Chapter 3

On Analogical Reasoning

3.1 Introductory remarks

The knowledge acquisition bottleneck in expert system development arises from the human nature that a person is unaware of the existence of her/his own knowledge. It is effective in overcoming this to support human creativity so that knowledge engineers or experts can consider the whole domain and can make the initial knowledge base *sparse*[13]. Knowledge acquisition techniques according to top-down methodology would work more effectively if this was done.

At Toshiba, we are studying on the difficulty of acquiring user needs in the context of software development. According to Toshiba's standpoint, it is effective to support human creativity for acquiring user needs, thus the author is developing an creativity support system called *Chie-no-Izumi*[1]. This is because this difficulty also arises from the users' lack of awareness of the existence of their own knowledge.

The features of *Chie-no-Izumi* are as follows[61].

- To model a human cognitive process of getting an idea
- To implement the model
- To support human creativity based on the model

The author is mainly studying the first and second items[60]. A model of human cognitive process of getting an idea is formally defined in this chapter. In order to do so, a general process of creative thinking[2] is defined

[1] Japanese term meaning "the fountain of wisdom".

[2] The author could not find any single word to describe the human cognitive process of getting ideas in English, in contrast to do that in Japanese, which is done by *hassou*.

and next analogy-based creative thinking is defined as special case of it. Then, a definition of *paraphrasing-based analogical reasoning*(**PAR**) is given as an algorithm to implement a sort of analogy-based creative thinking, and it is shown that **PAR** is a valid form of analogical reasoning in some sense. At the end of this chapter, some extensions to **PAR** are made, which are important from a practical view and their effects are shown with examples.

3.2 Creative thinking as machine learning problem

What is creative thinking? If we had no perspective on this (somewhat philosophical) question, we cannot discuss whether some algorithms implement creative thinking or not. Here is the definition of creative thinking.

Definition 3.2.1 *Assume a consistent theory ι over an appropriate language L. When γ which is a subset of ι and information*[3] *on ι are given, to infer some sentences belong to the difference set between ι and γ is called* creative thinking.

This definition assumes that there is an ideal knowledge base ι, and that the knowledge base γ which a human or a computer has is a subset of it, namely, γ has no knowledge which does not belong to ι. Intuitively, creative thinking is an action that is made by a human or a computer to make their own γ similar to ι.

This is a very general definition. We can characterize creative thinking with relations between ι and γ or with oracles. The following example is one aspect of creative thinking.

Example 3.2.2 *Suppose that a consistent theory ι consists of Horn clauses over the first order language L with finite function symbols and predicate symbols. Assume an oracle which tells the truth values of arbitrary ground atoms with respect to a model of ι. Obtain ι.*

Although there are words for what is got by it (*inspiration, discovery, idea,...*) or ability to do it (*creativity*), the process itself is not given a word. It is merely regarded as a special case of thinking. Thus in the thesis the author uses *creative thinking* to describe the process. Hereafter the phrase will be also used to express an English concept corresponding to *hassou* for references originally written in Japanese. In the previous paper[63], *abduction* was used instead, which might cause confusion in the thesis where *abduction* will be used to describe a particular mathematical form of logical reasoning.

[3] This is often called an *oracle*.

This is the very problem of model inference[70]. We can regard this as creative thinking according to definition 3.2.1. Indeed, a study in creativity engineering[29] shows the discovery of the periodic law by Mendeleev as an example of such creative thinking[4]. In this case, there are functions such as atomic weight etc., predicates such as many chemical properties, ordering relations on numbers etc. and atoms as constants. Mendeleev could have obtained the truth values of chemical properties on atoms by experiments. Here, these played a part of oracles. And Mendeleev got an idea(ι) that chemical properties appear in the order of atomic weight.

A study in creativity engineering gives *analogical reasoning*, *universalization* and *reasoning to limit* as well as *systemization* as the four patterns of creative thinking. Among them, analogical reasoning is regarded as the most important one. According to the scheme of definition 3.2.1, analogy-based creative thinking is defined as follows.

Definition 3.2.3 *Suppose a consistent set ι of sentences over an appropriate language L, where $|\iota| > 2$. For ι, let $D_1, \ldots, D_n (n \geq 2)$ be non-empty sets such as $\iota = D_1 \cup \ldots \cup D_n$ and $D_i \cap D_j = \phi (i \neq j)$. $\{D_1, \ldots, D_n\}$ is called a* partition[5]. *Elements of a partition are called a* domain. *For some $i \neq j$, if s_i such that belongs to a deductive closure of D_i and s_j such that belongs to a deductive closure of D_j are identical upon a theory Th, we say D_i and D_j are analogous with respect to s_i,s_j upon Th, or D_i and D_j are analogous by the evidence s_i,s_j on Th.*

Suppose that L is the whole of the Horn sentences and that D_i and D_j are analogous with respect to $s_{1i}, \ldots, s_{mi}$ and $s_{1j}, \ldots, s_{mj}$ upon Th. Let D'_j be the subset of D_j such that $s_{1j}, \ldots, s_{m-1j}$ can be deduced, but s_{mj} cannot be deduced from it. When $D_i \cup D'_j$ are given, to obtain a set of sentences S, included by D_j and not included by D'_j, such that s_{mj} can be deduced from $D'_j \cup S$ and S is identical to a set of sentences in D_i upon Th, is called analogy-based creative thinking.

[4] This is called *"systemization-based" creative thinking.*

[5] The concept was called *division* in the previous paper[63]. The change has been made in accordance with a suggestion by Professor Bipin Indurkhya.

We can characterize analogy-based creative thinking by the form of S[6] or relations between D_j and D'_j. Let us consider the following example.

$$\begin{array}{rcrcll}
\iota & = & D_1 & \cup & D_2 & \\
\gamma & = & D_1 & \cup & D'_2 & \\
D_1 & = & \{f(a',d') & & & (1) \\
 & & g(a',c') & & & (2) \\
 & & r(c',d') & \leftarrow & p(d',c') & (3)\} \\
D'_2 & = & \{g(a,c) & & & (4) \\
 & & f(a,d) & & & (5) \\
 & & p(d,c) & & & (6) \\
 & & p(d,c) & \leftarrow & f(a,d), g(a,c) & (7)\} \\
D_2 & = & D'_2 & \cup & \{r(c,d) \leftarrow p(d,c) & (8)\}
\end{array}$$

In this example, $r(c,d)$ can be deduced from ι but not from γ. Since we can see identity between (1) and (5), and between (2) and (4) upon theory Th_1 such that $(a' \approx a) \wedge (d' \approx d) \wedge (c' \approx c)$ upon it where $\approx$ represents the identity relation, D_1 and D'_2 are analogous. Here, the sentence which is identical to (3) upon Th_1 is (8). If we infer (8) from γ, we can deduce $r(c,d)$ from it.

The reason why $r(c,d)$ cannot be deduced from γ is the lack of implication(8) which belongs to ι. Haraguchi[21] gave a formal definition to these cases of analogical reasoning. We can also regard them as creative thinking according to definition 3.2.3.

Now we introduce another case of analogy-based creative thinking.

Example 3.2.4 *Assume a set ι of sentences with the form* $\forall x_1, \ldots, x_m(p(x_1, \ldots, x_m) \Leftrightarrow q_1(x_1, \ldots, x_m) \wedge \ldots \wedge q_n(x_1, \ldots, x_m))$. *We call sentences with this form* paraphrasings *on p. Assume that ι can be partitioned into domains D_i and D_j, where D_i and D_j are analogous with respect to paraphrasing on p_i and p_j and paraphrasing on q_i and q_j upon the same theory Th, where paraphrasing on q_i has the symbol p_i and paraphrasing on q_j has the symbol p_j. When $\gamma = D_i \cup D'_j$ (where $D'_j = D_j - \{$paraphrasing on $q_j\}$) is given, obtain ι.*

A study in creativity engineering gives an episode of the discovery of natural selection by Darwin as an example of such creative thinking. Namely,

[6] To specify the form of S corresponds to *bias* in inductive inference[22].

suppose γ such as

$$
\begin{array}{rcll}
\iota & = & D_i \cup D_j & \\
\gamma & = & D_i \cup D'_j & \\
D_i & = & \{improve\_breed(z) \Leftrightarrow & \\
 & & \quad artif\_variant(z) \wedge survive(z). & (1) \\
 & & artif\_variant(x) \Leftrightarrow & \\
 & & \quad variant(source(x), x) \wedge make(man, x). & (2)\} \\
D_j & = & \{natural\_selection(z) \Leftrightarrow & \\
 & & \quad natural\_variant(z) \wedge survive(z). & (3)\} \cup D'_j \\
D'_j & = & \{natural\_variant(x) \Leftrightarrow & \\
 & & \quad variant(source(x), x) \wedge make(nature, x). & (4)\}
\end{array}
$$

are given. Since (2) and (4) are identical upon the theory such that $artif\_variant$ and $natural\_variant$ are identical and *man* and *nature* are identical upon it (we denote this theory as Th_2), we can say D_i and D'_j are analogous with respect to (2),(4) upon Th_2. In this example, we want to infer (3) from $D_i \cup D'_j$ by transforming (1) using theory Th_2. While this case is similar to the last one in that γ has information on partitioning of ι and has evidence that the two domains are analogous, this case is different from the last one because γ does not have all the predicates in ι. This seems to be a special case because many machine learning studies assume that all predicates needed to form an objective theory are known[7]. In this case, the learning system may infer what sort of predicate the missing one is, but cannot name it—it is done only by an oracle.

The author thinks that a model of human creative thinking is a combination of various models. We cannot say that we should *always* infer implications or paraphrasings by analogy between domains. But it is considered important to formulate individual models because they can be regarded as some aspects of human creative thinking.

Observing brainstormings[8], the author is led to believe that the pair of an eye-catcher and a catch phrase[9] invite creative thinking by the brainstormers. This is well explained by the hypothesis that the evidence of analogy between domains of ι is the pair of an eye-catcher and a catch phrase, namely paraphrasing. Therefore, we can model an aspect of human creative thinking in brainstorming by a framework of analogical reasoning which obtains evidence of analogy from paraphrasing. In this chapter, we formulate this as *paraphrasing-based analogical reasoning*.

[7] For example, in MIS with theoretical term, all predicates can be divided into an experimentable one or an a priori known one[70].

[8] A method to generate ideas within a group.

[9] A short explanation for the eye-catcher.

3.3 Paraphrasing-based analogical reasoning (PAR)

3.3.1 Definition of PAR

In **PAR**, an object is a theory consisting of paraphrasings, such as a subset of the Horn theory. First we will define some fundamental terms. Some of them follow AST[25]. The author will also describe the relationship between **PAR** and AST.

Definition 3.3.1 *Constant symbols, predicate symbols and function symbols are all called* tokens.

Definition 3.3.2 (Vocabulary[25]) *let* Γ *be a countable set of* types $\{f_0, f_1, \ldots, p_0, p_1, \ldots\}$*. Here,* f_n *shows a function with n arguments, while* p_n *shows a predicate with n arguments. For a set of tokens* V *and mapping* $\zeta : V \to \Gamma$*, we call a pair* $[V, \zeta]$ vocabulary.

Definition 3.3.3 *For token* t *with type* $p_n (n \geq 0)$ *and atoms* $L_1, \ldots, L_m$*, a set of Horn clauses with the following form is called a* paraphrasing of t.

$$\begin{array}{rcl} \{t(x_1, \ldots, x_n) & \leftarrow & L_1(x_1, \ldots, x_n), \ldots, L_m(x_1, \ldots, x_n). \\ L_1(x_1, \ldots, x_n) & \leftarrow & t(x_1, \ldots, x_n). \\ \cdots\cdots\cdots & \cdots & \cdots\cdots\cdots \\ L_m(x_1, \ldots, x_n) & \leftarrow & t(x_1, \ldots, x_n).\} \end{array}$$

where no variable is in $L_1(x_1, \ldots, x_n), \ldots, L_m(x_1, \ldots, x_n)$ *other than* $x_1, \ldots, x_n$ *and* t *does not occur in* $L_1(x_1, \ldots, x_n), \ldots, L_m(x_1, \ldots, x_n)$.

Here, we call $\{L_1(x_1, \ldots, x_n), \ldots, L_m(x_1, \ldots, x_n)\}$ *the* body *of the paraphrasing. We use the following notation for this paraphrasing.*

$$t(x_1, \ldots, x_n) \Leftrightarrow L_1, \ldots, L_m.$$

Paraphrasing is a special case of derivation *in AST such that it has restrictions on the right hand side of the equivalence symbol. We denote a set of tokens included by* S *as* $T(S)$*, for a body of paraphrasing or a set of paraphrasings* S.

Definition 3.3.4 *Assume that a finite set* W *of paraphrasings and a partition* $C = \{D_1, \ldots, D_n\}$ *of it are given, where* $|W| \geq 2$*. We denote* $\bigcup_{u \in W} T(\{u\})$ *as* T_w*. We make a distinction between two subsets of* T_w*, namely* T_{we} *and* T_{wi}*, where* $T_{we} = \{e | e \in T_w \wedge \exists i, j (i \neq j \wedge e \in T(D_i) \wedge e \in T(D_j))\}$, $T_{wi} = T_w - T_{we}$.

Elements of T_{we} *are called* external tokens *of* C. *Elements of* T_{wi} *are called* internal tokens *of* C [10]. *We denote* $\{t | t \in T_{wi} \wedge t \in T(D_i)\}$ *as* I_i.

In the remaining part of this chapter, we assume that all sets of paraphrasings are finite and have more than two elements, and that for a partition $C = \{D_1, \ldots, D_n\}$, there is at most one i such that $I_i = \phi$.

Definition 3.3.5 *For vocabulary* $[V, \zeta]$, *let us consider the following conditions among an equivalence relation* R *over* V *and a partition* $C = \{D_1, \ldots, D_m\}$ *of set* W *of paraphrasing over* $[V, \zeta]$.

$$\begin{aligned} \forall x, y(R(x, y) \quad &\supset \quad \zeta(x) = \zeta(y)) \\ \forall x, y(R(x, y) \quad &\supset \quad (x = y \wedge (x \in T_{we} \vee \exists i(x \in I_i)) \vee \\ &\qquad (x \neq y \wedge \exists i, j(x \in I_i \wedge y \in I_j \wedge i \neq j))) \end{aligned}$$

If and only if these conditions hold true, R *is called a* correspondence of tokens *on* C *over* $[V, \zeta]$.

Definition 3.3.6 *For a partition* $C = \{D_1, \ldots, D_m\}$ *of set* W *of paraphrasings over* $[V, \zeta]$, *we denote an equivalence class of token* t *according to the correspondence of tokens* R *as* $T_c R(t)$.

Theorem 3.3.7 *For* $t \in T_w$, *there exists at most one* u *such that* $u \in T_c R(t) \cap I_i (i = 1, \ldots, m)$, *for all* i.

Proof By definition 3.3.6. ■

We can regard the correspondence of tokens as the extension of *admissible mapping*[25][11] so that it can deal with three or more domains. This is because all the external tokens are equivalent only to themselves according to definition 3.3.5. The equivalence classes of tokens give some sets of tokens that are identical among domains upon a given correspondence of tokens over a given vocabulary. The function which picks out a token belonging to a given domain from an equivalence class is regarded as an extension of T-MAP. Now we define such a function as the *mapping of tokens*.

Definition 3.3.8 *For a partition* $C = \{D_1, \ldots, D_m\}$ *of set* W *over vocabulary* $[V, \zeta]$ *and an equivalence class* $T_c R(t)$ *of* $t \in T_w$ *upon the correspondence of tokens* R *over* $[T_w, \zeta]$, *the* mapping of tokens $T_c R_j$ *to* D_j *upon* R *is a partial function from* $\bigcup_{i=1,\ldots,m} I_i \cup T_{we}$ *to* $I_j \cup T_{we}$ *that is only defined for* t *such as* $\exists u(u \in T_c R(t) \cap (I_j \cup T_{we}))$ *and has such* u *as a value.*

[10] These definitions of internal/external tokens are somehow different from [25].

[11] Type-preserving mapping from internal tokens to internal tokens.

For a body of paraphrasing or a set of paraphrasing S, when T_cR_j is defined for all the elements in $T(S)$ we denote a set such that all the tokens in S are replaced by its image of T_cR_j as $T_cR_j(S)$.

Two correspondences of tokens R_1,R_2 are said to be consistent *when the following condition holds true.*

$$\begin{aligned}\forall j, t(t \in T_w \supset \ & (T_cR_1(t) \cap I_j = \phi) \\ & \vee (T_cR_2(t) \cap I_j = \phi) \vee (T_cR_{1j}(t) = T_cR_{2j}(t)))\end{aligned}$$

When R_1, R_2 are consistent, we denote the relation represented by the following graph as $R_1 \cup R_2$.

$$\begin{aligned}GR_1 \cup GR_2 \ & \cup\{(x,y)|(x,z) \in GR_1 \wedge (z,y) \in GR_2\} \\ & \cup\{(y,x)|(x,z) \in GR_1 \wedge (z,y) \in GR_2\}\end{aligned}$$

where GR_1, GR_2 are graphs of R_1, R_2 respectively.

Theorem 3.3.9 *For consistent correspondences of tokens R_1 and R_2, $R_1 \cup R_2$ is a correspondence of tokens.*

Proof Evident according to the last two definitions. ■

Definition 3.3.10 *For the correspondence of token R on the partition C of set W of paraphrasings over $[V, \zeta]$, $\bigcup_{j=1,\ldots,m} \{T_cR_j(\{s\})\}$ is called the equivalence class of s upon R where s is an arbitrary paraphrasing over $[V, \zeta]$.*

Definition 3.3.11 *For the partition $\{D_1, \ldots, D_n\}$ of set W of paraphrasings over $[V, \zeta]$, both of the followings are called* T-equivalence.

1. *The relation represented by the graph $\bigcup_{t \in T_w} \{(t,t)\}$.*
2. *For two different domains D_i and D_j, suppose $s_1 \in D_i$ and $s_2 \in D_j$ are paraphrasings of t_1 and t_2 whose bodies are b_1 and b_2, respectively, and $\exists x, y(y \in T_cR(x) \wedge x \in T(b_1) \wedge y \in T(b_2))$ holds true for a given T-equivalence R. If a correspondence of tokens θ such as is consistent with R can make equivalent s_1 and s_2 in the equivalence class of paraphrasings upon $R \cup \theta$, then $R \cup \theta$ is such a T-equivalence.*

We define the size of T-equivalence R with respect to D_i, D_j as $|\{(x,y)|(x,y) \in GR \wedge x \in I_i \wedge y \in I_j\}|$ where GR denotes a graph of R. We denote this as $|R|_{ij}$. Sometimes we represent T-equivalence by a graph such that symmetric and reflexive relations are eliminated.

There are the following features in the definition of T-equivalence. The second item of the definition claims that if we want to make a new T-equivalence based on a given one, we should take two paraphrasings such that contain equivalent tokens upon the given one. Thus, at first we can make T-equivalence only from two paraphrasings such that contain the same external tokens. The syntactic support for the validity of T-equivalence is its type-preserving property. Semantic supports for the validity of T-equivalence are the identity of external tokens and that only such extensions are allowed that extend the semantic supports for the validity of basic T-equivalence by making equivalent the two paraphrasings that are partially identical upon the basic T-equivalence.

Before defining **PAR**, we assume an oracle. Suppose a set ι of paraphrasings over a given $[V, \zeta]$ which includes a given set W of the paraphrasings as its proper subset. We call ι an *intended theory*. Here, the oracle accepts a set Y of atoms consisting of tokens $t \in T_w$ and variables, and returns a token of type p_n in $V - T_w$ whose paraphrasing in $\iota - W$, if it exists. It is called a *godparent*.

Definition 3.3.12 *We assume that the following formula holds true, for a set W of paraphrasings over $[V, \zeta]$,its partition C and given T-equivalence on C.*

$$\exists i, j, b, t, t' (b \in D_i \wedge t \in T(b) \wedge t' = T_c R_j(t))$$

Here, a paraphrasing-based analogical reasoning *from b with the base domain D_i and the target domain D_j are defined by a procedure* reason(b, i, j, R).

Algorithm 3.1 reason$(b, i, j,$ **var** $R)$

Suppose $b = (t(x_1, \ldots, x_n) \Leftrightarrow PARA)$
$PARA' \leftarrow$ inner$(PARA, i, j, R)$
if $(t'(x_1, \ldots, x_n) \Leftrightarrow PARA') \notin D_j$ **then**
    $t' \leftarrow$ godparent$(PARA')$
end
$\theta \leftarrow$ relation represented by graph $\{(t, t'), (t', t)\}$
if R and θ are consistent **then**
    $R \leftarrow R \cup \theta$
    return $t'(x_1, \ldots, x_n) \Leftrightarrow PARA'$
else
    return `FAIL`
end

Algorithm 3.2 inner($PARA, i, j$, **var** R)

```
if ∃t ∈ T(PARA) ∧ T_cR_j(t) is not defined then
    k ← 0
    let L as a list of all tokens with same type as t, and L's element ∈ T(D_j)
    if such a token does not exist then
        return FAIL
    end
    repeat
        cand ← FAIL
        k ← k + 1
        if length of L < k then
            return FAIL
        end
        let t' as k-th element of L
        θ ← relation represented by graph {(t, t'), (t', t)}
        if R and θ are consistent then
            R ← R ∪ θ
            cand ← inner(PARA, i, j, R)
        end
    until cand ≠FAIL
else
    return T_cR_j(PARA)
end
```

The procedure godparent shows an oracle introduced above.

Theorem 3.3.13 *For a set W of paraphrasings, its partition $C = \{D_1, \ldots, D_n\}$ and given T-equivalence R on C, if a* **PAR** *from b with the base domain D_i and the target domain D_j infers $b' \neq$ FAIL and renew R into R', then $T_cR'_j(\{b\}) = \{b'\}$ where $\{D_1, \ldots, D_{j-1}, D_j \cup \{b'\}, D_{j+1}, \ldots, D_n\}$ is a partition of $W \cup \{b'\}$.*

Proof According to the definition of inner, it is evident that $PARA' = T_cR'_j(PARA)$ holds where $PARA$ is a body of b and $PARA'$ is a body of b'. Due to the definition of reason we can see that $T_cR'_j(t) = t'$ where tokens t and t' are paraphrased by b and b', respectively. Therefore, we can say $T_cR'_j(\{b\}) = \{b'\}$. ∎

Theorem 3.3.13 shows that **PAR** is valid with respect to given T-equivalence because it extends the size of analogy represented by T-equivalence between the domains. Here, we regard $|R|_{ij}$ as the size of analogy between D_i and D_j upon T-equivalence R. Notice that we use the concept *consistency of* *correspondence* in order to guarantee the conservativeness of the new analogy against basic analogy, instead of using *logical consistency between the new sentence and the target domain*, which is often used by other analogical reasoning systems. The former is decidable, but the latter is not. A set of paraphrasing is always consistent[12]. And we can show that we need not to consider the consistency of object theories in order to discuss the validity of **PAR** using the following theorem. This claims that a set obtained from a set of paraphrasings by **PAR** has a model such as a conservatively extended one of the old set.

Theorem 3.3.14 *For s set W of paraphrasings over $[V, \zeta]$, its partition and given T-equivalence R and model $\mathcal{M}$[13] of W, when* **PAR** *from b with the base domain D_i and the target domain D_j infers a paraphrasing b' of token t', $\mathcal{M} \cup I'_t$ is a model of $W \cup b'$, where I'_t is an appropriate interpretation of t'.*

Proof If b' belongs to W then $W = W \cup b'$. In this case, we can pick an interpretation of $t' \in \mathcal{M}$, so that $\mathcal{M} \cup I'_t$ is a model of $W \cup b'$. This is because $\mathcal{M} = \mathcal{M} \cup I'_t$ and $\mathcal{M}$ is a model of W. If b' does not belong to W, we can make I'_t as assignments of truth values for instantiations of conjunction of the body of b' over the arguments of t'. So we have I'_t such that makes the theorem hold. ■

3.3.2 Related works

Now, *augmentation* with respect to predicates on AST will be introduced. Augmentation with respect to predicates is defined as follows. If f, which is consisting T-MAP[14], is defined for all the tokens in Y of derivation $\forall x_1, \ldots, x_n(i_1(x_1, \ldots, x_n) \Leftrightarrow Y)$ and not defined for i_1, then add i_2, corresponding to i_1, and a derivation $\forall x_1, \ldots, x_n(i_2(x_1, \ldots, x_n) \Leftrightarrow f(Y))$ into the target domain.

If we restrict ourselves to consider only the conjunctions of atoms as Y, this is similar to **PAR** with two domains. But there are some differences.

1. The semantics of external tokens. In AST, external tokens are defined as such tokens that are internal tokens of other domains but occur in

[12] Maybe some predicates have no extension.

[13] We assume that a model of the theory is represented by a set of interpretations of tokens.

[14] A pair of admissible mapping f and a set of sentences in the base domain such that all internal tokens have an image of f.

the domain. They are used to make relationships between domains. In **PAR**, external tokens are external for any domain. This is related to a practical extension of **PAR** which is mentioned in the next section.

2. The difference between T-MAP and T-equivalence. These two concepts are similar like admissible mapping and the correspondence of tokens. While T-MAP requires the coherency[15] in order to keep itself valid, it need not consider consistency in **PAR**, as we can see in the last part of the last section. The existence of paraphrasings syntactically corresponding to each other in each domain guarantees the validity of **PAR**.

3.4 Discussions

When **PAR** can infer all the paraphrasings belong to $\iota - W$ from W such that it is a subset of the intended theory ι, we can say **PAR** is valid as a model of creative thinking. To realize this, we should make a constraint to W and ι. In this section, some necessary conditions for it are considered.

3.4.1 Input to the system

Evidently, all the paraphrasings belonging to $\iota - W$ should belong to some equivalence classes on a partition C of ι which has more than one element, and at least one element of those should belong to W. Moreover, a given partition $C' = \{D'_1, \ldots, D'_n\}$ of W should satisfy the condition $\forall i \exists j (D_j \supseteq D'_i)$ against $C = \{D_1, \ldots, D_n\}$. This condition claims that C' is a partition obtained by a natural restriction of C, so to speak, we can get a partition on ι with the same form. This seems to be too optimistic. Now, we show that to specify a subset of T_w such as external tokens in ι is equivalent to satisfying the above requirement.

Let us see the definition of domain and internal tokens from another viewpoint. Intuitively, internal tokens are proper to the domain, while external tokens are used among domains. Therefore, we can regard a domain as a classified theory on internal tokens. We can obviously construct a partition $C = \{D_1, \ldots, D_n\}$ when we are given a set W of paraphrasings and $I_1, \ldots, I_n$[16].

Next, let us relax the restriction. When we are given W and a set T_{we} of external tokens of unknown partition $C = \{D_1, \ldots, D_n\}$, how about recon-

[15] *Logical* consistency between $f(S)$ and the target domain.

[16] Because there exists at most one i such that $I_i = \phi$.

structing the partition? Suppose that $T(D_i)$ and $T(D_{i+1})$(where $1 \leq i < n$) has no token belonging to T_{we}. A set of external tokens of C and a partition $C' = \{D_1, \ldots, D_{i-1}, D_i \cup D_{i+1}, D_{i+2}, \ldots, D_n\}$ are the same. So, we cannot decide which is the original partition, C or C'.

Definition 3.4.1 *When a partition $C = \{D_1, \ldots, D_m\}$ of W whose external token set is ε satisfies the following, we call C a* canonical partition *with respect to W and ε.*

$$\begin{aligned}\forall n(\quad & (D_i \in C) \supset \\ & \forall I((I \subsetneq I_i) \supset \quad \{s|s \in D_i \wedge T_i(s) \subseteq I\} \neq \\ & \qquad\qquad\qquad\quad \{s|s \in D_i \wedge \exists t(t \in T_i(s) \wedge t \in I)\}))\end{aligned}$$

where $T_i(s)$ denotes $\{t|t \in T(s) \wedge t \notin \varepsilon\}$.

A canonical partition is the finest partition among a family of partitions with the same external token set, such that all domains in it cannot be partitioned any more.

Theorem 3.4.2 *A canonical partition with respect to W and ε is unique, if it exists.*

Proof Suppose that $C = \{D_1, \ldots, D_n\}, C' = \{D'_1, \ldots, D'_m\}$ are canonical partitions with respect to W and ε. According to the definition of internal tokens, the following holds for $1 \leq i \leq n$.

$$D_i = \{r|r \in W \wedge T_i(r) \subseteq I_i\} = \{r|r \in W \wedge \exists t(t \in T_i(r) \wedge t \in I_i)\} \qquad (1)$$

Let s be an arbitrary paraphrasing in D_i and assume $s \in D'_j$. According to the definition of domains, j is unique. The following holds true.

$$D'_j = \{r|r \in W \wedge T_i(r) \subseteq I'_j\} = \{r|r \in W \wedge \exists t(t \in T_i(r) \wedge t \in I'_j)\} \qquad (2)$$

Let us consider the following procedure check(k) where $Y_1 = \{s\}$ and $k = 1$.

Algorithm 3.3 check(k)

$Z \leftarrow \bigcup_{r \in Y_k} T_i(r)$
if $Z = I'_j$ **then**
  return k
end
$Y_{k+1} \leftarrow \{r|r \in D'_j \wedge \exists t(t \in T_i(r) \wedge t \in Z)\}$
return check(k+1)

We can say from (1) and (2) that $Z \subseteq I'_j \wedge Z \subseteq I_i$ is always true during the execution of check(k), and therefore $Y_k \subseteq D'_j \wedge Y_k \subseteq D_i$ for every k. Since C' is a canonical partition, an assignment statement to Y_{k+1} in the procedure always gives a Y_{k+1} such that $Y_k \subsetneq Y_{k+1}$. Accordingly, check(1) must halt. When the procedure halts, it is true that $Z = I'_j$. So we can say $I_i = Z$, because C is a canonical partition. Therefore, we can see that $I_i = I'_j$. From this and the definition of internal tokens, we have $D_i = D'_j$. Here, we can say $\forall i \exists j (D_i = D'_j)$, because i is arbitrary. Using a symmetrical argument, we can show $\forall i \exists j (D'_i = D_j)$. So we have $C = C'$. ■

Theorem 3.4.2 claims that we can construct a partition C of W from the given W, and the external token set ε if C is a canonical partition with respect to W and ε. This is done by the following procedure div(W, ε, ϕ).

Algorithm 3.4 div(W, ε, ϕ)

if $W = \phi$ **then**
    return D
else
    $k := |D|$
    $b :=$ one paraphrasing in W
    $i := \{t | t \in T(\{b\}) \wedge t \notin e\}$
    Suppose that $D = \{D_1, \ldots, D_n\}$
    if $\exists j (I_j \supseteq i)(1 \leq j \leq k)$ **then**
        $D_j := D_j \cup \{b\}$
    else
    If $\forall j (I_j \cap i = \phi)(1 \leq j \leq k)$ **then**
        $k := k{+}1$
        $D_k := \{b\}$
    else
        let L as a list of j such that $I_j \cap i \neq \phi$
        $D' := D - \bigcup_{j \in L} D_j$
        Renumber suffix of D' and suppose that $D' = \{D'_1, \ldots, D'_l\}$
        $k := l + 1$
        $D'_k := \bigcup_{j \in L} D_j \cup \{b\}$
        $D := D'$
    end
    return div($W - b, e, D$)
end

Theorem 3.4.3 *If a partition C is a canonical one with respect to W and*

ε, then `div`(W, ε, ϕ) *returns C.*

Proof Details are eliminated. The main points are:

- If div(W,ε,ϕ) returns a set which has two or more elements then it is a canonical partition.
- Also by theorem 3.4.2.

■

Theorem 3.4.4 *Assume that a partition $C = \{D_1, \ldots, D_n\}$ of set ι of paraphrasings is a canonical one with respect to ι and ε. For a subset W of ι and a given T_{we} such that $\{t|t \in T_w \wedge t \in \varepsilon\}$, we have $\forall i \exists j (D_j \supseteq D'_i)$ where $C' = \{D'_1, \ldots, D'_m\}$ is returned by* `div`(W, T_{we}, ϕ).

Proof Using mathematical induction on $|\iota - W|$. If $|\iota - W| = 0$, then $C = C'$ so we can show that the theorem holds because `div`(W, T_{we}, ϕ) returns C according to theorem 3.4.3. Next, we assume that when $|\iota - W| = k$ the theorem holds for $k < |\iota|$. When $|\iota - W| = k + 1$, we can consider three cases on $C'' = \{D''_1, \ldots, D''_l\}$ returned by `div`$(W \cup b, T_{we} \cup E_b, \phi)$, where $b \in \iota - W, E_b = \{t|t \in T(\{b\}) \wedge t \in \varepsilon\}$.

1. Case of $|C''| = |C'|$. We can say $\forall i \exists j (D''_j \supseteq D'_i)$ from the definition of `div` and the assumption of induction claims $\forall i \exists j (D_j \supseteq D''_i)$, and therefore the theorem holds.

2. Case of $|C''| > |C'|$. We can say $\forall i \exists j (D'_i = D''_j) \wedge \exists i (D''_i = \{b\})$ from the definition of `div` and the assumption of induction claims $\forall i \exists j (D_j \supseteq D''_i)$, and therefore the theorem holds.

3. Case of $|C''| < |C'|$. We have $\forall i (\exists j (D''_i = D'_j) \vee (D''_i = \bigcup_{k \in L} D'_k))$, where $L = \{k|(\{t|t \in T(D'_k) \wedge t \notin T_{we} \cup E_b\} \cap \{t|t \in T(\{b\}) \wedge t \notin E_b\}) \neq \phi\}$, from the definition of `div` and reason $\forall i \exists j (D''_j \supseteq D'_i)$ from this, and the assumption of induction claims $\forall i \exists j (D_j \supseteq D''_i)$, and therefore the theorem holds.

■

Theorem 3.4.4 claims that if the partition C is a canonical one of the intended theory ι, we can obtain a set C' of subsets of W naturally reduced from C in the sense mentioned at the beginning of this subsection, where given W such as a subset of ι and T_{we}. A partition is not always made, because `div` can return a set with a single element.

The result of this subsection is interesting mainly in a practical sense. Suppose that a set W of paraphrasings is given. Unless we have information on the partition of the intended theory, we may try to make a partition with an arbitrary T_{we} and perform **PAR** on it. It is possible for **PAR** to give a different result on a different partition. To give T_{we} is regarded as giving a viewpoint for partitioning. Suppose that a person has two intended theories that has W as a common subset. To partition from plural viewpoints gives **PAR** a way to reach plural intended theories from one W. The example described in the next section will illustrate this point more clearly.

3.4.2 Examples when PAR does not work well

In this subsection, some examples in which **PAR** suffers will be given. In the remainder of this subsection, the author will introduce a new notation of paraphrasing. We denote a paraphrasing by a tree. A node shows a token. A node with a label of a capital letter is an external token. The others are internal tokens. In all subtrees of tree, a root node of subtree is paraphrased by its immediate descendants. We show an example of tree representations in figure 3.1 for W as follows.

$$\begin{aligned} W \;=\; \{a(x) &\Leftrightarrow b(x), c(x). \\ b(x) &\Leftrightarrow d(x), e(x). \\ f(x) &\Leftrightarrow g(x), h(x).\} \end{aligned}$$

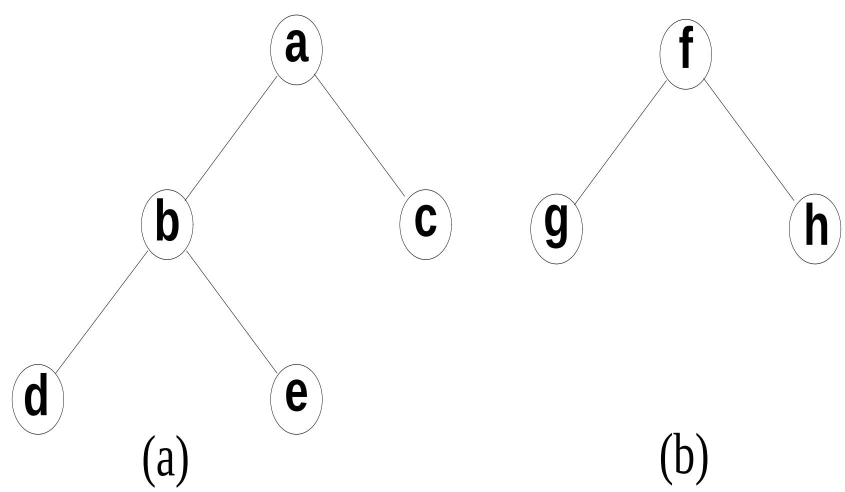

Figure 3.1: Tree representation of set of paraphrasings

Mutual recursive paraphrasing

In figure 3.2, we can obtain the part of the broken lines by **PAR** from (1) with the base domain (a) and the target domain (b), where T-equivalence $\{(a, a'), (b, b')\}$ are given. On the other hand, if we are given T-equivalence $\{(a, h'), (i, i')\}$, we can extend the T-equivalence to $\{(a, h'), (i, i'), (d, d')\}$ by **PAR** from (4). Next, T-equivalence can be extended to $\{(a, h'), (i, i'), (d, d'), (b, b')\}$ by **PAR** from (3). Finally, **PAR** from (2) is tried, but it fails because $\{(a, a')\}$ is inconsistent with the given T-equivalence.

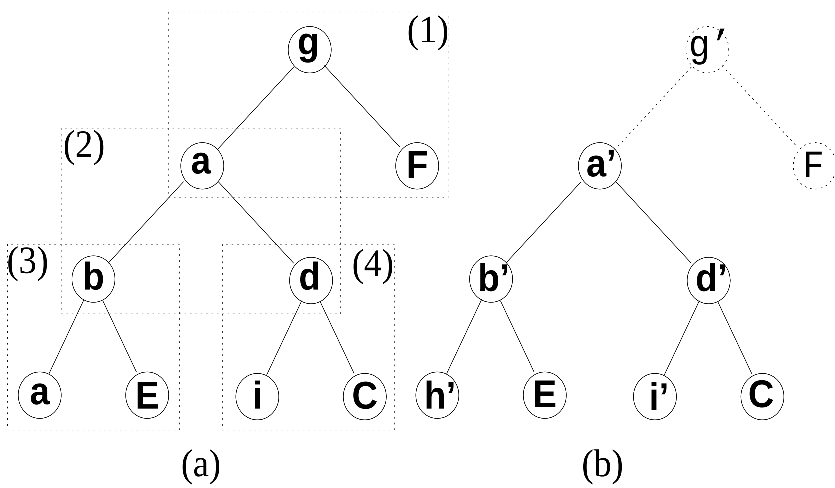

Figure 3.2: Mutual recursive paraphrasing problem on **PAR**

This is an example that two domains, which are analogous at a relatively higher level, are reduced their analogy by discovering correspondence at a lower level. T-equivalence detected at a lower level is a useless and harmful analogy (UHA) for **PAR** . In this case, UHA is detected because there are mutual recursive paraphrasings on a and b in the base domain. The size of T-equivalence is a similar concept to the *maximal analogy* by Haraguchi, but a large T-equivalence is not always useful for **PAR** because such situations can happen. We think it is related with the *systematicity principle*[16] that T-equivalence detected at a lower level comes to UHA.

Multiple paraphrasing

In figure 3.3, we can obtain the part of the broken lines by **PAR** from (1) with the base domain(a) and the target domain (b), where T-equivalence$\{(b, b'), (d, d')\}$ are given. On the other hand, if we are given T-equivalence$\{(f, f')\}$, we can extend T-equivalence to $\{(f, f'), (b, h'), (i, i')\}$ by **PAR** from (2), but this **PAR** gives nothing new to the target domain. Moreover, because of inconsistency between the correspondence(b,b') and this T-equivalence, we cannot make **PAR** from (1), which is possible for a relatively small T-equivalence$\{(b, b'), (d, d')\}$.

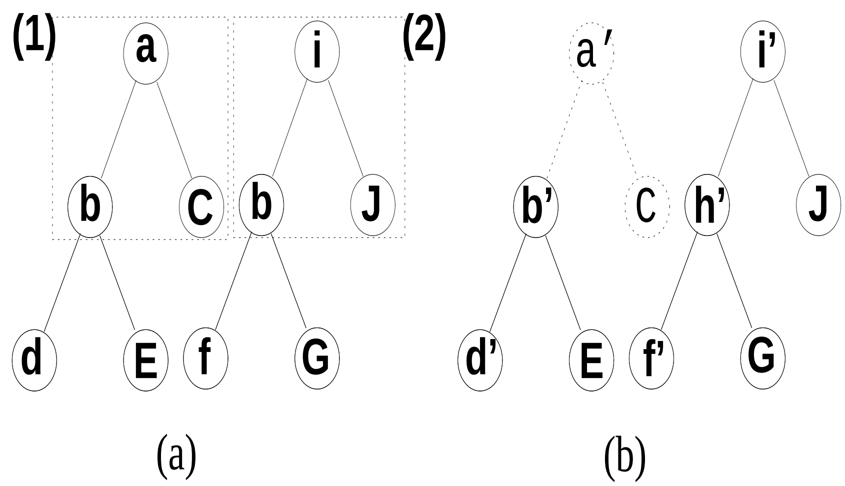

Figure 3.3: Multiple paraphrasing problem on **PAR**

In this case, UHA is found since there exist two paraphrasings of b in the base domain. And also because the structures of the paraphrasings to be identical for a large T-equivalence are too similar, there is no new knowledge given to the target domain. This is another example of useless large T-equivalence.

3.5 Examples

In this section, the author shows the behavior of **PAR** system with two examples. The **PAR** system has been implemented on J-3100SGT UX/386, with IF/Prolog and C. The **PAR** system accepts a set of paraphrasings and a set of external tokens as input, and making dialogues with an oracle(user), performs **PAR**.

3.5.1 "Meat and fish" partition

Figure 3.4 shows an input file for an example used in this subsection. An input file is a list of external tokens followed by paraphrasings.

```
[block,uncut,slice,boil,unboiled,pan,roast,unroasted,frying_pan].
    meat_block(x)  ⇔  meat(uncut(x)),block(uncut(x),x).   (1)
    meat_slice(x)  ⇔  meat(uncut(x)),slice(uncut(x),x).   (2)
    fish_block(x)  ⇔  fish(uncut(x)),block(uncut(x),x).   (3)
    fish_slice(x)  ⇔  fish(uncut(x)),slice(uncut(x),x).   (4)
 bouillabaisse(x)  ⇔  fish_block(unboiled(x)),
                      boil(unboiled(x),x,pan).            (5)
         steak(x)  ⇔  meat_slice(unroasted(x)),
                      roast(unroasted(x),x,frying_pan).   (6)
```

Figure 3.4: Input file 1

Two domains are obtained from this. One is {(1),(2),(6)}(called the *meat* domain), and another is {(3),(4),(5)}(called the *fish* domain).

The **PAR** system detects T-equivalence between the domains from (1) and (3). Although a token `fish` can correspond with the p_1-typed token other than `meat`, The **PAR** system chooses it because it assumes that (1) and (3) belong to the same equivalence class of paraphrasings. Therefore, T-equivalence {`(fish,meat)`,`(fish_block,meat_block)`} are detected. Next, the **PAR** system tries to make **PAR** from (5) with the base domain *fish* and the target domain *meat*. So, it sends the following question to an oracle.

```
I know the term <<bouillabaisse(X)>>
  in context of
bouillabaisse(X) <=> fish_block(unboiled(X)),
                     boil(unboiled(X),X,pan)
  (belongs to domain FISH)
is transformed into the term <<izUNKNOWN(X)>>
  in context of
izUNKNOWN(X) <=> meat_block(unboiled(X)),
                 boil(unboiled(X),X,pan)
  (belongs to domain MEAT).
Please complete this:
```

`izUNKNOWN` shows that the **PAR** system does not know the corresponding token. Here, this is the very concept of *stew*. So, the user may respond to the question above as follows.

```
stew(X).
```

The **PAR** system is satisfied with this answer and searches the next question.

3.5.2 “Block and slice” partition

It is a feature of **PAR** that its analogical reasoning can be controlled by external tokens even when the same paraphrasings are given. Let us give the **PAR** system an input file as in figure 3.5.

```
[meat,fish,uncut,boil,unboiled,pan,roast,unroasted,frying_pan].
    meat_block(x)  ⇔  meat(uncut(x)),block(uncut(x),x).    (1)
    meat_slice(x)  ⇔  meat(uncut(x)),slice(uncut(x),x).    (2)
    fish_block(x)  ⇔  fish(uncut(x)),block(uncut(x),x).    (3)
    fish_slice(x)  ⇔  fish(uncut(x)),slice(uncut(x),x).    (4)
 bouillabaisse(x)  ⇔  fish_block(unboiled(x)),
                      boil(unboiled(x),x,pan).             (5)
         steak(x)  ⇔  meat_slice(unroasted(x)),
                      roast(unroasted(x),x,frying_pan).    (6)
```

Figure 3.5: Input file 2

We obtain two domains from this also. One is {(1),(3),(5)}(called the *block* domain), and another is {(2),(4),(6)}(called the *slice* domain).

This time the **PAR** system detects T-equivalence {`(slice,block)`, `(meat_slice,meat_block)`} from (1) and (2). Then, the **PAR** system tries to make **PAR** from (6) with the base domain *slice* and the target domain *block*. So, it presents the following question to an oracle.

```
I know the term <<steak(X)>>
  in context of
steak(X) <=> meat_slice(unroasted(X)),
             roast(unroasted(X),X,frying_pan)
  (belongs to domain SLICE)
is transformed into the term <<izUNKNOWN(X)>>
  in context of
izUNKNOWN(X) <=> meat_block(unroasted(X)),
                 roast(unroasted(X),X,frying_pan)
  (belongs to domain BLOCK)
Please complete this:
```

Here, this is the very concept of *dice steak*[17]. So, the user may respond to the question above as follows.

```
dice_steak(X).
```

[17] Grilled small blocks of beef. This is a very popular dish in a Japanese *family* restaurant.

The **PAR** system is satisfied with this answer and searches the next question.

3.6 Summary

The author has regarded creative thinking as a process which can be dealt with by machine learning theories, and have defined paraphrasing-based analogical reasoning as a framework for creative thinking on theories consisting of paraphrasings. We have implemented a prototype system according to this definition, and have confirmed the effect of **PAR**. A method to make partitions based on given external tokens are practically useful as shown in section 3.5, and discussed in subsection 3.4.1. But there are still many problems in **PAR** as mentioned in subsection 3.4.2.

The current largest problem in **PAR** is that the size of T-equivalence does not ensure its usefulness for **PAR**. Perhaps a domain-dependent measure of the usefulness of T-equivalence should be implemented.

When we regard **PAR** as a practical creativity support system, we can assume other oracles than the godparent. For example, an oracle accepts some internal tokens and gives corresponding tokens to them under limited conditions. This sort of oracle is called a *ferryman*. Now we are studying on the extension of **PAR** with ferryman with respect to internal tokens without its paraphrasing.