TR-220

A Procedure Resolving Anaphoric Ambiguity by Finding the Most Informative Interpretation

by

T. Ukita, K. Sumita, and S. Amano
(Toshiba Co.)

December, 1986

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

A Procedure Resolving Anaphoric Ambiguity by Finding the Most Informative Interpretation

Teruhiko Ukita, Kazuo Sumita, and Shinya Amano

Information Systems Laboratory,
Research and Development Center, Toshiba Corp.
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki,
210 Japan

A Procedure Resolving Anaphoric Ambiguity by Finding the Most Informative Interpretation *

ABSTRACT

An inferential approach to resolving anaphoric ambiguity is described. The procedure finds the most plausible anaphoric referent, based on three general principles of understanding discourse; non-contradiction, relevance, and optimality. For ambiguous anaphora, which cannot be identified by syntactic and semantic constraints, the procedure produces a tentative interpretation, whose internal representation corresponds to a case-frame of a predicate. Tentative interpretations are produced for all combinations of referent candidates. Then, the procedure inspects them and excludes improper interpretations in accordance with non-contradiction with the previous sentences. As the step for optimal selection, the procedure identifies anaphoric referents by finding an interpretation which has causal relation with previous messages; causal relation is used to supplement some shortage of information and to give an either affirmative or negative explanation to the interpretation. Combined with a frame-based knowledge representation system, the procedure has been implemented in an experimental question-answering system, and is being tested for real data.

---

* This work is a part of a project on Fifth Generation Computer Systems (FGCS).

I. INTRODUCTION

In building a question-answering system accepting natural language input, it is necessary to deal with many problems, such as anaphoric references and ellipses, indirect speech act, fragmental input, conversation initiative, etc. Among these problems, determining anaphora and complementing ellipses are the most fundamental problems which deter understanding a user's message, because they are the first stage in a question-answering system.

The anaphora and ellipsis problems have been analyzed and managed from various points of view (e.g., [Webber 80]), including syntactic and semantic constraints, inferential approach to find antecedents, and use of focus* to resolve discourse ambiguity. Though focus is a comprehensible concept to explain anaphoric reference, its extraction process is not well-established. Sidner proposed a "bootstrapping" procedure which decides focus by verifying its consistency with the contexts [Sidner 83]. In her approach, the inference function is used to verify consistency of the focused object with the contexts. Using inference in such a way seems insufficient, because her procedure cannot select a preferable interpretation when there are plural "consistent" interpretations. It can be seen that extracting proper focus from the discourse context is a process of understanding the discourse itself.

---

* The term focus here means local context according to [Sidner 83] or "local center" according to [Grosz, et al. 83].

In this paper, the authors explore a way of using the inference function to resolve anaphoric ambiguity. In the subsequent sections, principles to resolve anaphoric ambiguity are discussed and a practical procedure, using a knowledge representation system, is described.

II. PRINCIPLES TO RESOLVE DISCOURSE AMBIGUITY

The problem of resolving anaphoric ambiguity can be viewed as selection of the most plausible interpretation corresponding to a new sentence. In this process, the inference function is required to decide the referents as well as to extract implicit antecedents. This paper focuses on general principles to guide a procedure to disambiguate anaphoric referent and on a way of using inference in determining it.

"Coherency" had been proposed as a general concept to explain discourse structures [Hobbs 77; Grosz, et al. 83], and various phenomena in discourse, including anaphoric identification, are discussed in accordance with this concept. Though it is an attractive concept, more precise principles are needed to construct a practical procedure to resolve anaphoric ambiguity, specifically to design a procedure based on a practical knowledge representation system.

There have been researches carried out on controlling inference in understanding discourse. For example, Rieger arranged types of inference and proposed a method of reference establishment in a heuristic way [Rieger 75]. Schank proposed "interestingness" in understanding a story and used numerical scores to control inference [Schank 79]. Joshi and Weinstein used "appropriateness", which is defined by a number of common objects between sentences [Joshi and Weinstein 79]. Wilensky proposed basic principles (e.g., ACHIEVE AS MANY GOALS AS POSSIBLE) to control inference and implemented them by the mechanism of meta-goals and meta-plans [Wilensky 83], where his main theme was understanding a story by finding goal-plan chains between sentences. Such researches can be viewed as an effort to make a consistent representation within practical time and space for a large chunk of sentences, such as a story.

A psychologist proposed general principles of understanding sentences [Uchida 82]; they are, consistency of a new interpretation with a hearer's knowledge base, optimality of the new representation, and open-endedness of the knowledge representation system. The former two principles are stated to construct a proper representation for the new sentence. The third principle is used to detect unnaturalness in the whole interpretation and then to reconstruct the interpretation. For identifying anaphoric referent, consistency and optimality principles are important. This will be discussed more precisely from the viewpoint of anaphoric disambiguation. In this paper, the consistency concept is divided into two concepts; non-contradiction and relevance.

Non-contradiction is the most fundamental property that the new interpretation should have. The syntactic and semantic constraints can be viewed as a basic filter, which protects us from adding a superficially contradictory representation. In general, the new added interpretation should not violate consistency with the previous speakers's utterances, as well as a knowledge base of the system. Though an intentional violation of consistency might give a hearer a cue to detect some unusual effect, such as a metaphor, it may be assumed that a speaker's utterance is not self-contradictory or contradictory with a knowledge base, at least within the scope of a simple question-answering system. When this assumption is posed, the machine system can exclude tentative interpretations which are contradictory to previous messages and/or the built-in knowledge base.

Relevance is a principle in which a new internal representation must have some links with the previous interpretations and the knowledge base. This principle is required to avoid adding a completely irrelevant interpretation, even if it is not in contradiction with the previous interpretation. In general, the smaller a system scope is, the more important this principle comes to be, because the knowledge base for the system is small and, therefore, a tentative interpretation could be irrelevant to the knowledge base, with high possibility.

In resolving anaphoric ambiguity, the relevance principle can be used to exclude tentative interpretations which have no relation with the knowledge base. For example, consider an example of the following sentences:

"A bug was in a program and the computer stopped. I started it again."

For the second sentence, an interpretation of "I started the bug again" is possible, even for a situation where the knowledge base only consists of information on computer systems. Though this interpretation is not unnatural from the viewpoints of syntax or semantics, it seems quite trivial, unless the interpretable world is so wide that it includes fairy tales or some kinds of competitions. For a usual task, which covers a limited domain, such as information retrieval on computer systems, the system can exclude such kinds of irrelevant interpretations, even if they satisfy non-contradiction principle.

Optimality can be viewed as a principle wherein the system selects the most informative interpretation. When there are more than two possibilities for an anaphoric referent, the system must decide which candidate is preferable. For example, consider the following simple sequence of sentences.

"John phoned Jack. He stayed home."

In this example, an interpretation of "He = Jack" is preferable to that of "He = John". This preference comes from the fact that these two sentences can be interpreted as saying "John succeeded in communicating with Jack by telephone". In addition, we must notice that, even if the second sentence in the example is negated, we observe a similar phenomenon, whose conclusive item is a negative statement in this case.

This example shows that the inference function is required to draw some conclusions regarding items not spoken explicitly. In general, it can be said that the system should search for an interpretation which can supplement a shortage of information or which can produce effective information; in a word, the system should find the most informative interpretation. To do so, the system can use the inference function, based on causal relations prepared in the knowledge base for the system. In resolving anaphoric ambiguity, though the inference function has been used to find inferential antecedents, it should also be used to decide an informative interpretation.

III. PROCEDURE

In this paper, a procedure to resolve anaphoric ambiguity follows the principles described above; non-contradiction, relevance, and optimality. Figure 1 shows the procedure flow resolving anaphoric ambiguity, which consists of four major steps; anaphora detection, referent candidates extraction, non-contradiction test, and optimal candidate selection.

The procedure uses a knowledge representation system, which the authors' colleagues have been implementing [Konno, et al. 86]. The knowledge representation system represents an object and an event as a "schema" in a similar way as the frame [Minsky 75], and causality relations between events by if-then rules (thoughout the explanation of the procedure in this section, the terminology related to the knowledge representation system will be simplified to as general a level as possible).

The procedure has been constructed as a part of an experimental question-answering system, whose present task is guidance in operating household electric appliances, such as VTR and TV. Every input sentence is analyzed syntactically and semantically, and represented as a case-frame representation which corresponds to an event [Sano and Amano 87]. After this analysis, the procedure described here will be invoked.

3.1 Anaphora detection

As the first step of disambiguating anaphora, the procedure detects anaphoric indicators for a sentence. Pronouns and definite articles, which are explicit anaphoric indicators, must be detected. At the same time, nouns themselves are extracted as candidates for anaphoric representations. In addition to them, omitted obligatory cases of the predicate in a sentence are detected as ellipses. In a situation involving a question-answering system, pragmatic ellipses can also be detected, when an interpretation attained for an input sentence is insufficient to form a command for task execution.

After the procedure detects the anaphoric and elliptical indicators, it produces a temporary instance schema for every indicator. The procedure then inserts the temporary schema into the slot representing case of the predicate. At this point, for instance, a sentence "He pushed the switch" is represented as shown in Fig. 2.

3.2 Referent candidate extraction

For every anaphoric indicator, the procedure searches for candidates corresponding to the indicator. The procedure searches for "instance schemata", which belong to the same class as the indicator and belong to a super-class of the indicator's class (including some other relations, such as a whole-part link). In searching for candidates, the procedure uses constraints written at every slot that restrict the class of objects, to reduce the number of objects to be searched for. For example, the semantic template of a word "push" in the knowledge base is shown in Fig. 2. In this case, a slot value for a "push" agent is restricted to being human by the additional description of "value-class". When a subject of a sentence is omitted (in Japanese, this is frequently done), the procedure searches for instance schemata belonging to the human class. In general, the number of new tentative representations for the sentence becomes plural, because most verbs usually have more than two case objects. This search process is performed by the knowledge representation system invoked by the procedure.

The relevance principle has to be satisfied at this point, because the procedure should succeed in finding some relations between a case slot and possible objects which have previously appeared. (Otherwise, the procedure fails to identify the anaphoric referent.)

3.3 Non-contradiction test

Non-contradiction is verified by two subsequent steps. The first step is test for non-contradiction within a single sentence, where the corresponding inner representation has more than two undetermined slot values (case objects for a predicate). Though it is impossible to state all legal combinations of objects, this step is effective to rule out improper combination of objects for a multi-meaning verb, such as 'have', and to decide the precise meaning of the word.

The second step in this process is scanning non-contradiction between the new hypothetical interpretation and the previous user's messages. In general, the procedure must verify non-contradiction between the new interpretation and the whole knowledge base. However, in such a system whose task is diagnosis, it may be usual that a user's input is contradictory with regard to a knowledge base, because fault symptoms are abnormal and contradictory with normal state. Thus, at present, the procedure verifies non-contradiction between a new interpretation and the previous user's messages alone. For a systematic solution to this problem, the procedure and the supporting knowledge representation system should be enhanced so that they can manage contradiction explicitly as the logical truth maintainer of TMS (Truth Maintenance System) [Doyle 79].

As an example of this step, assume a sequence of sentences:

" . . .

I pushed the recording button.

(But) I didn't push it. "

Deciding anaphoric referent of "it" in the second sentence, the procedure excludes a possibility "it = recording button", and inspects other candidates. Figure 3 shows internal representations for these interpretations, which are easily detected as contradiction, because these two representations belong to the same class and have the same entries, except an entry of negation.

3.4 Optimal candidate selection

When more than two interpretation possibilities remain, the procedure tries to find causal links which can be produced from the interpretations. By finding the interpretation which has some causal links with the previous sentences, the procedure decides the most informative interpretation. In inspecting causal connectivities for interpretations, the procedure searches for rules*, which can be classified into two groups; one is a rule which has both previous and current sentences in the conditional part and another is a rule that has the previous sentences in the conditional part and the current one in the conclusive part.

---

* Rules are used to represent operation procedures as well as causal relations in our knowledge representation system.

Assume that X denotes an event in a tentative interpretation being processed, A is an event or a conjunction of events corresponding to input sentences, and B represents an inferable, goal event or an event which explicitly denotes objective of an action in a sentence. The procedure searches for rules in the order shown below ($\neg X$ means a negation of X).

(1) if A then $\neg X$,

(2) if A & $\neg X$ then B

(3) if A & X then $(\neg)B$

(4) if A then X.

The first and the last rules show direct causal links between the previous interpretation (=A) and a hypothetical one (=X). The others are rules that have interpretations in conditional part.

The order of rules to be searched is decided so that it might correspond to the magnitude of reducing ambiguity; e.g., explicit negation of a plausible event can be considered to give a hearer more information than a positive interpretation for the event.

Consider an example sequence of sentences:

"I pushed the play-back button in front of the VTR.

(But) The tape was not played back."

To decide the elliptical referent of a subjective actor (=VTR) in the second sentence, the procedure finds a rule such as follows (in practice, terms in rules are represented by schemata in our system and the rule has more terms):

if (a person pushes the play-back button)

then (VTR plays back a tape).

In this case, the rule is classified to category (1), which has a (doubly) negated event for the input sentence in the conclusive part. After finding the above rule and verifying that the VTR has the play-back button, the procedure recognizes the agent of the second sentence as the VTR.

Let's inspect another example:

"Though I pushed the button, the VTR didn't play back the tape."

Processing "the button" in the first part of the sentence, the procedure recognizes it as a button to play back, from various kinds of buttons that the VTR mounts on its control panel. In this case, the above rule is used as a category (3) rule.

It is important to note that the procedure, at present, searches for rules whose terms in the conditional part are restricted, so that they do not include events that can be inferred from rules in the knowledge base. This restriction is introduced in order to avoid an intractable state of confusion in controlling the inference.

After the procedure identifies an anaphoric referent, it merges the identified interpretation into the previous one by replacing the link pointer in the event schema. Then, the procedure goes on to the next sentence.

IV. CONCLUDING REMARKS

An inferential approach to resolving anaphoric ambiguity is described. In the paper, general principles to resolve anaphoric ambiguity were discussed. They are non-contradiction, relevance, and optimality. Based on these principles, the procedure detects anaphora occurence, extracts referent candidates, and verifies non-contradiction and optimality for a tentative interpretation, for every input sentence. After extracting referent candidates, the procedure produces a tentative interpretation, represented as a case-frame, for a combination of the candidates. Then, the procedure excludes an improper interpretation, which is in contradiction with the previous sentences. In the last step, the procedure finds the most informative interpretation, in the sense that some meaningful causal links are obtained. In this paper, causal rules are not only used to find direct relations between sentences, but also used to evaluate tentative interpretations by giving a negative or positive explanation to the interpretation.

The procedure has being implemented as a Japanese sentence understander in an experimental question-answering system, in combination with the syntactic and semantic analyzer, and the knowledge representation system. The procedure described in the paper uses a lot of facilities in the knowledge representation system; e.g., representing a sentence interpretation by a schema (frame), matching a schema against other schemata, representing causal relations by if-then rules, and so on. It might be worth noting, however, that these functions, required by the procedure, are tractable within the scope of the current knowledge representation art.

There is a problem remaining for further study. At present, the procedure described in this paper identifies anaphoric and elliptical referents for objects, and cannot deal with a problem of phrase or sentence ellipses. This problem seems serious, because the procedure in this paper uses a case-frame representation as a basis for the processes, and cannot produce any tentative interpretations for such ellipses. For this problem and, more generally, for the problem of fragmental input, a kind of expectation-driven mechanism would be needed.

ACHNOWLEDGMENT

The authors gratefully wish to thank Dr. T. Yokoi of ICOT (Institute for New Generation Computer Technology) for his encouragement during this research.

REFERENCES

[Doyle 79]
Doyle, J., "A Truth Maintenance System", Artificial Intelligence, Vol.12, pp.231-272 (1979).

[Grosz, et al. 83]
Grosz,B.J., Joshi,A.K. and Weinstein,S., "Providing a Unified Account of Definite Noun Phrases in Discourse", Proc. 21st Annual Meeting of the Association for Computational Linguistics, PP.44-50 (1983).

[Hobbs 77]
Hobbs, J.R., "Coherence and Interpretation in English Texts", Proc. IJCAI-77, pp.110-116 (1977).

[Joshi and Weinstein 81]
Joshi,A.K. and Weinstein,S., "Control of Inference: Role of Some Aspects of Discourse Structure - Centering", Proc. IJCAI-81, pp.385-387 (1981).

[Konno, et al. 86]
Konno, S., Ishida, K. and Amano, S., "A Plan of Knowledge Representation System for Discourse Understanding", Proc. 33rd Meeting, Information Process. Soc. Japan, 4L-5 (1986) (in Japanese).

[Minsky 75]
Minsky, M., "A Framework for Representing Knowledge", in P.H. Winston (Ed.), The Psychology of Computer Vision, New York : McGraw-Hill (1975).

[Rieger 75]
Rieger,C.J., "Conceptual Memory and Inference", in R.C. Schank (Ed.), Conceptual Information Processing, North-Holland / American Elsevier (1975).

[Sano and Amano 87]
Sano, H. and Amano, S., "A Theory of Word Class for Analysis of Japanese Sentence", Proc. 34rd Meeting, Information Process. Soc. Japan (1987) (in Japanese, to appear).

[Shank 79]
Shank,R.C., "Interestingness: Controlling Inferences", Aritificial Intelligence, Vol.12, pp.273-297 (1979).

[Sidner 83]
Sidner,C.L., "Focusing in the Comprehension of Definete Anaphora", in M.Brady and R.C.Berwick (Eds.), Computational Models of Discourse, MIT Press (1983).

[Uchida 82]
Uchida, N., "Understading Sentences and Knowledge", in Y.Saeki (Ed.), Inference and Understanding, Tokyo Daigaku Shuppan (1982) (in Japanese).

[Webber 80]
Webber, B.L., "Syntax beyond the Sentence: Anaphora", in R.J. Spiro, et al. (Eds), Theoretical Issues in Reading Comprehension, Lawrence Erlbaum Associates, Inc. (1980).

[Wilensky 83]
Wilensky,R., "Planning and Understanding", Addison-Wesley Publishing Company (1983).

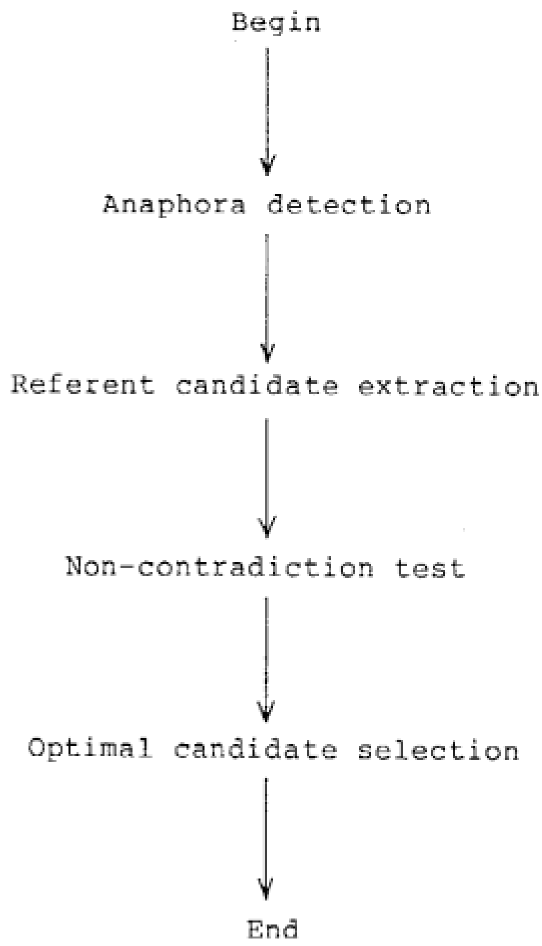

Figure 1. Procedure flow

```
push#1
    class: push
    tense: PAST
    agent: man#1
    object: button#1
```

```
man#1
    class: man
```

```
button#1
    class: button
```

(a) instance schemata for "He pushed the button".

```
push
    super-class: event
    agent: [value-class: human]
    object : [value-class: button]
    . . .
```

```
man
    super-class: human
    sex: male
    . . .
```

```
button
    super-class: parts
    . . .
```

(b) class schema examples built in the knowledge base

Figure 2. A representation example of schemata for a sentence "He pushed the button."
(Entries followed by "#" are instances.)

```
push#1
    class: push
    tense: PAST
    agent: author#1 ( = I )
    object: button#1
```

(a) instance schema for "I pushed the button".

```
push#2
    class: push
    tense: PAST
    agent: author#1 ( = I )
    object: button#1 (tentative referent candidate for "it")
    negation: YES
```

(b) tentative interpretation for " I didn't push it."

Figure 3. A tentative interpretation example

外部発表許可願

昭和61年12月10日
新世代コンピュータ技術開発機構

題目	Towards Qualitative Physics			
発表者（代表者に○印）	所属	氏名	所属	氏名
	第5研究室	大木 優 古川 康一		
発表期日	昭和61年12月14日			
発表先	認知科学会 シンポジウム　　TR-221			
発表の理由・目的	技術研究の優先主張・(討議結果の研究へのフィードバック)・技術解説／宣伝 その他（　　）			
発表内容	研究開発事項の成果・(研究開発事項の中間結果)・研究開発事項の構想 一般事項・その他（　　）			
連絡事項（開発題目との関係など）				

外部機関との関係	(無)	①外注 ②委員会・WG ③共同研究 ④その他（　　）	契約書等の題名	相手方	契約日
			相手方 役職名 氏名 の許可を 年 月 日 口頭／書面 で得た		

承認印／所見など	所長記入欄	研究部長記入欄	室長記入欄
	承認 (印)	承認 (印)	承認 (印)

○印のみ報告者記入

ICOT Technical Report	1. 報告番号 TR-221
②. 標題 Towards Qualitative Physics	③. 報告年月日 昭和61年12月1日 4. 活動分類 所内 WG（WG名：　） その他（　）
⑥. 報告者氏名・所属 第5研究室　大木　優 古川　康一	5. 承認印 所長 / 研究部長 / 室長

⑦. 概要（できるだけ和文のもの、英文のもの双方つける）

We hope to make computers that can solve high-level physical problems, for example design of devices and prediction of physical phenomena. We think that "understanding" of the physical world is important if computers are to solve such high-level problems. We are developing a reasoning system to perform basic reasoning about the physical world. In this paper, we describe a reasoning system which grasps relations among physical objects and predicts the next state of a physical phenomena (we call the reasoning system Qupras (for Qualitative physical reasoning system)). Qupras has two knowledge representations. One of them is for knowledge of physics corresponding to the physical laws found in physics textbooks. The other is knowledge about objects, which are components of physical devices or subsystems of the physical world. Qupras reasons relations among physical objects using restricted sets of physical laws, just as human beings do. Reasoning in Qupras is based on qualitative reasoning, but it handles physical variables quantitatively as well as qualitatively. This is because almost all physical laws are quantitative expressions, and we want to avoid losing quantitative information, which is often more comprehensive than qualitative information.

（和文の概要の中に、下記事項の１以上記載のこと．）

(1) 発表の内容には，ノウハウを含んでいないこと．

(2) 特許，実用新案等工業所有権○○件出願中（または予定）．

(3) 既承認の○○○と同一のものであること．

⑧. キーワード

Qualitative Reasoning　Understanding of the Physical worlds

9. 保秘コメント（特許・ノウハウ関係） 適（1. 出願関係なし，2. 出願手続済），不適（1. 出願完了，2. ノウハウ）	10. 総頁数 3

注）査読結果を反映したいと思いますので
4月上旬頃に本文を差し換えます

Towards Qualitative Physics

Masaru Ohki and Koichi Furukawa

ICOT Research Center,
Institute for New Generation Computer Technology,
Mita Kokusai Bldg. 21F, 1-4-28, Mita,
Minato-ku, Tokyo, 108, Japan

Abstract

We hope to make computers that can solve high-level physical problems, for example design of devices and prediction of physical phenomena. We think that "understanding" of the physical world is important if computers are to solve such high-level problems. We are developing a reasoning system to perform basic reasoning about the physical world. In this paper, we describe a reasoning system which grasps relations among physical objects and predicts the next state of a physical phenomena (we call the reasoning system Qupras (for Qualitative physical reasoning system)). Qupras has two knowledge representations. One of them is for knowledge of physics corresponding to the physical laws found in physics textbooks. The other is knowledge about objects, which are components of physical devices or subsystems of the physical world. Qupras reasons relations among physical objects using restricted sets of physical laws, just as human beings do. Reasoning in Qupras is based on qualitative reasoning, but it handles physical variables quantitatively as well as qualitatively. This is because almost all physical laws are quantitative expressions, and we want to avoid losing quantitative information, which is often more comprehensive than qualitative information.

1. Introduction

One of the targets of artificial intelligence is to implement computers which have intellectual capabilities similar to those of human beings. We hope to produce computers that can solve high-level problems such as design of devices, diagnoses of devices, and prediction of physical phenomena. However, we have to implement on computers various capabilities which human beings have. One such capability is "understanding". We think that understanding of the physical world is important for computers if they are to solve high-level problems. Many high-level capabilities of human beings, such as learning and designing, are based on understanding. Thus, solid understanding should form the basis of implementing these capabilities on computers.

This paper describes a reasoning system that grasps relations among physical objects and predicts the next states of physical phenomena. We call the reasoning system Qupras (Qualitative physical reasoning system). The physical world can be explained on several levels. Explanation at the level of the quark is perhaps one of the deepest. On the other hand, explanation using experiential knowledge is rather shallow. Physical laws given in physics textbooks are at the level of general knowledge, and as such they are widely used to explain relations among physical objects. For example, if a person versed in physics is given an explanation of a device in terms of physical laws, he will probably understand it. When he comes upon a physics problem which he has never encountered before, he will try to solve the problem using the physical laws he knows, because he does not have the knowledge peculiar to the problem, the compiled knowledge. Thus, we can say that physical laws are primitive knowledge to understand or solve physical problems.

The laws of physics constitute primitive knowledge of the physical world. We think it best to use knowledge at this level of physical laws to specify relations among objects. Reasoning in Qupras is based on qualitative reasoning [de Kleer 84, Kuipers 84], with the added feature that it can handle physical variables quantitatively also. If we use the more comprehensive quantitative information, it is possible to decrease ambiguity at transition to subsequent states resulting from the arithmetic operations on use signs only. This is one of the problems in qualitative reasoning in which physical variables are treated qualitatively. Qupras has two primitive representations. One represents physical laws and the other represents (physical) objects. Using these representations Qupras reasons on:

(1) relations among objects which are components of physical systems, and
(2) the next states of the systems following transition.

In Section 2 we describe the structure of Qupras, the representation of Qupras, and reasoning in Qupras. Section 3 gives an example analyzed by Qupras. Finally, we discuss Qupras along with related research and unresolved issues.

2. Qupras

2.1 The Structure of Qupras

Qupras consists of a representation section and a reasoning section. The representation section contains knowledge of physics and objects, and is dependent on applications. The reasoning section determines the physical relations that hold among the objects and their next subsequent states. The structure of Qupras is given in Figure 1.

```
The Representation Section
+----------------------------------------------------+
|     1. Object : representation of objects          |
|            (1) Applied conditions                  |
|            (2) Relations                           |
|                                                    |
|     2. Physics : representation of physics laws    |
|            (1) Objects                             |
|            (2) Applied conditions                  |
|            (3) Relations                           |
+----------------------------------------------------+

 The Reasoning Section
+---------------------------------------------------------+
| 1. Qualitative reasoning                                |
|        (1) Propagation                                  |
|            Determine status of physical system          |
|            at a given time.                             |
|        (2) Prediction                                   |
|            Determine the physical variables,            |
|            and values which should change.              |
| 2. Qualitative/quantative evaluation of inequalities    |
|    Evaluate inequalities, i.e., applied conditions,     |
|    using relations.                                     |
+---------------------------------------------------------+
```

Figure 1. Structure of Qupras

The following is a brief description of the representation section, its details will be given later. The representation of objects in this section mainly consists of applied conditions and relations. Applied conditions correspond to conditions for the existence of the objects. Objects satisfying these conditions are called active objects. Relations are expressed as relative equations which include physical quantities when the objects are active. Representations of systems of physics consist of objects, applied conditions, and relations. The objects are those necessary to apply the system of physics. The representations of applied conditions and relations are similar to those of objects. Applied conditions are those required to activate a system of physics, and those systems which are satisfied with their conditions and whose necessary objects are activated are called the active systems of physics. When a given system of physics is active, its relations are relative equations holding among physical quantities of objects specified in the physical representation, and they correspond to physical laws.

The reasoning section consists of qualitative reasoning and qualitative/quantitative evaluation for inequalities. There are two kinds of qualitative reasoning, propagation and prediction, similar to those given in previous papers [Kuipers 84, de Kleer 84]. Both quantitative information and qualitative information are used in both types of reasoning. Propagation reasoning determines the state of the physical world at a given time (or during a given time interval). Prediction reasoning determines physical variables that change with time and infers their values at the next given point in time. Then propagation reasoning predicts the next states of the physical world using the results of prediction reasoning.

Physical laws are specified in quantitative expressions, and qualitative/quantitative evaluation of inequalities (i.e., quantitatively as well as qualitatively) determines the conditions on objects and physical representations. These conditions are expressed as binomial relations or terms. Conditions are evaluated using relations in active systems of physics and active objects. The unit performing this qualitative/quantitative evaluation is called the expression evaluator. It determines conditions using quantitative values as much as possible, since quantitative values are known or are obtained from the relations. Qupras differs from the former qualitative reasoning systems[Kuipers 84, Forbus 84, de Kleer 84] at the point where it quantitatively handles physical variables. One of the merits to handling physical variables quantitatively is that it is not always appropriate to handle all physical variables qualitatively. Treating all variables qualitatively (even when the values of variables are quantitatively known) may lead to ambiguity, because quantitative information is abandoned to handle all variables qualitatively, though the quantitative data is move comprehensive. There are some cases when variables have to be treated quantitatively to understand physical phenomena. For instance, if we have understood the relation between the life of an electric battery and consumption of power per unit time quantitatively, we can easily infer that the life of the battery is reduced by half when consumption per unit time doubles. Another merit of quantitative treatment is that it is not necessary to transfer quantitative expressions to qualitative expressions [de Kleer 84]. Qupras uses quantitative physical laws directly, so it dose not have to translate the physical laws to qualitative representations. If only the qualitative information is known, Qupras can also use that qualitative information.

2.2 The Representations of Qupras

In Qupras, physical objects are described by the predicate "object" and systems of physical laws are described by the predicate "physics". The initial states ,which specify objects involved and initial facts, are described by the predicate "initial_state". "initial_state" is regarded as a problem which Qupras has to solve.

"object" specifies parts, attributes, conditions and relations. The parts specify the components of the object. Attributes are physical quantities. The conditions and relations are applied conditions and relative equations among the physical quantities of the object, respectively. An example of an "object" is given in Figure 2, a specification of a boiler. The variable "Boiler" in the first line is used to describe itself in the "object" description. The boiler consists of two parts. One part is a container holding

```
object boiler:Boiler
   parts_of
      Container - container ;
      Heat_source - heat_source ;
   relations
      on(container!Boiler,heat_source!Boiler) ;
      melting_point@container!Boiler <
                  temperature@heat_source!Boiler ;
  end.
```

Figure 2. Description of a boiler in Qupras

liquid, and the other is a heat source heating the container. The first term in the "part_of" description specifies a variable to describe the part in the "object" description. That is, an instance of a part is shown as a variable. The second term gives the name of the object description. The relations are those holding between attributes and between parts, the first of which states that the container of the boiler is on its heat source, and the second that the melting point of the container is below the temperature of the heat source.

We use some special notations in Qupras. "<partname>!<variable>" indicates an instance of the part whose name is <partname>. The <variable> is a variable indicating the whole. The notation "<attribute>@<instance>" is used to show the value of the <attribute> in the <instance>, which is an instance of an object. We use other notations also. "ddt(<variable>" is used to indicate the temporal derivative of the <variable>. "coincidence(<variable 1>,<variable 2>)" says that <variable 1> and <variable 2> are working together. That is, it states that if <variable 1> has a value, <variable 2> simultaneously has the corresponding value, and vice verse.

The predicate "physics" specifies objects, conditions and relations of the physical system in question. Objects are those to which this system of physics applies. The syntax of "object" within "physics" is the same as that of the "part_of" within "object". The conditions are the applied conditions to activate the system of physics. Binomial relations and terms can be described in the conditions. Several relations, equality(=), no equality(<>), equality of sign(:=:), no equality of sign(:<>:) and inequalities(>,>=,<,=<), can be specified in binomial relations. The expression evaluator tries to prove the conditions using the relations of active objects and active systems of physics. If conditions are proved to be true, the conditions are regarded as satisfied, otherwise the conditions are regarded as unsatisfied. If all the objects specified in the "object" are not active and all the applied conditions are not satisfied, the "physics" is not activated. In the relations, corresponding to physical laws, it is possible to describe relations whose right sides are general expressions and whose operators are the same as those of the binomial relations, in addition to the binomial relations and the terms. The relations in an active "physics" and in an active "object" are regarded as truths that hold in the physical world. Figure 3 is an example of a "physics" illustrating the specification in Qupras of the physics of "heat flow" from a heat source to a heat sink. This "physics" of heat

```
physics heat_flow
  objects
     Heat_source - heat_source ;
     Container - container ;
  conditions
     on(Container,Heat_source) ;
      temperature@Heat_source <> temperature@Container ;
  relations
     ddt(heat@Container) :=: temperature@Heat_source -
                            temperature@Container ;
 end.
```

Figure 3. Descriptipon of the physics of "heat flow"

flow needs two objects, a container and a heat source. There are two conditions to activate this "physics". One condition is the locational relation between the container and the heat source, and it says that the container must be on the heat source, while the other condition is the temperature relation between the two, and it says that the temperature of the heat source must be different from the temperature of the container. The relation of this "physics" shows that the sign of the temporal derivative of the container's heat is equal to that of the difference between the temperature of the heat source and the temperature of the container.

2.3 Reasoning in Qupras

There are two reasoning mechanisms in Qupras as described in Section 2.1 above. One is qualitative reasoning and the other is qualitative/quantitative evaluation of inequalities performed by the expression evaluator. We describe the latter first.

The expression evaluator is used to test whether the conditions in the "object" and "physics" descriptions are proved by the known relations obtained from active objects and active systems of physics. Relations are given as expressions and terms. We ignore the terms in this discussion, because their evaluation is very simple. The expression evaluator evaluates the conditions as follows:

(1) Before evaluation of a condition, first examine the values or signs of each variable on the left and right side in a condition.
(2) If the values or the signs are obtained, try to test the condition using them. If the condition can be tested and its result is not suspension, return the result. If the result is suspension, this shows that it is impossible to evaluate the condition using the values or the signs. For example, in the case when only the signs of two variables, "X" and "Y", are known to be positive, it is impossible to evaluate a binomial relation "X > Y".
(3) If the result is suspension, logically evaluate the conditions using the known relations. In this step, the relations are regarded as logical relations.

Here are some examples. The first example is evaluation of the following binomial relation:

X > Y

using the relations below:

X = 5 + C.
Y = A - B.
A < B.
C = 5.

The expression evaluator tries to determine the values of "X" and "Y". The value of "X" can easily be shown to be 10 by evaluating 5 + C, but finding the value of "Y" is not so easy. It is necessary to calculate the expression "A - B" in order to get the value of "Y". The values of "A" and "B" are not obtained directly. The expression evaluator easily determines the sign of "A - B" as negative from the relation "A < B", and the sign of "Y" is thus found to be negative. Finally, the first binomial relation "X > Y" is proved from the value of "X" and the sign of "Y". If the sign of "Y" is positive, the expression evaluator cannot evaluate the binomial relations.

The next example is also evaluation of the same binomial relation:

X > Y

but the known relations are different:

X > A.
A > B.
B > Y.

The expression evaluator tries to obtain the values of "X" and "Y" in the same way as the previous example. But the values and signs of both variables cannot be directly determined. Therefore, the expression evaluator tries to evaluate the binomial relation using transition rules. There are many transition rules, for example:

X > A, A > Y → X > Y.
X > A, A = Y → X > Y.
X >= A, A > Y → X > Y.

Now, the binomial relation can be proved by applying the first transition rule twice. Qupras has transition rules for failure in addition to the above transition rules, because the capabilities of the expression evaluator are not complete. That is, even if the expression evaluator cannot prove a binomial relation, this does not mean that the binomial relation is false. Transition rules for failure are:

X > A, A > Y → failure of X < Y.
X > A, A = Y → failure of X < Y.

The expression evaluator is not very powerful now. One of its weak points is that it cannot evaluate an expression that simultaneously requires both qualitative and quantitative treatment. For instance, the above binomial relation "X > Y" cannot be proved from the relations below:

X = A + 5.
Y = A + 3.

Now let us turn to qualitative reasoning in Qupras. We have already stated that there are two types of qualitative reasoning in this system, propagation and prediction. Propagation reasoning is used to find the active objects that satisfy the given facts and known relations, and the systems of physics holding among the active objects at one time or during a time interval. Qupras performs this reasoning as follows:

(1) Try to find inactive objects whose conditions are satisfied by the given facts and the known relations using the expression evaluator.
(2) If such inactive objects are found, change them to active objects and register their relations as known relations.
(3) Next, try to find the inactive systems of physics whose necessary objects are active and whose conditions are satisfied by the given facts and known relations.
(4) If such inactive systems of physics are found, change them to active and register their relations as known relations.
(5) If any remaining inactive object or system of physics was activated in the last sequence through steps (1) to (4), repeat steps (1) to (4). If not, terminate.

Prediction reasoning is used to determine the physical variables changing with time from known relations which are the result of the propagation reasoning. Then the new values or the new intervals of the changing variables at the next specified time or during the next time interval are sought. Qupras updates the given facts according to the sought values or intervals. The updated facts are used as the initial facts at the next initiation of propagation reasoning. We describe the procedure of this reasoning briefly below.

(1) Find the physical parameters changing with time. They are the arguments of the ddt operator in the known relations. The dependent physical parameters are specified by the "coincidence" operator.
(2) Next, try to find the values or the range of values to which the parameters change. They are as follows:
 (a) the values required for currently inactive objects and inactive systems of physics to become active.
 (b) the values required for currently active objects and active systems of physics to become inactive.
(3) Select the nearest value to the current value from the values found in each group.
(4) Perform steps (5) and (6), for the nearest value of each group.
(5) Change the value of the changing physical variable to the nearest value and remove the facts contradicting the nearest value from the given facts.
(6) Perform the next propagation reasoning using the updated facts.

Qupras does not use quantity spaces as in QPT (qualitative process theory) [Forbus 84]. Qupras finds the information corresponding to quantity spaces in step (2). If there are several groups which have the changing variable in step (4), it is an ambiguity in Qupras which generates several next states. But Qupras may be able to decrease ambiguity by using quantitative information of physical parameters.

3. Example

We discuss an examples, it is a boiler. Consider the simple boiler shown Figure 4. Qupras infers the existing objects in the

```
+           +
|           |
|           |
|-----------|
|  Water    |    T
|           |
+-----------+


\/\/\/\/\/\/\/    Ts
  \/\/\/\/\/
    \/\/

  heat_source
```

Figure 4. Boiler

boiler and the systems of physics holding among the objects, and predicts the next states of the boiler. There are four object descriptions for the boiler, container, water and heat source required to specify the boiler system. The container description in Figure 5 is given as another example of an object description.

```
object container:Container
   attributes
      melting_point ;
      capacity ;
      temperature ;
      mass ;
      content ;
   conditions
      temperature@Container < melting_point@Container ;
  end.
```

Figure 5. Container description

The description has a condition specifying the condition of existence of the container. It shows that if the temperature of the container is equal to or greater than the melting point of the container, the container is broken. A physics for the boiler system was shown in Figure 3. There are seven systems of physics for the boiler, as follows:

(1) heat flow
   Shown in Figure 3.
(2) boiling
   Boiling the liquid in the container.
(3) melting
   Melting of the container.
(4) change of temperature of a container with liquid
   This system of physics covers temperature change of the container and liquid due to heat flow.
(5) change of the temperature of an empty container
   This system of physics consists of the physical law that an empty container changes when subjected to heat flow.
(6) containing
   This system of physics specifies the relations of the physical variables between a container and its contents, using the "coincidence" and the "object" predicates. The "object" predicates are used to check whether an attribute of an object is an object. Its description is shown in Figure 6.
(7) emptiness
   This is the physical law which gives the conditions for emptiness of the container.

Qupras has to be supplied with a description of an initial state to begin reasoning. An example of an initial state is shown in Figure 7. It specifies that the temperature of the container and the water is 10 degrees, the boiling point of the water is less than the melting point of the container, the melting point of the container is less than the temperature of the heat source, and so

```
physics containing
  objects
     Container = container ;
  conditions
     object(content@Container) ;
     volume@(content@Container) > 0 ;
  relations
     volume@(content@Container) = volume@Container ;
     coincidence(temperature@Container,
                           temperature@(content@Container)) ;
     coincidence(ddt(heat@(content@Container)),
                           ddt(heat@Container)) ;
  end.
```

Figure 6. Description of the physics "containing"

```
initial_state initial
  objects
     Boiler = boiler ;
     Water = water ;
  facts
     mass@container!Boiler > 0 ;
     te.perature@container!Boiler = 10 ;
     temperature@Water = 10 ;
     boiling_point@Water = 100 ;
     boiling_point@Water < melting_point@container!Boiler;
     volume@Water > 0 ;
     content@container!Boiler = Water ;
     melting_point@container!Boiler <
                         temperature@heat_source!Boiler ;
  end.
```

Figure 7. Initial state description

on. The facts and relations of the initial state can be represented qualitatively and quantitatively.

Using the description of the objects, physics, and the initial state, Qupras first finds the active objects and their relations to the boiler system at the initial time. The initial state and the transition from it deduced by Qupras are shown Figure 8. There are three active objects (circled) and the three systems of physics (in broken line squares) in the initial state. When the initial state is known, it is possible to find physical parameters that vary over time. The temperature of the water and the heat of the water are the changing physical parameters in the initial state. It is impossible to find the next value of the heat of the water, because nothing happens even if the heat is changed. So Qupras cannot predict the next state according to the heat of the water. On the other hand, it is possible to find the next value of the temperature of the water, because the physics of "change of temperature of a container with liquid" will be activated. The predicted value is greater than 10 degrees and less than its the boiling point. Therefore, Qupras can predict the next state of the boiler system according to the predicted temperature. Qupras reasons the next states based on the current state. Finally Qupras can reason the transition of states of the boiler system as shown in Figure 9 from the initial state given in Figure 7.

4. Discussion

We have discussed Qupras which reasons relations among objects in the physical world and the next states qualitatively and quantitatively using knowledge of objects and systems of physics. Qupras is capable of primitive level understanding. In this section, we discuss related issues and future research on Qupras.

4.1 Related research

Qupras uses qualitative reasoning, propagation and prediction reasoning, corresponding to the qualitative reasoning mechanism developed by [Kuipers 84, de Kleer 84]. But there are several differences between Qupras and their systems. They

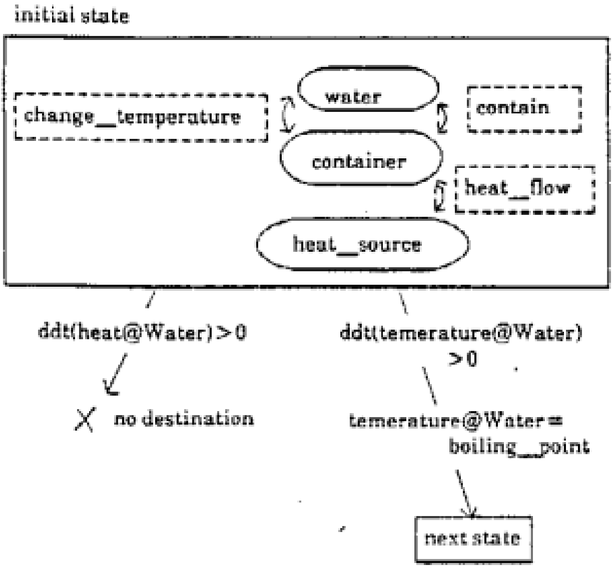

Figure 8 Initial state of the boiler

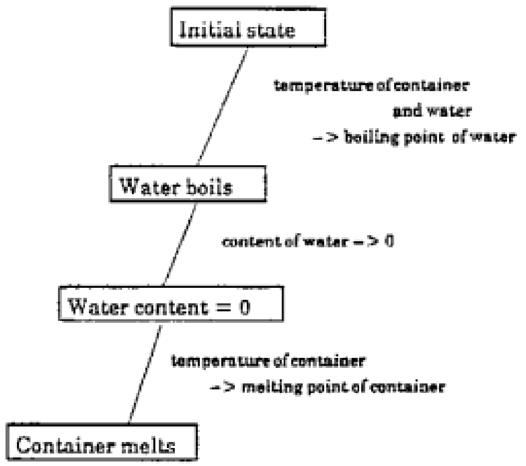

Figure 9 Boiler transitions

take the behavior of differential equations into consideration, while Qupras deduces the behavior of the given physical system from its initial state. Both systems are given the set of the differential equations of the total system from the start, but Qupras is not given them and finds them during the propagation reasonings. Neither system can deal with changes in differential equations, because they are given initially. The evaporation of water in a boiler is one example of this. Moreover, Qupras deals with physical variables quantitatively as well as qualitatively and thereby makes it possible to decrease ambiguity, while the previous two systems only deal with physical variables qualitatively.

The syntax of Qupras is similar to that of QPT with some differences. QPT is based on processes to follow changes in the physical world, but Qupras is based on physics to focus attention on relations among objects. So QPT does not deduce individual views from using knowledge from other individual views, because there is no propagation reasoning in QPT. Qupras can deduce relations among objects from applying physical laws repeatedly. For example, Qupras can reason on the physics of temperature change from the physics of heat flow. Qupras does not need to use quantity spaces as QPT does. The reason for this is that we cannot always determine the quantity spaces when trying to solve a problem.

4.2 Subsequent research

At present, Qupras is not powerful enough to solve many problems in the ph; ;ical world. We have to enhance Qupras before trying problems. There are many issues which we did not implement in the current version of Qupras. We are now studying the following important issues:

(1) Inheritance

Knowledge representation in Qupras does not have an inheritance mechanism. In general, an inheritance mechanism is required to represent knowledge effectively. We think that it is possible to describe more abstract representations by introducing an inheritance mechanism.

(2) Generic expressions

Some physical laws are represented by generic expressions, for example Kirchoff's voltage law is often represented as:

$$V = V1 + V2 + ... + Vn$$

This expression is a general representation because the number n is undefined. The number n is different in each problem and is determined by the number of connected devices. So we must introduce a mechanism which determines the number n for any given problem to allow generic expressions.

(3) Quantitative operations in prediction reasoning

We do not deal with changes in time in prediction reasoning quantitatively, even if a temporal derivative and the amount of its change are quantitatively known. Dealing with time changes quantitatively would decrease ambiguity resulting from qualitative treatment.

We plan to introduce a theorem prover for mathematics because the Qupras expression evaluator is not powerful enough. We also plan to introduce manipulation of numerical expressions as in [Apte 86], in order to manipulate physical laws more flexibly.

Acknowledgment

This research is based on discussions in the ICOT working group FAI (Foundations of Artificial Intelligence). We wish express our thanks to its head, Prof. Mizoguchi (Science Univ. of Tokyo), and the members of FAI. We would also like to thank Toshihiko Miyazaki at ICOT, who helped us improve Qupras. Finally we must express our thanks to Kazuhiro Fuchi, Director of ICOT Research Center, who provided us with the opportunity of doing this research in the Fifth Generation Computer Systems Project.

References

[Bowen 82] Bowen, D.L., Pereira,L.M., Pereira,F.C.N. and Warren,D.H.D.: User's Guide to DECsystem-10 Prolog. Dept of Artificial Intelligence, University of Edinburgh (1982).

[de Kleer 84] de Kleer,J. and Brown,J.S. : Qualitative Physics Based on Confluence, Artificial Intelligence 24, 7-83, 1984.

[Forbus 84] Forbus, K.D. : Qualitative Process Theory , Artificial Intelligence 24, 85-168, 1984.

[Kuipers 84] Kuipers, B. : Commonsense Reasoning about Causality:Deriving Behavior from Structure, Artificial Intelligence 24, 169-203, 1984.

[Apte 86] Apte, Chidanand and Hong, Se June : Using Qualitative Reasoning to Understand Financial Arithmetic, AAAI-86, 942-948, 1986.