Handling Pragmatic Information With A Reversible Architecture

Masato Ishizaki

NTT Communications and Information Processing Laboratories
1-2356, Take, Yokosuka, Kanagawa, 238-03 Japan
E-mail: ishizaki%nttnly.ntt.jp@relay.cs.net

Abstract

This paper proposes a reversible architecture to handle not only syntactic and semantic information but also pragmatic information. Existing architectures cannot represent pragmatic information explicitly, and lack reasoning capability given insufficient information. I argue that the techniques of plan representation and approximate reasoning are, in the enhanced argumentation system proposed here, effective for solving these problems.

1. Introduction

Reversibility or bi-directionality of grammars seems to play a quite important role in natural language processing. It reduces the cost of constructing a grammar; we need to use only one grammar instead of two for parsing and generation. Cost here includes not only the making of grammar rules but also verifying the rules and the algorithms for parsing and generation. Reversibility differs from bi-directionality: the former requires the same mechanism and grammar for parsing and generation; the latter requires just the same grammar as shown Figure 1(Noord,1990).

Pragmatic information is not rigidly defined; rather it is thought of as information other than syntactic and semantic information. It is indispensable for explaining many linguistic phenomena from cleft sentences to discourse structures. As pragmatics cannot restrain language in a strict manner like syntax, it must be processed differently; that is, a distinction must be made between constraints that need to be fully satisfied and those that do not. Plan representation seems to be appropriate for collecting different level information. A plan consists of preconditions, constraints, plan expansion (usually termed the body), and effects. The relationship between preconditions and constraints parallels that between pragmatic and syntactic information. Thus, the difference between preconditions and constraints can be easily modeled.

Handling pragmatic information clearly depends on assumption of belief: Generating referring expressions requires inferencing the hearer's belief (Appelt,1985); Producing text requires the usage of a one-sided mutual belief[1](Moore et al.,1989); the listener's inference about the speaker's belief greatly helps to resolve anaphora or to analyze the speaker's intention. In any case, belief becomes a condition for further inference; however, it is difficult if not impossible to confirm the assumed belief. Thus, a new mechanism based on a new architecture is needed. Approximate reasoning (Elkan,1990) is suitable for this purpose. Processing can continue, even if some preconditions are not fully satisfied; they are held as assumptions[2]. This approach seems to be very natural . For example, in conversations, the speaker should conceptualize the listener's

[1] One-sided mutual belief is one half of mutual knowledge, so to speak, namely the set of those pieces of mutual knowledge that constitute the knowledge of one speaker (Bunt,1989: 60).

[2] Since approximate reasoning can fail, assumptions must be held explicitly for further inference. Plan representation is adequate for that reason.

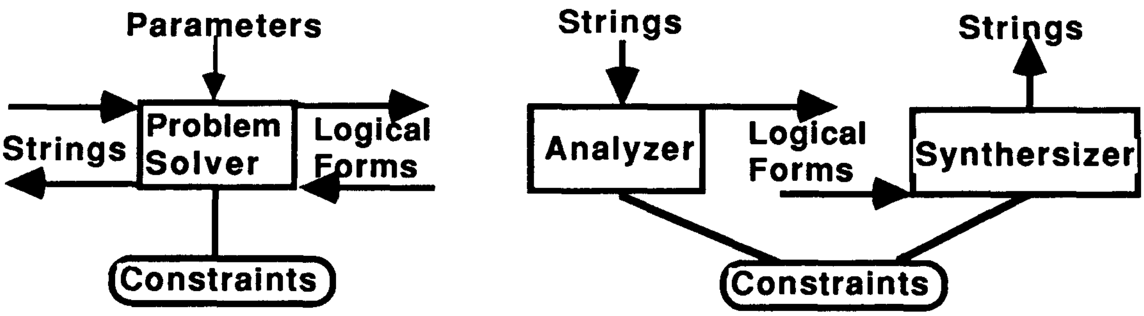

Figure 1. Reversible And Bi-directional Architectures.

understanding. In most conversations, however, the speaker does not keep confirming the other's belief because this would disrupt the conversation flow.

This paper describes a reversible architecture to handle not only syntactic and semantic information, but also pragmatic information. Existing architectures cannot represent pragmatic information explicitly, and lack reasoning capability given insufficient information. I argue that the techniques of plan representation and approximate reasoning in the argumentation system introduced here are effective for solving these problems.

First, the difficulties of existing architectures have in handling pragmatic information are described. Next, plan representation of linguistic information and approximate reasoning are mentioned in the context of the argumentation system. Third, parsing and generation examples are shown. Finally, the problem of proposed data structure, the decomposition of semantic representation, the role of syntactic information and the difference between active and passive vocabulary are discussed.

2. Existing Architectures For Reversible And Bi-directional Grammar

Shieber proposed a uniform architecture for sentence parsing and generation based on the Early type deduction mechanism (Shieber,1988). He parametrized the architecture with *the initial condition, a priority function on lemmas,* and *a predicate expressing the concept of successful proof.* Shieber remedied the inefficiency of the generation algorithm in his uniform architecture to introduce the concept of semantic head (Shieber et al.,1989). Although Definite Clause Grammar (DCG) is reversible, its synthesis mode is inefficient. Dymetman and Strzalkowski approached the problem by compiling DCG into efficient analysis and synthesis programs (Dymetman et al.,1988) (Strzalkowski,1990). The compilation is realized by changing goal ordering statically. Since Shieber's, Dymetman's and Strzalkowski's architectures are based on syntax deduction, they have difficulties in handling pragmatic information.

Dependency propagation was suggested for parsing and generation in (Hasida et al.,1987). His idea was developed using horn clauses similar to PROLOG. The word *dependency* indicates the states where variables are shared by constraints[3]. Problem solving or parsing and generation can be modeled by resolving the dependencies. Dependency resolution is executed by fold/unfold transformations. Dependency propagation is a very elegant mechanism for problem solving, but it seems to be difficult to represent syntactic, semantic and pragmatic information with indiscrete constraints. In addition, to that, since dependency propagation is a kind of co-routine process, programs are very hard to debug and so constraints are tough to stipulate.

Ait-Kaci's typed unification was applied to a reversible architecture (Emele and Zajac ,1990). All features in sign are sorted and placed in hierarchical structures. Parsing and generation can be executed by rewriting the

[3] Constraints are represented by the usual PROLOG predicates.

features into their most specific forms. Their mechanism greatly depends on the hierarchy of information, but with information other than syntactic, especially pragmatic, it is hard to construct the hierarchy.

3. Introduction To the New Reversible Architecture

3.1. The Linguistic Objects

We introduce the linguistic object *sign* which incorporates syntactic, semantic and pragmatic information. *Sign* is represented by feature structures and consists of features *phn, syn, sem* and *prag*. *Phn* represents surface string information for words, phrases and sentences. *Syn* stands for syntactic information like the part of speech and subcategorization information using HPSG. HPSG inherits the fundamental properties of Generalized Phrase Structure Grammar(GPSG). That is, HPSG makes use of a set of feature-value pairs, feature constraints and unification to stipulate grammar instead of rewriting rules for terminal and nonterminal symbols. The major difference between HPSG and GPSG is that subcategorization information is stored in lexical entries, instead of being stored in grammar rules (Pollard et al.,1987,1990). *Sem* denotes semantic information or logical forms. Logical forms are expressed by the semantic representation language proposed by Gazdar (Gazdar et al,1989). Since the language is a feature representation of Woods' representation (Woods,1978), it has the advantages that it can represent quantifier scope ambiguities. It consists of the features *qnt*, *var*, *rest*, and *body* : *qnt* is for quantifier expressions; *var* is for variables bound by the *qnt*; *rest* is for restrictions of the *var* ; while *body* represents the predication of the logical form. *Prag* delineates pragmatic information. Pragmatic conditions are not necessarily true but are held as assumptions. Uniqueness and novelty conditions in cleft sentences are instances of the conditions.

```
phn: "It was the girl that a boy  loved"
syn: pos:  verb
     subcat:subc([])
sem: qnt: indefinite
     var: X
     rest: arg0: X
           pred: BOY
     body: qnt: definite
           var: Y
           rest: arg0: Y
                 pred: GIRL
           body: arg0: X
                 arg1: Y
                 pred: LOVED
prag: [novel(Y), unique(Y)]
Sign
```

Figure 2. Linguistic Object Example.

Figure 2 shows an example of the linguistic object *sign*. The feature *phn* indicates that surface string is "It was a girl that the boy loved". *Syn* represents that: 1) the part of speech is verb; 2) subcategorization information is satisfied. *Sem* shows that: 1) the quantifier at the top level is indefinite; 2) the property of the variable X is a boy; 3) the property of the variable Y bounded by the quantifier definite is a girl; 4) the boy loved the girl. *Prag* mentions that the variable Y is constrained with uniquness and novelty conditions.

3.2. The Plan Representation of Linguistic Objects

To handle syntactic , semantic, and pragmatic information, our generator represents them as plans. Plans are composed of preconditions, constraints, plan expansion, and effects. Preconditions include pragmatic information which are the criteria needed to select a plan. Constraints include syntactic conditions such as the head feature principle and conditions on surface strings. Plan expansion contains sub-semantic expressions for effects, which are complete semantic expressions. Constraints and preconditions are similar, but differ in that the former must be satisfied, but the latter are retained as assumptions if not satisfied.

Figure 3 describes a plan relating to the semantic information LOVED. No preconditions exists because the expression "loved" has no pragmatic information. Constraints indicate that: 1) the part of speech equals verb; 2) the subcategorization information is subc([Sbj,Obj]); 3) The *sem* features of Sbj and Obj are semantic arguments of predicate LOVED; 4) the surface string is "loved". There is no plan expansion because lexical information does not need to be expanded. Effects mention semantic expression of LOVED.

```
precond: []
const: (Sign:syn:pos = verb),
       (Sign:syn:subcat = subc([Sbj,Obj])),
       (Arg0 = Sbj:sem), (Arg1 = Obj:sem),
       (Sign:phn = "loved")
eplan: []
effect: arg0: Arg0
        arg1: Arg1
        pred: LOVED
```

Figure 3. Plan Example.

3.3. An Argumentation System For Planning

A plan recognition scheme, named the argumentation system, was proposed by Konolige and Pollack (Konolige and Pollack ,1989). It can defeasibly reason about belief and intention ascription[4], and can process preferences over candidate ascriptions. The framework is so general and powerful that it can perform other processes other than belief and intention ascription. For example, Shimazu has shown that it can model parsing mechanism

The argumentation system consists of arguments. An argument is a relation between a set of propositions (the premises of the argument), and another set of propositions (the conclusion of the argument) (Konolige and Pollack,1989: 926). The system has a language containing the following operators: t(P) which indicates the truth of the proposition P; bel(A,PF) which mentions that agent A believes plan fragment PF; int(A,PF) which shows that agent A intends plan fragment PF; exp(A,P) which means that agent A expects proposition P to be true; and by(Actexp1,Actexp2,Pexp) which signifies the complex plan fragment which consists of the action expression Actexp2, by doing action expression Actexp1 while propositional expression Pexp is true[5]. The arguments to the operators, action expressions inform(S,H,Sem) and utter(S,H,Str), are introduced to mimic informing and uttering activities: the former is designated such that speaker S informs hearer H about semantic content Sem; the latter indicates speaker S utters string Str to hearer H.

Plan expansion, effects and constraints mentioned in subsection 3.2 correspond to Actexp1, Actexp2 and Pexp, respectively. To represent the difference between preconditions and constraints, the operator by is revised to include preconditions as the fourth argument. Thus, the new operator by(Actexp1,Actexp2,Pexp1,Pexp2) is defined as the complex plan fragment, consisting of doing Actexp2 (effect(s)), by doing Actexp1 (plan expansion) while Pexp1 (constraints) is true, and Pexp2 (precondition(s)) is true or held as assumptions. The plan in figure 2 was redefined by using axiom (1)[6].

Axiom (2) shows another example cor-

[4] Defeasible reasoning and approximate reasoning are very similar, but differ in that: the former addresses the result after rule application; the latter considers just rule application.

[5] Action expressions are formed from an action name and parameters; Propositional expressions are formed from a property name and parameters.

[6] Because of space limitations, abbreviations are used as necessary. For example, Pos is taken to mean the value of the features of part _of_speech of syntactic information of sign.

responding to the context free grammar for a cleft sentence.

Axiom (1):

```
t(by(utter(S,H,"loved"),
      inform(S,H,LOVED),
      ((Pos=verb),
       (Subcat=subc([Sbj,Obj])),
       (Sbj:sem=Arg0),
       (Obj:sem=Arg1)),
      ())
```

Axiom (2):

```
t(by((inform(S,H,LF1[7]),
      inform(S,H,LF2[8])),
     inform(S,H,LF[9]),
     ((Pos=Pos2),(Sign1=Sbj),
      (Subcat=subc([])),
      (Slash=sl([])),
      (Slash2=sl([Obj])),
      (Phn="It was"+Phn1+"that"+Phn2)),
     (Prag))).
```

Plan expansion and effects indicate that if speaker S wants to inform hearer H about LF, the speaker should inform the hearer about LF1 and LF2, while observing constraints 1) -4). Constraints state that: 1) the part of speech of Sign equals one of Sign2; 2) Subcategorized and slash information of Sign is nil; 3) Subcategorized information of Sign2 equals nil; 4) Slash information of Sign2 is equivalent to Obj; 4) a surface string consists of the string "It was", the string relating to Sign1, the string "that" and the string relating to Sign2. Other axioms which are necessary for explaining parsing and generation examples are listed in the appendix.

4. Reversibility In Proposed Architecture

4.1. Sentence Parsing

Parsing techniques were simulated using an argumentation system in (Shimazu,1990). Since he faithfully tried to model existing techniques, many parsing oriented terms such as complete and addition were introduced. This seems to be the cause of the difficulty he experienced in integrating parsing with other processes.

Argument (a):

```
        belasc[10]
t(P)   -----> bel(S,P) .
```

Argument (b):

```
bel(S,by(PE,E,C,PR)),int(S,PE),
exp(S,C),exp1(S,PR)

 by2'
-----> int(S,by(PE,E,C,PR)),
       int(S,E).
```

[7] LF1 is designated as:

```
Sign1: sem: qnt: Q2
            var: Y
            rest: arg0: Y
                  pred: YRest.
```

[8] LF2 is designated as:

```
Sign2: sem: qnt: Q1
            var: X
            rest: arg0: X
                  pred: XRest
            body: arg0: Arg0
                  arg1: Arg1
                  pred: Pred.
```

[9] LF designated as:

```
Sign: sem: qnt: Q1
           var: X
           rest: arg0: X
                 pred: XRest
           body: qnt: Q2
                 var: Y
                 rest: arg0: Y
                       pred: YRest
                 body: arg0: Arg0
                       arg1: Arg1
                       pred: Pred.
```

[10] The expression above the arrow indicates the class of an argument.

Since syntactic, semantic and pragmatic information can be represented with the new by relation, arguments (a) and (b) enable us to simulate parsing: (a) says that true propositions can be ascribed to a speaker's belief; (b) states that, if a speaker is assumed to believe that E is an effect of performing plan expansion PE, while constraint C is true and precondition PR is assumed to be true[11], then it is plausible that his reason for doing PE is his intention to do E.

Parsing is executed as follows: first, axioms whose constraints match an input word are collected; second, the axiom which satisfies the constraint is selected (preconditions are asserted); third, an effect, or semantic information is derived using an instance of argument (b); fourth, another instance of the argument is applied to the effect and the effect which was already derived to obtain a new effect. If the application cannot proceed further, a new word is analyzed; Lastly, if all words in a sentence are analyzed successfully, the execution is complete.

Parsing is exactly the same as plan recognition in the sense of (Konolige and Pollack,1989:925):

> *Plan recognition is essentially a "bottom-up" recognition process, with global coherence used mostly as an evaluative measure, to eliminate ambiguous plan fragments that emerge from local cues.*

Maximizing head elements can realize right association and minimal attachment, but handling semantic ambiguities, that is to clarify global coherence, is a further issue.

4.2. Sentence Generation

Generation can be simulated using arguments (a) and (c). (c) says that, if a speaker believes that E is an effect of performing plan expansion PE, while constraint C is true and precondition PR is assumed to be true, and he intends to do E, then it is plausible that his intention PE is to achieve E.

Argument (c):

bel(S,by(PE,E,C,PR)),int(S,E),
exp(S,C),exp1(S,PR)

by3'
-----> int(by(PE,E,C,PR)),
int(S,PE)

Generation is executed in a similar way to parsing except that axioms are collected using semantic information and the result is a string[12]. Figure 4 describes the generation process. The input linguistic object is equivalent to the object in Figure 2 whose surface string information is parameterized. The generation result is the input object with the instatiated surface string, that is Figure 1. In Figure 4, axiom (2) creates subgoals related to the variable Y and others (corresponding to (2) and (3)) because the semantic and pragmatic information of the input equals the effect and preconditions of the axiom. As the head features propagate to the linguistic object (2), execution addressing object (2) is preferred. The axiom (6) constructs subgoals by referring to the objects whose semantic information is related to the features qnt, var, rest and the logical form concerned with the bound variable X. The head feature preference makes the generator execute axioms about object (4). This results in the axiom (1) of lexical information. Similar to the above process, the remaining subgoals (5) and (2) are executed. Finally, the surface string "It was a girl that the boy loved" is obtained.

5. Discussions

As mentioned in (Emele and Zajac,1990), the proposed approach inevitably leads to the consequence that the data structure becomes slightly complicated. However, due

[11] Action expression exp1(A,P) means that agent A expects proposition P to be assumed to be true if not fully satisfied.

[12] Because of space limitation, action expressions inform and utter are omitted in the figure.

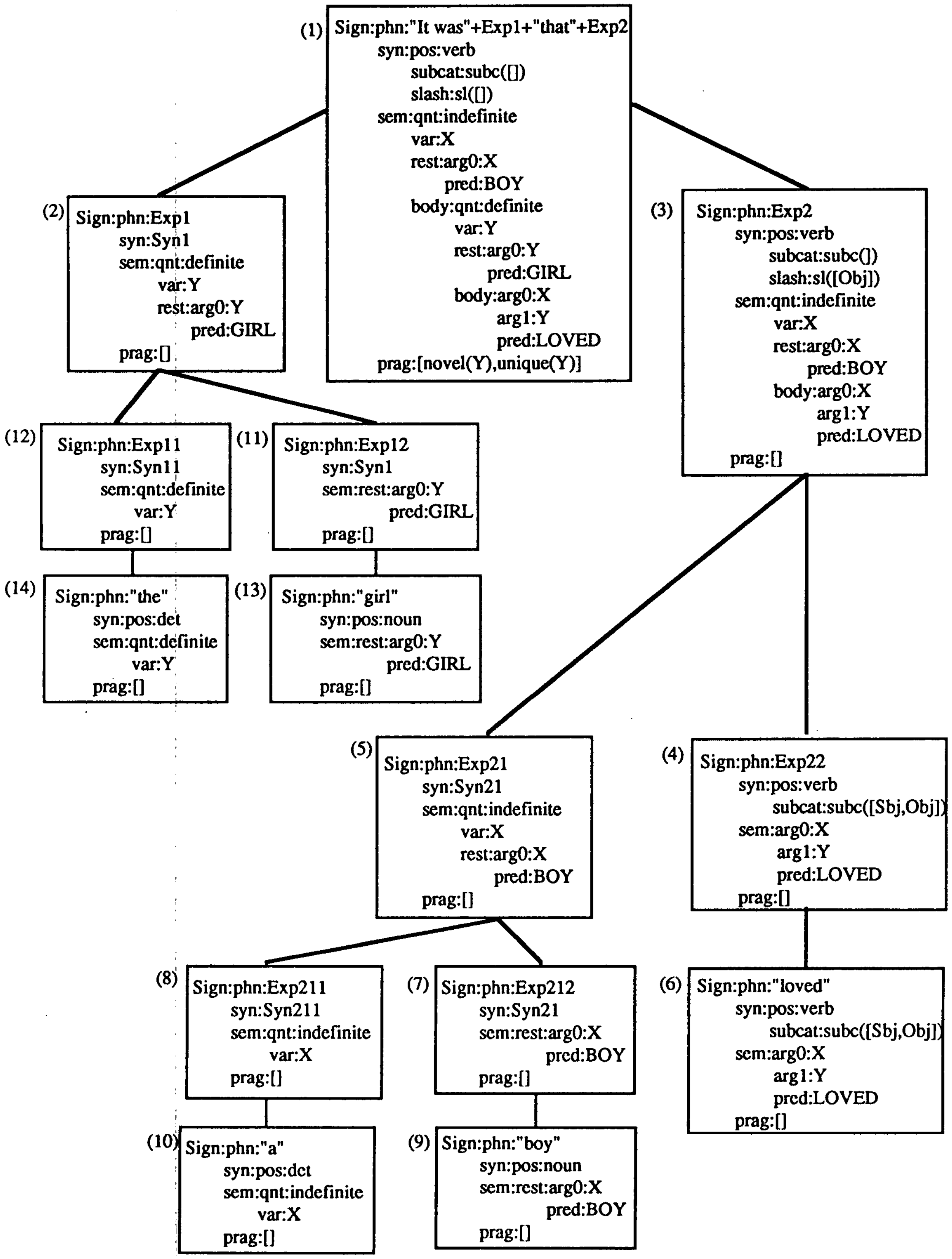

Figure 4. Generation Example.

to the segregation of the structure such as the distinction between preconditions and constraints, the task of developing rules can be done independently. Thus, if you want, you can concentrate on developing grammar rules irrespective of the pragmatic information. If desired, however, pragmatics can be used to precisely stipulate some linguistic phenomena (Delin,1990).

The semantic representation utilized here depends on the strong assumption that it can be systematically decomposed. It is advantageous that the assumption supports symmetry as discussed in (Noord,1990) and naturally realizes semantic indexing which leads to efficient processing (Calder,1989). However, it limits the representation capability for semantic processing (Shieber et al.,1989). The problems of semantic representation are still difficult, so their study is an ongoing task.

Syntactic information, or grammar rules in the paper is neutral in the sense that only one kind of rules, or axioms are sufficient for both parsing and generation; but their difference lies in their usage of information. In the case of parsing, syntactic information is used as a local cue to derive the semantic and pragmatic information. In the case of generation, it is used to prevent the production of ungrammatical strings. This difference appears to mirror asymmetry between writing and reading (or speaking and hearing). The reading process lets unknown words be hypothesized by referring to neighboring words that are understood. Writing, on the other hand, is a process in which unknown words cannot be developed by examining adjacent words. Hypothesizing is used both for parsing and generation, but its role in these processes is different. It is used to derive a coherent interpretation from all words in the parsing, while it is used to smooth the conversational (or text) flow in generation. The difference of the hypothesis use seems to be one of the factors in explaining this asymmetry. The proposed architecture is certain to provide a basis to examine this claim in the sense that it integrates linguistic processes with a reasoning mechanism.

6. Conclusion

This paper has proposed a reversible architecture to handle not only syntactic and semantic information but also pragmatic information. Existing architectures cannot represent pragmatic information explicitly, and lack reasoning capability given insufficient information. I argue that the techniques of plan representation and approximate reasoning in the enhanced argumentation system are effective for solving these problems.

Acknowledgments

The author wishes to extend his sincere gratitude to Tsuneaki Kato and Yoshihiko Hayashi for discussing this architecture. He also thank Masanobu Higashida and Yoichi Sakai for encouraging him to study this subject.

Appendix

Axiom (3):

```
t(by(utter(S,H,"the"),
     inform(S,H,definite),
     ((Pos=det),(Adjacent=noun)),
     ())).
```

Axiom (4):

```
t(by(utter(S,H,"boy"),
     inform(S,H,BOY),
     (Pos=noun),
     ())).
```

Axiom (5):

```
t(by(utter(S,H,"a"),
     inform(S,H,indefinite),
     ((Pos=det),(Adjacent=noun)),
     ())).
```

Axiom (6):

```
t(by(utter(S,H,"girl"),
     inform(S,H,GIRL),
     (Pos=noun),
     ())).
```

Axiom (7):

```
t(by((inform(S,H,LF11),
      inform(S,H,LF12)),
     inform(S,H,LF1),
     ((Pos1=Pos12),
      (Phn1=Phn11+Phn12)),
     ())).
```

Axiom (8):

```
t(by((inform(S,H,LF21),
      inform(S,H,LF22)),
     inform(S,H,LF2),
     ((Pos2=Pos21),
      (Subcat2=subc([])),
      (Slash2=sl([Obj])),
      (Subcat22=subc([Sbj,Obj])),
      (Phn2=Phn21+Phn22)),
     ())).
```

Bibliography

(Appelt,1985) Douglas E. Appelt. 1985. "Planning English Sentences". Cambridge University Press.

(Calder et al.,1989) Jonathan Calder, Mike Reape, and Henk Zeevat. 1989. "An Algorithm for Generation in Unification Categorial Grammar". Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics. Manchester, U.K.. 10-12, April. 233-240.

(Delin,1990) J.L.Delin. 1990. "A Multi-Level Account of Cleft Constructions in Discourse". Proceedings of the 13th International Conference on Computational Linguistics. Aug. 20-25. Helsinki, Finland. 83-88.

(Dymetman et al.,1989) Marc Dymetman and Pierre Isabelle. 1989. "Reversible Logic Grammars For Machine Translation". Proceedings of the 2nd International Conference on Theoretical and Methodological Issues In Machine Translation of Natural Languages. Jun. 12-14. Pittsburgh, Pennsylvania, U.S.A..

(Elkan,1990) Charles Elkan. 1990. "Incremental, Approximate Planning". Proceedings of the 8th National Conference on Artificial Intelligence. Jul.29-Aug.3. Boston, MA, U.S.A.. 145-150.

(Emele and Zajac,1990) Martin Emele and Remi Zajac. 1990. "Typed Unification Grammars". Proceedings of the 13th International Conference on Computational Linguistics. Aug. 20-25. Helsinki, Finland. 293-298.

(Gazdar et al.,1989) Gerald Gazdar and Chris Mellish. 1989. "Natural Language Processing in PROLOG". Addison-Welsley Publishing Company.

(Hasida,1987) Hasida Koiti. 1987. "Dependency Propagation: A Unified Theory of Sentence Comprehension and Generation". Proceedings of the 10th International Joint Conference on Artificial Intelligence. Aug. 23-28. Milan, Italy. 664-670.

(Konolige and Pollack,1989) Kurt Konolige and Martha E. Pollack. 1989. "Ascribing Plans To Agents - Preliminary Report - ". Proceedings of the 11th International Joint Conference on Artificial Intelligence. Aug. 20-25. Detroit, Michigan, U.S.A.. 924-930.

(Moore et al,1989) Johanna D. Moore and Cecile Paris. 1989. "Planning Text For Advisory Dialogue". Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics. Jun. 26-29. Van couver, British Columbia, Canada. 203-211.

(Pollard et al.,1987) Carl Pollard and Ivan A. Sag. 1987. "An Information-Based Syntax and Semantics (Volume 1)". CSLI

Lecture Notes. Number 13.

(Pollard et al.,1990) Carl Pollard and Ivan A. Sag. 1990. "An Information-Based Syntax and Semantics (Volume 2)". ms.

(Shieber,1988) Stuart M. Shieber. 1988. "A Uniform Architecture For Parsing and Generation". Proceedings of the 12th International Conference on Computational Linguistics. Aug. 22-27. Bonn, Germany. 614-619.

(Shieber et al.,1989) Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C.N. Perreira. 1989. "A Semantic-Head-Driven Generation Algorithms for Unification-Based Formalisms". Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics. Jun. 26-29. Van couver, British Columbia, Canada. 7-17.

(Shimazu,1990) Akira Shimazu. 1990. "Japanese Sentence Analysis as Argumentation". Proceedings of the 13th International Conference on Computational Linguistics. Aug. 20-25. Helsinki, Finland. 259-264.

(Strzalkowski,1990) Tomek Strzalkowski. 1990. "How To Invert A Natural Language Parser Into An Efficient Generator: An Algorithm For Logic Grammars". Proceedings of the 13th International Conference on Computational Linguistics. Aug. 20-25. Helsinki, Finland. 347-352.

(Woods,1978) W.A.Woods. 1978. "Semantics and Quantification in Natural Language Question Answering". Advances in Computers. Vol. 17. M.Yovits (ed.). Academic Press.