A Study of Cognitive Processes of Natural Language Understanding

自然言語理解における認知プロセスに関する研究

森　辰則

December 25, 1990

A Study of Cognitive Processes of Natural Language Understanding

自然言語理解における認知プロセスに関する研究

A Dissertation submitted to
The Division of Electrical and Computer Engineering,
Faculty of Engineering,
Yokohama National University
in Partial Fulfillment of the Requirements
for The Degree of Doctor of Engineering

By
Tatsunori Mori
森 辰則

December 25, 1990

Title	A Study of Cognitive Processes of Natural Language Understanding (自然言語理解における認知プロセスに関する研究)
Author	Tatsunori Mori (森 辰則)
Date	February 20, 1991

Signature of Author Tatsunori Mori

Certified by

Professor	Hideki Imai (Thesis Supervisor)	Hideki Imai
Professor	Takashi Sekiguchi	Takashi Sekiguchi
Professor	Yasutaka Dohi	Yasunori Dohi
Associate Professor	Hiroshi Nakagawa (Thesis Supervisor)	Hiroshi Nakagawa
Associate Professor	Hiroshi Arisawa	H. Arisawa
Associate Professor	Naoyoshi Tamura	Naoyoshi Tamura

Abstract

A Study of Cognitive Processes
of Natural Language Understanding

From the viewpoint of computational linguistics, of which main purpose is to formalize human processes of language understanding as computational processes, this dissertation proposes three models which formalize three different aspects of natural language understanding respectively.

First, we propose a model of metaphor understanding in which contexts play an important role. The model of metaphor understanding proposed by this dissertation deals not only with understanding an isolated metaphorical sentence, but also with understanding metaphorical expressions in a discourse. We adopt the formalization based on situation semantics, in which by dealing with the interpretation of a metaphorical expression and the information in the expression separately the metaphor understanding is done according to the hearer's resource.

Secondly, we propose a human parsing model of natural language from the point of view of performance, or actual usage of language. Several researchers have paid attention to some remarkable phenomena in the human parsing process, since these phenomena may be a clue to the explication of the high efficiency on the human parsing process. On the hypothesis that these phenomena should be due to delicate aspects of performance, we propose a parsing model based on connectionist model, which is founded on interaction among a large number of simple computing element.

Finally, we propose a model of common sense reasoning, which is used for inference by world knowledge, or background knowledge, in natural language understanding. Some mechanism of common sense reasoning, such as circumscription, is necessary in order to make a natural language understanding system be able to comprehend some natural language expression more deeply, that is, to make the system be able to obtain some useful information, which human extracts from some expression by using his/her own background knowledge, especially common sense. While circumscription is a rule for the first order logic, we propose an application of circumscription to logic programming languages, such as Prolog. We also present an application of circumscription to the programming language Uranus which is an extended logic programming language with the multiple world mechanism. The application of circumscription to Uranus is useful to formalize a common sense reasoning on some frame-style knowledge base.

Contents

List of Figures

Acknowledgments

This work is here not because of my effort alone, but thanks to a lot of people who have supported me, in academic or personal way, especially during this couple of years.

First of all, I would like to thank professor Hiroshi Nakagawa, my supervisor, heartily. He has given me many pieces of advice on both academic and personal matter earnestly for last six year. I also thank professor Hideki Imai and professor Yasutaka Dohi for their very meaningful advice.

Discussions with people in Nakagawa laboratory, which I have been a member of for these six years, have been constantly stimulating me. I am also indebted to people in the division of electrical and computer engineering for their help.

People in Hozumi Tanaka laboratory, Department of Computer Science, Tokyo Institute of Technology, let me attend their reading meeting. I had significant times with them.

Parts of this dissertation would not have been possible without the discussion in AIUEO, an informal seminar on artificial intelligence, and ICOT NLU, a working group of the Institute for New Generation Computer Technology.

Finally, I would like to thank all people who give assistance to me, personally or secondhand.

Chapter 1

Introduction

Natural language is the medium of communication which has been established naturally in order for our to use it for mutual understanding, and it has changed with the times. For example, Japanese, which is author's mother tongue, is one of natural languages. Without any special effort, one can understand expressions in his/her mother tongue[1]. For linguists and psychologists, the main goal of researches of natural language is to make models which can explain such an ability of language comprehension.

However, as computers was invented, computer scientists, especially AI[2] researchers, participated in the research on a new line of approach. The methodology is to formalize processes of language understanding as calculation and establish natural language understanding system on computer in order to validate the model. Conversely, it is possible that fruitful results of modeling the processes as computation are returned to both the psychology field and the linguistic field. This interdisciplinary field called *computational linguistics* is one of the key themes in cognitive science[All87]. The major goal of establishing a language

[1] Needless to say, in order to master the mother tongue, he/she had to learn it somehow in his/her early childhood. A number of psychologists have observed and studied how children obtain the ability of language comprehension. According to it, while children of three or so can rapidly acquire the ability both to understand their mother tongue and to use it, after the period one has to make a special effort to acquire foreign languages[And80].

[2] Artificial Intelligence

understanding system is to use a natural language, which seems to be the most natural way of communication for us, for one of media of communication between human and machines. While the technology for controlling machines by computers is at fairly high level, between human and computers, there exist some semantic gaps because of the difference between languages being used. It is expected that natural language understanding systems are developed on computers to serve as the bridge over the gap, that is, as the interface between human and machines. A number of computational models of natural language understanding have been proposed so far, and each of them has its own characteristic. However they seems to share the following basic organizations, in which there exist some different realm according to information treated[FM86, TT88, Nom88].

- *Morphological analysis*
  This realm concerns how words are constructed out of more basic meaning units called *morphemes*. For example, it deals with conjugation, declension, inflection, and so on.
- *Syntactic analysis*
  According to a grammer, which describes the syntactic structure of a natural language, this realm concerns how words can be put together to form sentences that look correct syntactically in the language[Sel85, Shi86, Cho88, GKPS85, Gun87a, KB82, SSS88, Gun87b, Tuj86, Ots87].
- *Semantic analysis* (in one sentence)
  This realm concerns what words mean and how these meanings combine in sentences to form sentence meanings[NN86, Ken85, NF83].
- *Discourse analysis*
  This realm concerns how sentence meanings combine in a discourse to form a total meaning of the discourse[BB83, BP83, Bar89]. For example, anaphora of pronouns is included in this realm.
- *Pragmatic analysis*
  This realm concerns how a context affects the interpretation of a expression and what information the expression carries[Lev83]. For example, this realm deal with speech act, metaphor understanding, and so on[Yam86, Fau85, LJ80, Yam88, Sak85].

Moreover, the following realm is related to the realms described above.

- Analysis based on world knowledge
  This realm concerns how the general knowledge about the structure of the world are used for inference in order to, for example, maintain coherence of the above realms, complete insufficient information, and so on[Tsu87].

As described above, computational models of natural language understanding can be regarded as the complex system consisting of a number of different realms. However we should rather regard that processes of these analysis are combined not in a sequential manner, but in the manner of non-directional interaction in order to understand some language expressions[Has89]. Since there exist many unsolved problems in each realms, a number of researchers are going on studying them in order to establish more sophisticated natural language understanding systems.

From the viewpoint of computer linguistics, of which main purpose is to formalize human processes of language understanding as computational processes, this dissertation proposes the following three models which formalize three different aspects of natural language understanding respectively.

- *A metaphor understanding model*
  (Chapter 2. A Formalization of Metaphor Understanding in Situation Semantics)
- *A parsing model*
  (Chapter 3. A Parsing Model based on Connectionist Model)
- *A model of common sense reasoning*
  (Chapter 4. A Model of Common Sense Reasoning Based on Circumscription)

In Chapter 2, we will propose a model of metaphor understanding in which contexts play an important role. Metaphor is one of rhetorical methods. However, it is used not only in literary writings, but also in many conversations in our everyday life as effective means of communication. Therefore, metaphor understanding is one of the most important themes which we must deal with to have a sophisticated natural language understanding system. Metaphor understanding should be regarded not as a simple process of semantic analysis, but the complex of several processes, such as semantic analysis, discourse analysis, inference based on world knowledge, and so on. From the view of this, formalizing metaphor understanding provides us good material for establishing an integrated system of language understanding.

The model of metaphor understanding proposed by this dissertation deals not only with understanding an isolated metaphorical sentence, but also with understanding metaphorical expressions in a discourse. In order to deal with the information obtained from a metaphorical expression in a discourse, it is necessary that the part of semantics which deals with metaphor understanding should harmonize with the rest of semantics which deals with the ordinary discourse understanding. It is a typical case of the metaphor understanding in a discourse that the information which the hearer obtains from a metaphorical expression depends on the context in which the expression is uttered. Generally speaking, the information in an expression may vary with hearer's circumstance, or *resource*, such as contexts, background knowledge, and so on. The context sensitivity like this can be explained by drawing a distinction between the interpretation of an utterance and the information in the utterance, which is one of the ideas in situation semantics. Accordingly we will adopt the formalization based on situation semantics, in which by dealing with the interpretation of a metaphorical expression and the information in the expression separately the metaphor understanding is done according to the hearer's resource.

In Chapter 3, we will propose a human parsing model of natural language from the point of view of performance, or actual usage of language. Several researchers have paid attention to some remarkable phenomena in the human parsing process, since these phenomena may be a clue to the explication of the high efficiency on the human parsing process. On the hypothesis that these phenomena should be due to delicate aspects of performance, we will propose a parsing model based on *connectionist model*, which is founded on interaction among a large number of simple computing element. A connectionist parser according to the model will basically work in real time, and its behavior is fairly similar to one of the human parsing. Moreover, we will show that our connectionist parser can explain some interesting phenomena in the human parsing process, such as cognitive difficulties to recognize garden path sentences or deeply nested sentences, preference of structurally ambiguous sentences, and so on.

In Chapter 4, we will propose a model of *common sense reasoning*, which is used for inference by world knowledge, or background knowledge, in natural language understanding. Some mechanism of common sense reasoning, such as *circumscription*, is necessary in order to make a natural language understanding system be able to comprehend some natural language expression more deeply, that is, to make the system be able to obtain some useful information, which human extract from some expression by using his/her own background knowledge, especially *common sense*. In language understanding, it is hard for us to own perfect knowledge of our circumstance. However, we can infer some useful information from such incomplete knowledge. Circumscription is a rule of conjecture based on incomplete knowledge that can be used for 'jumping to certain conclusions'. Since such a reasoning should be one of the most important kind of reasoning which we usually do in our daily life, it will be expected that the reasoning is useful for formalizing an aspect of reasoning in natural language understanding. While circumscription is a rule for the first order logic, in this dissertation, first we will propose an application of circumscription to logic programming languages, such as Prolog. By this method, common sense reasoning can be done using the knowledge base which is built with one of logic programming languages. Logic programming languages, which are based on a subset of the first order logic called *horn set*, are well suited to natural language processing because of its characteristics in symbol manipulation, such as the unification mechanism, and so on. It is also one of characteristics of logic programming languages that a program written in them can be regarded as the knowledge base which naturally includes both facts and rules.

We also present an application of circumscription to the programming language *Uranus* proposed by Nakashima[NTS84, Nak85b], which is an extended logic programming language with the multiple world mechanism. The multiple world mechanism allows programmers both to divide program definition space into several parts called *worlds* and to organize them in a hierarchical manner like the frame representation. Since a knowledge base used in natural language understanding should consist of complicated information, some highly structured schema, such as the frame representation, is necessary to describe it. The application of circumscription to Uranus is useful to formalize a common sense reasoning on some frame-style knowledge base.

Chapter 2

A Formalization of Metaphor Understanding in Situation Semantics

2.1 Introduction

Metaphor — one of rhetorical methods — is used not only in literary writings, but also in many conversations in our everyday life as effective means of communication[Yam88, LJ80, Lak87b]. Therefore, metaphor understanding is one of the most important themes which we must deal with to have a sophisticated natural language understanding system. The metaphor understanding process has been discussed from various points of view by a number of researchers[TIT89, Ind87, GFS87, OF87, Pla87, Lak87a, Nun87, Hol89].

It is the essence of metaphors that one thing or idea (we will call it α) is expressed by means of something else (β). For example, in a metaphorical phrase "A man like a wolf", "a man" corresponds to α and "a wolf" corresponds to β. In the rest of this chapter, we will use the term *target* (or *target domain*) to refer to a thing α and *source*(*source domain*) to refer to β. According to the interaction theory of metaphors proposed by Black, an "implicative complex", which is a set of inferences that can be drawn about a domain, is associated with the domain. A metaphorical expression works by projecting an implicative complex of a source domain upon a target domain[Ind87]. There are a number of studies of the metaphor understanding based on this theory, in which such projections, or correspondences, are found by an analogical mapping[GFS87, Hol89, Ind87]. Most of these studies, however, deal only with understanding an isolated metaphorical sentence, and regard the metaphor understanding process just as establishing correspondences by some analogical mapping. Indeed the establishment of correspondences is the main part of the metaphor understanding process, but in order to deal with the information obtained from a metaphorical expression in a discourse, it is necessary that the part of semantics which deals with metaphor understanding should harmonize with the rest of semantics which deals with the ordinary discourse understanding. There are no researches based on such a viewpoint, that is, a framework which refers to the analysis of the metaphorical expression in a discourse.

In this chapter, in view of this point, we present a new theory which can account for understanding metaphorical expressions in a discourse. It is a typical case of the metaphor understanding in a discourse that the information which the hearer obtains from a metaphorical expression depends on the context in which the expression is uttered. For example, let us consider the following sentence:

With a new strategy, I wiped him out. (2.1)

This sentence can convey some different kind of information according to contexts in which it is embedded:

Case-1. A description about a war:
The head of a winning state utters the sentence about the victory, referring to the head of the defeated state.

Case-2. A description about an argument:
Someone utters the sentence about the argument in which he/she refuted his opponent.

"I argued with Mr.B about his theory last night."
"How did it come out ?" (2.2)

Case-3. A description about a game:
The manager of a soccer team utters the sentence about the result of the game, referring to the manager of the opponent team.

etc.

As this example shows, the sentence (2.1) connotes different situations according to the contexts in which it is uttered. These connotations of the sentence (2.1), however, seem to share some common structure, rather than to be quite different from each other.

Generally speaking, the information in an expression may vary with hearer's circumstance, or *resource situations*, such as contexts, background knowledge, and so on. The context sensitivity like this can be explained by drawing a distinction between the interpretation of an utterance and the information in the utterance, which is one of the ideas in situation semantics. Accordingly we adopt the formalization based on situation semantics, in which by dealing with the interpretation of a metaphorical expression and the information in the expression separately the metaphor understanding is done according to the hearer's resource. That is, an actual meaning connoted by either a word or a phrase in a metaphorical expression may vary from its ordinary meaning. As a result, metaphorical expressions can convey some other new information. We would like to formulate this variation as a process of introducing new resources, namely, establishing correspondences between a source domain and a target domain, with which a hearer can project information about a source domain onto a target domain.

2.2 The Outline of Metaphor Understanding in Situation Semantics

In our theory, metaphor understanding consists of the following steps and semantic representations with which this metaphor understanding process are represented in terms of situation semantics:

Step-1 Hear a fragment uf^i of an utterance and construct states of affairs corresponding to it by the convention of the language use. Then a described situation s^i_{uf} corresponding to uf^i is obtained.

Step-2 By supposing that uf^i describes the same situation which is described by the current context s_d^{i-1}, we obtain a situation s_d^{new} such that:

$$s_d^{new} = s^i_{uf} = s_d^{i-1} \tag{2.3}$$

Then examine whether the situation s_d^{new} satisfies the following requirements:

1. Every relation in the situation s_d^{new} satisfies the appropriateness of its argument assignment.
2. The situation s_d^{new} is coherent.
3. All of constraints holding in the situation s_d^{new} are satisfied.

Step-3 If the requirements in Step-2 are satisfied, let the new context s_d^i be s_d^{new} and increase i by 1, then go to Step-1.

Step-4 Since the requirements in Step-2 are not satisfied at this point, introduce a new resource to cancel the inappropriateness and to obtain a meaningful interpretation. In metaphor understanding, as described later, the new resource is a set of constraints which represents the correspondence between the source domain and the target domain. Thus, the new situation s_r^i in which these constraints hold is regarded as a new resource situation. Now pay attention to states of affairs carried relative to the resource situation s_r^i by the described situation s^i_{uf}, which is a situation related to the source domain in this case. Since the situation $s^{i,t}_{uf}$ supporting these states of affairs corresponds to a situation related to the target domain, return to Step-2 after replacing the described situation s^i_{uf} with $s^{i,t}_{uf}$.

Not only in ordinary sentence understanding, but also in metaphor understanding, it does not seem to be plausible that the hearer begins to interpret a sentence just after he/she finished hearing all of the sentence. As described in Step-1, whenever the hearer hears/reads a fragment of an utterance such as a noun phrase, a verb (phrase), and so on, he/she starts to interpret the fragment under the current context made by the discourse.

It is Step-2 that is the process which integrates fragments of interpretation newly obtained with the current context. The way of the integration depends on the convention of the language used. For example, the grammer of the language is one of the constraints which are effective within one sentence.

In Step-4, the described situation s^i_{uf} constitutes a part of the context about the source domain, which is going on in parallel with the main context. We call this context about the source domain the *parallel context* and denote it s_p^i. If there already exists a parallel context s_p^{i-1}, Step-2 is also applied to the parallel context and s^i_{uf}. When the requirements in Step-2 are satisfied, a new parallel context s_p^i such that

$$s_p^i = s_p^{i-1} = s^i_{uf} \tag{2.4}$$

is made. In this case, the situation which includes both s_r^i and the resource situation which has already introduced for the parallel context s_p^{i-1} serves as a new resource situation. When one of the requirements in Step-2 is not satisfied, a new metaphor has been introduced and another parallel context is made. By allowing for multiple parallel contexts which describe multiple source domains of metaphors, the case that several different metaphors are introduced one after another into a discourse can be accounted for.

2.3 The Details of Each Step

This section will give details of these steps roughly described in the previous section through the process of understanding the example sentence (2.1)

"With a new strategy, I wiped him out."

in the Case-2, in which a topic about an argument is described.

2.3.1 Construction of a Fragment of a Described Situation

As described in the previous section, we assume that whenever the hearer hears/reads a fragment of an utterance, he/she interprets the fragment to some extent. While how big the unit of fragments to be interpreted is depends on the language used and his/her ability in the language use, it seems that the most primitive unit is the phrase, which is determined by the grammar of the language, such as a noun phrase and a verb phrase. Assuming that the phrase is the unit of an utterance fragment, the example sentence is regarded as the following sequence of three fragments:

$$[\text{With a new strategy}]_{PP},\ [\text{I}]_{NP}\ [\text{wiped him out}]_{VP}$$

As the result of the constraint satisfaction such as the grammer of the language, the described situation s^i_{uf} corresponding to each fragment described above is obtained as follows:

$$s^1_{uf} \models \langle\langle \textit{with, agent:A, object:ST} \rangle\rangle \wedge \langle\langle \textit{strategy, object:ST} \rangle\rangle \quad (2.5)$$

$$s^2_{uf} \models \langle\langle \textit{speaker, agent:A} \rangle\rangle \quad (2.6)$$

$$s^3_{uf} \models \langle\langle \textit{wipe\_out, agent:A, patient:B} \rangle\rangle \quad (2.7)$$

We also assume the following situation s^0_d as the initial context corresponding to the context (2.2):

$$s^0_d \models \langle\langle \textit{argue, agent:A, participant:B, object:TH} \rangle\rangle \wedge \langle\langle \textit{theory, object:TH} \rangle\rangle \wedge \langle\langle \textit{own, agent:B, object:TH} \rangle\rangle \quad (2.8)$$

In the following subsections, first we will show how the described situation s^1_{uf} is interpreted by Step-2, Step-3 and Step-4 in the process described above. Then we will describe about the interpretation of s^2_{uf} and s^3_{uf} briefly.

2.3.2 Examining the Appropriateness of a Described Situation

According to Step-2, let us consider the situation s^{new}_d such that:

$$s^{new}_d = s^1_{uf} = s^0_d \quad (2.9)$$

This situation is a candidate for the context of s^2_{uf}. If only the condition (2.9) is imposed on s^{new}_d, the situation satisfies the requirements in Step-2. The hearer, however, usually uses some background knowledge to fill the lack of information, whenever he/she interprets an utterance fragment. For example, the typical situation supporting a state of affairs *argue*, and the knowledge that what state of affair is involved by the state of affair *strategy*, are derived from these background knowledge. In the following part of this subsection, first we show how to represent background knowledge, then we consider how s^{new}_d is interpreted with the background knowledge which we usually have in our mind.

2.3.2.1 Representing Background Knowledge

An utterance itself usually gives little information to a hearer. The hearer can receive useful information from an utterance with some background knowledge, rather than from the utterance only. In this chapter, by the term *resources* we mean some information which does not appear in utterances and can be used to interpret utterances by a hearer. We will pay attention particularly to the following resource, from which the hearer will be able to obtain useful information for metaphor understanding:

> The domain knowledge of a concept such as the typical context in which the concept is used

Since the speaker assumes that the hearer has sufficient knowledge about a source domain to retrieve some information intended by the speaker from an expression related to the source domain, the speaker may inform the hearer about something related to the current context, that is, the target domain intentionally by comparing the two domain. Therefore, in order to formulate metaphor understanding, it should be assumed that a hearer has some knowledge both of source and target domains to a certain extent. Such a knowledge includes common sense and typical scenes, which the hearer is usually supported to have or be able to associate in his/her mind.

We treat this background knowledge as a situation type, that is,

$$[s \mid s \models \sigma] \quad (2.10)$$

which somehow the hearer has. Then an instance of a typical scenes actually used as a resource becomes the situation which is of this situation type, where parameters are anchored to fit individual situations. These resource situations may also include knowledge such as causal relations between states of affairs in typical scenes.

We also define the type hierarchy in terms of the relation *subtype* in order to express the conceptual hierarchy as follows:

- R_1 *subtype* R_2 is true, if all assignments of the type R_1 are of the type R_2.
- Types in the same type hierarchy can have the same assignment, only if they have a common lower bound.

2.3.2.2 Interpretation with Background Knowledge

We suppose that the noun "strategy" recalls the situation type T_{war} to the hearer's mind as the background knowledge of the "war"[1]. Suppose also that T_{war} is the type of the situation which supports the following constraints about a strategy:

$$\langle\langle \Rightarrow, \langle\langle strategy, object{:}x\rangle\rangle, \langle\langle method, object{:}x, event\text{-}type{:}ET_{use\ m.p.}\rangle\rangle\rangle\rangle \tag{2.11}$$

where

$$ET_{use\ m.p.} = [s \mid s \models \langle\langle use, agent{:}y, object{:}mp\rangle\rangle \wedge \langle\langle mil\text{-}power, object{:}mp\rangle\rangle] \tag{2.12}$$

Now, a positive constraint $\Rightarrow$ is defined as follows. If $s_0 \models \langle\langle \Rightarrow, \sigma(\vec{x}), \tau(\vec{x})\rangle\rangle$ then for every situation $s \trianglelefteq s_0$ and every anchor $f : \vec{x} \rightarrow Obj(s)$ such that $s \models \sigma(f)$, there is a situation s' such that $s' \models \tau(f)$. A negative constraint $\perp$ is also defined as follows. If $s_0 \models \langle\langle \perp, \sigma(\vec{x}), \tau(\vec{x})\rangle\rangle$ and there is a situation $s \trianglelefteq s_0$ and an anchor f such that $s \models \sigma(f)$, then there is no situation s' such that $s' \models \tau(f)$.

[1] To do this, such a mechanism as the association is necessary in implementation.

For the initial context s_d^0, we also suppose that to the hearer's mind, the verb "argue" recalls the type T_{argue} of the situation, in which the following states of affairs hold, as the background knowledge of the "argue":

$$\langle\langle argue, agent{:}a, participant{:}p, object{:}x\rangle\rangle \wedge \langle\langle use, agent{:}a, object{:}y\rangle\rangle \wedge \langle\langle reasoning, object{:}y\rangle\rangle \tag{2.13}$$

Moreover, we suppose that T_{misc} is the situation type which represents the miscellaneous background knowledge and a situation of T_{misc} supports the following constraints:

$$\langle\langle \Rightarrow, \langle\langle with, agent{:}a, object{:}x\rangle\rangle \wedge \langle\langle method, object{:}x, event\text{-}type{:}[s \mid s \models \sigma]\rangle\rangle, \sigma(agent{:}a)\rangle\rangle \tag{2.14}$$

$$\langle\langle \perp, \langle\langle mil\text{-}power, object{:}y\rangle\rangle, \langle\langle reasoning, object{:}y\rangle\rangle\rangle\rangle \tag{2.15}$$

The state of affairs (2.14) represents the constraint about the case that the preposition "with" occurs with a "method". The state of affairs (2.15) represents the constraint about the conceptual structure.

Now, let us apply the background knowledge described above to s_d^{new} such that (2.9), that is, assume the following conditions:

$$s_d^{new}{:}T_{argue} \tag{2.16}$$

$$s_d^{new}{:}T_{war} \tag{2.17}$$

$$s_d^{new}{:}T_{misc} \tag{2.18}$$

The proposition (2.16) leads to:

$$s_d^{new} \models \langle\langle argue, agent{:}A, participant{:}B, object{:}TH\rangle\rangle \wedge \langle\langle use, agent{:}A, object{:}y\rangle\rangle \wedge \langle\langle reasoning, object{:}y\rangle\rangle \tag{2.19}$$

where we assume that the state of affair *argue* supported by s_d^{new} was merged with one in T_{argue}.

On the other hand, for (2.17) and (2.18) imply that the constraints (2.11) and (2.14) hold in s_d^{new}, there exists the following situation s'_d:

$$s'_d \models \langle\langle method, object{:}ST, event\text{-}type{:}ET_{use\ m.p.}\rangle\rangle \wedge \langle\langle use, agent{:}A, object{:}mp\rangle\rangle \wedge \langle\langle mil\text{-}power, object{:}mp\rangle\rangle \tag{2.20}$$

Assuming that states of affairs carried relative to the constraints, that is, the right hand side of (2.20) is used as parts of the main context, the condition $s'_d = s_d^{new}$ should be satisfied. Under the condition, it seems to be natural that we presume the state of affair *use* in (2.19) is merged with one in (2.20). For this presumption leads to:

$$s_d^{new} \models \langle\langle reasoning, y \rangle\rangle \wedge \langle\langle mil\text{-}power, y \rangle\rangle \tag{2.21}$$

it is clear that the situation s_d^{new} violates the constraint (2.15). Since the requirement 3 of Step-2 is not satisfied, to assume the condition (2.9), that is, $s^1_{uf} = s^0_d$ has turned out to be false. Therefore s^1_{uf} constitutes the parallel context s^1_p.

$$s^1_p = s^1_{uf} \tag{2.22}$$

Generaly speaking, there exists some inappropriateness about the content in metaphorical expressions. What a pair of domains is compared with in a metaphor understanding is shown by the background knowledge recalled using associations. In the example, the initial context s^0_d and the new described situation s^1_{uf} recall the background knowledge T_{argue} of an argument and the background knowledge T_{war} of a war respectively. This means that the target domain is the "argument", and the source domain is the "war". Finally we obtain the following conditions as the result of Step-2:

$$s^0_d \neq s^1_{uf} \tag{2.23}$$

$$s^0_d{:}T_{argue} \wedge s^0_d{:}T_{misc} \tag{2.24}$$

$$s^1_{uf}{:}T_{war} \wedge s^1_{uf}{:}T_{misc} \tag{2.25}$$

That is, the new described situation s^1_{uf} constitutes a parallel context which is not equal to the main context s^0_d. The main context s^0_d and the described situation s^1_{uf} are the situations related to an argument and a war, respectively.

2.3.3 Introducing New Resources to Get Meaningful Information

In the previous subsection, we examined the inappropriateness which might occur when a described situation is combined with the hearer's

circumstances. In metaphor understanding, however, the clue is what kinds of information is carried by an utterance and is combined with the context to make a new context. But, as described in the previous subsection, the hearer's background knowledge of both a war and an argument is insufficient for information which contributes to the main context to be able to obtained from s^1_{uf}. Therefore, it can be presumed that the hearer should introduce some new resource to get meaningful information.

In addition to the interaction theory mentioned in the section 2.1, the following two studies serve to figure out the new resource. First, Lakoff[LJ80] points out that about the example, we can consider that *a part of conceptual network in the concept "war" partially characterizes the concept "argument"*. Secondly, the pragmatic function proposed by G.Nunberg, whose importance is pointed out by G.Fauconnier[Fau85], is worthy of notice. The pragmatic function connects objects whose characters are different from each other according to psychological, cultural, domestic pragmatic grounds. Nunberg shows that a reference to an object can be done by another object which is appropriately connected with the former object. Both studies suggest that the correspondence is important. As the consequence we come up with having the following supposition:

> First, the hearer introduces some correspondence between a source domain and a target domain as a new resource. Then, according to this resource, he/she converts information, that is, he/she converts some states of affairs in the source domain into the corresponding states of affairs in the target domain[2].

In the case of simile understanding or metaphor understanding, what connects the source domain with the target domain is a set of correspondences such as those which are given by a general metaphor [3] and are made by some analogical mapping. On the other hand, in metonymy understanding and so on, pragmatic functions may give some correspondences.

[2]From now on, let the terms "source domain" and "target domain" mean also situations related to a source domain and a target domain, respectively.

[3]See also the section 2.3.3.2 for further information about general metaphors.

Figure 2.1: Construction of an Information Converter

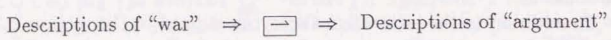


Figure 2.2: Conversion of Information

2.3.3.1 A Description of Correspondences between a Target Domain and a Source Domain

In this section, we will show what correspondences we should deal with and how to express them. Since a source domain and a target domain are generally distinct from each other, in a strict sense we should treat correspondences between the schemes of individuation for each situation in order to express correspondences between the situations. However, since it is important for metaphor understanding what kinds of information about the target domain are obtained from descriptions about the source domain, the following constraints are sufficient to express correspondences between the source domain and the target domain:

$$\langle\langle \Rightarrow, \langle\langle r_{source}, \vec{x}; p_s \rangle\rangle, \langle\langle r_{target}, \vec{y}; p_t \rangle\rangle \rangle\rangle \tag{2.26}$$

where s_{target} and s_{source} are the situations related to the target domain and the source domain respectively, and $r_{target} \in Rel_{s_{target}}$, $r_{source} \in Rel_{s_{source}}$. The hearer's attunement to these constraints means that he/she is aware of the correspondence between the two domain by comparing them. Intuitively, these constraints are descriptions of the hearer's inference or association between a source domain and a target domain.

2.3.3.2 How to Obtain Correspondences

How can we obtain the constraints which represent the correspondence between the source domain and the target domain ? General metaphors[LJ80] are an important concept relevant to this question. Lakoff points out that a metaphorical expression in a certain text is an instance of a general metaphor. For instance, the example sentence can be regarded as an instance of the general metaphor "ARGUMENT IS WAR". At first sight, it seems to be sufficient for metaphor understanding to prepare all of correspondences obtained from all of general metaphors. However, for metaphorical expression can be and moreover might be extended flexibly by the speaker according to the context, it seems to be difficult that we prepare universal set of correspondences. If so, how can we get correspondences, which should be set up according to the context ? In the next section, we show that some heuristics can give us some useful correspondences.

2.3.3.3 Heuristics for Analogical Mapping in Metaphor Understanding

Basically, there may be many possible correspondences between two domains. However, it seems that we use some heuristics to establish correspondences which can be used to retrieve some useful information from metaphorical utterances during our interpretation process. Let us enumerate some heuristics useful for metaphor understanding:

H1 Coherence of information

There are no incoherent situations when constraints, representing correspondences, are applied.

H2 Similarity between corresponding states of affairs

This heuristic is based upon our assumption that it is difficult even also for human in metaphor understanding to find correspondences between two domains containing no pairs of states of affairs closely related to each other. On this assumption, we expect that some corresponding pairs of states of affairs may be found easily.

H2.1 Make correspondences between the same states of affairs in two domains as far as possible.

H2.1.1 Especially, when a state of affairs $\sigma = \langle\langle r, \vec{x}; p\rangle\rangle$ holding in the source domain does not appear in the target domain, check whether the state of affairs can be used in the target domain, that is, checking whether the assignments to the argument roles of the state of affairs are appropriate in the target domain. If the condition is satisfied, the following constraint becomes one of the assumption:

$$\langle\langle \Rightarrow, \sigma, \sigma\rangle\rangle \tag{2.27}$$

H2.2 Make correspondences between *similar* states of affairs in two domains as far as possible, when the heuristic H2.1 is not applicable. Similarity among relations is defined in terms of the relation *subtype*, by which the type hierarchy is defined.

H2.2.1 If in the target domain the assignment is not appropriate for the state of affairs σ obtained by applying the heuristic H2.1.1, generalize the relation r by obtaining a super type r_{sup} such that:

$$\sigma' = \langle\langle r_{sup}, \vec{x}; p\rangle\rangle \tag{2.28}$$

$$r \ \mathit{subtype} \ r_{sup} \tag{2.29}$$

from the type hierarchy and check whether the assignment is appropriate for this super type. If the condition is satisfied, the following constraint becomes one of the assumption:

$$\langle\langle \Rightarrow, \langle\langle r, \vec{x}; p\rangle\rangle, \langle\langle r_{sup}, \vec{x}; p\rangle\rangle\rangle\rangle \tag{2.30}$$

This generalization, however, gives us a weaker assumption.

H2.2.2 If the heuristic H2.2.1 is applicable, examine subtypes r_{sub} of the relation r_{sup} obtained by applying H2.2.1 such that:

$$\sigma'' = \langle\langle r_{sub}, \vec{x}; p\rangle\rangle \tag{2.31}$$

$$r_{sub} \ \mathit{subtype} \ r_{sup} \tag{2.32}$$

If the assignment is appropriate for one of the subtypes, r_{sub}, the following constraint becomes one of the assumption:

$$\langle\langle \Rightarrow, \langle\langle r, \vec{x}; p\rangle\rangle, \langle\langle r_{sub}, \vec{x}; p\rangle\rangle\rangle\rangle \tag{2.33}$$

This assumption is stronger than the result of H2.2.1.

H3 Coherence of objects in two domains

Each object in one domain corresponds to an object in another domain according to correspondences between relations. There must be no incoherence in these correspondences between objects. For example, suppose that the relations r^1_{target} and r^2_{target} in the target domain correspond to the relations r^1_{source} and r^2_{source} in the source domain respectively. Suppose also that the following states of affairs hold in the target domain and the source domain respectively.

$$\langle\langle r^1_{target}, w, x\rangle\rangle \wedge \langle\langle r^2_{target}, y, z\rangle\rangle \tag{2.34}$$

$$\langle\langle r^1_{source}, l, m\rangle\rangle \wedge \langle\langle r^2_{source}, n, o\rangle\rangle \tag{2.35}$$

While it is an appropriate case that the equations

$$x = y \ \ \& \ \ w \neq z$$
$$m = n \ \ \& \ \ l \neq o$$

are obtained from some other relations, it is an inappropriate case that the equations

$$x = y \ \ \& \ \ w \neq z \tag{2.36}$$

$$m \neq n \ \ \& \ \ l = o \tag{2.37}$$

are obtained, because objects in one domain do not correspond to objects in another domain coherently.

H4 Isomorphism of constraints

Make correspondences which preserve the isomorphism of constraints in two domains. For example, Suppose the following causal relations hold in the target domain and a source domain respectively.

$$\langle\langle \Rightarrow, A_{target}, B_{target} \rangle\rangle \quad (2.38)$$
$$\langle\langle \Rightarrow, A_{source}, B_{source} \rangle\rangle \quad (2.39)$$

It seems to be natural to assume that the consequence B_{target} corresponds with the consequence B_{source}, if it is known that the premise A_{target} corresponds to A_{source}, and vice versa.

H4.1 Strengthening the definition of constraints

Since the definition of the positive constraint, which is described in the section 2.3.2.2, is weaker than the implication of the predicate logic, we cannot combine several constraints into a new constraint. Therefore, the heuristic H4, which depends on the isomorphism of constraints, not always work well. In order to cope with this, we introduce the heuristic which strengthens the definition of the positive constraint only while the heuristic H4 is applied to make some correspondences. The strengthening is achieved by the following two restrictions.

- The situation s' which supports the state of affairs τ, which is carried relative to a constraint, should be a part of the situation s_0, which supports the constraint. That is, add the condition $s' \trianglelefteq s_0$ to the definition of the positive constraint.
- This definition should be bidirectional. Replace "if" in the definition of the positive constraint with "if-and-only-if".

These may correspond to the plausible restriction on the domain of consideration, with which we usually do not have to examine unrelated matters.

These restrictions contribute to the derivation of the following relations, which can be used in H4, rather than, to finding quit new constraints[4].

[4]The size of search spaces for finding quit new constraints based on these restrictions is often very large.

- Transitivity

$$s \models \langle\langle \Rightarrow, \sigma, \sigma' \rangle\rangle \wedge \langle\langle \Rightarrow, \sigma', \sigma'' \rangle\rangle$$
$$\rightsquigarrow \; s \models \langle\langle \Rightarrow, \sigma, \sigma'' \rangle\rangle \quad (2.40)$$

- Monotonicity

$$s \models \langle\langle \Rightarrow, \sigma, \sigma' \rangle\rangle \;\rightsquigarrow\; s \models \langle\langle \Rightarrow, \sigma \wedge \sigma'', \sigma' \rangle\rangle \quad (2.41)$$

- Conjoining of consequence

$$s \models \langle\langle \Rightarrow, \sigma, \sigma' \rangle\rangle \wedge \langle\langle \Rightarrow, \sigma, \sigma'' \rangle\rangle$$
$$\rightsquigarrow \; s \models \langle\langle \Rightarrow, \sigma, \sigma' \wedge \sigma'' \rangle\rangle \quad (2.42)$$

- Disjoining of consequence

$$s \models \langle\langle \Rightarrow, \sigma, \sigma' \wedge \sigma'' \rangle\rangle$$
$$\rightsquigarrow \; s \models \langle\langle \Rightarrow, \sigma, \sigma' \rangle\rangle \wedge \langle\langle \Rightarrow, \sigma, \sigma'' \rangle\rangle \quad (2.43)$$

- Weakening

$$s \models \langle\langle \Rightarrow, \sigma, \sigma' \rangle\rangle$$
$$\rightsquigarrow \; s \models \langle\langle \Rightarrow, \sigma \wedge \sigma'', \sigma' \wedge \sigma'' \rangle\rangle \quad (2.44)$$

where $\rightsquigarrow$ means that assuming the restriction described above, the right hand side is derived from the left hand side logically.

The following points may be paid attention to when these heuristics are used.

- We consider that a *partial* correspondence between two domains will be enough for metaphor understanding. Here, by "*partial*" we mean:
  - One domain is allowed to have some relations which correspond to no relations in another domain.

- All argument roles about a relation do not have to be considered. That is, it may be sufficient that projections of relations in one domain correspond to projections of relations in another domain. Because an object which corresponds to another object in the source domain does not have to exist in target domain as far as it is not referred in metaphorical expressions.

- Some measure, that is, some evaluating functions for application of heuristics may be required for the efficient search of a plausible correspondence.

2.3.3.4 An Example of Establishing Correspondences

Let us examine correspondences for the example sentence. As described in the section 2.2, the described situation s^1_{uf} forms a certain part of the parallel context s^1_p in parallel with the main context s_d. Therefore correspondences we should obtain are those that connect states of affairs which hold in s^1_{uf} to states of affairs which meet the requirements of Step-2 in the context s^0_d relative to the conditions (2.23), (2.24) and (2.25).

First let us spell out the states of affairs which hold in s^1_{uf} under the condition (2.25). The states of affairs which hold in s^1_{uf} are given in (2.5). By applying the constraints in T_{misc} to s^1_{uf}, it is derived that there exists a situation s'_d such that (2.20). Now, supposing that s'_d is also used as a part of the parallel context, the condition $s'_d = s^1_{uf}$ should be satisfied. Under the condition, states of affairs holding in s^1_{uf} are as follows:

$$s^1_{uf} \models \langle\langle \textit{with, agent:A, object:ST} \rangle\rangle \wedge \quad (2.45)$$
$$\langle\langle \textit{strategy, object:ST} \rangle\rangle \wedge \quad (2.46)$$
$$\langle\langle \textit{method, object:ST, event-type:}ET_{use\ m.p.} \rangle\rangle \wedge \quad (2.47)$$
$$\langle\langle \textit{use, agent:A, object:mp} \rangle\rangle \wedge \quad (2.48)$$
$$\langle\langle \textit{mil-power, object:mp} \rangle\rangle \quad (2.49)$$

Let us construct the constraints which connect the states of affairs (2.45),...,(2.49) to the states of affairs which may hold in the target

domain. For the states of affairs (2.45) and (2.48), the following constraints are obtained by the heuristic H2.1, which generates correspondences related to the states of affairs which can be used also in the target domain:

$$C^1_{w\to a} = \{\langle\langle \Rightarrow, \langle\langle \textit{with, agent:A, object:ST} \rangle\rangle, \langle\langle \textit{with, agent:A, object:ST} \rangle\rangle \rangle\rangle, \quad (2.50)$$
$$\langle\langle \Rightarrow, \langle\langle \textit{use, agent:A, object:mp} \rangle\rangle, \langle\langle \textit{use, agent:A, object:y} \rangle\rangle \rangle\rangle\} \quad (2.51)$$

To the states of affairs (2.49), (2.47) and (2.46), however, the heuristic H2.1 cannot be applied, since they cannot be used in the target domain for the following reasons:

- (2.49) violates the constraint (2.15).
- (2.47) leads to (2.49) with both (2.45) and (2.14).
- (2.46) leads to (2.47) with (2.11).

For the state of affair (2.49), the correspondence constraint is obtained by comparing s^1_{uf} with the main context s^0_d under the correspondences (2.50) and (2.51). Under the condition (2.24) the main context s^0_d is as follows:

$$s^0_d \models \langle\langle \textit{argue, agent:A, participant:B, object:TH} \rangle\rangle \wedge \langle\langle \textit{theory, TH} \rangle\rangle \wedge \langle\langle \textit{own, agent:B, object:TH} \rangle\rangle \wedge \langle\langle \textit{use, agent:A, object:y} \rangle\rangle \wedge \langle\langle \textit{reasoning, object:y} \rangle\rangle \quad (2.52)$$

Since the parameter mp in the source domain is connected to the parameter y in the target domain by (2.51), by applying the heuristic H3, the following constraint is obtained:

$$C^2_{w\to a} = \{\langle\langle \Rightarrow, \langle\langle \textit{mil-power, object:mp} \rangle\rangle, \langle\langle \textit{reasoning, object:y} \rangle\rangle \rangle\rangle\} \quad (2.53)$$

Since the relation *method* itself can be used also in the target domain, for the state of affair (2.47) the correspondence constraint is obtained by translating the situation type, which is in the argument of the state of affair and is related to the source domain, into the situation type related to the target domain. Since the correspondence constraints (2.51) and (2.53) are applied to the any situation s, such that $s{:}ET_{use\ m.p.}$, related to the target domain, it is derived that there exists a situation s' such that $s'{:}ET_{use\ reas}$ for every situation s, where

$$ET_{use\ reas} = \quad [s \mid s \models \langle\langle use, agent{:}y, object{:}reas\rangle\rangle \wedge \langle\langle reasoning, object{:}reas\rangle\rangle] \tag{2.54}$$

We suppose the following constraint, for s' can be regarded as a situation related to the target domain:

$$C^3_{w\to a} = \{\langle\langle \Rightarrow, \langle\langle method, object{:}ST, event\text{-}type{:}ET_{use\ m.p.}\rangle\rangle, \langle\langle method, object{:}ST, event\text{-}type{:}ET_{use\ reas}\rangle\rangle\rangle\rangle\} \tag{2.55}$$

Lastly we examine the case of the state of affairs (2.46). It seems to be natural that we assume the following constraint about the "logic" holds in a situation s which is of the situation type T_{argue} about the "argument".

$$\langle\langle \Rightarrow, \langle\langle logic, object{:}x\rangle\rangle, \langle\langle method, object{:}x, event\text{-}type{:}ET_{use\ reas}\rangle\rangle\rangle\rangle \tag{2.56}$$

Now, applying the heuristic H4 to the constraints (2.11) and (2.56), the following correspondence constraint are obtained:

$$C^4_{w\to a} = \{\langle\langle \Rightarrow, \langle\langle strategy, object{:}ST\rangle\rangle, \langle\langle logic, object{:}ST\rangle\rangle\rangle\rangle\} \tag{2.57}$$

We have obtained all constraints which connect each state of affairs holding in s^1_{uf} to a state of affairs related to the target domain "argument".

2.3.4 Metaphor Understanding with Resources Related to Correspondences

The resource situation s^1_r in Step-4 becomes the situation which supports the constraints $C^1_{w\to a}, \ldots, C^4_{w\to a}$ in the previous subsection. In

order to apply these constraints to s^1_{uf}, suppose the condition $s^1_r = s^1_{uf}$. Then it is derived that there exists the following situation $s^{1,t}_{uf}$:

$$s^{1,t}_{uf} \models \langle\langle with, agent{:}A, object{:}ST\rangle\rangle \wedge \tag{2.58}$$
$$\langle\langle logic, object{:}ST\rangle\rangle \wedge \tag{2.59}$$
$$\langle\langle method, object{:}ST, event\text{-}type{:}ET_{use\ reas}\rangle\rangle \wedge \tag{2.60}$$
$$\langle\langle use, agent{:}A, object{:}y\rangle\rangle \wedge \tag{2.61}$$
$$\langle\langle reasoning, object{:}y\rangle\rangle \tag{2.62}$$

This situation corresponds to a described situation related to the target domain. Return to Step-2 after replacing the described situation s^1_{uf} with $s^{1,t}_{uf}$. Then all requirements in Step-2 are satisfied under the condition $s^{1,t}_{uf} = s^0_d$. Therefore $s^{1,t}_{uf}$ forms part of the new main context s^1_d such that:

$$s^1_d = s^0_d = s^{1,t}_{uf} \tag{2.63}$$

Generally speaking, the semantic representation of metaphor understanding can be formulated as the following constituents:

- A described situation s^i_{uf} related to the source domain, that is , a part of the parallel context
- An application of the background knowledge of the source domain $s^i_{uf}{:}T_{source}$
- A resource situation s^i_r which supports constraints representing the correspondence between two domains
- A described situation $s^{i,t}_{uf}$ related to the target domain, which satisfies the following conditions:

$$s^i_r = s^i_{uf} \tag{2.64}$$
$$s^i_{uf} \Vdash_{s^i_r} \sigma \tag{2.65}$$
$$s^{i,t}_{uf} \models \sigma \tag{2.66}$$

- The following condition, which represents the construction of a new context:

$$s^i_d = s^{i-1}_d = s^{i,t}_{uf} \tag{2.67}$$

- An application of the background knowledge of the target domain $s_d^i{:}T_{target}$

2.3.5 Understanding the Remaining Described Situations

In this section, we will briefly examine the interpretation for the remaining described situations s_{uf}^2 and s_{uf}^3. First, let us consider s_{uf}^2 such that (2.6). The described situation s_{uf}^2 satisfies each requirement in Step-2 under the condition $s_{uf}^2 = s_d^1$, which is imposed in Step-2. Thus we obtain a new main context s_d^2 as follows:

$$s_d^2 = s_d^1 = s_{uf}^2 \tag{2.68}$$

Secondly, let us examine s_{uf}^3 such that (2.7). By taking s_{uf}^3 into account, we should describe the situation types T_{war} and T_{argue}, which have been already used as the background knowledge, in more detail. For the situation type T_{war}, we suppose that a situation of T_{war} holds not only (2.46), but also the following states of affairs:

$$\begin{aligned}
&\langle\langle \Rightarrow, \langle\langle \mathit{wipe_out, agent{:}a, patient{:}b, object{:}z}\rangle\rangle, \\
&\quad \langle\langle \mathit{own, agent{:}b, object{:}z}\rangle\rangle \wedge \\
&\quad \langle\langle \mathit{make, agent{:}a, event{:}E_{not_func}}\rangle\rangle \wedge \\
&\quad \langle\langle \mathit{can, agent{:}b, event\text{-}type{:}ET_{repair}}; -\rangle\rangle\rangle\rangle \wedge && (2.69) \\
&\quad \langle\langle \mathit{purpose, agent{:}a, event{:}E_{make_not_func}}\rangle\rangle && (2.70)
\end{aligned}$$

where

$$\begin{aligned}
&E_{not_func}{:}ET_{not_func} && (2.71)\\
ET_{not_func} &= [s \mid s \models \langle\langle \mathit{function, object{:}z}; -\rangle\rangle] && (2.72)\\
ET_{repair} &= [s \mid s \models \langle\langle \mathit{repair, agent{:}b, object{:}z}\rangle\rangle] && (2.73)\\
&E_{make_not_func}{:}ET_{make_not_func} && (2.74)\\
ET_{make_not_func} &= [s \mid s \models \langle\langle \mathit{make, agent{:}a, event{:}E_{not_func}}\rangle\rangle] && (2.75)
\end{aligned}$$

Similarly, for the situation type T_{argue}, we suppose that a situation of T_{argue} holds not only (2.13), but also the following states of affairs:

$$\begin{aligned}
&\langle\langle \Rightarrow, \langle\langle \mathit{refute, agent{:}a, patient{:}b, object{:}w}\rangle\rangle, \\
&\quad \langle\langle \mathit{own, agent{:}b, object{:}w}\rangle\rangle \wedge \\
&\quad \langle\langle \mathit{prove, agent{:}a, event{:}E_{not_func}}\rangle\rangle\rangle\rangle \wedge && (2.76)\\
&\quad \langle\langle \mathit{purpose, agent{:}a, event{:}E_{prove_not_func}}\rangle\rangle \wedge && (2.77)\\
&\quad \langle\langle \mathit{make, agent{:}a, event{:}E_{not_func}}; -\rangle\rangle && (2.78)
\end{aligned}$$

where

$$ET_{prove\_not\_func} = [s \mid s \models \langle\langle \mathit{prove, agent{:}a, event{:}E_{not\_func}}\rangle\rangle] \tag{2.79}$$

and (2.78) means "in an argument, an agent does not damage anything. " Since the constraint about the state of affairs *wipe_out* holds in a situation of the type T_{war}, which represents the source domain, we can suppose that s_{uf}^3 is a described situation related to the target domain, namely, a part of the parallel context as the following conditions:

$$\begin{aligned}
&s_{uf}^3{:}T_{war} && (2.80)\\
&s_{uf}^3 \neq s_d^2 && (2.81)\\
&s_p^3 = s_{uf}^3 = s_p^1 && (2.82)
\end{aligned}$$

Since the constraint (2.69) is applied to s_{uf}^3 by these conditions, it is derived that there exists a situation $s_{uf}^{3'}$ such as:

$$\begin{aligned}
s_{uf}^{3'} \models\ &\langle\langle \mathit{own, agent{:}B, object{:}z}\rangle\rangle \wedge && (2.83)\\
&\langle\langle \mathit{make, agent{:}A, event{:}E_{not_func}(f)}\rangle\rangle \wedge && (2.84)\\
&\langle\langle \mathit{can, agent{:}B, event\text{-}type{:}ET_{repair}(f)}; -\rangle\rangle && (2.85)
\end{aligned}$$

where f is an anchor such that $f(a) = A$ and $f(b) = B$. We also assume that $s_{uf}^{3'}$ constitutes a part of the parallel context, that is,

$$s_p^3 = s_{uf}^3 = s_p^1 = s_{uf}^{3'} \tag{2.86}$$

Now, the constraints which connect the states of affairs (2.7), (2.70), (2.83), (2.84) and (2.85) which hold in s_{uf}^3 to the states of affairs of the target domain are obtained as follows:

$$\begin{aligned}
&C_{w \to a}^5 = \\
&\quad \{\langle\langle \Rightarrow, \langle\langle \mathit{own, agent{:}B, object{:}z}\rangle\rangle,
\end{aligned}$$

$$\langle\langle own, agent{:}B, object{:}TH\rangle\rangle\rangle\rangle, \tag{2.87}$$

$$\langle\langle \Rightarrow, \langle\langle make, agent{:}A, event{:}E_{not\_func}(f)\rangle\rangle, \langle\langle prove, agent{:}A, event{:}E_{not\_func}(f)\rangle\rangle\rangle\rangle, \tag{2.88}$$

$$\langle\langle \Rightarrow, \langle\langle can, agent{:}B, event\text{-}type{:}ET_{repair}(f); -\rangle\rangle, \langle\langle can, agent{:}B, event\text{-}type{:}ET_{repair}(f); -\rangle\rangle\rangle\rangle, \tag{2.89}$$

$$\langle\langle \Rightarrow, \langle\langle purpose, agent{:}A, event{:}E_{make\_not\_func}(f)\rangle\rangle, \langle\langle purpose, agent{:}A, event{:}E_{prove\_not\_func}(f)\rangle\rangle\rangle\rangle, \tag{2.90}$$

$$\langle\langle \Rightarrow, \langle\langle wipe\_out, agent{:}A, patient{:}B, object{:}z\rangle\rangle, \langle\langle refute, agent{:}A, patient{:}B, object{:}TH\rangle\rangle\rangle\rangle\} \tag{2.91}$$

The constraints (2.87),(2.89) and (2.90) are obtained by applying the heuristic H2.1. The constraint (2.88) is obtained as the result of the heuristic H3 by taking account of the correspondence (2.90). The constraint (2.91) is obtained by using the heuristic H4 under the correspondence (2.69), (2.76),(2.87), and (2.88). Thus, the situation supporting the constraints $C^5_{w \to a}$ becomes the resource situation s^3_r. In order to apply the constraints to s^3_{uf}, suppose the condition $s^3_r = s^3_{uf}$. Then there exists a situation $s^{3,t}_{uf}$ such that:

$$s^{3,t}_{uf} \models \langle\langle own, agent{:}B, object{:}TH\rangle\rangle \wedge \langle\langle prove, agent{:}A, event{:}E_{not\_func}(f)\rangle\rangle \wedge \langle\langle can, agent{:}B, event\text{-}type{:}ET_{repair}(f); -\rangle\rangle \wedge \langle\langle refute, agent{:}A, object{:}TH\rangle\rangle \tag{2.92}$$

This situation corresponds to the described situation related to the target domain.

As the result of understanding the example sentence, the situation which satisfies the following condition is obtained as the main context:

$$s^0_d = s^{1,t}_{uf} = s^2_{uf} = s^{3,t}_{uf} \tag{2.93}$$

and the situation which satisfies the following condition is obtained as the parallel context:

$$s^1_{uf} = s^3_{uf} \tag{2.94}$$

where

$$s^0_d \neq s^1_{uf}$$
$$s^0_d{:}T_{argue} \wedge s^0_d{:}T_{misc}$$
$$s^1_{uf}{:}T_{war} \wedge s^1_{uf}{:}T_{misc}$$

2.4 Conclusion

In this chapter, we give a formal description of metaphor understanding in terms of situation semantics. In this formulation, we propose the model in which the hearer introduces correspondences between a target domain and a source domain as a new resource to obtain useful information about the target domain in terms of source domain. We expect that this formal description of metaphor understanding may serve as a guide to natural language understanding systems which can understand discourses with metaphors. Since we have concentrated upon a formal description in this chapter, we have not mentioned how to search a plausible correspondence. The concrete method of how to search a plausible correspondence will be developed in our future work. (As related works of plausible search in metaphor understanding, several methods have been proposed[TIT89, Ind87].)

Chapter 3

A Parsing Model based on Connectionist Model

3.1 Introduction

In order to make clear the human parsing mechanism for natural language sentences, the *human parsing process* has been discussed from various points of view by a number of researchers. From the viewpoint of *efficiency*, on the ground that we can usually parse sentences in real time, several fast parsing algorithms, such as the top-down parser[PW80], left-corner parser[JL83], chat parser, shift-reduce parser[Shi83], Earley's parser[Ear70], Tomita's parser[Tom87] and so on, has been developed[SSS88]. On the other hand, several researchers have paid attention to some remarkable phenomena in the human parsing process, since these phenomena may be clues to the explication of the high efficiency on the human parsing process. These phenomena include *cognitive difficulties to recognize garden path sentences or deeply nested sentences*, and *preference of structurally ambiguous sentences*[1]. In order to explain these phenomena, some parsing algorithms introduce *heuristics* which agree with these phenomena. For example, in some models, such as Marcus's wait-and-see parser[Mar80] and Shieber's shift-reduce parser[Shi83], new parsing algorithms based on some heuristics are proposed. Some other models are built by introducing some ad-hoc 'princi-

[1]See 3.4 for the details of each phenomenon.

ples' into one of efficiency-oriented parsing algorithms. However, these parsing models have not yet succeeded to explain all of phenomena, described above, under one simple integrated principle. Since these phenomena should be due to delicate aspects of *performance*, or actual usage of language, it is hard to model them with mechanisms which uniformly and sequentially handle all of sentences generated by some formal grammer rules, such as context free grammers.

In this chapter, in view of this point, we will propose a parsing model based on *connectionist model*[2], in order to explain such delicate aspects of performance, which cannot be model by a formal symbolic paradigm, such as some grammers. CM is a massively parallel computational model, in which many simple computing elements are work simultaneous by interacting each other. Therefore, a parser based on CM may have several intermediate syntactic structures at one time, and moreover these intermediate structures can interact each other. In our CM parser proposed in this chapter, each syntactical category, like a noun phrase, a verb phrase and so on, is represented as one computing element in CM, and the state of activation of a computing element is regarded as the degree to which the hearer now pays attention to the phrase structure corresponding to the computing element. In other words, we stand for the what is called *localist* view, in which one symbol corresponds to one computing element. In order to model the process in which, as a new word is recognized, phrase structures grow larger dynamically, we introduce the *linking mechanism*, which makes connections dynamically. While previous CM parsers, for example, Waltz's natural language understanding model[WP85], Fanty's parser[Fan85] and so on, take account of activation patterns only on given phrase trees, with the linking mechanism our parser furthermore can represent the strength of connections on a phrase structure, which are dynamically made from substructures, as one activation pattern. So, with this parser, we can analyze interesting phenomena described above in detail. Moreover, Since a subnetwork corresponding to a phrase structure rule is directly assigned to a brunch of a parse tree, the *predication for the next category* and the *reachability* of the predicted category from a

[2]Hereafter, we will use the acronym 'CM' to refer to this model, for short.

new word[3] are expressed as the states of activation of some computing elements.

3.2 Survey of Connectionist Model

Connectionist Model is a parallel computing model in which a large number of simple computing elements[4] interact each other. Each unit u_i is an n-input/1-output processor with one internal state $a_i(t)$, which is called an *activation value*. The activation value is passed through an output function f_i to produce an output value $o_i(t)$. This output value can be seen as passing through a set of unidirectional *connections*(or *link*) [5] to other units. There is associated with each connection a real number, usually called the *weight* or *strength* of the connection. The weight of the connection from the unit u_j to the unit u_i is designated by w_{ji}. A connection of which weight w_{ij} is positive is called the *activation connection* and negative one is called the *inhibition connection*. The activation value $a_i(t+1)$ of the unit u_i at the next time $t+1$ is given as follows:

$$a_i(t+1) = \sum_j w_{ji} \times f_j(a_j(t)) \qquad (3.1)$$

Another kind of units, called *sigma-pi units*, have the net input which is given by the weighted sum of the products of a set of individual inputs, or $\sum w_{ji} \Pi f_k(a_k)$. Such a multiplicative connection allows one unit to *gate* another[6]. Sigma-pi units can be used to convert the output level of a unit into a signal that acts like the weight of a connection connecting some other units. Thus, sigma-pi units allow for a dynamically programmable network in which the activation value of some units determine what another network can do.

While CM is an application of the results of recent research on neural networks to a model of our cognitive process from the view of the

[3]If the new word can be the leftmost leaf of one of subtrees of which root is the predicted category, the predicted category is *reachable* from the new word.

[4]They may be simply called *units*. We will denote them as circles (○) in our diagrams.

[5]They are denoted by arrows (→) in our diagrams.

[6]These multiplicative connections are denoted by triangles (△) in our diagram.

parallel processing, we should not consider that our cerebrum just has the mechanism described in this chapter as some organic part. Rather, in our parsing model, we should regard CM as a scheme with which we can appropriately explain how the *focus of attention* shift. That is, the state of activation of a unit should be regarded as the degree of attention to something and a network of units should be regarded as constraint on the shift of attention.

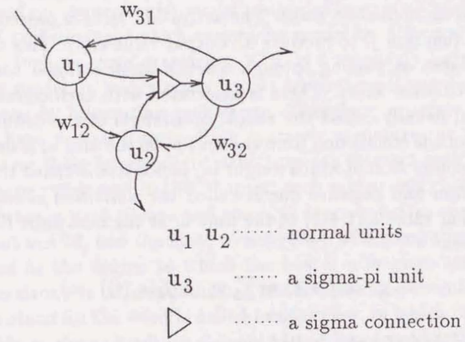


Figure 3.1: Connectionist model

3.3 Parsing Mechanism

Our parsing model is founded on the following hypothesis about the characteristic of the human parsing process:

Immediate Effect Hypothesis
Whenever the parsing process encounters a new word, without any delay the process constructs all of syntactic structures which the process can do from both previous structures and the word.

We consider that this hypothesis should be one of reason for *efficiency* of the human parsing process[7]. It is derived form adopting the hypothesis that all of possible structures for structurally ambiguous sentence should be deal with by the parsing process simultaneously[WP85]. Humans experience an increased processing load with ambiguous language, which suggests that humans compute multiple readings at least in some sense. Another interesting phenomenon in language interpretation is that humans can usually entertain only one interpretation of an ambiguous sentence at a time, but can easily 'flip' between interpretations[WP85]. From these facts, we can consider that all of possible structures for a structurally ambiguous sentence are exclusive mutually, and at a time humans are conscious of only one structures from among them. Since in our model the degree of attention to a structure is represented as the activation of the unit corresponding to the structure, we would like to adopt the following hypothesis:

Preference Hypothesis
Among structures for a structurally ambiguous sentence, the most activated structure are preferred to others.

3.3.1 Representation of parsing trees in CM

Given a network of units and some entities to be represented, there are at least two scheme to represet these entities. One way is called a *local* representation, another is called a *distributed* representation. As mentioned in the section 3.1, we adopt the former scheme. In our local representation scheme, we suppose that one unit is assigned to each syntactic category[8] and its state of activation shows to what degree the corresponding structure have been made. We will call these units *category units*. In other words, a parse tree is represented as a network in which appropriate activating category units are connected. In order to parse sentences by CM network, we have to realize a set of phrase structure rules in terms of CM network. For example, let us consider

[7] This standpoint may share the same origin with the efficiency thesis in Hashida[Has85].

[8] To be exact, one unit are assinged to one part of a parse tree, which belongs to a syntactic category.

the following phrase structure rule:

$$< C > \rightarrow < A >, < B > \quad (3.2)$$

This rule contains at least the following two constraints upon the category A, B and C:

1. The category A is prior to the category B.
2. The category C consists of the category A and B.

We can realize these constraint in CM network as follows. As for the constraint 2, since the dependency of the category C upon its subcategories A,B corresponds to the dependency of the activation of corresponding units, we can express the dependency simply by two activation connections, that is, the connection between the unit for C and the unit for A and the connection between the unit for C and the unit for B. On the other hand, since only with normal activation connections we cannot control the order of activation of two units, as for the constraint 1, we introduce a new kind of connection which restricts the order of activation of two units. With this connection, a category unit can 'trigger' the next category, that is, the full activation of the former unit permits the latter unit to be activated. So we call this kind of connection *trigger connection* and denote it by "$\xrightarrow{t}$". Now, the phrase structure rule (3.2) can be translated into a subnetwork of CM in Figure 3.2. We call this subnetwork a *phrase structure network*, and a *PSN* for short, hereafter. In this figure, the weight w_{CA} (w_{AC}) and w_{CB} (w_{BC}) are positive numbers less than 1, which express how important the category A and B are for the category C, respectively. The weight w_{AB} is also a positive numbers less than 1, which expresses to what degree the category B is predicted when the category A is occurred. Trigger connections can not be realized by normal units, but by sigma-pi units. Using a trigger connection, which is realized by a sigma-pi unit, a phrase structure network can be expressed in more detail as Figure 3.3.

3.3.2 Basic Mechanisms to Construct Parse Trees

When phrase structure networks, described the previous section, are connected appropriately to grow a parse tree according to each input

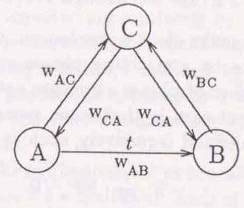


Figure 3.2: A phrase structure network

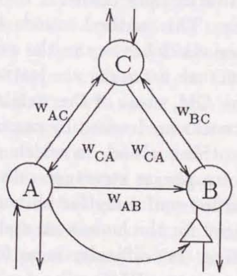


Figure 3.3: A phrase structure network realized by a sigma-pi unit

word in a sentence, we can parse the sentence as the result. In this section, we will present basic mechanisms to do this.

3.3.2.1 Copying Phrase Structure Networks

Our final goal is to make clear a mechanism of building a parse tree for a whole sentence by connecting phrase structure networks. For this purpose, a number of phrase structure networks of same type are required. The typical case is that some parse structure rules in the given grammer are defined recursively, such as the following rules:

$$\mathrm{S} \rightarrow \mathrm{NP}, \mathrm{VP} \quad (3.3)$$
$$\mathrm{NP} \rightarrow \mathrm{NP}, \overline{\mathrm{S}} \quad (3.4)$$
$$\overline{\mathrm{S}} \rightarrow \mathrm{Comp}, \mathrm{S} \quad (3.5)$$

The simplest method is to prepare a number of copies of a phrase structure network for each phrase structure rule in advance. For example, ten phrase structure networks of the rule (3.3), ten phrase structure networks of the rule (3.4), and so on. When a parser reads a sentence, it select some phrase structure networks from these prepared phrase structure networks, and connects them to make a parse tree of the input sentence. This method sounds all right. Unfortunately, this method has a serious deficiency in the explanation of the process in which phrase structure networks are learned. From the the point of view of learning in CM, most of the weights of connection links in phrase structure networks are learned by parsing or recognizing a number of sentences. In this method, in which many copies of a phrase structure network for a phrase structure rule should be prepared, the weights must be learned uniformly for all copies. But this uniformly learning is too artificial for the human mental learning processes.

A solution avoiding this difficulty is as follows. Only one phrase structure network is prepared for each phrase structure rule and all learning processes are done on it. In parsing, when a parser needs a phrase structure network of some rule, the parser makes copies of the phrase structure network and connects them appropriately. A network copying mechanism can be implemented as an application of the *connection information distribution* (CID) mechanism[MRtPRG86]. The basic parts of the mechanism are shown and labeled in Figure 3.4. The CID mechanism consists of three parts, that is, a *central module* (CMD), a *connection activation system* (CA) and a set of *programmable modules* (PM). A central module is a network with permanent connections, of which weights are adjusted appropriately by the learning processes. On the other hand, a programmable module is a network with programmable connections, which are realized by sigma connections in sigma-pi units. The purpose of a connection activation system is to distribute the connection information of the central module to each programmable modules.

In order to grasp the mechanism, let us consider the case that the left input on a PM in Figure 3.4 is activated. Each of the connections 1,...,4 from the CA is one part of a multiplicative connection in the PM, so that the actual weight of the multiplicative connection is in proportion to the activation value on the connection from the CA. At first, the activation values on the connections 1,...,4 are 0, and the weights of the connections between the inputs and the outputs in the PM are 0. Therefore, the activation at the left input of the PM are propagated not to outputs, but to the left input of the CMD and the left input of the CA. According to the permanent connections in the CMD to be copied, the activation at the left input is propagated to the upper output of the CMD. The CAS is used to extract the connection information in the CMD from both the outputs of the CMD and the outputs of PM's. In this case, since the upper output of the CMD and the left input of the CA are activated, the left upper unit in CA is activated and this activation is propagated to the connection 1. The activation can be regarded as the construction of the left upper connection in the PM, for the weight of the connection is proportional to the value of the activation. So the activation to the left input of the PM is propagated to the upper output through the connection. In this case, we can consider that the weight of left upper connection in the PM is copied from the corresponding connection in the CMD.

3.3.2.2 Connecting Phrase Structure Networks

To grow a parse tree according to the inputs of words, we need a mechanism to generate connections dynamically. Unfortunately, CM has

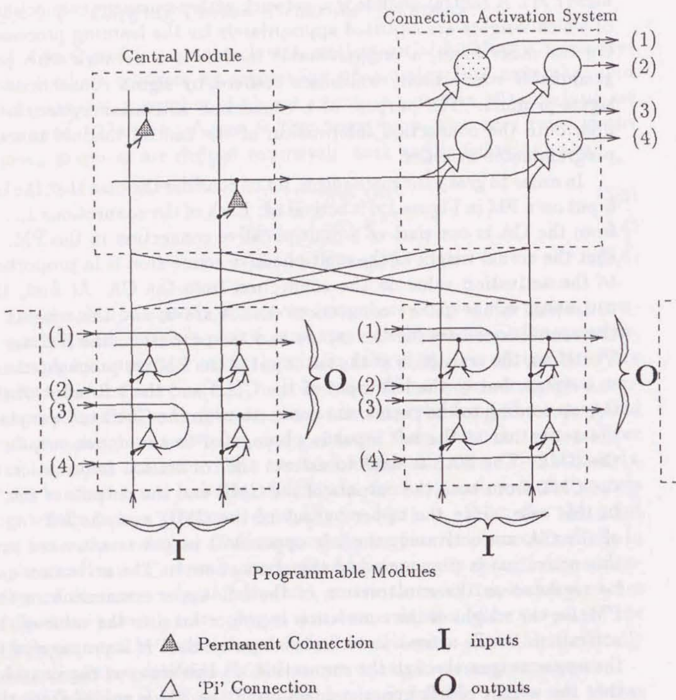


Figure 3.4: A simple connection information distribution (CID) mechanism

not yet had this mechanism. Instead of this mechanism, we can use sigma-pi units to change weights of linking connections dynamically. If a connection of which actual weight is normally 0 can be changed to the positive weighted connection by the activation of a control unit, we can consider that an activation connection is *generated virtually*. We call the linking mechanism by the notion of this virtual generation *connectors*. The basic function of a connector is to link one activated unit to another unit which is ready to be activated. In order to do this without any supervisor, we use the following two kinds of connector:

1. *Requesting connectors*
   When the unit which has this connector is activated, the connector sends the request for linking to all of accepting connectors which is associated with the requesting connectors, and waits for the response 'accepted' from these accepting connectors. Where, these operations are expressed as spreading activation via some connections. The connector establishes connections with the accepting connectors which reply the message 'accepted' to it.

2. *Accepting connectors*
   An accepting connector responds to requests from the requesting connectors which are associated with the connector. When a accepting connector receives requests from requesting connectors, if the unit which has the accepting connector is permitted to be activated, the accepting connector sends the message 'accepted' to each of the requesting connectors of which request meets the *acceptance condition*

The acceptance condition in accepting connectors can be regarded as the algorithm for the dynamically linking mechanism.

3.3.3 Parsing Algorithm

The algorithm of our CM parser consists of the following parts:

- The connections in each of phrase structure networks, which correspond to one of the phrase structure rules.
- The connectors which link phrase structure networks dynamically.

Our parser adopts the following type of context free grammers as a set of phrase structure rules:

$$< C > \rightarrow < A >, < B > \tag{3.6}$$

$$< D > \rightarrow \text{`word'} \tag{3.7}$$

where $< A >$, $< B >$, $< C >$ and $< D >$ are syntactic categories, and 'word' is one of the lexical items in a given lexicon. This kind of restricted context free grammar is called *Chomsky normal form*[BW84]. It is known that for any context free language there exists at least one context free grammer in Chomsky normal form which can generate the context free language.

For convenience, We will refer to categories in the same position with $< A >$ as *A-categories* and also we will refer to units corresponding to A-categories as *A-units*. We also use the same notation to $< B >$ and $< C >$ and treat $< D >$ just like $< C >$.

Figure 3.5 shows our network for parsing. It consists of the following parts:

- *An original phrase structure network*, which is a central module in the CID mechanism.
- A number of *'blank' phrase structure networks*, which are programmable modules in the CID mechanism and of which connection information are copied from the original phrase structure network.
- *A lexicon*, which is a set of subnetworks corresponding to lexical items.

In this parser, inputing a word of a sentence is expressed as activating the category unit in the lexicon corresponding to the word. We also assume that the input to the category unit, which is given at the point corresponding to the time when the word is inputted, is not retained at a constant value, but dumping by a certain time constant. This hypothesis can be considered that when we read/hear a sentence, stimuli by all of the recognized words are not given at a time, rather, each of these stimuli is recognized for a certain time span and the focus of

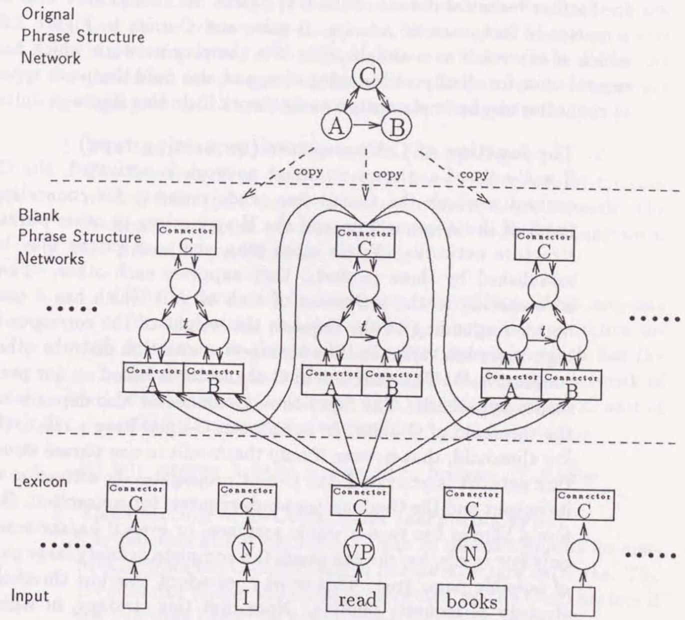


Figure 3.5: The connectionist model parser

recognition moves whenever a new word is given as a new stimulus. In this parser, we use three types of connector, that is, A-connectors (accepting type), B-connectors (accepting type) and C-connectors (requesting type) for each phrase structure network in order to connect phrase structure networks dynamically[9]. While in this chapter we omit the further technical details of the CM parser, we should note that as a matter of fact, each of A-units, B-units and C-units in Figure 3.5, which is expressed as a simple unit, is a complex network which has several units for all of possible categories, and also note that each types of connector can be implemented as a network including sigma-pi units.

1. **The function of C-Connectors (requesting type)**
   If a C-unit of a phrase structure network is activated, the C-connector which the C-unit has sends requests for connection to all of the A-connectors and the B-connectors in other phrase structure networks. While more than one connections may be established by these requests, they suppress each other. That is, according to the activation of each of unit which has a connector responding to the request, the weight of the corresponding connection varies and the strongest connection disturbs other connections[10]. This function of C-connector is based on our preference hypothesis. The function of C-connector also depends on the threshold of C-units. In our parser, C-units have a relatively low threshold, that is, even if only the A-unit in one phrase structure network is activated, the C-unit propagates its activation to its output and the C-connector sends requests for connection. Before a human has read a whole sentence, or even if he/she reads only few words, he/she can predicts a complete or fairly large part of possible parse tree. This is why we adopt this low threshold strategy of requests sending. Note that this strategy, in which even from incomplete subtrees its upper parts in a parse tree can be built, includes the check of the reachability of the predicted category as follows. When a word is inputted, the incomplete trees which has the category of the word as the leftmost leaf grow upward. If the root of a grown tree can be linked with the category unit which has been triggered as a predicted category, the tree, which may be part of the admissible tree of a whole sentence, can receive the activation feedback from the unit of the predicted category and the activation of the tree is strengthened. On the other hand, other grown trees, of which root cannot be linked with the predicted category, cannot receive such a activation feedback and its activation will dump. In other words, whenever a new word is inputted, several partial trees which include one of the path from the category of the word to the predicted category are built and their states of activation are strengthened.

[9] See also Figure 3.5.

[10] The suppression is denoted as a dotted arrow (<· · · · >) between connections.

2. **The function of A-Connectors (accepting type)**
   The function of an A-connector is to receive a request for connection from the C-connector in a phrase structure network. The A-connector replies the message 'accepted' to the C-connector if the following condition are satisfied:

   - The A-connector has not yet received any other requests for connecting, and it is possible to copy the connection information to the phrase structure network, which has the A-connector, from the original phrase structure network of which A-category is same with the category of the C-unit of which C-connector is requesting.

   By this copying, a parse tree grows in bottom-up manner.

3. **The function of B-Connectors (accepting type)**
   The function of a B-connector is also to receive requests for connection from the C-connectors in phrase structure networks. The B-connector replies the message 'accepted' to the C-connectors if the following condition are satisfied:

   - The B-category of the B-unit is same with the category of the C-unit of which C-connector is requesting.
   - The B-unit has been triggered and it is ready to be activated.

As mentioned before, each units in a phrase structure network is actually a network. A B-unit has several internal units in order to accept multiple requests. These internal units are connected each other with inhibition connections. Therefore, if more than one connections are established by the B-connector, these candidates for the predicted category suppress each other[11]. This inhibition is also based on the preference hypothesis.

3.3.4 Parsing on the CM Parser

To summarize the CM parser described above, we sketch a process in which the CM parser parses the following sentence to generate the parse tree in Figure 3.6:

"*I read books.*"

The phrase structure rules used in this example is as follows:

$$S \rightarrow N, VP \quad (3.8)$$

$$VP \rightarrow V, N \quad (3.9)$$

1. When the CM parser reads "I", the unit N_1 is activated and the C-connector of the unit N_1 sends requests for connection to A-connectors and B-connectors in other units. While there is no B-unit which has been triggered to be ready to be activated, there exists the phrase structure rule (3.8), which has the category N as the A-category, and its connection information can be copied from the original phrase structure network to a blank phrase structure network, that is, the programmable module which is not used at that point. Then, the A-connector which is associated with the copied phrase structure network replies the message 'accepted' to the C-connector which is requesting, and the connection between these connectors is established. Since the unit N_2 is fully activated, the unit triggers the next category unit VP_1 and the unit VP_1 is ready to be activated now. The unit S is also activated

[11]In our diagrams, the inhibition among units are denoted as square brackets ('[···]').

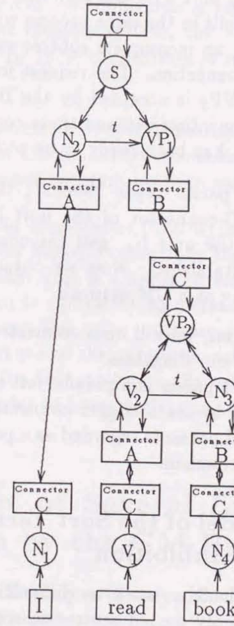


Figure 3.6: An example parse tree made by the CM parser

and its C-connector sends requests for connection. Since, however, there is no VP units which is ready to be activated and not rule of which A-category is VP, no connection is established.

2. When the CM parser reads "read", the incomplete subtree of which root is the unit VP_2 and in which the unit N_3 is ready to be activated is built in the same process with 1. As described the previous section, an incomplete subtree such as VP_2 is also able to request for connection. The request for connection from the C-connector of VP_2 is accepted by the B-connector of the unit VP_1 and the connection between these connectors is established, for the unit VP_1 has been ready to be activated.

3. When the CM parser reads "books", the request for connection from the C-connector of the unit N_4 is accepted by the B-connector of the unit N_3, and the connection between these connectors is established. Now we obtain a parse tree for the example sentence as a CM network.

For compact expressions, we will omit connectors and redundant units for explanations from our diagrams.

Intuitively, our CM parser is a parallel left corner parser. Speaking more precisely, owing to use a trigger connection, which predicts the next category, our CM parser is regarded as a parallel left corner parser with a prediction mechanism.

3.3.5 The Model of the Sort Term Memory based on Field Inhibition

In our cognitive difficulties, such as difficulties to recognize garden path sentences or deeply nested sentences, some performance limitations should play an important role. If all of possible candidates are held in the process of language recognition, these cognitive difficulties would not be observed. As one of such performance limitations, we pay attention to the limitation of the *short term memory*. It is well know that a human memory system consists of at least two levels, namely, the *short term memory* and the *long term memory*. It is also known that the capacity of the short term memory is limited to 7 ± 2 chunks.

In the CM, an implementation of short term memory has not yet been cleared. Intuitively, however, the limitation of the short term memory may depends on the mutual inhibition among memory chunks. We implement this limitation of the short term memory as follows:

1. The normal activation connection is regarded as the strong interaction in the memory system, and a network in which units are connected by these connections is regarded as one chunk of memory.

2. In order to express the effect of mutual inhibition, we suppose that there exist the inhibition connections of which weight is a very small negative constant between each of all units implicitly.

We call the inhibition connections in 2 *hidden connections*. We also call the effect of all of hidden connections the *field inhibition*. If the output function of each unit is a linear threshold function, the effect of the field inhibition is equivalent to increasing the threshold value of the output function. The more units are activated, the higher the threshold of each unit is and the output of each unit shows a tendency to be suppressed by the field inhibition. For example, according to the field inhibition, in order of lower activation, possible candidate parse trees are weeded out.

3.4 Analysis of Several Linguistic Phenomena by the CM Parser

In this section, we will analyze the several interesting linguistic phenomena with the CM parser.

3.4.1 Recognition of deeply nested sentences

Basically the limitation of the short term memory, which is also adopted by our CM parser, can explain why deeply nested sentences are hard to recognize for us. Such as the following sentence, however, several sentences cannot be explained by the limitation of the short term memory:

The man who the girl who the dog chased liked laughed.

When we read this sentence, at the latter half of the sentence, we find difficulty in recognizing the sentence, even if the sentence is not so much deeply nested[12]. That is, while the hearer clearly remembers all of the nouns, which should be subjects of each relative clause, he/she may not be able to associate each noun with the corresponding verb appropriately.

Figure 3.7 shows a network which is built just after the CM parser reads the word 'chased'. Here, the category VP_2 and VP_1 are remain being incomplete. When the CM parser reads the word 'like', the category unit for the word will connected with ether VP_2 or VP_1. It is depends on the strength of prediction for these categories which connection will win. That is, the connection to the unit which is triggered more strongly will win. Since the strength of a trigger to a unit is determined by the activation of the previous category unit, now let us compare the states of activation of the units, NP_3 and NP_1. When "chased" is inputted, the category NP_3 is completed. The CM parser, however, was occupied with one embedded sentence, after NP_4, which is the main constituent, or *head*, of NP_3, was completed. Therefore, since the activation propagated from the input is dumping by the passage of time and the field inhibition is also effective, the activation of the unit NP_3 is dumping. Where, we suppose that in a phrase structure network the activation of the head unit contributes to the activation of the C-unit more than another sister unit, that is, the connection between the C-unit and the head unit is stronger. Roughly speaking, the activation of the unit NP_3 depends almost upon the activation of the unit NP_4, because NP_4 is the head of NP_3. In the network of the subtree of which root is NP_4, the input activations corresponding to its constituent words, 'the' and 'girl', have dumped and the activation of the unit NP_4 is supported only by the feedback in the network of the subtree. Moreover, the network of the subtree also influenced by the effect of the field inhibition.

On the other hand, while the unit NP_2, which is the head unit of NP_1, has been completed and S_1 is also almost completed, the activation

[12]Can you show us what the subjects for verbs '*liked*' and '*laughed*' are, immediately ?

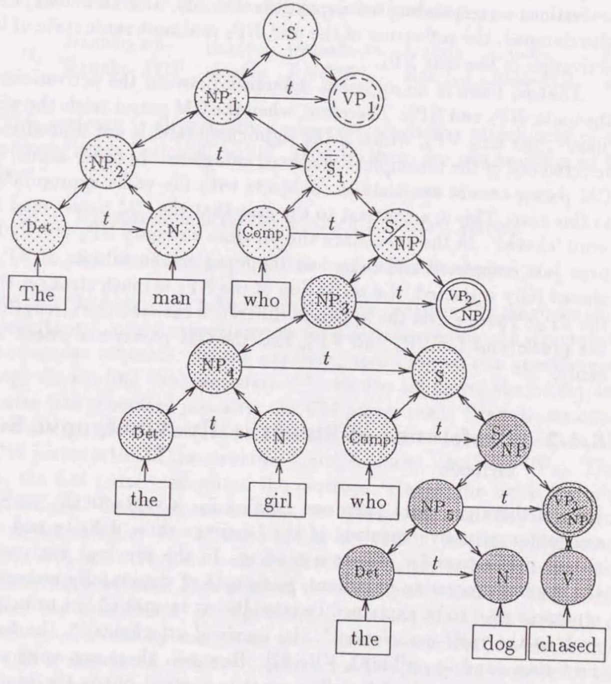


Figure 3.7: The parse tree just after the CM parser reads the word 'chased'

of the unit NP_2 is dumping because of the same reason in the case of NP_3. Now, we pay attention to the unit NP_2, which is the head of NP_1 and supports the activation of the unit NP_1. Since the subtree NP_2 has just a same structure with the subtree NP_4 and the input activations corresponding to its constituent words, 'the' and 'man', have also dumped, the activation of the unit NP_2 is almost same state of the activation of the unit NP_4.

That is, there is no apparent difference between the activations of the units, NP_3 and NP_1. Therefore, when the CM parser reads the word 'liked' , the unit VP_2, which is the right candidate, is not immediately selected out of the incomplete predicted categories. In other words, the CM parser cannot associate the subjects with the verbs appropriately, in this case. This is a contrast to the case that the CM parser read the word 'chased'. In the case, since the previous category NP_5 of VP_3 has been just completed and moreover the input to the subtree of NP_5 is almost fully activated, the activation of the NP_5 is much stronger than the NP_3. There exists the apparent difference between the strengths of the predictions for VP_3 and VP_2, and the CM parser can select VP_3 easily.

3.4.2 Preference of Structurally Ambiguous Sentences

Even if there are more than one reading for a syntactically ambiguous sentence, native speakers of the language show definite and consistent preferences for a certain reading. In the previous analyses by the symbol processing paradigm, preference of structurally ambiguous sentences used to be explained by introducing several ad hoc principles, such as the *right association*[13], the *minimal attachment*[14], the *lexical preference*, and so on[Shi83, FBK82]. However, there are some problems about these principles. The most important one is the problem of which should be used in parsing a given sentence. Since our parser is based on CM, preference of structurally ambiguous sentences is uniformly explained by each of the activation of the units which constitute

[13]This principle is also called the *lower attachment*

[14]This principle is also called the *higher attachment*

the parse tree for a given sentence. Roughly speaking, this preference mechanism with the activation is regarded as the minimal attachment principle in some cases and as the right association principle in other cases.

Let us consider the following Japanese sentence:

(i)	Hanako-wa	isoide	watashi-ga	eranda	hon-wo	katta
	Hanako-*TOP*	*quickly*	*I-NOM*	*selected*	*book-ACC*	*bought*

This sentence is structurally ambiguous about the attachment of the adverb 'isoide'(*quickly*)[Tok85]. That is, there are two readings as follows:

(ii) a. Hanako bought the book which I selected *quickly*.

b. Hanako *quickly* bought the book which I selected.

As described above, it is known that native speakers of a language show definite and consistent preferences for a certain reading of a structurally ambiguous sentence. In the example, according to our questionnairing, the reading (iib) is preferred[15]. Figure 3.8 shows the incomplete parse tree generated just after our CM parser reads "Hanako-wa isoide watashi-ga eranda hon". Just after the word 'eranda' was inputted, the CM parser selected the structure which includes VP_{1b} and VP_{2a}. That is, the CM parser recognized the sentence "Hanako-ha isoide watashi-ga eranda." In the successive step, the CM parser reads the word 'hon' and the category NP of the word is activated. At this point, since both NP_a and NP_b have been triggered and ready to be activated, two connections between each units of these categories and the unit of NP 'hon' are established. As described in Section 3.3.3, the strength of each connections by connectors depends on the activation of the unit which is associated with the C-connector accepting the request for the connection. When there exist several possible connections, they mutually inhibit each other to decrease each strength of others. Since the activation of the unit NP 'hon' is propagated equally to both the unit NP_a and the unit NP_b, let us examine the states of activation of the

[15]This questionnairing was conducted on 14 students majoring in computer science. 13 students preferred the reading (iib).

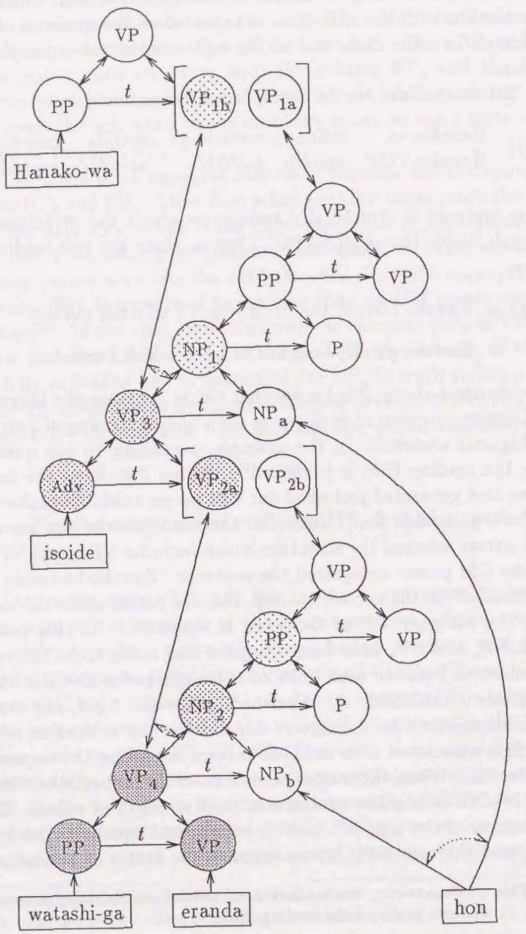


Figure 3.8: The parse tree just after the CM parser reads the word 'hon'

categories triggering these units. The subtree VP_4 has been completed just before the CM parser read the word 'hon', and the unit VP_4 is highly activated. On the other hand, the activation of the unit VP_3 is partly supported by the activation of the unit VP_4 and the unit Adv, which is the left daughter of VP_3, is less activated than VP_4 because of both the passage of time and the field inhibition. Therefore, the connection to NP_b wins and the reading (iib) is preferred. In our CM parser, a newly inputted category is easy to connect to the most activated category unit among the units triggered. We call this tendency the *attracting effect*.

Besides the example described above, many cases including the following sentences can be explained by the attracting effect:

(iii)	Hanako-ha	Taro-to	Jiro-wo	nagutta
	Hanako-*TOP*	Taro- *and*/*with*	Jiro-*ACC*	*hit(PAST)*

There are two readings and (iva) is preferred:

(iv) a. Hanako hit both Taro and Jiro.

b. Hanako and Taro hit Jiro.

(v)	utsukushii	Kanojyo-no	Utagoe
	beautiful	her	singing voice

There are two readings and (via) is preferred:

(vi) a. her beautiful singing voice

b. the singing voice by her. She is beautiful.

From the point of view of *modifiers*, the previous principles, such as the lower attachment, the higher attachment, and so on, show where a modifier is attached to. For example, in (iii) the structure in which the postposition phrase 'Taro-to' is attached low is preferred, on the other hand, in (v) the structure in which the adjective 'utsukushii' is attached high is preferred. Because the preference is considered from the viewpoint of the attaching point of the modifier which occurred in the past, mutually incoherent principles, such as the higher attachment and the lower attachment, are needed and so they should be used properly. In our CM parser, these preferences can be uniformly explained by interaction among the activation of the units, that is, the attracting effect on the newly inputted word and the principle "the most activated structure wins". However, there exists some case of preference which cannot be explain by our CM parser. In these case, these effects are incoherent each other and it would be hard to explain such cases only with syntactic information.

3.4.3 Garden Path Sentences

Let us consider the following sentence:

(vii) The cotton clothing is made of grows in Mississippi.

While one can parse this sentence easily until the word 'grows' is encountered, he/she suddenly falls into cognitive difficulty of parsing it just after reading 'grows'. The kind of sentence such as the example (vii) is called *garden path sentence*. While at first, we can smoothly parse the sentence of this kind in one reading, we suddenly fall into cognitive difficulty of parsing the moment we read one word in the sentence because we cannot change the reading to the another right one. Based on the phenomenon in garden path sentences, Marcus[Mar80] advocates the modified Determinism Hypothesis as follows:

> There is enough information in the structure of natural language in general, and in English in particular, to allow left-to-right deterministic parsing of those sentences which a native speaker can analyze without conscious effort.

and in this theory, errors in comprehension of garden path sentences is explained by the breakdown of simple and limited serial mechanism. However, the sentence (vii) does not always lead a native speaker of English down the garden path. Crain and Steedman have shown how easy it is to prevent backtracking by adding context to garden path sentences. For example, if the sentence (vii) is embedded in the appropriate context as follows:

(viii) a. Cotton is grown in several of the Southern States.

b. The cotton clothing is made of is grown in Mississippi.

the latter is no longer a garden path sentence. Therefore, leaving room for the effect of contexts, we should discuss difficulty of recognizing *isolated* garden path sentences. Figure 3.9 shows a network generated just after the CM parser reads "The cotton clothing is made of". In this figure, we omit trigger links. As the figure shows, when the word 'cotton' is inputted, both the category unit Mod(modifier) and the category unit NP are activated according to ambiguity of 'cotton'. Then, the category NP and $\overline{S}$(complement) are predicted as the candidates of next category. In the following step, in which the word "clothing" is read, the reachability of these predicted categories from the word is examined. At this point, the degree of the reachablity of NP is much higher than one of $\overline{S}$ because of the following reasons:

- The word 'clothing' is just a NP.
- From the viewpoint of the reachablity of S from the NP 'The cotton', as the following category of the NP, the category VP is triggered much highly than the category $\overline{S}$.
- The connections in the path [NP→S/NP→$\overline{S}$] are not so much strong.

Therefore, the structure including NP_a is preferred, or activated, and the structure including both $\overline{S}$ and NP_b is little activated. When the CM parser reads the word "grows", the structure including NP_a should be rejected and select another structure. Since the another structure, which include a complement, is little activated, the CM parser cannot switch the reading to the structure. That is, the CM parser is just led down the garden path.

Now, in order to examine the effect of context on garden path sentences, let us suppose a simple context sensitive mechanism, such as the mechanism proposed by Waltz et al[WP85]. The mechanism proposed by Waltz et al. can deals with simple context information by corresponding a certain context to the activation of one unit. Let us

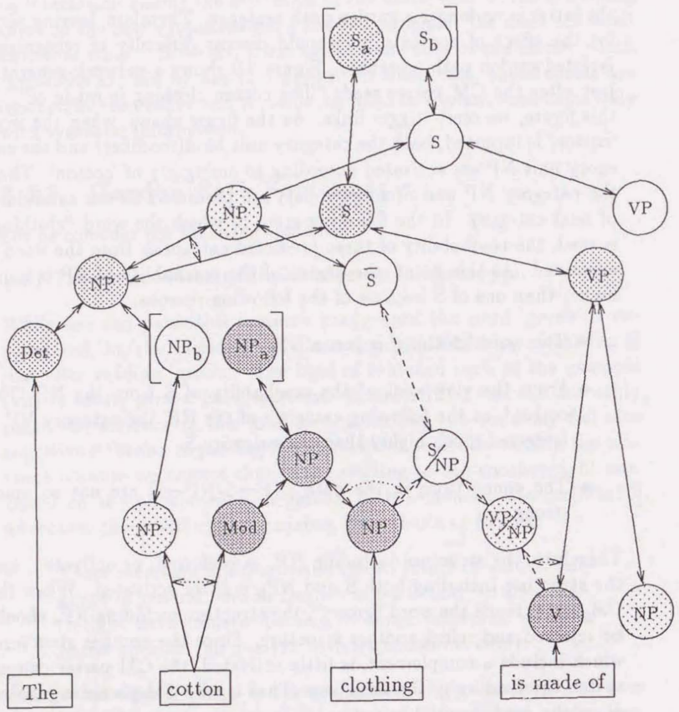


Figure 3.9: The parse tree just after the CM parser reads the word 'of'

consider (viii) by supposing that our CM parser has such a context handling mechanism. Since the sentence (viiia) makes the word "cotton" be activated as a noun, if the sentence (viiia) precedes the sentence viiib, in Figure 3.9 the unit N standing for the noun 'cotton' is much more activated than the unit Mod standing for the modifier 'cotton'. This bias effect of the context promotes the degree of prediction of $\overline{S}$, and the complement structure can be easily built.

3.5 Conclusion

In this chapter, we proposed a parsing model based on CM and we also showed that several phenomena of linguistic performance can be uniformly explained by the interaction of activation. Since in our CM parser, one of phrase structure rules is expressed as a subnetwork of CM and these subnetworks are connected dynamically by a dynamically linking mechanism, delicate aspects of parsing performance can be uniformly explained by the interaction of activation. Moreover, we showed that our CM parser can explain some interesting phenomena in the human parsing process, such as cognitive difficulties to recognize garden path sentences or deeply nested sentences, preference of structurally ambiguous sentences, and so on.

However, our model may be insufficient for linguistic phenomena beyond the scope of context free grammers. These phenomena includes *unbounded dependency.* The unbounded dependency is a phenomena that two parts in a syntactic structure are associated with each other and there can be exists any large subtree between them. For example, the relation between an empty category and a relative pronoun in a relative clause is one of these phenomena. In order to explain these phenomena, we should extend our model. As for the mechanism of our CM parser, the implementation of the dynamically linking mechanism is somewhat problematic. Since the dynamically linking mechanism in our CM parser does not generate new connections, but varies the strength of each connections, it is necessary to prepare all of possible connections in advance. If we do not adopt the linking mechanism, we will have to introduce a supervisor to generate new connections. Solving the problem how to generate new networks dynamically in CM may lead us to a harmonious paradigm including both a symbol processing approach and a connectionist model approach.

Chapter 4

A Model of Common Sense Reasoning Based on Circumscription

4.1 Introduction

Some mechanism of *common sense reasoning* is necessary in order to make a natural language understanding system be able to comprehend some natural language expression more deeply, that is, to make the system be able to obtain some useful information, which human extract from some expression by using his/her own background knowledge, especially *common sense.*

To formalize such a human common sense reasoning, several kinds of model have been proposed by a number of researchers[MD80, Rei80, Doy79, dK86, She84]. In these approaches, circumscription proposed by McCarthy[MaC80, MaC84], which is a form of *non-monotonic reasoning,* is one of the most promising model. Circumscription is a rule of conjecture that can be used by our or system for 'jumping to certain conclusions'. Namely, the objects that can be shown to have a certain property P by reasoning from certain facts A are all the objects that satisfy P. Since such a reasoning should be one of the most important kind of reasoning which we usually do in our daily life, it will be expected that the reasoning is useful for formalizing an aspect of reasoning in natural language understanding.

While circumscription is applied to a sentence of first order logic, unfortunately, circumscription itself is expressed in second order logic. It is known that there is no general algorithm to prove a theorem of the second order logic. Therefore, the main technical problem is that circumscription cannot be directly handled by theorem provers for first order, which is usually based on the *resolution principle*[Llo87]. However, the series of researches by Lifschitz shows that there exist classes of circumscription formulas expressible in the first order language which cover some of the intended applications. For these class, circumscription is computable by some resolution theorem provers.

In this chapter, first we will propose a application of circumscription to logic programming languages, such as Prolog. By this method, common sense reasoning can be done using the knowledge base which is built with one of logic programming languages. Logic programming languages, which are based on a subset of the first order logic called *horn set*, are well suited to natural language processing because of its characteristics in symbol manipulation, such as the unification mechanism, and so on. It is also one of characteristics of logic programming languages that a program written in them can be regarded as the knowledge base which naturally includes both facts and rules.

From the viewpoint of knowledge representation allowing for common sense reasoning, circumscription has the merit of needing no other special logic systems to describe some knowledge besides the ordinary first order logic. On the other hand, the major drawback is that the first order logic, which including horn set, can deal with only a single axiom space and so it cannot realize the structured knowledge base efficiently. The efficiency of reasonings in our daily life may result from restricting domains to be concerned adequately. The restriction comes not only from some of common sense reasoning, but also from the fact, which has been supported by several cognitive experiences[And80], that our knowledge consists of a number of modules and these modules are appropriately structured. It is one of the most important requirement for a scheme of knowledge representation that it can make a knowledge base into several appropriate parts and organize these parts. In view of this point, we also present an application of circumscription to the programming language *Uranus* proposed by Nakashima[NTS84, Nak85b, Nak85a], which is an extended logic programming language with the multiple world mechanism. The multiple world mechanism allows programmers both to divide program definition space into several parts called *worlds* and to organize them. It has been pointed out by V.Lifschitz[Lif85] that specificness of concepts plays an important role in assigning priorities of minimization to different kind of abnormality. We will show that structuring a conceptual hierarchy by the multiple world mechanism makes specificness of concepts clear and contributes to the assignment of priorities. By this method, we can deal not only with intra-world knowledge, but also with inter-world knowledge in an uniform way, that is, circumscription.

4.2 Circumscription

In this section, we will explain the general idea of circumscription, its formalization and its application to logic programming languages such as Prolog[MaC80, MaC84, Lif84, Lif85, Nak87, Nak89].

4.2.1 The general idea of circumscription

Let us consider the following conversation:

(ix) a. "I have not climbed Mt. Fuji."

b. "What ? You are not a Japanese !"

c. "I'm sorry. I'm not a Japanese, too..."

In the earlier part of this conversation, to the statement of Mr. A:

$$\neg\text{has-climbed}(A, \text{Mt. Fuji}). \tag{4.1}$$

Mr. B applied one of his subjective inference rules as follows:

$$\forall x.[\neg\text{has-climbed}(x, \text{Mt. Fuji}) \supset \neg\text{Japanese}(x)]. \tag{4.2}$$

$$\forall x.[\neg\text{like}(x, \text{high-school-baseball}) \supset \neg\text{Japanese}(x)]. \tag{4.3}$$

$$\forall x.[\text{staple-food}(x, \text{rice}) \supset \text{Japanese}(x)]. \tag{4.4}$$

that is, he use the rule (4.2) to infer ¬Japanese(A) and said, (ixb). Here, by (ixc), Mr. C intended not to state that he really is not a Japanese, but to inform others that he also had not climbed Mt. Fuji, that is,

$$\neg\text{has-climbed}(\text{C}, \text{Mt. Fuji}). \qquad (4.5)$$

Mr. A and Mr. B must be aware of that.

Note that at that point Mr. C supposed that both Mr. A and Mr. B has nothing but the following rule:

$$\forall x.[\neg\text{has-climbed}(x, \text{Mt. Fuji}) \subset \neg\text{Japanese}(x)] \qquad (4.6)$$

and indeed both Mr. A and Mr. B inferred (4.5). Since Mr. B's original rule is (4.2), at the point when (ixc) was interpreted, the following equation held:

$$\forall x.[\neg\text{has-climbed}(x, \text{Mt. Fuji}) = \neg\text{Japanese}(x)] \qquad (4.7)$$

That is, though Mr. B had not only (4.2) but also (4.3) and so on as the rules related to ¬Japanese(x), *only* the rule (4.2) was paid attention to as if there is nothing but (4.2). Moreover, a new rule (4.6) seemed to be produced. As described above, in inference in our everyday life including natural language understanding, *according to circumstance*, we usually restrict domains to be concerned adequately and produce new rules. Circumscription is a formalization of such common sense reasonings for knowledge written in the first order logic. Basically, circumscription makes up for the lack of information by regarding a given axiom of the first order language, which corresponds to all information *we know*, as all information *being true* and reinforce rules like the rule (4.2) was strengthened by transforming it into (4.7).

4.2.2 Formalization of Circumscription

Let AB be a tuple of predicate constants, Z a tuple of function and/or predicate constants disjoint with AB, and let $A(AB, Z)$ be a sentence. We denote the *(parallel) circumscription of AB in $A(AB, Z)$ with variable Z* by:

$$\text{Circum}[A(AB, Z); AB; Z] \qquad (4.8)$$

and it is defined as follows[Lif85]:

$$\text{Circum}[A(AB, Z); AB; Z] = \\ A(AB, Z) \wedge \neg\exists ab \exists z.[A(ab, z) \wedge (ab < AB)] \qquad (4.9)$$

Here ab,z are tuples of variables similar to P,Z. When Z is empty, we omit it. If U,V are n-ary predicates, then $U \leq V$ stands for $\forall x (Ux \supset Vx)$. If U,V are tuples $U_1, \ldots, U_m$ and $V_1, \ldots, V_m$ respectively, then $U \leq V$ stands for:

$$U_1 \leq V_1 \wedge \ldots \wedge U_m \leq V_m \qquad (4.10)$$

Futhermore, $U = V$ stands for $U \leq V \wedge V \leq U$, and $U < V$ stands for $U \leq V \wedge \neg(V \leq U)$.

In order to make targets of circumscription uniform, McCarthy introduced *abnormality predicates*, which express some abnormality[MaC84]. Abnormality predicates allow us represent general knowledge which may include some abnormality. For example, the knowledge, "*In general,* birds can fly", is written as:

$$\forall x.[\text{bird}(x) \wedge \neg\text{ab}(x) \supset \text{fly}(x)] \qquad (4.11)$$

Where the abnormality predicate ab expresses the abnormality of the characteristic "can fly" of birds.

In applications to the formalization of common sense reasoning, $A(AB, Z)$ is the conjunction of the axioms, AB is the list of abnormality predicates, and Z is the list of symbols that we intend to characterize by means of circumscription.

4.2.3 Computable Circumscription

Lifschitz showed that circumscription is computable for the subset of the first order language called *separable formulas*, that is, circumscription of formulas of the class can be reduced to the first order logic. Since in logic programming languages like Prolog a conjunction of *horn clauses*, which is a subset of first order language, is regarded as a program, if the program is a separable formula then it is just a *solitary formula*, which is the special case of separable formulas and defined as follows. We say that $A(AB)$ is solitary with respect to AB if it has the following form:

$$N(AB) \wedge \{U \leq AB\} \tag{4.12}$$

where $N(AB)$ contains no positive occurrences of $AB_1, \ldots, AB_m$, and U is a tuple of predicates not containing $AB_1, \ldots, AB_m$. Then the result of circumscription is given by the formula[Lif85]:

$$\begin{aligned}\text{Circum}[N(AB) \wedge \{U \leq AB\}; AB] \equiv \\ N(U) \wedge \{U = AB\}\end{aligned} \tag{4.13}$$

Every circumscription with a non-empty Z can be reduced to a circumscription with the empty Z:

$$\begin{aligned}\text{Circum}[A(AB, Z); AB; Z] \equiv \\ A(AB, Z) \wedge \text{Circum}[\exists z.A(AB, z); AB]\end{aligned} \tag{4.14}$$

The problem with this trick is that the first argument of circumscription in (4.14), generally, contains new second-order quantifiers. However, if $A(AB, z)$ is solitary with respect to z, that is:

$$A(AB, z) \equiv [Nn(AB, z) \wedge \tag{4.15}$$
$$\{Uu(AB) \leq z\}] \tag{4.16}$$

then the following equation held and the second-order quantifiers can be eliminated.

$$\exists z.A(AB, z) \equiv Nn(AB, Uu(AB)) \tag{4.17}$$

That is, in (4.15), z is replaced with the extension $Uu(AB)$[1] of z given by (4.16). From the view of *program transformation in logic programming*[ST83, FM87], (4.17) can be considered that by *unfolding* the goal z in (4.15) is partially evaluated with respect to the clause (4.16) to eliminate z with preserving validity.

[1] In Prolog, it corresponds to the bodies of the clauses of which head are z.

In order to evade conflicts of minimizing abnormality predicates with each other, McCarthy[MaC84] proposed to establish priorities between different kinds of abnormality. This type of circumscription is called *Prioritized circumscription* and denoted by:

$$\text{Circum}[A; AB_1 > \cdots > AB_k; Z] \tag{4.18}$$

In (4.18), the predicates in AB_i should be minimized at higher priority than the predicates in AB_{i+1}. Any prioritized circumscription can be written as a conjunction of parallel circumscriptions, as follows:

$$\begin{aligned}\text{Circum}[A; AB_1 > \cdots > AB_k; Z] \equiv \\ \bigwedge_{i=1}^{k} \text{Circum}[A; AB_i; AB_{i+1}, \cdots, AB_k, Z]\end{aligned} \tag{4.19}$$

4.3 Introducing Abnormality Predicates by Finding Inconsistency

While our target logic programming language of circumscription is based on Uranus[NTS84, Nak85b, Nak85a], as "negation", it has not only *negation as failure* (NAF)[2][She84, Llo87], which is usually used as a negative operation, but also *logical negation*[3], which is used to represent negative knowledge explicitly[Nak86, KN85]. Therefore, in the worst case, knowledge written in this programming system has some inconsistency. In general, it is known that introducing both logical negation to logic programming languages and some mechanism to check inconsistency causes deterioration of efficiency. However, since detecting inconsistency can be regarded as a chance of knowledge acquisition and plays an important role in inference, introducing logical negation should be indispensable.

In order to detect inconsistency with respect to a predicate, a logic programming system like Uranus, of which inference is usually based on *backward reasoning*, first executes the predicate as goal. Next, if the execution succeeds then the system executes the goal of its negation to

[2] We will denote this type of negation by the predicate THNOT.
[3] We will denote this type of negation by the predicate NOT.

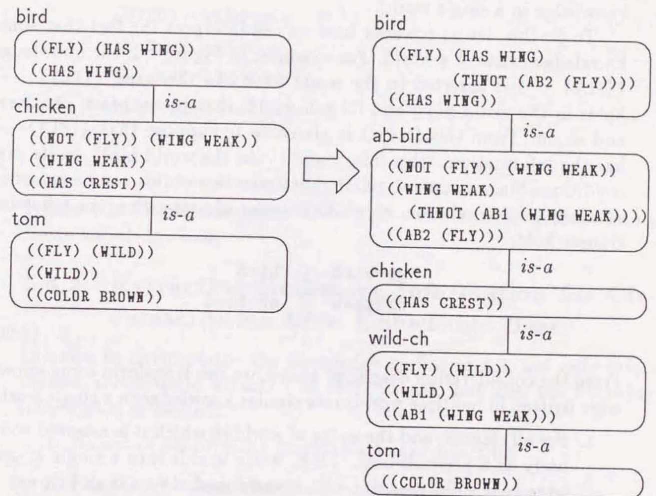
check inconsistency. However, checking consistency of all predicates in a program causes the system to waste both memory resource and time resource. In order to avoid this drawback, we introduce into Uranus the inconsistency checking predicate `C`, by which users can explicitly specify a goal to be checked.

Now, we explain the process of resolving inconsistency. First, the system has a user, or teacher, to show which goal is wrong in the world where the inconsistency occurred, the positive goal or the negative goal. Next, by the interaction with the user, system examines all knowledge which was used in the process of deriving the inconsistency to traces the inconsistency to its origin. At least one of these knowledge must be abnormal knowledge in the world where the consistency occurred. In order to resolve the inconsistency, system explicitly shows that the knowledge, which is the cause of inconsistency, is abnormal in the world. To do this, we adopt abnormality predicates and use them as Figure 4.1 shows. While the method of searching the origin of inconsistency is same as *Dependency Directed Backtracking* in Doyle's *TMS*[Doy79], the method of resolving it by abnormality predicates is characteristic of our system. Now, we use the notation of the S-expression to write programs in conformity with Uranus, hereafter.

In Figure 4.1, both `ab-bird` and `wild-ch` are the worlds which the process of resolving inconsistency generates under the direction of the user.

4.4 Circumscription of Hierarchical Knowledge with Inheritance

In the previous section, we proposed the method to introduce information about existence of abnormal things according to detecting inconsistency. Human inference does not only detect existence of abnormal things but also recognize the contents of it. Circumscription formalizes just this process. This section presents the method of circumscribing hierarchical knowledge.



(a) The predicate `fly` is inconsistent in the world `tom` and `chicken`.

(b) Resolution of inconsistency by abnormality predicates

Figure 4.1: Resolution of inconsistency by abnormality predicates

4.4.1 How to Deal with Information about Worlds

Circumscription is a inference rule for the first order logic, that is, a single axiom space. For this reason, in order to circumscribe some knowledge written in a programming language with the multiple world mechanism, it is necessary to transform the knowledge into the similar knowledge in a single world.

To do this, let us consider how we can interpret the fact that some knowledge exists in a world. For example, in Figure 4.1, the knowledge `((FLY) ···)` is asserted in the world `bird`. In this case, `((FLY) ···)` holds in the world `bird` and its sub-world, that is, `ab-bird`, `chicken`, and so on. From this fact, it is plausible to suppose that `((FLY)···)` has the information, "this information is in the world bird", in its pre-condition. Since a sup-sub relation between two worlds can be regarded as a implication relation, as to inheritance of properties, the following clauses hold:

$$\begin{array}{rcl} \texttt{ab-bird} & \leq & \texttt{bird} \\ \texttt{chicken} & \leq & \texttt{ab-bird} \\ & \vdots & \end{array} \tag{4.20}$$

From the consideration described above, we can transform some knowledge written in multiple worlds into similar knowledge in a single world.

1. For all clauses, add the name of world in which it is asserted to its body as a precondition. Then, unite worlds into a single knowledge base. The meaning of a transformed clause is as follows:

   If all of its preconditions, or body, hold and the clause itself is in the world indicated in its body, then its consequence hold.

2. Express the hierarchy of worlds as a conjunction of implication relations, such as (4.20), and add the conjunction to the single knowledge base.

According to this transformation, the knowledge in Figure 4.1 can be regarded as the knowledge in Figure 4.2[4].

[4]In this figure, we omit expressions which will not appear in our following explanation.

$$\begin{array}{ll}
\texttt{(HAS WING)} \wedge \texttt{(THNOT (AB2 (FLY)))} \wedge \texttt{bird} \leq \texttt{(FLY)} & (4.21) \\
\texttt{bird} \leq \texttt{(HAS WING)} & (4.22) \\
\texttt{(WING WEAK)} \wedge \texttt{ab-bird} \leq \texttt{(NOT (FLY))} & (4.23) \\
\texttt{(THNOT(AB1(WING WEAK)))} \wedge \texttt{ab-bird} \leq \texttt{(WING WEAK)} & (4.24) \\
\texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(FLY)} & (4.25) \\
\texttt{ab-bird} \leq \texttt{bird} & (4.26) \\
\texttt{chicken} \leq \texttt{ab-bird} & (4.27) \\
\texttt{wild-ch} \leq \texttt{chicken} & (4.28) \\
\vdots &
\end{array}$$

Figure 4.2: Transformation of a multiple world knowledge base into a single knowledge base

4.4.2 Extracting Necessary Information for Circumscription from Knowledge Base

In order to circumscribe the knowledge in Figure 4.2, not only information about world attached to each clauses, but also the following information is needed.

1. Names of abnormality predicates to circumscribe.
2. Priorities between abnormality predicates for minimization.
3. Name of predicates which are allowed to vary in the process of minimization.

As for 1, since all of abnormality predicates are introduced by resolving inconsistency, the system can easily maintain names of abnormality predicates. As for the others, we should introduce some heuristics.

4.4.2.1 Priorities between Abnormality Predicates for Minimization

Priorities between abnormality predicates for minimization are important information, because they will influence definitions of abnormality predicates after circumscription. We can use *positions of fact-type clauses of abnormality predicates* to assign these priorities. That is, the higher priority is assigned to the abnormality predicate of which fact-type clause is positioned at lower world in the hierarchy. Since the world in which the fact-type clause of an abnormality predicate is asserted is the highest one of worlds in which the inconsistency related to the abnormality predicate occurred, the abnormality predicate of which fact-type clause is asserted in lower world has narrower effect, and it may be natural to suppose that the abnormality predicate is assigned higher priority. For example, in Figure 4.1, the predicate `AB1` should be minimized at higher priority than the predicate `AB2`.

4.4.2.2 Selecting Predicates Allowed to Vary in Circumscription

The predicates we intended to characterize by means of circumscription are selected to for predicates which are allowed to vary in the process of minimization. Here, we pay attention to the effect of resolving inconsistency. In order to use the effect of resolving inconsistency by circumscription, first, we should add the following predicates to the predicate allowed to vary:

1. The predicate on which an inconsistency occurred, that is, the predicate of the clause on which an inconsistency occurred.

For example, in Figure 4.1, the predicate `FLY` corresponds to this type.

Secondly, when an abnormality predicate is attached not to the clause on which the inconsistency occurred, but to another clause which is used in the middle of the process of proving the former clause, the following predicates also should be added to the predicate allowed to vary:

2. All predicates which occurred in the path of the proof tree, from the clause to which the abnormality predicate is attached, to the clause on which the inconsistency occurred.

In order for the effect of resolving inconsistency to influence the definition of an abnormality predicate, when we calculate the definition of the abnormality predicate, we should make the abnormality predicate

occur virtually in the clause on which the inconsistency occurred. To do this, we may repeat unfolding one of the goal in the body of the clause on which inconsistency occurred, until the abnormality predicate occurs in the clause. Since, as described in section 4.2, elimination of the predicates allowed to vary is equivalent to unfolding, we should select the predicates allowed to vary according to 2. For example, at the world `ab-bird` in Figure 4.1 the abnormality predicate is attached not to the clause of the predicate `FLY`, on which an inconsistency occurred, but to the clause of the predicate `WING`, which is used in the process of proving the former clause. Since, as Figure 4.3 shows, unfolding the predicate `WING` makes the abnormality predicate appear virtually in the body of the clause of the predicate `FLY`, the effect of eliminating the predicate `FLY`, on which inconsistency occurred, can influence the definition of the abnormality predicate. Finally, in the case of Figure 4.1, we should

(WING WEAK) ∧ ab-bird ≤ (NOT (FLY))

↑

unfold

|

(THNOT (AB1···)) ∧ ab-bird ≤ (WING WEAK)

⇓

(THNOT (AB1···)) ∧ ab-bird ≤ (NOT (FLY))

Figure 4.3: The case that an abnormality predicate is attached to the clause which is used in the process of proving inconsistency

select not only `FLY` but also `WING` for the predicates allowed to vary.

4.4.3 Reorganization of Worlds According to the Result of Circumscription

Let A be a conjunction of horn clauses in Figure 4.2. Then, circumscription of the knowledge in Figure 4.1 is equivalent to the following expression:

$$\mathrm{Circum}[A; \mathtt{AB1} > \mathtt{AB2}; \mathtt{FLY}, \mathtt{WING}] \tag{4.29}$$

that is, circumscription of AB1,AB2 in A. In this circumscription, the abnormality predicate AB1 is minimized at higher priority than AB2 and the predicates FLY,WING are allowed to vary. In order to calculate this, first, the prioritized circumscription is transformed into a conjunction of parallel circumscriptions by (4.19), and secondly, the predicates allowed to vary are eliminated by (4.14).

$$\begin{aligned} &\text{Circum}[A; \texttt{AB1} > \texttt{AB2}; \texttt{FLY}, \texttt{WING}] \equiv \\ &\quad A \wedge \text{Circum}[\exists\texttt{AB2}\exists\texttt{FLY}\exists\texttt{WING}.A; \texttt{AB1}] \wedge \\ &\quad \text{Circum}[\exists\texttt{FLY}\exists\texttt{WING}.A; \texttt{AB2}] \end{aligned} \tag{4.30}$$

Thirdly, (4.17) eliminates the second-order predicate variables FLY, WING and AB2 Now, in (4.30):

$$\begin{aligned} &\exists\texttt{FLY}\exists\texttt{WING}.A \equiv \\ &\quad \exists\texttt{FLY}\exists\texttt{WING}.[\\ &\quad\quad \{\texttt{(HAS WING)} \wedge \texttt{(THNOT (AB2 (FLY)))} \wedge \texttt{bird} \leq \texttt{(FLY)}\} \wedge \\ &\quad\quad \{\texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(FLY)}\} \wedge \\ &\quad\quad \{\texttt{(WING WEAK)} \wedge \texttt{ab-bird} \leq \texttt{(NOT (FLY))}\} \wedge \\ &\quad\quad \{\texttt{(THNOT (AB1 (WING WEAK)))} \wedge \texttt{ab-bird} \leq \texttt{(WING WEAK)}\} \wedge \\ &\quad\quad Rest] \end{aligned} \tag{4.31}$$

where, the expression *Rest* includes all expressions which will not be used in the explanation. Let us transform (4.31) into:

$$\texttt{(WING WEAK)} \wedge \texttt{(THNOT (NOT (FLY)))} \wedge \texttt{ab-bird} \leq \texttt{(FALSE)} \tag{4.32}$$

for the preparation to unfold to eliminate the predicates allowed to vary. When the literal is moved over the implication symbol, as the above expression shows, THNOT is attached to the literal and the literal (THNOT (NOT (FLY))) appears. Here, (THNOT (NOT···)) can be regarded as a modal operator which is true if the logical negation of its argument can not be proved. In general, when the knowledge is consistent, the domain in which (FLY) is true should be included by the domain in which (THNOT (NOT (FLY))) is true. Therefore, in eliminating the predicates allowed to vary, consistency on the predicate FLY can be preserved by unfolding in which (THNOT (NOT (FLY))) and (FLY)

are dealt with as if they are equivalent. In order to use the effect of resolving inconsistency, the predicate on which inconsistency occurred should be added to the predicates allowed to vary. To preserve consistency on the predicate FLY, (4.32) is transformed into:

$$\texttt{(WING WEAK)} \wedge \texttt{(FLY)} \wedge \texttt{ab-bird} \leq \texttt{(FALSE)} \tag{4.33}$$

In the expression A, unfold the predicates FLY,WING to eliminate them, and reduce goals of world name. Then, we obtain the following expression:

$$\begin{aligned} &\exists\texttt{FLY}\exists\texttt{WING}.A \\ &\quad \equiv \{\texttt{(THNOT (AB1 (WING WEAK)))} \wedge \texttt{ab-bird} \wedge \texttt{(HAS WING)} \\ &\quad\quad \wedge\texttt{(THNOT (AB2 (FLY)))} \wedge \texttt{bird} \leq \texttt{(FALSE)}\} \wedge \\ &\quad\quad \{\texttt{(THNOT(AB1(WING WEAK)))} \wedge \texttt{ab-bird} \wedge \texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(FALSE)}\} \wedge \\ &\quad\quad Rest \\ &\quad \equiv \{\texttt{(HAS WING)} \wedge \texttt{(THNOT (AB1 (WING WEAK)))} \wedge \texttt{ab-bird} \leq \texttt{(AB2 (FLY))}\} \wedge \\ &\quad\quad \{\texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(AB1 (WING WEAK))}\} \wedge Rest \end{aligned} \tag{4.34}$$

Because, in the expression ∃FLY∃WING.A, AB2 is included only in the first term, the following expression holds:

$$\begin{aligned} &\exists\texttt{AB2}\exists\texttt{FLY}\exists\texttt{WING}.A \equiv \\ &\quad \exists\texttt{AB2}.[\texttt{(THNOT(AB1(WING WEAK)))} \wedge \texttt{(HAS WING)} \wedge \texttt{ab-bird} \leq \texttt{(AB2(FLY))}] \wedge \\ &\quad \wedge\{\texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(AB1 (WING WEAK))}\} \wedge Rest \end{aligned} \tag{4.35}$$

Since the expression in the square brackets [ ] of (4.35) is true if the following expression holds, the first term of (4.35) is always true.

$$\texttt{(AB2 (FLY))} \;=\; \texttt{(TURE)} \tag{4.36}$$

Therefore, we can eliminate the existential quantifier of AB2 as follows:

$$\begin{aligned} &\exists\texttt{AB2}\exists\texttt{FLY}\exists\texttt{WING}.A \equiv \\ &\quad \{\texttt{(WILD)} \wedge \texttt{wild-ch} \leq \texttt{(AB1 (WING WEAK))}\} \wedge Rest \end{aligned} \tag{4.37}$$

This is equivalent to the calculation of AB1 in the condition as if the abnormality predicate AB2, which is in the upper world, is true. By

(4.13), we obtain the following expression:

$$\begin{aligned}&\text{Circum}[\exists \texttt{AB2}\exists \texttt{FLY}\exists \texttt{WING}.A;\texttt{AB1}] \equiv\\ &\quad [\{(\texttt{WILD}) \wedge \texttt{wild-ch}\} = (\texttt{AB1 (WING WEAK)})] \qquad (4.38)\\ &\quad \wedge Rest\end{aligned}$$

To change the implication relation ($\leq$) in (4.13) to the equivalence relation (=) is circumscription of the predicate AB1. Similarly, we also obtain:

$$\begin{aligned}&\text{Circum}[\exists \texttt{FLY}\exists \texttt{WING}.A;\texttt{AB2}] \equiv\\ &\quad [\{(\texttt{HAS WING}) \wedge (\texttt{THNOT (AB1 (WING WEAK))}) \wedge \texttt{ab-bird}\}\\ &\qquad = (\texttt{AB2 (FLY)})] \qquad (4.39)\\ &\quad \wedge[(\texttt{WILD}) \wedge \texttt{wild-ch} \leq (\texttt{AB1 (WING WEAK)})] \wedge Rest \qquad (4.40)\end{aligned}$$

By (4.38) and (4.39), the abnormality predicates AB1 and AB2 are eliminated and the following expression is obtained:

$$\begin{aligned}&\text{Circum}[A;\texttt{AB1} > \texttt{AB2};\texttt{FLY},\texttt{WING}] \equiv\\ &[(\texttt{HAS WING}) \wedge \texttt{bird} \wedge (\texttt{THNOT ab-bird}) \leq (\texttt{FLY})] \wedge \qquad (4.41)\\ &[(\texttt{HAS WING}) \wedge (\texttt{WILD}) \wedge \texttt{wild-ch} \leq (\texttt{FLY})] \wedge \qquad (4.42)\\ &[(\texttt{WILD}) \wedge \texttt{wild-ch} \leq (\texttt{FLY})] \wedge \qquad (4.43)\\ &[(\texttt{WING WEAK}) \wedge \texttt{ab-bird} \leq (\texttt{NOT (FLY)})] \wedge \qquad (4.44)\\ &[(\texttt{THNOT (WILD)}) \wedge \texttt{ab-bird} \leq (\texttt{WING WEAK})] \wedge \qquad (4.45)\\ &[\texttt{ab-bird} \wedge (\texttt{THNOT wild-ch}) \leq (\texttt{WING WEAK})] \wedge \qquad (4.46)\\ &[(\texttt{AB1 (WING WEAK)}) = \{(\texttt{WILD}) \wedge \texttt{wild-ch}\}] \wedge \qquad (4.47)\\ &[(\texttt{AB2 (FLY)}) = \{\{(\texttt{HAS WING}) \wedge (\texttt{THNOT (WILD)}) \wedge \texttt{ab-bird}\}\\ &\vee\{(\texttt{HAS WING}) \wedge \texttt{ab-bird} \wedge (\texttt{THNOT wild-ch})\}\}] \wedge \qquad (4.48)\\ &Rest] \qquad (4.49)\end{aligned}$$

In the process of circumscription of abnormality predicates in a conjunction of clauses including world names, world names are treated as normal predicates. As the above expression shows, subgoals of the form (THNOT *world-name*) appear in the result of circumscription. According to the definition of the goal of a world name, in which the goal of world name w is defined as the predicate which is true if the clause including w is at the world w, the goal (THNOT *world-name*) can be regarded as the predicate which is true if it is not proved that the clause including the goal is at the world *world-name*. That is, the goal indicates that the clause must not be at the world *world-name*. Since, moreover, the clauses which represent inheritance of properties are also applied to goals of world name, the goal (THNOT *world-name*) is false if the clause including the goal is at ether the world *world-name* or its sub-worlds. Therefore, the goal (THNOT w) indicates that the inheritance of the clause including the goal must be canceled at the world w. From the above consideration, the treatment for goals of world name after circumscription is as follows:

- Ordinary world name: w
  The world w indicates the world in which clauses including w should be asserted.
- World name with THNOT: (THNOT w)
  The world w indicates the world at which the inheritance of the clause including w should be canceled .

After circumscription, when the goal (THNOT wn) appears in a clause and the world wn is sub-world of the world wp included by the clause as the goal of ordinary world name, the inheritance of the clause must be canceled at the world wn. To do this, we may cut the is-a link, which is the inheritance path from the super-world, at the world wn, and make the world wn be one of sister worlds of the world wp, in which the clause is asserted. In our example, the result of circumscription is shown as Figure 4.4.

4.5 Control of Splitting Hierarchy

While in the previous section we proposed that the goal of world name with THNOT indicates the world at which we must cancel the inheritance, if for each of the goals of world name with THNOT the knowledge hierarchy is split, splitting may be overdone. In human learning process, change of concept hierarchy should be suppressed as far as possible.

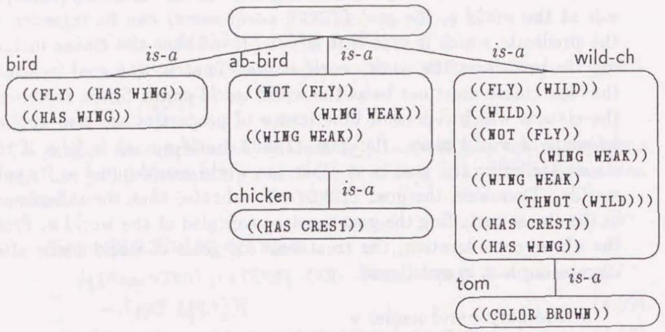


Figure 4.4: Organization of the world after circumscription

From this view, let us consider the method to suppress change of concept hierarchy as far as possible. For example, both (4.45) and (4.46), as described in the previous section,

$$\texttt{(THNOT (WILD))} \wedge \texttt{ab-bird} \leq \texttt{(WING WEAK)}$$

$$\texttt{(THNOT wild-ch)} \wedge \texttt{ab-bird} \leq \texttt{(WING WEAK)}$$

provide the definition of the predicate `WING` in the world `ab-bird`, because each of them has the goal `ab-bird` in its body. On the other hand, while (4.45) holds in all of the worlds which inherit knowledge from the world `ab-bird`, (4.46) does not hold in the worlds which are the world `wild-ch` and its sub-worlds. In the organization of worlds, as Figure 4.1 shows, while for the world `wild-ch` we cannot but choose the definition (4.45), for the world `ab-bird` we can also adopt (4.46), which influences the organization of worlds. Taking also (4.46) into account, the definition of the predicate `FLY` in the world `ab-bird`, which is a program of Uranus without goals of world name, is as follows:

```
((WING WEAK)(THNOT (WILD)))     (4.50)
((WING WEAK))                   (4.51)
```


Because (4.50) is redundant logically, after all, the definition is only (4.51). Then, according to the treatment for `(THNOT wild-ch)` in the body of (4.46), the organization of worlds in Figure 4.4 is obtained.

However, we can consider as follows. As the result of circumscription, the definition of the abnormality predicate `AB1` related to the predicate `WING` was as follows:

$$\texttt{(AB1 (WING WEAK))} \equiv \texttt{(WILD)} \wedge \texttt{wild-ch} \tag{4.52}$$

Since the predicate `WILD` is asserted in the world `wild-ch`,

$$\texttt{(WILD) is true.} \equiv \texttt{wild-ch} \tag{4.53}$$

Therefore, the definition of `AB1` is ether of the following expression:

$$\texttt{(AB1 (WING WEAK))} \equiv \texttt{(WILD)} \tag{4.54}$$

$$\texttt{(AB1 (WING WEAK))} \equiv \texttt{wild-ch} \tag{4.55}$$

The organization in Figure 4.4 results from the choice of (4.55). When (4.55) is chosen, by eliminating the abnormality predicate `AB1` only (4.45) is obtained[5]. In this case, since the goal `(THNOT wild-ch)` in (4.46) is not effective, the split of the hierarchy may be suppressed as far as possible and the organization of worlds is as Figure 4.5. In this case, the abnormality of the predicate `WING` is introduced into its clause, and the exception based on the existence of the world `wild-ch` is expressed in the world `ab-bird`. From the viewpoint of learning, while in the case of Figure 4.4 a new organization about the world `wild-ch` is introduced, in the case of Figure 4.5, a new rule is learned in the world `ab-bird`.

In general, when the following definition of an abnormality predicate is obtained as the result of circumscription:

$$(AB\ *) \equiv (P_1\ *) \wedge \cdots \wedge (P_n\ *) \wedge world \tag{4.56}$$

if the question $(P_1\ *) \wedge \cdots \wedge (P_n\ *)$ fails in the world $world_1$, which is the immediate super-world of the $world$, the following expression holds:

$$(AB\ *) \equiv (P_1\ *) \wedge \cdots \wedge (P_n\ *) \tag{4.57}$$

[5] This corresponds to (4.50) in the program.

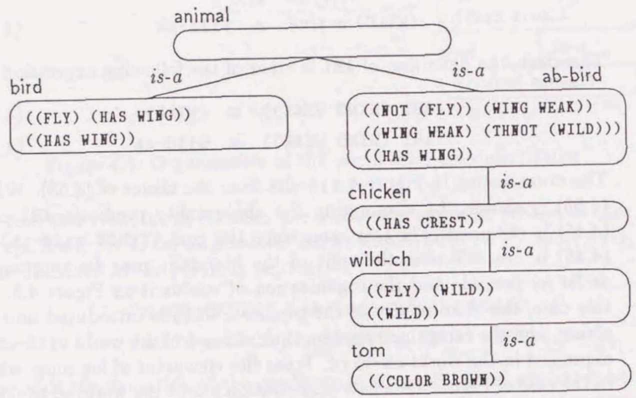


Figure 4.5: Suppressing split of the hierarchy

Because the world $world_2$ in which the question $(P_1\ *) \wedge \cdots \wedge (P_n\ *)$ succeeds is the world *world* or its sub-worlds and

$$\text{The question } (P_1\ *) \wedge \cdots \wedge (P_n\ *) \text{ succeeds.} \supset world_2 \supset world$$

holds. When the definition of the abnormality predicate is the same form as (4.57), we can suppress the split of the world hierarchy. Because several modes, such as Figure 4.4 and Figure 4.5, may be used properly, both reasoning and memory of human are so efficient.

4.6 Conclusion

In this chapter, we proposed a model of common sense reasoning based on an application of circumscription to logic programming languages, such as Prolog, as a way to make a natural language understanding system be able to comprehend some natural language expression more deeply, that is, to make the system be able to obtain some useful information, which human extract from some expression by using his/her own background knowledge. By this method, common sense reasoning can be done using the knowledge base which is built with one of logic programming languages. We also presented an application of circumscription to the programming language *Uranus* proposed by Nakashima. The application is useful to formalize a common sense reasoning on some frame-style knowledge base. By this method, we can deal not only with intra-world knowledge, but also with inter-world knowledge in an uniform way, that is, circumscription.

On the other hand, from the viewpoint of learning, we should note that circumscription does only "leaning by negative examples". When some leaning system adopts the application of circumscription we proposed, the system should need the mechanism of "leaning by positive examples", such as analogy, induction, and so on. For example, Shapiro's MIS (Model Inference System)[Sha82], which is the algorithm of an inductive leaning for horn clauses, gives us the mechanism to learn both facts and rules including variables from both positive facts and negative facts.

Chapter 5

Conclusion

In this dissertation, from the viewpoint of computational linguistics, of which main purpose is to formalize human processes of language understanding as computational processes, we proposed the models of three aspects of processes of natural language understanding.

In Chapter 2, we formalized metaphor understanding in terms of situation semantics as the process of introducing new resources, namely, establishing correspondences between a source domain and a target domain, with which a hearer can project information about a source domain onto a target domain. As a result, metaphorical expressions can convey some other new information. We expect that this formalization of metaphor understanding, which includes both semantics of metaphor and its processing, may serve as a guide to natural language understanding systems which can understand discourses with metaphors.

In Chapter 3, we proposed a parsing model based on CM and we also showed that several phenomena of linguistic performance can be uniformly explained by the interaction of activation. Since in our CM parser, one of phrase structure rules is expressed as a subnetwork of CM and these subnetworks are connected dynamically by a dynamically linking mechanism, delicate aspects of parsing performance can be uniformly explained by the interaction of activation. Moreover, we showed that our CM parser can explain some interesting phenomena in the human parsing process, such as cognitive difficulties to recognize garden path sentences or deeply nested sentences, preference of structurally ambiguous sentences, and so on.

In Chapter 4, we proposed a model of common sense reasoning based on an application of circumscription to logic programming languages, such as Prolog, as a way to make a natural language understanding system be able to comprehend some natural language expression more deeply, that is, to make the system be able to obtain some useful information, which human extract from some expression by using his/her own background knowledge. By this method, common sense reasoning can be done using the knowledge base which is built with one of logic programming languages. We also presented an application of circumscription to the programming language Uranus proposed by Nakashima. The application is useful to formalize a common sense reasoning on some frame-style knowledge base.

However, since three models proposed in this dissertation deal with only three aspects of natural language understanding, our models cannot cover all processes of natural language understanding. In view of the fact that our use of language is closely related to our thinking, the research of natural language understanding by computer is essentially hard to complete and takes very long time. For this reason, many researchers are working vigorously in this field. I hope this dissertation will contribute toward establishing the *true* model of natural language understanding, which has the same linguistic ability as human has.

Published Papers

Refereed Papers

[1] 中川 裕志, 森 辰則. 論理型言語における circumscription. 情報処理学会論文誌, Vol. 28, No. 4, pp. 330–338, 1987.

[2] 森 辰則, 中川 裕志. 多重世界機構における circumscription. 情報処理学会論文誌, Vol. 28, No. 5, pp. 437–445, 1987.

[3] 中川 裕志, 萱島 信, 森 辰則. 自然な構造の階層的知識. 人工知能学会誌, Vol. 3, No. 3, pp. 329–336, 1988.

[4] 森 辰則, 中川 裕志. Connectionist model による構文解析モデル. 情報処理学会論文誌, Vol. 30, No. 4, pp. 447–456, 1989.

[5] 森 辰則, 中川 裕志. 意味マッチングによる比喩理解モデル. 情報処理学会論文誌, Vol. 32, No. 3, 1991. (採録決定).

Conferences

[1] 中川 裕志, 萱島 信, 森 辰則. 階層を持つ知識の自然な再構成. In *Proceedings of the Logic Programming Conference '87*, pp. 125–336. ICOT, 6 月 1987.

[2] Hiroshi Nakagawa and Tatsunori Mori. A parser based on connectionist model. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)*, pp. 454–458, August 1988.

[3] 森 辰則, 中川 裕志. 状況意味論による比喩理解モデルの記述. 言語とその環境シンポジウム論文集. 電子情報通信学会, ソフトウェア科学会, 1月 1990.

Technical Reports, Annual Meetings, etc.

[1] 森 辰則, 中川 裕志. 論理型言語における circumscription. 知識工学と人工知能研究会報告 86-AI-46-10, 情報処理学会, 5月 1986.

[2] 森 辰則, 中川 裕志. Circumscription を用いた概念学習. 知識工学と人工知能研究会報告 86-AI-47-12, 情報処理学会, 7月 1986.

[3] 中川 裕志, 森 辰則. 多重世界機構における circumscription. 日本ソフトウェア科学会第3回大会論文集, pp. 45–48. 日本ソフトウェア科学会, 11月 1986.

[4] 萱島 信, 森 辰則, 中川 裕志. Circumscription による概念階層の再構成. 知識工学と人工知能研究会報告 87-AI-51-3, 情報処理学会, 3月 1987.

[5] 萱島 信, 森 辰則, 中川 裕志. 典型的という性質を用いた知識再構成. 知識工学と人工知能研究会報告 87-AI-52-2, 情報処理学会, 5月 1987.

[6] 森 辰則, 中川 裕志. 記憶資源制約下での connectionist model による自然言語処理. 日本ソフトウェア科学会第4回大会論文集, pp. 463–466. 日本ソフトウェア科学会, 11月 1987.

[7] 森 辰則, 中川 裕志. Connectionist model による自然言語理解モデル. 情報処理学会第36回 (昭和63年前期) 全国大会講演論文集, pp. 1199–1200. 情報処理学会, 3月 1988.

[8] 森 辰則, 中川 裕志. Connectionist model による構文解析モデル. 技術研究報告 (言語とコミュニケーション) NLC 88-1, 電子情報通信学会, 6月 1988.

[9] 森 辰則, 中川 裕志. Connectionist model による自然言語理解モデル. 日本認知科学会第5回大会発表論文集, pp. 54–55. 日本認知科学会, 6月 1988.

[10] 中川 裕志. 自然言語理解における常識推論機構に関する研究. 昭和63年度科学研究費 特定研究 (1) 「言語情報処理の高度化」 研究報告会発表資料, pp. 173–178, 2月 1989. (研究協力者: 田村 直良, 森 辰則).

[11] 森 辰則, 中川 裕志. 分散表現を用いた自然言語理解. 日本認知科学会第6回大会発表論文集, pp. 150–151. 日本認知科学会, 7月 1989.

[12] 森 辰則. 意味マッチング理論に向けて. 知識工学と人工知能研究会報告 89-AI-65-1-2-4, 情報処理学会, 7月 1989. 1989年夏のワークショップ ポジションペーパに基づく討論 「記号主義でどこまでいけるか?」.

[13] 森 辰則, 中川 裕志. 意味マッチングによる暗喩理解. 知識工学と人工知能研究会報告 89-AI-66-6, 情報処理学会, 9月 1989.

[14] 森 辰則, 中川 裕志. 意味マッチングによる暗喩理解. 情報処理学会第39回 (平成元年後期) 全国大会講演論文集. 情報処理学会, 10月 1989.

[15] 森 辰則. 状況意味論による比喩理解モデル. 比喩理解に関するミニ・ワークショップ (早稲田大学), 6月 1990.

[16] 森 辰則, 中川 裕志. 状況意味論による比喩理解モデル. 日本認知科学会第7回大会発表論文集, pp. 42–43. 日本認知科学会, 7月 1990.

[17] 森 辰則, 中川 裕志. 状況意味論による文脈を考慮した比喩理解モデル. 自然言語処理研究会報告 90-NL-78-11, 情報処理学会, 7月 1990.

[18] 森 辰則, 中川 裕志. 状況意味論による文脈を考慮した比喩理解モデル. 情報処理学会第42回 (平成3年前期) 全国大会講演論文集. 情報処理学会, 3月 1991. (発表予定).

Bibliography

[All87] James Allen. *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Menlo Park, California, 1987.

[And80] John R. Anderson. *Cognitive Psychology and its implications.* W.H.Freeman and Company, San Francisco, 1980. (富田他訳:『認知心理学概論』, 誠信書房 (1982)).

[Bar89] Jon Barwise. *The Situation in Logic*, volume 17 of *CSLI lecture notes.* CSLI, 1989.

[BB83] Michael Brady and Robert C. Berwick, editors. *Computational Models of Discourse.* The MIT Press series in artificial intelligence. The MIT Press, Cambridge, Massachusetts, 1983.

[BP83] Jon Barwise and John Perry. *Situations and Attitudes.* The MIT Press, Cambridge, 1983.

[BW84] Robert C. Berwick and Amy S. Weinberg. *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition.* The MIT Press, Cambridge, Massachusetts, 1984.

[Cho88] Noam Chomsky. *Language and Problems of Knowledge — The Managua Lectures.* The MIT Press, 1988.

[dK86] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence* **28**, pp. 127–162, 1986.

[Doy79] Jon Doyle. A truth maintenance system. *Artificial Intelligence* **12**, pp. 231–272, 1979.

[Ear70] J Earley. An efficient context-free parsing algorithm. *Communications ACM*, Vol. 6, No. 8, pp. 94–102, 1970.

[Fan85] Mark Fanty. Context-free parsing in connectionist networks. Technical Report TR174, Computer Science Department, The University of Rochester, 11 1985.

[Fau85] Gilles Fauconnier. *Mental Spaces: Aspects of Meaning Construction in Natural Language.* Bradford/MIT Press, 1985.

[FBK82] Marilyn Ford, Joan Bresnan, and Bonald M. Kaplan. A competence-based theory of syntactic closure. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, chapter 11, pp. 727–796. The MIT Press, Cambridge, Massachusetts, 1982.

[FM86] 古川, 溝口 (編). 自然言語の基礎理論, 知識情報処理シリーズ 第4巻. 共立出版, 1986.

[FM87] 古川 康一, 溝口 文雄 (編). プログラム変換, 知識情報処理シリーズ 第7巻. 共立出版, 1987.

[GFS87] Dedre Gentner, Brian Falkenhainer, and Janice Skorstad. Metaphor: The good, the bad and the ugly. In *Theoretical Issues in Natural Language Processing 3*, pp. 176–180, January 1987.

[GKPS85] G. Gazdar, E. Klein, G. Pullum, and I.A. Sag. *Generalized Phrase Structure Grammar.* Basil Blackwell, Oxford, 1985.

[Gun87a] Takao Gunji. *Japanese Phrase Structure Grammar.* Studies in Natural Language and Linguistic Theory. D.Reidel Publishing Company, Dordrecht, Holland, 1987.

[Gun87b] 郡司 隆男. 自然言語の文法理論. 産業図書, 1987.

[Has85] Koiti Hasida. *Bounded Parallelism.* PhD thesis, Univ. of Tokyo, 1985.

[Has89] 橋田 浩一. 制約と言語. ディスコースと形式意味論ワークショップ 論文集, pp. 161–170. ソフトウェア科学会「論理と自然言語」研究会, ICOT, 3 月 1989.

[Hol89] Keith J. Holyoak. Analogical mapping by constraint satisfaction. *Cognitive Science* **13**, pp. 295–355, 1989.

[Ind87] Bipin Indurkhya. Approximate semantic transference: A computational theory of metaphors and analogies. *Cognitive Science* **11**, pp. 445–480, 1987.

[JL83] P. N. Johnson-Laird. *Mental Models.* Cambridge University Press, Cambridge, 1983. (邦訳: 海保博之 監修: メンタルモデル).

[KB82] Bonald M. Kaplan and Joan Bresnan. Lexical-Functional grammer: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, chapter 4, pp. 173–281. The MIT Press, Cambridge, Massachusetts, 1982.

[Ken85] 白井 賢一郎. 形式意味論入門 — 言語・論理・認知の世界 —. 産業図書, 東京, 9 月 1985.

[KN85] 岸川 徳幸, 中川 裕志. 多重世界機構を用いた非単調論理. ソフトウェア基礎論研究会報告 85-SF-14-1, 情報処理学会, 1985.

[Lak87a] George Lakoff. Position paper on metaphor. In *Theoretical Issues in Natural Language Processing 3*, pp. 194–197, January 1987.

[Lak87b] George Lakoff. *Women, Fire, and Dangerous Things.* The University of Chicago Press, Chicago and London, 1987.

[Lev83] Stephen C. Levinson. *Pragmatics.* Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge, 1983.

[Lif84] Vladimir Lifschitz. Some results on circumscription. In *AAAI Workshop on Non-Monotonic Reasoning*, pp. 151–164, 1984.

[Lif85] Vladimir Lifschitz. Computing circumscription. In *Proceedings of 9th International Joint Conference on Artificial Intelligence*, pp. 121–127, 1985.

[LJ80] George Lakoff and Mark Johnson. *Metaphors We Live By.* The University of Chicago Press, Chicago and London, 1980.

[Llo87] J.W. Lloyd. 論理プログラミングの基礎. ソフトウェア サイエンス シリーズ. 産業図書, 東京, 1987. (邦訳: 佐藤, 森下).

[MaC80] John MaCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence* **13**, pp. 27–39, 1980.

[MaC84] John MaCarthy. Applications of circumscription to formalizing commonsense knowledge. In *AAAI Workshop on Non-Monotonic Reasoning*, pp. 295–324, 1984.

[Mar80] M. Marcus. *A Theory of Syntactic Recognition for Natural Language.* The MIT Press, Cambridge, 1980.

[MD80] Drew McDermott and Jon Doyle. Non-monotonic logic I. *Artificial Intelligence* **13**, pp. 41–72, 1980.

[MRtPRG86] J.L McClelland, D.E. Rumelhart, and the PDP Research Group. *Parallel Distributed Processing.* The MIT Press, Cambridge, 1986.

[Nak85a] 中島 秀之. 知識表現と Prolog/KR. ソフトウェア サイエンス シリーズ. 産業図書, 1985.

[Nak85b] 中島 秀之. 超時空プログラミングシステム uranus. 第 26 回 プログラミング・シンポジウム, pp. 13–23. 情報処理学会 プログラミング・シンポジウム委員会, 1985.

[Nak86] 中川 裕志. 論理型言語における explicit な否定. 第 27 回 プログラミング・シンポジウム, pp. 13–21. 情報処理学会 プログラミング・シンポジウム委員会, 1986.

[Nak87] 中川 裕志. 論理 + サーカムスクリプション = 常識推論. 人工知能学会誌, Vol. 2, No. 1, pp. 14–21, 3 月 1987.

[Nak89] 中川 裕志. 極小限定の研究動向 — 論理プログラムの意味論との関係 および計算化について —. 情報処理学会誌, Vol. 30, No. 6, pp. 684–685, 6 月 1989.

[NF83] 長尾 真, 淵 一博. 論理と意味, 岩波講座情報科学 第 7 巻. 岩波書店, 東京, 1983.

[NN86] 野村浩郷, 内藤昭三. 自然言語理解における意味表現. 情報処理 特集: 計算言語学, Vol. 27, No. 8, pp. 906–914, 8 月 1986.

[Nom88] 野村 浩郷. 自然言語処理の基礎技術. 電子情報通信学会, 1988.

[NTS84] 中島 秀之, 戸村 哲, 諏訪 基. Prolog/KR から Uranus へ — 多重世界機構の拡張 —. 知識工学と人工知能研究会報告 84-AI-36-2, 情報処理学会, 9 月 1984.

[Nun87] Geoffrey Nunberg. Poetic and prosaic metaphors. In *Theoretical Issues in Natural Language Processing 3*, pp. 198–201, January 1987.

[OF87] Andrew Ortony and Lynn Fainsilber. The role of metaphors in descriptions of emotions. In *Theoretical Issues in Natural Language Processing 3*, pp. 181–184, January 1987.

[Ots87] 大津 由紀雄 (編). ことばからみた心, 認知科学選書 第 13 巻. 東京大学出版会, 東京, 8 月 1987.