

Japanese Sentence Analysis as Argumentation

Akira Shimazu

NTT Basic Research Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180, Japan
shimazu%ntt-20.ntt.jp@relay.cs.net

Abstract

This paper proposes that sentence analysis should be treated as defeasible reasoning, and presents such a treatment for Japanese sentence analyses using an argumentation system by Konolige, which is a formalization of defeasible reasoning, that includes arguments and defeat rules that capture defeasibility.

1 Introduction

Sentence analyses are essentially reasoning processes which derive assumptions/expectations from observed input sentences. A syntactic structure extracted from a sentence by parsing is only a prediction, and may be invalidated by semantic or contextual analyses. This is because interpretation of a sentence requires the use of semantic and contextual analyses to determine its meaning, and because information expressed by an utterance is partial. Furthermore, even when utterances are not grammatical, it is impractical for a parser to reject them because of their ungrammaticality. Therefore, the following two desiderata can be considered for such sentence analyses: to select plausible candidates from among many possible candidates and to integrate, in a uniform manner, syntactic, semantic, and pragmatic processing.

From these viewpoints, this paper proposes that sentence analysis should be treated as defeasible reasoning, and presents such a treatment using an argumentation system [7], which is a formalization of defeasible reasoning, that includes arguments and defeat rules that capture defeasibility. In particular, this paper discusses treatments of chart parsing [5], case analyses, and interpretation of Japanese noun phrases with adnominal particles. Since there is a continuity from syntactic analysis (parsing) to semantic and contextual analyses when viewed as reasoning processes, we use the word analysis rather than parsing.

2 Underlying Frameworks

2.1 Sentence Analysis as Deduction

Mental processes can be viewed as reasoning processes that are invoked by observations of external environments and interactions with other agents. Reasoning has been generally formalized and implemented as deduction frameworks. Even parsing and

generation can be formalized as deduction [12] [15]. This treatment has several advantages, including, in particular, theoretical cleanliness, harmony with semantics and pragmatics, generalization of parsing, gap and unbounded dependency treatments that avoid the addition of specific mechanisms. The deductive formalisms for parsing proposed by Shieber correspond to chart parsing [5]. We describe deduction rules for parsing [15], which satisfy our present requirements for describing sentence analysis and defeat rules. The basic inference rules are prediction and completion.

The inference rule of prediction is as follows.

$$\frac{[a \leftarrow b\gamma, i, j, \alpha] \quad b \rightarrow \beta}{[b \leftarrow \beta, j, j, -]}$$

The inference rule of completion is as follows.

$$\frac{[a \leftarrow b\gamma, i, j, \alpha] \quad [b \leftarrow j, k, \beta]}{[a \leftarrow \gamma, i, k, \alpha b]}$$

Here, $[a \leftarrow \beta, i, j, \alpha]$ represents an edge where i is its starting position, j is its ending position, and where α is analysed. $b \rightarrow \beta$ represents a grammar rule. To be precise, these rules are schemata. In contrast to these rules, grammar rules in DCG themselves function as deduction rules.

2.2 Argumentation System

Many types of common sense reasoning are said to be defeasible; such reasoning involves inferences that are plausible on the basis of current information, but that may be invalidated by new information. Konolige defined a simple and natural system that formalizes such reasoning. This formalization used arguments specified by schemata. He showed how the Yale Shooting Problem and the plan recognition problem can be treated in an intuitively satisfying manner by using the argumentation system ARGH [7], [8]. According to [8], the ARGH is a formal system, in the sense that its elements are formal objects. It is similar in many respects to so-called justification-based Truth Maintenance Systems, but differs in the diversity of argumentation allowed, and in the fact that arguments for a proposition and its negation may coexist without contradiction. Formally, an argument is a relation between a set of propositions (the premises of the argument), and another set of propositions (the conclusion of the arguments). Argumentation is started with an initial set

of facts. Then, argument schemata are used to construct plausible arguments. The process of deciding which arguments are valid is carried out using defeat rules. Although there are other formalizations for defeasible reasoning, such as abduction [1], [3], since our main concern is to clarify inferences in sentence analysis and the relations between them, we use the argumentation system here, without consideration of the alternatives.

3 Sentence Analysis as Argumentation

Sentence analysis is comprised of reasoning processes which derive assumptions/expectations from observed input sentences. From such a viewpoint, sentence analysis is really abduction rather than deduction:

$$\text{Background} + \text{Assumption} \vdash \text{sentence}$$

Therefore, various decisions pertaining to the assumption expectations are carried out in the sentence analysis processes. These decisions may be invalidated later in the processes as the analysis becomes further elaborated. The basic decisions are performed, when syntactic structures and semantic structures (logical forms) are extracted along with their contextual analyses. The important point here is that we can view these analysis processes as decisions in a defeasible reasoning process; in this paper, we model this point with an argumentation process. Basic arguments in analysis and related defeat rules are presented in the following.

3.1 Chart Parsing as Argumentation

Based on the framework that defines chart parsing as deduction, we define arguments corresponding to fundamental rules of top-down chart parsing, prediction, and completion steps, as follows.

$$\begin{aligned} & \text{member}([a \leftarrow b\gamma, i, j, \alpha], \text{Agenda}), \\ & \text{rule}(b \rightarrow \beta) |s_n \xrightarrow{\text{predict}} \\ & \text{member}([b \leftarrow \beta, j, j, -], \text{Agenda}) |s_{n+1} \end{aligned}$$

Here, *Chart* and *Agenda* respectively denote a chart and an agenda list as in usual implementations of chart parsing. Lower case roman and Greek letters indicate schema variables.

$$\begin{aligned} & \text{member}([a \leftarrow b\gamma, i, j, \alpha], \text{Agenda}), \\ & \text{member}([b \leftarrow j, k, \beta], \text{Chart}) |s_n \xrightarrow{\text{complete}} \\ & \text{member}([a \leftarrow \gamma, i, k, \alpha b], \text{Agenda}) |s_{n+1} \end{aligned}$$

This is for cases where an inactive edge $[b \leftarrow j, k, \beta]$ is in *Chart*. Cases where the inactive edge is in *Agenda* are described similarly. Both of the above arguments may be satisfied when applicable.

Since, in a chart parsing algorithm, an edge from the agenda must be removed and added to the chart when the above arguments are applied, the following argument is necessary.

$$\begin{aligned} & \text{member}([a \leftarrow \beta, i, j, \alpha], \text{Agenda}) |s_n \xrightarrow{\text{addition}} \\ & \neg \text{member}([a \leftarrow \beta, i, j, \alpha], \text{Agenda}), \\ & \text{member}([a \leftarrow \beta, i, j, \alpha], \text{Chart}) |s_{n+1} \end{aligned}$$

Here, we assume that propositions continue to hold unless they are denied. That is,

$$\varphi |s_n \xrightarrow{\text{persist}} \varphi |s_{n+1}$$

A *subsume argument* is necessary to keep edges nonredundant. This is one characteristic of chart parsing.

$$\begin{aligned} & \text{member}(\phi, \text{Agenda}), \text{already-in}(\phi, \text{Chart}) |s_n \\ & \xrightarrow{\text{subsume}} \neg \text{member}(\phi, \text{Agenda}) |s_{n+1} \end{aligned}$$

Only when the *subsume argument* does not hold, is the *prediction* or *completion argument* permitted. Therefore, the following defeat rule is necessary.

When both a *subsume argument* and a *prediction/completion argument* are possible, the former defeats the latter.

One of the important characteristics of chart parsing is that it can control the order of parsing processes, that is, the order of edge selections from the agenda. This aspect is suited for treating defeasible reasoning. To incorporate such control, we modify the *prediction* and *completion arguments*. First, we select an edge from *Agenda* and put it on a list called *Cache*. Then, we apply the *prediction* and *completion arguments* to the edge in *Cache*, and add the resulting edges into *Agenda*. The selection argument is as follows.

$$\begin{aligned} & \text{member}(\phi, \text{Agenda}) |s_n \xrightarrow{\text{select}} \\ & \neg \text{member}(\phi, \text{Agenda}), \\ & \text{member}(\phi, \text{Cache}) |s_{n+1} \end{aligned}$$

The *edge addition argument* is modified by replacing *Agenda* with *Cache*.

Several *selection arguments* are generally possible because of plural edges in *Agenda*. Selections are classified according to types of edges. The following is classification of *selection arguments* based on types of edges in the premise of the arguments [15].

$$\begin{aligned} \text{prediction-type: } & \phi = [a \leftarrow \beta, i, i, -] \\ \text{active-type: } & \phi = [a \leftarrow \gamma, i, j, \beta] \\ \text{inactive-type: } & \phi = [a \leftarrow i, k, \beta] \\ \text{lexical-type: } & \phi = \text{lexical inactive edge} \end{aligned}$$

where we list only ϕ in *member*(ϕ , *Agenda*) instead of listing the whole *selection argument*.

3.2 Simulation of LR Parsing for English

For selections of instances of *selection argument*, that is, selection of edges from the agenda, we have the following defeat rules based on [15], which guide the parser to determine an appropriate syntactic structure of English sentences as the first solution. The deductive parser by [15] simulates LR parsing, which reflects right association and minimal attachment readings.

- (1) If there is more than one possible argument, prediction-types defeat lexical-types, which defeat active-types, which defeat passive-types.

- (2) If (1) does not fully order possible arguments, arguments with items ending farther to the right defeat the others.
- (3) If (1) and (2) together do not fully order possible arguments, arguments with items constructed from the instantiation of longer rules defeat the others.

Shieber devised the above preferences based on correspondences between an LR parser and a chart parser, and on preferences of shift/reduce and reduce/reduce conflicts in English [13].

4 Japanese Sentence Analysis

4.1 Simulation of LR Parsing for Japanese

For Japanese sentences, however, the above defeat rules are inappropriate. Japanese sentences have the following characteristics.

- When we read/hear a Japanese sentence from left to right, we usually relate the word/phrase just read to the next word.
- A Japanese sentence generally have a recursive structure derived by a rule *modifier + head* [2].

These two characteristics result in a tendency for Japanese sentences to have left branching structures like $[[[[[[[neko\ ga]\ oikaketa]\ nezumi]\ ga]\ tabeta]\ sakana]\ wa]\ shinsendatta]$ (The fish that the rat that the cat chased ate was fresh.) [9]. We can capture the left branching characteristics by the strategy of *reduce* preference when *shift/reduce* conflicts occur against Shieber's strategy. Here, these arguments do not mean that a Japanese sentence always has a left branching structure, but they do mean that the preferable reading tend to results in the left branching structure, provided that linguistic constraints are satisfied. For example, for *Taro ga kouen ni iku* ('Taro Subj park Goal go. "Taro goes to a park."), the structure is $[[[Taro\ ga]_{Pp}\ [[kouen\ ni]_{Pp}\ iku]_V]_S]_S$, and is not left branching, since *Taro ga* is not related to *kouen*. In this case, we try to combine *Taro ga* with *Kouen*, and since a relation between *Taro ga* and *Kouen* does not hold, the above structure is tried.

To simulate the strategy of *reduce* preference when *shift/reduce* conflicts occur, the following three rules in addition to (1) replace rules (2) and (3) for a proper treatment of Japanese.

- (4) If (1) does not fully order possible arguments, arguments with longer items defeat the others. (Length is defined as ending position minus starting position.)
- (5) If (1) and (4) together do not fully order possible arguments, arguments with items starting farther to the left defeat the others.
- (6) If (1), (4) and (5) together do not fully order possible arguments, arguments with items pushed into the agenda earlier defeat the others.

Rules (4) and (5) are based on the preference for left branching structures. Because these preferences tend to prevent the parser from proceeding to the right, rule (6) is necessary for longer phrases. These rules have been tested for basic sentences [17], some of which are syntactically ambiguous. For example, there are many Japanese noun phrases that have the following pattern.

$$\begin{array}{l} N_1\ no\ N_2\ no\ \dots\ no\ N_k \\ N_1\ poss\ N_2\ poss\ \dots\ poss\ N_k \\ N_k\ of\ N_{k-1}\ of\ \dots\ of\ N_1 \end{array}$$

Generally there can be $2^{n-1}(2n-3)!!/n!$ possibilities for this noun phrase, which is computed by dependency combinations like $((((N_1\ no\ N_2)\ no)\ \dots\ no)N_n)$.

4.2 Feature Incorporation

Contemporary parsing technology is based on complex feature structures. Chart parsing uses such linguistic constraints presented by features when completion and/or prediction steps are applied as in [14]. Accordingly, for example, a completion argument for cases where an inactive edge is in the chart is as follows.

$$\begin{array}{l} member([a \leftarrow b\gamma, i, j, \alpha, e], Cache), \\ member([b \leftarrow, j, k, \beta, f], Chart), \\ unify(e, [b : f], g) |_{s_n} \xrightarrow{\text{complete}} \\ member([a \leftarrow \gamma, i, k, \alpha\beta, g], Agenda) |_{s_{n+1}} \end{array}$$

where e, f and g are feature structures, and $unify(x, y, z)$ means that z is the result of unifying x and y .

Feature structures uniformly represent various linguistic constraints such as subcategorizations, gaps, unbounded dependencies, and logical forms. A problem of this representation scheme is that it describes all possible constraints in one structure and deals with them at once. This is inefficient with many copy operations due to unifications of unnecessary features that do not contribute to successful unification [6]. Thus treatments such as strategic unification [6] have been developed.

It seems that a preferable approach is to treat linguistic constraints piecewise, taking into consideration abductivity of parsing, uniform integration of various linguistic processings, and the problem of a unification-based approach. From this point of view, we describe such treatments as, especially, incorporation of word properties, case analyses, composition of logical forms, and interpretation of noun phrases with adnominal particles. This section discusses the incorporation of word properties, and the following section the others.

Word properties are incorporated using lexical arguments when a lexical edge is in *Cache*. For example, semantic categories of *Taro* (boy's name) are incorporated using the following lexical argument.

$$\begin{array}{l} member([Np \leftarrow, 0, 1, Taro, X], Cache), \\ scategory(Taro, Animate) |_{s_n} \\ \xrightarrow{\text{nounprop}} scategory(X, Animate) |_{s_{n+1}} \end{array}$$

where the edge representation is redefined adding the identifier X for the edge. $category(x, c)$ means that x 's semantic category is c .

Likewise, proposition $particle(x, p)$ is introduced for edge $[P \leftarrow, i, j, p, x]$ corresponding to a particle.

Properties of constituents are generally propagated to their mother. For example, since the above *Taro* and *ga* (subject case particle) are combined to make a postpositional phrase (Pp), their properties are propagated to the postpositional phrase, and used for case analyses.

$$\begin{aligned} &member([Pp \leftarrow, i, j, Np P, x y z], Cache), \\ &category(y, c), particle(z, p) |s_n \\ &\xrightarrow{ppprop} ppcategory(x, p, c) |s_{n+1} \end{aligned}$$

where $ppcategory(x, p, c)$ means that postpositional phrase Pp identified as x has particle p , usually a case particle, and semantic category c .

A subcategorization frame for a verb is introduced as follows ¹.

$$\begin{aligned} &member([S \leftarrow, j, k, v, x], Cache), \\ &subcat(v, role, p, c) |s_n \\ &\xrightarrow{subcatprop} subcat(x, role, p, c) |s_{n+1} \end{aligned}$$

where $subcat(v, role, p, c)$ means that verb v subcategorizes for a postpositional phrase with particle p and semantic category c . For example, $subcat(X, Subj, Ga, Animate)$ is introduced for edge $[S \leftarrow, 2, 3, asobu, X]$ corresponding to verb *asobu* (play). This is an argument for an intransitive verb. Here, for simplicity, we use the intransitive case. Arguments for plural case roles can be represented in a similar manner by just adding extra $subcat$ predicates for the other case roles like $subcat(v, role_2, p_2, c_2)$.

Like the property propagation of postpositional phrases, when the above edge $[S \rightarrow, j, k, v, z]$ is combined with active edge $[S \rightarrow S, i, j, Pp, x y z]$, a subcategorization frame is propagated for later use, as follows.

$$\begin{aligned} &member([S \leftarrow, i, j, Pp S, x y z], Cache), \\ &member([S \leftarrow, j, k, v, z], Chart), \\ &subcat(z, role, p, c) |s_n \\ &\xrightarrow{subcatprop} subcat(x, role, p, c) |s_{n+1} \end{aligned}$$

4.3 Case Analysis Arguments

Two important characteristics of Japanese sentences are that it exhibit free word order, and that it has zero pronouns, i.e., subjects or objects which are not explicitly expressed, but are supplied from the context. Accordingly, case particles and semantic categories of head nouns are necessary to analyze relations between postpositional phrases (Pp) and verbs (v). In some cases, only modal particles are used instead of case particles [11]. Therefore, semantic categories are important for subcategorization or case analysis. These characteristics of Japanese inevitably necessitate defeat rules for practical analyses.

¹Here, we assume that a verb itself can be a Japanese sentence, and use Japanese grammar rules including $S \rightarrow v$, and $S \rightarrow PpS$ [17].

Two basic arguments of case analysis are a rule for obligatory cases (subcategorization) and a rule for optional cases (adjunction).

Subcategorization

The argument for obligatory case analysis is as follows.

$$\begin{aligned} &member([S \leftarrow, Pp S, i, j, x y z], Cache), \\ &subcat(z, role, p, c), ppcategory(y, p, c) |s_n \\ &\xrightarrow{subcat} relation(z, y, role) |s_{n+1} \end{aligned}$$

where $relation(z, y, role)$ means that the postpositional phrase y is the case $role$ of phrase z . For example, when there is $ppcategory(Y, Ga, Animate)$ corresponding to postpositional phrase Pp with identifier Y , and there is $subcat(Z, Subj, Ga, Animate)$ corresponding to sentence S with identifier Z , we get $relation(X, Y, Subj)$.

Adjunction

The argument for optional case analysis is as follows.

$$\begin{aligned} &member([S \leftarrow, Pp S, i, j, x y z], Cache), \\ &adjunction(y, role, p, c), ppcategory(y, p, c) |s_n \\ &\xrightarrow{adjunction} relation(z, y, role) |s_{n+1} \end{aligned}$$

where $adjunction(y, role, p, c)$ means that postpositional phrase y modifies sentence z in the relation $role$ when y has the postposition particle p and the semantic category c . ² The property $adjunction(y, role, p, c)$ is introduced for particles or adverbial nouns.

No case relation holds when the above arguments do not hold, which is represented by the following argument.

$$\begin{aligned} &member([S \leftarrow, Pp S, i, j, x y z], Cache), \\ &subcat(z, role, p, c) |s_n \\ &\xrightarrow{crfailure} \neg relation(z, y, role) |s_{n+1} \end{aligned}$$

There is a similar argument for an adjunct case. The above argument always holds when it is applicable, but it should be defeated when the *subcategorization* or *adjunction argument* holds. Thus, we have the following defeat rule.

If a *subcategorization* or *adjunction argument* holds, the *case relation failure argument* is defeated.

When a *case relation failure argument* holds, it is preferable to retract the premise edge which triggered case relation analyses. This is represented by the following argument.

$$\begin{aligned} &member([S \leftarrow, Pp S, i, j, x y z], Cache), \\ &\neg relation(z, y, role) |s_n \xrightarrow{retract} \\ &\neg member([S \leftarrow, Pp S, i, j, x y z], Cache) |x \end{aligned}$$

²Strictly speaking, there are correlations between types of adjunctive phrases (Pp) and types of sentences (S) [10]. Here, we do not represent such correlations for simplicity.

Composition of Logical Forms

Like case analyses, composition of logical forms is treated as follows.

$$\begin{aligned} & \text{member}([S \leftarrow, i, j, Pp\ S, x\ y\ z], \text{Cache}), \\ & \text{lf}(z, p(a')), \text{lf}(a, a'), \text{relation}(z, a, r) \mid s_n \\ & \xrightarrow{\text{lf comp}} \text{lf}(x, p(a')) \mid s_{n+1} \end{aligned}$$

This is an argument for an intransitive verb where $\text{lf}(x, x')$ is introduced by lexical edge introductions, and means that the logical form of the constituent x is x' . The premise predicates of this argument are satisfied provided that instances of $\text{relation}(z, y, \text{role})$ and $\text{lf}(y, y')$ hold. For the case of *Taro ga asobu* (“Taro subj-case play, “Taro plays”), $\text{lf}(X, \text{play}(\text{Taro}))$ holds when $\text{lf}(Z, \text{play}(a'))$, $\text{lf}(Y, \text{Taro})$, and $\text{relation}(Z, Y, \text{Subj})$ hold.

4.4 Plausible Case Analysis

The above two rules result in the possibility that a given Pp may fill both obligatory and optional cases. On the other hand, the requirements $\text{subcat}(y, \text{role}, p, c)$, $\text{adjunction}(y, \text{role}, p, c)$, and $\text{ppcategory}(y, p, c)$ in the above rules are too strict for practical linguistic processings, since there are noun phrases with modal particles, no particles, and no strict category matches. Therefore, we relax the requirement $\text{ppcategory}(y, p, c)$ replacing it with one of the following conditions. That is, if some of the arguments having the following conditions hold, a given Pp can fill the corresponding case roles.

- (a) $\text{ppcategory}(y, p, c)$,
- (b) $\text{ppcategory}(y, p, c'), \text{isa}(c', c)$,
- (c) $\text{ppcategory}(y, p, c'), \neg \text{isa}(c', c)$,
- (d) $\text{ppcategory}(y, p', c), m(p')$,
- (e) $\text{ppcategory}(y, p', c'), m(p'), \text{isa}(c', c)$,
- (f) $\text{ppcategory}(y, p', c'), m(p'), \neg \text{isa}(c', c)$,

where $\text{isa}(c', c)$ means that c is a super semantic category of c' , and $m(p')$ means that p' is a modal particle.

Thus, when we replace the requirement condition in the two arguments given above with conditions (a) – (f), we obtain twelve arguments for case analysis. This results in the possibility that some constituent may be analyzed as filling more than one possible case role. Therefore, we need defeat rules to select the appropriate case analysis argument. The following are two basic defeat rules.

- (1) Generally, the strength order is (a) > (b) > (c) > (d) > (e) > (f) except for the following condition (2). (c) and (f) do not hold for optional cases.
- (2) If both obligatory and optional cases fill (a) or (b), the obligatory case defeats the optional case. That is, $(a)_{ob} > (b)_{ob} > (a)_{op} > (b)_{op}$.

The fact that (c) and (f) cannot be satisfied by optional cases means that semantics is important when optional information is expressed. Rule (2) means

that syntax is important when case particles are expressed explicitly.

For the sentence

Watashi mo non-da.
I modal-particle drink-past.
I drank (something), too.

an argument using (d) concludes that *watashi mo* is the subject, while one using (f) concludes that it is the object. As (d) defeats (f), *watashi mo* is determined to be the subject. For the sentence

Budoushu mo non-da.
wine modal-particle drink-past.
(Someone) drank wine, too.

the reverse conditions hold.

For noun phrases with relative clauses constructed by $Np \rightarrow S\ Np$, the Np on the right of S may be a case element of S . In such cases, we use property $\text{ppcategory}(x, p, c)$ with variable p , which is not instantiated when applied, and it is assumed that only (a) and (b) hold.

4.5 Interpretation of Japanese Noun Phrase *A no B*

Integration of syntactic, semantic, and pragmatic processings is an interesting and complex problem [3], and the treatment by the argumentation framework is a promising approach to this problem. As for such a problem, interpretation of Japanese noun phrase patterns of the type *A no B*, which abound in Japanese [16], is a good testbed.

A no B, which consists of two nouns A and B with an adnominal particle *no*, and which has at least the same ambiguity as *B of A*, is generally interpreted by assuming an appropriate predicate [16]. For example, *densha no mado* (a window of a train) is interpreted as *densha* (train) *ni* (Loc) *aru* (be) *mado* (window), supplementing a verb *aru* (be). *A no B* is generally ambiguous when taken out of context as *Hanako no e* (“the picture of Hanako” or “Hanako’s picture”) with a range of possible semantic relations including possession, producer, purchase, and content.

We can interpret semantic relations of *A no B* by using arguments in a similar way as before. For example, from the following sentence

Hanako wa e o kakimasu.
Hanako paints a picture.

the propositions $\text{lf}(X_1, \text{Paint}(\text{Hanako}, O))$ and $\text{lf}(X_2, \text{Picture}(O))$ hold. In this context, we can interpret an *A no B* relation of the following sentence

Kono Hanako no e wa kireida.
This picture of Hanako is beautiful.

For the second sentence, the $\text{relation}(Y, z, \text{No})$ and $\text{lf}(Y, \text{Hanako})$ hold for an edge corresponding to Pp (*Hanako no*), and $\text{lf}(Z, \text{Picture}(O))$, $\text{lf}(Z, p(a', O))$, $\text{lf}(a, a')$, $\text{relation}(a, Z, \text{No})$ for an edge Np (*e*). Then we have propositions $\text{relation}(Y, Z, \text{No})$ and $\text{lf}(X, p(\text{Hanako}, O))$ based on the following argument.

$$\begin{aligned} & \text{member}([Np \leftarrow, Pp \ Np, i, k, x \ y \ z], \text{Cache}), \\ & \text{lf}(z, p(y', z')), \text{relation}(y, z, No), \text{lf}(y, y') \mid s_n \\ & \xrightarrow{\text{AnoB}} \text{lf}(x, p(y', z')) \mid s_{n+1} \end{aligned}$$

Finally, we get *Paint(Hanako, O)* using the following argument,

$$\begin{aligned} & \text{relation}(y, z, No), \text{lf}(z, p(a_1, a_2)), \\ & \text{lf}(c, q(a_1, a_2)) \mid s_n \\ & \xrightarrow{\text{int-no}} \text{lf}(z, q(a_1, a_2)) \mid s_{n+1} \end{aligned}$$

and thereby complement the meaning of *Hanako no e* by extrapolating the verb *Paint*.

If it is learned that Hanako in fact bought the picture, and not painted it, the final interpretation is defeated using the same framework.

5 Conclusion

We have presented an argumentation-based model of Japanese sentence analysis, which is essentially abductive. We believe that this model is well suited for sentence analyses including various linguistic processings under conditions where information expressed by utterances is partial and its interpretation depends on context, for the following reason. Since the argumentation system is incremental and has the ability to cope with resource limitations [8], the analysis systems based on this argumentation system can return an appropriate decision that has been derived to that point.

The original heuristics to which arguments and defeat rules are formally described have been tested with about a thousand sentences over a period of more than five years. For case analysis, arguments and defeat rules that handle zero pronouns [4] could be introduced, thereby making reasoning about case analysis much more precise. Generally speaking, defeat rules for case analyses are based on the idea that, for new information, syntactic constraints are preferred, and, for old information, semantic and pragmatic constraints preferred. Finally, arguments such as those presented by [8] will also be necessary. Such arguments should be integrated with the arguments described in this paper.

Acknowledgment

I would like to thank Douglas Appelt, Jerry Hobbs, Yasuhiro Katagiri, Masahito Kawamori, Kiyoshi Kogure, Kurt Konolige, Shozo Naito, Martha Pollack, and Ikuo Takeuchi for their discussion, comments, and improvements on this research and paper.

References

- [1] D. E. Appelt. Weighted abduction as an inference method for plan recognition and evaluation. In *Proceedings of the Second International Workshop on User Modeling*, 1990.
- [2] T. Gunji. *Japanese Phrase Structure Grammar*. Reidel, 1987.
- [3] J. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, 1988.
- [4] M. Kameyama. A property-sharing constraint in centering. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, 1986.
- [5] M. Kay. *Algorithm Schemata and Data Structures in Syntactic Processing*. Technical Report CSL-80-12, Xerox PARC, 1980.
- [6] K. Kogure. Strategic lazy incremental copy graph unification. In *Proceeding of the 13th International Conference on Computational Linguistics*, 1990.
- [7] K. Konolige. Defeasible argumentation in reasoning about events. In *Proceeding of the International Symposium on Machine Intelligence and Systems*, Torino, Italy, 1988.
- [8] K. Konolige and M. E. Pollack. Ascribing plans to agents. In *Proceeding of the 11th International Joint Conference on Artificial Intelligence*, 1989.
- [9] S. Kuno. *Japanese Grammar* (in Japanese). Taishu-kan, 1973.
- [10] F. Minami. *Structure of Contemporary Japanese* (in Japanese). Taishu-kan, 1986.
- [11] S. Naito, A. Shimazu, and H. Nomura. Classification of modality function and its application to Japanese language analysis. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, 1985.
- [12] F. N. Pereira and D. H. D. Warren. Parsing as deduction. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983.
- [13] S. M. Shieber. Sentence disambiguation by shift-reduce parsing technique. In *Proceeding of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983.
- [14] S. M. Shieber. Using restriction to extend parsing algorithms for complex-feature-based formalisms. In *Proceeding of the 23rd Annual Meeting of the Association for Computational Linguistics*, 1985.
- [15] S. M. Shieber. A Uniform architecture for Parsing and Generation. In *Proceeding of the 12th International Conference on Computational Linguistics*, 1988.
- [16] A. Shimazu, S. Naito, and H. Nomura. Semantic structure analysis of Japanese noun phrases with adnominal particles. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 1987.
- [17] A. Shimazu and S. Naito. Preference reading models of Japanese sentences (in Japanese). In *Technical Reports NL-73-9 of Natural Language Processing Special Interest Group of Information Processing Society of Japan*, 1989.