{t} ! \text{p} ;
    \text{a\_stem} = \{\text{sp} \Rightarrow v3.a\_stem ! \text{np.\text{meaning}} ; \ldots \} ; \}
\]

\section*{2.5.3.1 Verbs with NP and VP/S/QS/AP complements}

A VPSlash can also be formed by means of verbs with two complements – NP and one of the following categories: VP (V2V; \textit{e.g.} \textit{to beg}), S (V2S; \textit{e.g.} \textit{to answer}), QS (V2Q; \textit{e.g.} \textit{to ask}) or AP (V2A; \textit{e.g.} \textit{to paint}).

S and QS complements are attached to the verbs in the same way as for VS and VQ (see 2.5.2.2). NP complements also precede the verb and are followed by the particle \textit{に} “ni”.

\[ \text{SlashV2S v2s s} = \{
    \text{s} = \{\text{sp, st, t, p} \Rightarrow v2s.s ! \text{st} ! \text{t} ! \text{p} ;
    \text{prep} = \"\text{に}\" ;
    \text{obj} = \{\text{st} \Rightarrow \text{s.\text{subj}} ! \text{Ga} ! \text{st} \text{+} \text{s.\text{pred}} ! \text{Plain} \text{+} \"\text{と}\" ; \ldots \} ; \}
\]

\[ \text{SlashV2Q v2q qs} = \{
    \text{s} = \{\text{sp, st, t, p} \Rightarrow v2q.s ! \text{st} ! \text{t} ! \text{p} ;
    \text{prep} = \"\text{に}\" ;
    \text{obj} = \{\text{st} \Rightarrow \text{qs.s\_plain\_pred} ! \text{Ga} ! \text{st} ; \ldots \} ; \}
\]

An AP complement of V2A is attached to it in the adverbial form. NPs are followed by the particle \textit{を} “o” as complements of transitive verbs.

\[ \text{SlashV2A v2a ap} = \{
    \text{s} = \{\text{sp, st, t, p} \Rightarrow v2a.s ! \text{st} ! \text{t} ! \text{p} ;
    \text{prep} = \"\text{を}\" ;
    \text{obj} = \text{ap}\_adv ; \}
\]

VPs precede the verbs of the V2V category in the nonpast plain form together with the construction \textit{ように} “you ni”. Here we faced a the word order contradiction in building VPSlashes out of V2V and other verbs. The general scheme of constructing VPs with two complements can be the following:

We start composing the VP part (“[Receiver] ni [object] o give”) from the first NP (“[Receiver] ni”). We chose that order because we have to decide first what verb to choose in case of the verb give, since this depends on the Receiver (whether it is I or not). This algorithm is appropriate for all V3s, V2S, V2Q and V2A we have.

But in case of a complex VP composed by means of V2V we have, for example:

Mary begs John to buy a car.

We start from “[car] o [to buy]” and add “John” afterwards (through the ComplSlash rule). The order we have here (vpslash.обj！s！st！t！p) is inappropriate now, it would change the particles に“ni” and を “o”. So we added the Boolean label v2vType to the VPSlash category; if True, the correct particle is added to the object within the obj field, leaving the prep field empty.

ComplSlash vpslash np = {
    verb = \sp,a,st,t,p => case np.changePolar of {
        True => vpslash.s ! sp ! st ! t ! Neg ;
        False => vpslash.s ! sp ! st ! t ! p ;
    } ;
    prep = case np.needPart of {
        True => case vpslash.v2vType of {
            True => [] ;
            False => vpslash.prep ;
        } ;
        False => []
    } ;
    obj = \st => case vpslash.v2vType of {
        True => np.s ! st ++ vpslash.prep ++ vpslash.obj ! st ;
        False => vpslash.obj ! st ++ np.s ! st
    } ; ... } ;

2.5.3.2 Complementation of VPSlash with VV and V2V

The mechanisms of attaching the verbs of the VV (must, can, want) and V2V (beg) categories to verb phrases missing complements (VPSlash) are the same as those for VP. The verb want is also of special interest since it varies depending on the speaker (Me or SomeoneElse). Thus, VPSlash also got the parameter Speaker, whose value is then inherited by VP (see 2.5.2.1).

SlashVV v vpslash = case v.sense of {
    Abil => {
        s = \sp,st,t,p => vpslash.s ! sp ! Plain ! TPres ! ResJap.Pos ++
            “こと” ++ v.s ! sp ! st ! t ! p ; ... } ;
    Oblig => {
        s = \sp,st,t,p => vpslash.a_stem ! sp ++ “なければ” ++
            v.s ! sp ! st ! t ! Neg ; ... } ;
    Wish => {
        s = \sp,st,t,p => vpslash.i_stem ! sp ++ v.s ! sp ! st ! t ! p ;
        ... } ;
} ;
SlashV2VNP v2v np vpslash = {
    s = \sp,st,t,p => vpslash.s ! sp ! Plain ! TPres ! ResJap.Fos ++ "よ
    ++ v2v.s ! st ! t ! p ;
}

2.5.3.3 Complementation of VPSlash with the reflexive pronoun

Japanese has one main reflexive pronoun – 自分 “jibun”. Its meaning is both singular and plural, masculine and feminine. It can be used for humans and for warm-blooded animals. It cannot be used for fish, reptiles, insects or inanimate objects [Akiyama 2002]:

ひろしは健司に自分のことを話した。
[Hiroshi] WA [Kenji] [to] [self] NOMINALIZER [about] [talked]
Hiroshi wa Kenji ni jibun no koto o hanashita.
Hiroshi talked to Kenji about himself.

ReflVP vpslash = {
    verb = \sp,a,st,t,p => vpslash.s ! sp ! st ! t ! p ;
    prep = vpslash.prep ;
    obj = \st => "自分" ; ... ) ;
}

2.5.4 Copula, complements to copula

The Japanese copula です “desu” is used to indicate such things as condition, quality, number, characteristics or identity. It conjugates according to the following table:

<table>
<thead>
<tr>
<th>Style, tense</th>
<th>Affirmative</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain nonpast</td>
<td>だ “da”</td>
<td>ではない “dewa nai”</td>
</tr>
<tr>
<td>Plain past</td>
<td>だった “datta”</td>
<td>でなかった “dewa nakatta”</td>
</tr>
<tr>
<td>Polite nonpast</td>
<td>です “desu”</td>
<td>ではありません “dewa arimasen”</td>
</tr>
<tr>
<td>Polite past</td>
<td>でした “deshita”</td>
<td>ではありませんでした “dewa arimasen deshita”</td>
</tr>
</tbody>
</table>

Table 11. Japanese copula です

Adjectival phrases, noun phrases and adverbs can serve as complements to the copula and, together with it, form an intermediate category Comp, which is then turned into VP. The record type of Comp is mostly the same as that of VP, except that it has no Speaker parameter (since it is not complemented with V2V verbs directly) and no prep field.

Noun phrases or common nouns (NPs without determiners) simply precede the copula without any particles:

田中さんは先生です。
[Mr. Tanaka] WA [teacher] [is]
Tanaka-san wa sensei desu.
Mr. Tanaka is a teacher.

It should be noted that we changed the verb in the i_stem field: it is used to produce the construction want to be, which is translated into Japa-
nese by means of the verb なる “naru” – *want to become* ([NP] ni naru). So the *i-stem* value here would be になり “ni nari” (the particle に and the *i-stem* of the verb なる).

CompNP np = {
    verb = \(\lambda a, st, t, p \Rightarrow \text{mkCopula}.s ! st ! t ! p ; \)
    i_stem = \(\lambda a, st \Rightarrow \text{“になり”} ; \)
    obj = \(\lambda st \Rightarrow \text{np.s} ! st ; \ldots \) ;
}

Adverbs as complements to the copula cannot be used with です; instead, we should use the verbs いる “iru” for animate subjects and ある “aru” for inanimate ones. Unlike English, in Japanese there is a difference between the copula です, which means *to be, to equal*, and the verbs いる and ある, which mean *to be, to exist*. To distinguish いる and ある, we had to introduce the parameter *Animateness* with values *Anim* and *Inanim*.

mkExistV : VP = {
    verb = \(\lambda sp \Rightarrow \text{table} \{\)
        Anim => \(\lambda st, t, p \Rightarrow \text{(mkVerb “いる” Gr2)}.s ! st ! t ! p ; \)
        Inanim => \(\lambda st, t, p \Rightarrow \text{table} \{\)
            Resp => \(\lambda (TPres|TFut) \Rightarrow \text{table} \{\)
                Pos => "あります" ;
                Neg => "ありません" ;
            \} ; \ldots \} ; \ldots \} ; \ldots \}
}

The categories *Comp* and *VP* (as its successor) conjugate according to this parameter, and all categories related to *NP* (N, N2, Pron, etc) have the inherited label *anim*, whose value is stated in the concrete Lexicon. At the *Clause* level the subject and predicate agree in *Animateness*.

本はテーブルの上にあります。
[book] WA [table] [on] [is]
*Hon wa teeburu no ue ni arimasu.*
The book *is on the table.*

あなたは昨日東京にいましたか。
[you] WA [yesterday] [Tokyo] [in] [was] KA
*Anata wa kinou Toukyou ni imashita ka?*
*Were you in Tokyo yesterday?*

Adjectives in the function of predicates involve the copula です, but the conjugation of their predicate form does not always coincide with the forms of です taken alone. Nevertheless, we just use the adjectival predicate forms that are already predetermined in the paradigm:

お天気が良かったです。
[weather] GA [was good]
*Otenki ga yokatta desu.*
The weather *was good.*

CompAP ap = \(\lambda a, st, t, p \Rightarrow \text{ap.pred} ! st ! t ! p ; \ldots \) ;
2.6 Adverbs

2.6.1 Forming of adverbs

The GF syntax predetermines three ways to produce adverbs: by means of adjectives, noun phrases with prepositions and subordinate clauses formed by sentences and conjunctions when, if, because or that (belonging to the Subj category). As a rule, all adverbs except those represented by clauses immediately precede structures they are attributed to (NP, AP, VP, etc). Subordinate clauses cannot be attached directly to some part of the sentence, they are usually placed before the main clause, which results in ambiguity (see 2.5.2.4). To find out whether an adverb is a clause or not, we supplied it with the Boolean label prepositive. If True, it is stored in the prepositive field of the structure it is attributed to (this prepositive label type is not Boolean anymore; it is String with the Style parameter).

The adverbial form of adjectives is specified in their paradigms.

```plaintext
PositAdvAdj a = {s = \st => a.adv ;
     prepositive = False} ;
```

In the second way of the adverb production, a preposition is placed after a noun phrase.

```plaintext
PrepNP prep np = {s = \st => np.s ! st ++ prep.s ;
     prepositive = False} ;
```

To form a subordinate clause, one of the conjunctions of the Subj category (except if) is placed after the sentence that is aimed to become subordinate. The conjunction if usually has no independent equivalent in Japanese; instead, the sentence’s predicate is used in the conditional form (so-called ba-form; see 2.4.1, 2.5.1.2). For this reason, we have to keep the ba field in the S record type.

```plaintext
SubjS subj s = {
     s = \st => case subj.type of {
        If => s.ba ! (Wa | Ga) ! st ++ subj.s ;
        _ => s.s ! (Wa | Ga) ! st ++ subj.s
     } ;
     prepositive = True
} ;
```

The abstract syntax also contains some functional categories of adverbs that usually modify only one certain morphological category: adjective-modifying adverbs (AdA, e.g. almost, so, too, very), numeral-modifying adverbs (AdN, e.g. almost, at least, at most), comparative adverbs (CAdv, e.g. less, more, as), adverb directly attached to a verb (AdV, e.g. always) and interrogative adverbs (IAdv, e.g. how, how many, when, etc).

Comparison adverbs are supposed to serve as numeral adverbs as well. In the Japanese syntax we do not involve adverbs as a morphological category in the comparative function, we use the wordより “yori” (than) and positive or negative forms to indicate comparison relations. But numeral adverbs with the same meaning have separate lexical realizations: these are post-positives 以上 “ijou” (more) and 以下 “ika” (less). The adverb as in the
numeral-modifying function is close to *as many as* and can be translated as the Japanese postpositive もの “mono”. So the CAdv category has the special field s_adn for storing numeral adverbs. Unlike other numeral adverbs (*almost, at least, at most*), comparison adverbs in the numeral-modifying function are placed after the number (and its counter, if present), not before it. So the AdN category got the Boolean *postposition* label.

```plaintext
AdnCAdv cadv = {s = cadv.s_adn ; postposition = True} ;
```

Adjective-modifying adverbs can be also attached to regular adverbs (belonging to the Adv category). At the same time adverbs produced from adjectives can serve as adjective-modifying adverbs (*e.g.* extremely).

### 2.6.2 Comparative adverbs

Forming of structures with comparative adverbs is realized in the same way as in case of comparative adjectives (see 2.4.2). Specifically, adverbs also do not have any special comparative form; instead, the word より “yori” is attached to an NPs that something is compared to:

田中さんの方が木村さんより早く歩きます。

*Mr. Tanaka walks faster than Mr. Kimura.*

On the opposite, to translate the construction “*less + adv*” we use より and the negative adverbial form of adjective.

Structures with comparative adverbs having a sentence as an object of comparison are formed by means of the same words. The predicate in the subordinate sentence is taken in the plain form, though all other parts should stylistically correlate with the main clause.

```plaintext
ComparAdvAdjS cadv a s = {
 s = \st => case cadv.less of {
 True => s.subj ! Ga ! st ++ s.pred ! Plain ++ cadv.s ++ a.adv ! Neg ;
 False => s.subj ! Ga ! st ++ s.pred ! Plain ++ cadv.s ++ a.adv ! Pos
 } ; prepositive = False} ;
```

### 2.6.3 Conjunction of adverbs

Simple adverbs in a coordinating construction (with the conjunction *and*) are just attached one to another, without any particles. If adverbs are represented by subordinate clauses, they are separated by commas. In case of alternative conjunction (*or*) simple adverbs are joined by means of the particle か “ka”; clauses are separated by commas and construction あるいは “aruiwa” (*or, otherwise*).

```plaintext
BaseAdv x y = {
 and = \st => case <x.prepositive, y.prepositive> of {
 <False, False> => x.s ! st ++ y.s ! st ;
 _              => x.s ! st ++ “,” ++ y.s ! st
 } ;
 or = \st => case <x.prepositive, y.prepositive> of {
 <False, False> => x.s ! st ++ “か” ++ y.s ! st ;
 _              => x.s ! st ++ “, あるいは” ++ y.s ! st
 } ; ... } ;
```
2.7 Numerals

2.7.1 Cardinals

The GF grammar defines numerals from 1 to 999999.

Like in most Western languages, Japanese has units of tens (十 “juu”), hundreds (百 “hyaku”) and thousands (千 “sen”), but then comes one more unit − ten thousand (万 “man”). So when we say forty thousand (40 * 1000) Japanese say [four] man (4 * 10,000). Or the number 111,000 (one hundred eleven thousand) would be [eleven] man [one] sen (11 * 10,000 + 1 * 1000).

This should be taken in consideration in the linearization of the rule pot3plus : Sub1000 -> Sub1000 -> Sub1000000, which produces numbers 1001...9999 (m * 1000 + n). The first Sub1000 should be split into numbers corresponding to mans (tens of thousands) and sens (thousands) that form the whole number.

We had to add labels NumeralType, digit1 and digit2 to Sub100 (including numbers 1...99) and NumeralType, man and sen to Sub1000 (including numbers 1...999). According to the rules in the abstract, one-figure numbers, whose NumeralType is SingleDigit, cannot form mans directly; they can produce only sens. In case of two-figure numbers, we store their first and second digits in the fields digit1 and digit2. The first one will be the number of mans, the second one − the number of sens (type EndNotZero). If a two-figure number is divisible by ten (10, 20, 30, etc − type EndZero), the word sen and its zero number are obviously not pronounced. As for three-figure numbers, the man field stores their first two digits with the correct reading, and sen stores the last digit. Thus, for the number 111 man would be [eleven] and sen would be [one]. Again, if the last digit of a three-figure number is zero (110, 200, 350, etc − type EndZero), we do not pronounce the sen unit at all.

When we deal with numbers 100...199 and 1000...1999 in Japanese, we do not say one hundred..., one thousand..., but just hundred... (“hyaku”), thousand... (“sen”). So we check if Sub10 or Sub1000 mean one through the Boolean label is1 in order to remove it for numbers 100...199 and 1000...1999. The word one is preserved only for the item man.

However, if one hundred or one thousand are parts of a larger number (i.e. if they are not in the beginning), the word one is preserved. For this reason we created two separate fields s_init and s_mid for the cases when one hundred or one thousand are in the number’s beginning or some other position.

The meaning of the tenPlus label is explained in 2.3.1.1.

```
pot2 : Sub10 -> Sub1000 ; -- m * 100

pot2 d = {
    s_init = case d.is1 of {
        True => d.null ++ "百" ;
        False => d.s ++ "百"
    } ;
    s_mid = d.s ++ "百" ;
    numType = EndZero ;
```
man = case d.is1 of {
    True => d.null ++ "十" ;
    False => d.s ++ "十"
} ;
sen = [] ;
...
};
pot3plus : Sub1000 -> Sub1000 -> Sub1000000 ; -- m * 1000 + n

pot3plus d n = {
s = case d.numType of {
    EndZero => d.man ++ "万" ++ n.s_mid ;
    EndNotZero => d.man ++ "万" ++ d.sen ++ "千" ++ n.s_mid ;
    SingleDigit => case d.is1 of {
        True => d.null ++ "千" ++ n.s_mid ;
        False => d.s_init ++ "千" ++ n.s_mid
    }
} ;
...
};

A cardinal can be modified by certain adverbial groups (AdN). Some of them (at least, at most, almost) are placed before the cardinal in Japanese, others (less than, more than) follow the cardinal and the noun’s counter. So AdN gets the Boolean label postposition, and Card gets the label postpositive in which the corresponding adverb is stored and which is inherited by Num and Det. We cannot glue this postpositive to the cardinal immediately as they will be separated by a counter later, when an NP is produced.

AdNum adn card = case adn.postposition of {
    True => {
        s = card.s ;
        postpositive = adn.s ;
        ...
    } ;
    False => {
        s = adn.s ++ card.s ;
        postpositive = [] ;
        ...
    } ;
}

2.7.2 Ordinals

In terms of grammatical functions, ordinals are close to adjectives. Therefore, Ord has the same record type as Adj (see 2.4.1).

We chose the easiest way to form an ordinal in Japanese – the addition of 番目 “banme” to a cardinal. In the attributive position one should also add the particle の “no”.

OrdDigits d = {
pred = \st,t,p => d.s ++ "番目" ++ mkCopula.s ! st ! t ! p ;
attr = d.s ++ "番目の" ;
...
};

However, the combination “banme” often has the meaning “number one”, “number two”, etc. Talking about the order of things/people can be more fine-grained: the suffix 目 “me” is added to the number and counter combination: 一回目 “ikkaime” – the first time, 一人目 “hirime” – the first person, etc.
In the GF grammar Ord also includes superlative adjectives. The Japanese equivalent of most is "ichiban", which is placed in front of the adjective:

日本の一番有名な山は富士山です。
[NJapan] [of] [most] [famous] [mountain] WA [Fuji] [mountain] [is]
Nihon no ichiban yuumeina yama wa Fujisan desu.
The most famous mountain in Japan is Mt Fuji [Bunt 2003].

\[
\text{OrdSuperl} a = \\
\text{pred} = \text{\"一番\" ++ a.pred ! st ! t ! p ;} \\
\text{attr} = \text{\"一番\" ++ a.attr ; ... ) ;}
\]

2.8 Tense

Japanese is often described as having just two tenses: past and non-past; there is no separate future tense. This means that all forms of verbs and predicative adjectives for the present and future tenses (TPres and TFut) coincide. The conditional tense also corresponds to the Japanese non-past (yet, it is usually expressed through the ba-form of a verb or adjective).

Japanese also does not have special verb and adjective forms to express anteriority. Thus, in our grammar the English future perfect is replaced with the Japanese present tense and the present and past perfect correspond to the Japanese past tense, although, for the present perfect, this is not always a correct decision.

The English present perfect (especially perfect continuous) can be close semantically to the Japanese present continuous to express an action that began in the past and continues through the present:

京都に五年間住んでいます。
[Kyoto] [in] [five] [year] [during] [is living]
Kyouto ni gonen kan sunde imasu.
I have been living in Kyoto for five years.

However, we did not take into consideration these nuances, which are mostly semantics-based.

In fact, 16 possible tense forms provided by the GF syntax (4 tenses * 2 anteriority types * 2 polarity types) are reduced to 4 distinct Japanese forms (2 tenses * 2 polarity types).

2.9 Sentences

2.9.1 Declarative sentences

Japanese is a Subject-Object-Verb language. Thus, in a typical sentence, the subject comes first, the object next, and the verb last:
お母さんはパンを買いました。

The noun phrase representing the subject of the sentence is usually followed by the particles は “wa” or が “ga”. は is used to mark the topic of a sentence and to express contrast. が is often used to introduce a new subject. The choice of は or が in a particular case can be complicated by questions of context and the speaker’s intent:

寿司はおいしい。
[Sushi] WA [delicious]
Sushi: it’s delicious.

寿司がおいしい。
[Sushi] GA [delicious]
Sushi is delicious.

In these sentences, the topic marker は directs attention forward to the predicate (it’s delicious), and the subject particle が emphasizes what precedes it (sushi). This distinction is not always easy (nor indeed necessary) to convey in an English translation [Bunt 2003].

In our grammar we do not carry out a contextual analysis, so we provide usage of both particles through the parameter Particle.

The predicate in a clause should agree with the subject in animateness (this is relevant for choosing the correct equivalent for the verb to be, to exist – いる “iru” or ある “aru”; see 2.5.4) and the parameter Speaker (Me or SomeoneElse – for choosing the correct want to do form; see 2.5.2.1). Usually the subject and predicate also correlate stylistically, except when the clause is subordinate and the grammar rules require that the verb is plain.

The Clause unit in our grammar also includes all prepositives of the subject and predicate, which are placed in the beginning of the clause. These may be subordinate adverbs that are related to some parts of the sentence but yet could not be attached directly to them (see 2.3.6.4, 2.5.2.4).

Some NPs need neither は nor が after them since they already contain some other particle (usually も “mo”). Among those NPs are pronouns 誰も “daremo” (everybody or nobody, depending on the predicate’s polarity) and 何も “nanimo” (nothing). For this reason the NP record type has the Boolean field needPart. If False, no other particles are attached to the NP.
At the *Sentence* level a clause gets the *tense* and *polarity* values according to the predetermined polarity, temporal and aspectual features (see 2.8). Some NPs determine the verb’s polarity already at the *Clause* level. These are negative pronouns 誰も (*nobody*) and 何も (*nothing*) and NPs with the quantifier no or predeterminer not. To distinguish this type of NPs, we introduced the Boolean label *changePolar*; if True, the predicate turns into the negative form.

```plaintext
UseCl t p cl = {
  s = \part,st |- case t.a of {
    Simul => case cl.changePolar of {
      False => t.s ++ p.s ++ cl.s ! part ! st ! t.t ! p.b ;
      True  => t.s ++ p.s ++ cl.s ! part ! st ! t.t ! Neg
    } ;
    Anter => case t.t of {
      TPres => case cl.changePolar of {
        False => t.s ++ p.s ++ cl.s ! part ! st ! TPast ! p.b ;
        True  => t.s ++ p.s ++ cl.s ! part ! st ! TPast ! Neg
      } ;
      TPast => case cl.changePolar of {
        False => t.s ++ p.s ++ cl.s ! part ! st ! TPast ! p.b ;
        True  => t.s ++ p.s ++ cl.s ! part ! st ! TPast ! Neg
      } ;
      TFut  => case cl.changePolar of {
        False => t.s ++ p.s ++ cl.s ! part ! st ! TPres ! p.b ;
        True  => t.s ++ p.s ++ cl.s ! part ! st ! TPres ! Neg
      } ) ; ... } ;
}
```

This linearization explains why we differentiate the present and future tenses in our concrete grammar, though in fact they both correspond to the single non-past tense in Japanese. The reason is anteriority: the present anterior form is semantically closer to the Japanese past tense, whereas the future anterior form is replaced with the present tense.

### 2.9.2 Interrogative sentences

The word order for a simple question in Japanese is exactly the same as for a statement, but with addition of the question particle か “ka” at he end. か usually replaces the question mark:

池田さんは学生です。池田さんは学生ですか。
*Ms Ikeda* WA [student] [is] *Ikeda-san wa gakusei desu.* *Ikeda-san wa gakusei desu ka.*

Ms Ikeda is a student.

Is Ms Ikeda a student?

An interrogative sentence can be also used as a subordinate clause. In this case its predicate should be turned into the plain style. For this purpose, the QCl record type contains the field *s_plain_pred*:

```plaintext
s_plain_pred = \part,st,t,p => cl.subj ! part ! st ++
               cl.pred ! Plain ! t ! p ++ "か" ;
```
There is no need to store the subject and predicate of a question clause separately, like for the \textit{Cl} category; in our grammar there are no cases when some new words are inserted between them.

All interrogative words (pronouns, complements, determiners and quantifiers) are placed after the subject and before the verb phrase in a question.

川場さんはなぜ来ませんでしたか。
[Mr Kawaba] WA [why] [did not come] KA
Kawaba-san wa naze kimasen deshita ka.
Why didn’t Mr Kawaba come?

A question word is placed in the beginning of the sentence only if it is a subject question:

誰が眠ていますか。
[who] GA [is sleeping] KA
Dare ga nete imasu ka.
Who is sleeping?

Except morphological pronouns, the GF interrogative pronouns (\textit{IP}) category also includes combinations of interrogative determiners and CNs (noun phrases without determiners). Most commonly, determiners precede nouns, but the determiner \textit{how many} involves special grammar. For example, if the phrase \textit{how many songs} replaces the subject (i.e. it is a subject question) the scheme would be the following: [songs] が [how many]. That is, the word \textit{songs} is regarded as a subject, correspondingly marked with the subject particle が and followed by the question word:

卵がいくつありますか。
[egg] GA [how many] [is] KA
Tamago ga ikutsu arimasu ka.
How many eggs are there?

But if the same phrase replaces an object, the noun following \textit{how many} should not be marked with が, since a sentence already has a topic (with the particle は) or a subject (with が). In this case いくつ is followed by the possessive particle の and placed before the noun:

あなたはいくつの部屋を持っていますか。
[you] WA [how many] [room] O [is possessing]
Anata wa ikutsu no heya o motte imasu ka.
How many rooms do you have?

Therefore, the \textit{IP} category has to contain linearizations both for the subject and object positions – $s_{\text{subj}}$ and $s_{\text{obj}}$. To check if an IP contains an interrogative \textit{how many}, we introduced the Boolean label \textit{how8many}. 

41
In subject questions an IP with how many is not followed by が, since it already contains it:

```
QuestVP ip vp = {
  s = \part,st,t,p => case ip.how8many of {
    True => ip.s_subj ! st ++ vp.obj ! st ++ vp.prep ++
              vp.verb ! SomeoneElse ! ip.anim ! st ! t ! p ++ "か" ;
    False => ip.s_subj ! st ++ "が" ++ vp.obj ! st ++ vp.prep ++
              vp.verb ! SomeoneElse ! ip.anim ! st ! t ! p ++ "か"
  } ;
...);
```

### 2.9.3 Conjunction of sentences

In a coordinating construction sentences are joined by means of commas and the word そして "soshite" (and). The conjunction or is translated as それとも "soretomo" and is also preceded by a comma. The complexity of writing rules for rows of sentences is that we should create labels for all conjunctions and all fields that a normal S category has:

```
BaseS x y = {
  and = \part,st => x.s ! part ! st ++ "，そして" ++ y.s ! Ga ! st ;
  or = \part,st => x.s ! part ! st ++ "，それとも" ++ y.s ! Ga ! st ;
  teAnd = \part,st => x.te ! part ! st ++ "，" ++ y.te ! Ga ! st ;
  teOr = \part,st => x.s ! part ! st ++ "，それとも" ++ y.te ! Ga ! st ;
  baAnd = \part,st => x.ba ! part ! st ++ "，" ++ y.ba ! Ga ! st ;
  baOr = \part,st => x.s ! part ! st ++ "，それとも" ++ y.ba ! Ga ! st ;
  subj = \part,st => x.subj ! part ! st ;
  predAnd = \st => x.pred ! st ++ "，そして" ++ y.s ! Ga ! st ;
  predOr = \st => x.pred ! st ++ "，それとも" ++ y.s ! Ga ! st ;
  pred_teAnd = \st => x.pred_te ! st ++ "，" ++ y.te ! Ga ! st ;
  pred_teOr = \st => x.pred ! st ++ "，それとも" ++ y.te ! Ga ! st ;
  pred_baAnd = \st => x.pred_ba ! st ++ "，" ++ y.ba ! Ga ! st ;
  pred_baOr = \st => x.pred ! st ++ "，それとも" ++ y.ba ! Ga ! st ;
} ;
```
2.9.4 Subordinate clauses

2.9.4.1 Relative clauses

Formally, there is no distinction between relative clauses and noun modification by participles or adjectives. In all cases, an adjective, participle (which is just a plain verb form) or a clause in the normal word order is simply placed before the noun, i.e. unlike English there are no relative pronouns or changes in the word order.

As there are no relative pronouns (or any other indication of the case relation between the modifying section and the noun it modifies), it is sometimes only from the context (or knowledge of the world) that it can be determined [Kaiser 2004]:

僕が記事を書いたレストラン
[I] GA [article] O [wrote] [restaurant]
Boku-ga kiji-o kaita resutoran

(1) A restaurant about which I wrote an article
(2) A restaurant in which I wrote an article

The first translation is the context-free interpretation of choice, but the second one is possible with the proper context.

The RP (relative pronoun) category in the abstract syntax also includes those pronouns that are supplemented with prepositional phrases or genitives. However, since there are no relative pronouns in Japanese and their prepositions are omitted, the NP attached to the RP becomes the subject of a relative clause. Constructions like this are hardly possible in natural Japanese speech:

お母さんが寝ている人
[mother] GA [is sleeping] [man]
Okaasan ga nete iru hito

A man whose mother is sleeping

A relative sentence can modify not only an NP, but also the whole sentence. Here it is essential for us to know whether the subject of the relative clause is represented by a sole relative pronoun or some noun phrase. In the first case the subject in the Japanese relative clause is actually missing (since, again, since there are no relative pronouns), but in the sentences like whose mother is sleeping or whom John loves the subject is present. To differentiate these types of relative clauses we introduced the Boolean label missingSubj. If the subject is omitted, the main sentence is regarded as a subject of the relative clause. For example, the sentence She sleeps, which is good translated into Japanese can be interpreted as “The fact that she sleeps is good”. Therefore, the main clause should be substantivized by means of the word こと “koto” (which turns the whole clause into a noun phrase), then followed by the particle が “ga” (as a normal subject) and ended with the predicate of another clause, which actually becomes the main one, not subordinate:
彼女が寝ていることが良いです。

[she] GA [is sleeping] NOMINALIZER GA [is good]
Kanojo ga nete iru koto ga ii desu.
She sleeps, which is good. (or It is good that she sleeps.)

In case when the subject of the relative clause is not omitted, the main clause becomes not the subject, but the object of the relative clause’s predicate. For example, the Japanese translation of the sentence She sleeps, which he knows is structurally the same as He knows that she sleeps. The main clause (she sleeps) is also substantivized with こと and, as an ordinary object of a transitive verb, is placed between the subject and predicate of the clause he knows and followed by the particle を “o”:

彼は彼女が寝ていることを知っている。

[he] WA [she] GA [is sleeping] NOMINALIZER O [is knowing]
Kare wa kanojo ga nete iru koto o shitte iru.
She sleeps, which he knows. (or He knows that she sleeps.)

The rule for a sentence modified by a relative clause is linearized in the following way:

RelS sent rs = case rs.missingSubj of {
  True => {s = \part,st => rs.subj ! part ! st ++ sent.subj ! Ga ! st
         ++ sent.pred ! Plain ++ “ことが” ++ rs.pred ! Inanim ! st ; ...} ;
  False => {
    s = table {
      Wa => \st => rs.subj ! Wa ! st ++ sent.subj ! Ga ! st ++
        sent.pred ! Plain ++ “ことを” ++ rs.pred ! Inanim ! st ;
      Ga => \st => rs.subj ! Ga ! st ++ sent.subj ! Ga ! st ++
        sent.pred ! Plain ++ “ことを” ++ rs.pred ! Inanim ! st
    } ; ... } ;
} ;

2.9.4.2 Conjunction of relative clauses

The correct conjunction of Japanese relative clauses is possible only if subjects of all those clauses are missing (e.g. who is sleeping). The value of the missingSubj field of these RSs is True. In fact, there clauses are just verb phrases that can be easily joined. The meaning of the conjunction and is expressed by the verb’s te-form, so we take the te-form of a clause and put a comma after it. The verb of the last clause (i.e. verb phrase) is taken in the plain form. The meaning of the conjunction or between RSs of this type is expressed by the particle か “ka”, like between nouns or adjectives.

However, in Japanese it is practically impossible to join relative sentences that have subjects (e.g. whose mother is sleeping and with whom I walk). It is recommended to restructure these constructions and split them in two (or more) sentences. However, the GF abstract syntax determines that all RSs can join together. This inconsistency results in the ungrammatical output, unfortunately.
BaseRS x y = {
    and = \a,st => x.te ! a ! st ++ "," ++ y.s ! a ! st ;
    or = \a,st => case <x.missingSubj, y.missingSubj> of {
        <True, True> => x.s ! a ! st ++ "か" ++ "," ++ y.s ! a ! st ;
        _ => x.te ! a ! st ++ ",あるいは" ++ y.s ! a ! st
    } ; ... } ;

2.9.4.3 Other subordinate clauses

The grammatical realization of a subordinate clause depends on the
conjunction (the Subj category) that determines its relation to the main clause.

The grammar for sentences with conjunctions のに “noni” (although),
から “kara” (because) and と “to” (when) is the same: the subordinate clause
is placed in the beginning (its predicate is taken in the plain style), then it is
followed by the conjunction and the main clause.

In case of conjunction if, the predicate of the subordinate clause is turned
into the ba-form (positive or negative, depending on the polarity); no separate
word for if is used.

Conjunction that semantically turns the subordinate clause into an ob-
ject of the main clause’s predicate. Therefore, the subordinate clause (with
the plain predicate) is substantivized by means of the word こと “koto” and,
with the particle を “o”, is placed between the subject and predicate of the
main clause (unlike the cases with other conjunctions, where the subordinate
clause is placed before the main one).

To distinguish conjunctions, the Subj category has the field type with
the possible values That, If and OtherSubj:

SSsubjS s1 subj s2 = case subj.type of {
    If => {s = \part,st => s1.ba ! part ! st ++ subj.s ++
        s2.s ! Ga ! st ; ... } ;
    That => {s = \part,st => s1.subj ! part ! st ++ s2.subj ! Ga ! st
        ++ s2.pred ! Plain ++ subj.s ++ s1.pred ! st ; ... } ;
    OtherSubj => {s = \part,st => s1.subj ! part ! st ++ s1.pred ! Plain
        ++ subj.s ++ s2.s ! Ga ! st ; ... } ;
} ;
Chapter 3

Evaluation of the Japanese Resource Grammar

To assess the correctness of the developed Japanese grammar we used a test set that contains syntax trees corresponding to 583 test units (words, phrases and sentences). We automatically generated English and Japanese linearizations of these trees and assessed the correctness of the Japanese translations. The test set is composed of the test units developed in [Khegai 2006] and examples used in the synopsis of the RGL [GF Resource Grammar Library: Synopsis].

The test set is aimed not only at proof-reading but also reflection of the resource library’s coverage. It is organized in such a way that displays all the rules in the grammar and possible combinations of the categories. For example, it starts from simple trees connected with the AP category subsequently making them more complex:

Lang> Lang: UttAP(PositA warm_A)
LangEng: warm
LangJap: 暖かい

Lang> Lang: UttAP(ComparA warm_A (UsePron i_Pron))
LangEng: warmer than I
LangJap: 僕より暖かい

Lang> Lang: SentAP (PositA good_A) (EmbedS (UseCl (TTAnt TPres ASimul) PPos (PredVP (UsePron she_Pron) (UseComp (CompAdv here_Adv)))))
LangEng: good that she is here
LangJap: 彼女はここでいることが良い

The test set was organized in such a way that displays the usage of every category in the grammar. The numbers of test units corresponding to certain grammar categories are shown below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of test units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP – adjectival phrase</td>
<td>25</td>
</tr>
<tr>
<td>AdA – adjective-modifying adverb</td>
<td>5</td>
</tr>
<tr>
<td>AdN – numeral-modifying adverb</td>
<td>5</td>
</tr>
</tbody>
</table>

46
<table>
<thead>
<tr>
<th>Category</th>
<th>Number of test units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdV – adverb directly attached to verb</td>
<td>1</td>
</tr>
<tr>
<td>Adv – verb-phrase-modifying adverb</td>
<td>17</td>
</tr>
<tr>
<td>Ant – anteriority</td>
<td>2</td>
</tr>
<tr>
<td>CAdv – comparative adverb</td>
<td>3</td>
</tr>
<tr>
<td>CN – common noun (without determiner)</td>
<td>23</td>
</tr>
<tr>
<td>Card – cardinal number</td>
<td>4</td>
</tr>
<tr>
<td>Cl – declarative clause, with all tenses</td>
<td>42</td>
</tr>
<tr>
<td>CISlash</td>
<td>7</td>
</tr>
<tr>
<td>Comp – complement of copula, such as AP</td>
<td>7</td>
</tr>
<tr>
<td>Conj – conjunction</td>
<td>7</td>
</tr>
<tr>
<td>Det – determiner phrase</td>
<td>26</td>
</tr>
<tr>
<td>Dig</td>
<td>10</td>
</tr>
<tr>
<td>Digits – cardinal or ordinal in digits</td>
<td>4</td>
</tr>
<tr>
<td>IAdv – interrogative adverb</td>
<td>7</td>
</tr>
<tr>
<td>IDet – interrogative determiner</td>
<td>5</td>
</tr>
<tr>
<td>IP – interrogative pronoun</td>
<td>15</td>
</tr>
<tr>
<td>IQuant</td>
<td>1</td>
</tr>
<tr>
<td>Imp – imperative</td>
<td>9</td>
</tr>
<tr>
<td>Imp – imperative</td>
<td>3</td>
</tr>
<tr>
<td>NP – noun phrase (subject or object)</td>
<td>60</td>
</tr>
<tr>
<td>Num – number determining element</td>
<td>20</td>
</tr>
<tr>
<td>Numeral – cardinal or ordinal in words</td>
<td>6</td>
</tr>
<tr>
<td>Ord – ordinal number (used in Det)</td>
<td>7</td>
</tr>
<tr>
<td>PConj – phrase-beginning conjunction</td>
<td>4</td>
</tr>
<tr>
<td>Phr – phrase in a text</td>
<td>5</td>
</tr>
<tr>
<td>Pol – polarity</td>
<td>2</td>
</tr>
<tr>
<td>Predet – predeterminer (prefix Quant)</td>
<td>4</td>
</tr>
<tr>
<td>Prep – preposition, or just case</td>
<td>21</td>
</tr>
<tr>
<td>Pron – personal pronoun</td>
<td>9</td>
</tr>
<tr>
<td>Punct</td>
<td>3</td>
</tr>
<tr>
<td>QCl – question clause, with all tenses</td>
<td>38</td>
</tr>
<tr>
<td>QS – question</td>
<td>3</td>
</tr>
<tr>
<td>Quant – quantifier (‘nucleus’ of Det)</td>
<td>6</td>
</tr>
<tr>
<td>RCl – relative clause, with all tenses</td>
<td>25</td>
</tr>
<tr>
<td>RP – relative pronoun</td>
<td>1</td>
</tr>
<tr>
<td>RS – relative</td>
<td>10</td>
</tr>
<tr>
<td>S – declarative sentence</td>
<td>8</td>
</tr>
<tr>
<td>SC – embedded sentence or question</td>
<td>6</td>
</tr>
<tr>
<td>Category</td>
<td>Number of test units</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>SSlash</td>
<td>1</td>
</tr>
<tr>
<td>Sub100</td>
<td>3</td>
</tr>
<tr>
<td>Sub1000</td>
<td>3</td>
</tr>
<tr>
<td>Subj – subjunction</td>
<td>5</td>
</tr>
<tr>
<td>Tense – tense</td>
<td>4</td>
</tr>
<tr>
<td>Text – text consisting of several phrases</td>
<td>11</td>
</tr>
<tr>
<td>Unit</td>
<td>9</td>
</tr>
<tr>
<td>Utt – sentence, question, word...</td>
<td>16</td>
</tr>
<tr>
<td>V2 – two-place verb</td>
<td>3</td>
</tr>
<tr>
<td>V2A – verb with NP and AP complement</td>
<td>1</td>
</tr>
<tr>
<td>V2Q – verb with NP and Q complement</td>
<td>2</td>
</tr>
<tr>
<td>V2S – verb with NP and S complement</td>
<td>2</td>
</tr>
<tr>
<td>V2V – verb with NP and V complement</td>
<td>2</td>
</tr>
<tr>
<td>V3 – three-place verb</td>
<td>1</td>
</tr>
<tr>
<td>VA – adjective-complement verb</td>
<td>1</td>
</tr>
<tr>
<td>VP – verb phrase</td>
<td>34</td>
</tr>
<tr>
<td>VPSlash – verb phrase missing complement</td>
<td>9</td>
</tr>
<tr>
<td>VQ – question-complement verb</td>
<td>1</td>
</tr>
<tr>
<td>VS – sentence-complement verb</td>
<td>1</td>
</tr>
<tr>
<td>VV – verb-phrase-complement verb</td>
<td>6</td>
</tr>
<tr>
<td>Voc – vocative or “please”</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>583</strong></td>
</tr>
</tbody>
</table>

*Table 12. The grammar categories and the numbers of test units displaying their forming and usage*

As we can see, the more “complex” the category is (i.e. the more rules are used to assemble or modify it), the more test units are needed to verify the rules corresponding to it. Thus, the *NP* category is the leader in the number of rules through which NPs are formed or modified. To display all those variants, 60 phrases were included in the test set only for NP. However, it is obvious that usage of some categories can be tested also when they are components of some categories of a higher rank. Therefore, test units for higher-order categories (such as *S, Phr* or *Utt*) additionally verify the usage of their constituents (*NP, AP, VP*, etc).

In some examples, several linearizations of the same syntax tree are possible. Most usually the reason for this is that we provided the categories of style and particles *wa/ga* in the Extra syntax. Sentences of the polite and plain style in Japanese are equally possible, the same is for the particles *wa/ga* marking the topic of the subject of the sentence. The system shows only the first obtained result.
The test has confirmed our assumption that some grammatical constructions predetermined in the abstract syntax are hardly possible, if not impossible, to implement in Japanese. For example, since Japanese has no relative pronouns, it is undesirable to use complex relative clauses. The following structure converted into Japanese is formally grammatical, but its semantics is totally lost as there is no way to denote the relation between the main and subordinate clauses:

Lang> Lang: ExistNP (DetCN (DetQuant IndefArt NumSg) (RelCN (UseN woman_N) (UseRCl (TTAnt TPres ASimul) PPos (RelSlash (FunRP possess_Prepp (DetCN (DetQuant DefArt NumSg) (UseN2 mother_N2)) IdRP) (SlashVP (UsePN john_PN) (SlashV2a love_V2)))))))
LangEng: there is a woman the mother of whom John loves
LangJap: ジョンがお母さんを愛する女はいる

As we can see, not only a relative pronoun is missing, but also the preposition determining the relation of mother to woman. In the natural Japanese language, one would restructure this phrase and possibly split it into successive clauses.

Very often in Japanese it is only the context that can determine the correct sense of a relative clause (see 2.9.4.1). For example, formally, the structures the table on which the cat is sitting and the table under which the cat is sitting are translated into Japanese in one and the same way. If the context (or knowledge about the world) is not enough, one should restructure the phrase.

In addition to this, there are some other restrictions of using subordinate clauses in Japanese, particularly conjoining them (see 2.9.4.2).

Some examples clarified the sense and usage of some words from the abstract Lexicon. Thus, we first assumed that a 2-place verb speak involves the preposition about (e.g. speak about weather). But the phrase speaks twelve languages from the test set made it clear that the verb speak was meant to be transitive. Semantic connotations are not the focus of the resource grammar, though the Japanese grammar is often sensitive to them (e.g. the chapter on the “giving-and-receiving” verbs can be found in any basic Japanese grammar book; see 2.5.3).

Except of the above mentioned nuances, no serious obstacles have been revealed in the application of the Japanese resource grammar. Testing the Japanese resource grammar was an important and final step in the debugging process.
Chapter 4

Conclusion

4.1 Related work on building Japanese computational grammars

The novel thing about GF with respect to a logical framework is that it adds a mapping from abstract terms to concrete linearizations. Previous works on building Japanese computational grammars were not aimed to fit in some common grammatical rules that could be extrapolated to other languages in order to establish compatibility.

Due to this fact, the starting point of building a Japanese grammar was the Japanese linguistics as such, with its peculiarities and distinctions from grammars of the European languages. For example, the Japanese Phrase Structure Grammar [Fujinami 1996] operates with not only categories that are traditional for computational grammars (noun phrase, verb phrase, sentence), but also involves categories that cannot be found in other languages, e.g. the particle category, corresponding to the Japanese particles “wa”, “ga”, “o”, etc. Among the linguistic phenomena under investigation are word order variation, gaps in a sentence and relativization, which are peculiar to Japanese, but not all other languages.

Masuichi et al. (2003) developed a Japanese parsing system with a linguistically fine-grained grammar based on the Lexical-Functional Grammar (LFG) formalism. Basing on the hand-coded grammar, this parser covers over 97% of real-world text. It shows roughly equivalent performance on the dependency accuracy to the standard parsers such as statistics-based Cabocha [Kudo and Matsumoto 2002], yet proposing a more language universal level of analysis. It also provides reasonably accurate results of case detection.

Some Japanese grammars were developed to parse particular kinds of text. For example, the Verbmobil HPSG grammar for Japanese [Siegel 2000] is a robust and efficient grammar for Japanese spoken language. It deals with basic phenomena of Japanese like various nominal and verbal constructions as well as phenomena of spoken Japanese like topicalization, honorification and zero pronouns. This grammar contains a clear concept of segments and phrases, however it cannot be considered universal, being a part of a machine translation system for dialogs.
Another special purpose grammar system is the English to Japanese Medical Speech Translator built on the basis of REGULUS 2 [Rayner 2003]. This prototype system translates spoken yes-no questions about headache symptoms from English to Japanese, using a vocabulary of about 200 words.

4.2 Prospects for future development

- More sensitive treatment to stylistic differentiation.
  In comparison with the European languages, the Japanese grammar has stronger dependence on the pragmatics of the idea that one wants to express. A number of grammatical constructions are predetermined to deliver one and the same meaning, but for different situations. Very often it depends on the speaker and the addressee, their social status, age, gender, etc. It is desirable to continue the coverage of this stylistics-dependent grammar whenever possible in the GF grammar.
- More grammatical forms of verbs and predicative adjectives.
  Only the most essential forms of verbs and predicative adjectives are covered by the Japanese resource grammar. By means of structural words, we avoided using some other grammatical forms that are peculiar to the Japanese language. For example, presumptive, past presumptive, conditional and causative forms were left aside, though they are often used in Japanese natural speech.
- Elliptical constructions.
  The subject (especially a personal pronoun) in a Japanese sentence is often omitted if its meaning is clear from the context. For the moment, the possibility to drop subjects is not provided in the Japanese resource grammar to avoid additional ambiguity, since Japanese verb forms do not reflect person, number, gender or other properties of noun phrases that can be observed in the European languages.
- Idiomatic grammar constructions.
  There are a lot of special grammar constructions that are used in Japanese, but not characteristic of the European languages. For example, special grammar is provided for expressing obligation, advice, prohibition, uncertainty and other connotations. Constructions like these can be stored in the Extra section of the grammar, which is poorly developed at the moment. Although the GF grammar is not meant to deal with semantics, this branch of linguistics is indissolubly connected with the Japanese grammar.

4.3 Results

The main result of the conducted research is building the Japanese grammar as a part of the GF Resource Grammar Library. The grammar covers all the categories and rules of the GF abstract syntax, thus providing the full correlation with the resource grammars of other languages in the GF library.

Testing and evaluation of the developed grammar showed the high percentage of correct English-Japanese translations. At the same time, a number of inconsistencies were revealed, which can be explained by at least two facts:
- Japanese is typologically distant from the European languages;
- the GF abstract syntax is based on the linguistic regularities, some of which are peculiar to the European languages.

Therefore, building of the Japanese resource grammar is also an experiment in the field of language learning and interlanguage comparison. The observed linguistic phenomena made us to draw the following conclusions:
- Japanese has no universal grammar that suits every speech situation. One should operate with at least two styles – plain and honorific. At the same time, a Japanese sentence (non-elliptical) can contain the particle “wa” (topic marker) or particle “ga” (subject marker); the grammar should cover both variants. Thus, the S (sentence) category is finalized by means of the new categories Level and Part introduced in the Extra module.
- In general, Japanese has enough grammatical and semantic resources to cover all the rules in the GF abstract syntax, though it involves some special morphological categories that are out of scope of the GF abstract syntax (e.g. particles), while some other categories are missing in Japanese (e.g. relative pronouns).
- Substantial obstacles in linearization of rules were observed only in one, yet important grammar segment – forming and conjoining of subordinate clauses. Complex subordinate constructions are avoided in the Japanese natural language.
- The necessity to make choices and find compromises in linearizing one or another word or grammatical rule once again proved the conventional principle of translation: one should translate not a word to a word, but a sense to a sense. Following this rule would maximize the correctness of grammar structures even within a restricted domain.

The obtained results can contribute to the discussion on the universal properties of languages and propose to reconsider some rules in the GF abstract syntax.
Bibliography


