Creative Problem Solving by Concept Generation Using Relation Structure

Katsutoshi Kanamori and Hayato Ohwada

Tokyo University of Science, Chiba, Japan
{katsu, ohwada}@rs.tus.ac.jp

Abstract. The purpose of our work is to achieve creative knowledge processing. In this paper, we focus on the formulation of concept generation and its use in problem solving. We propose a method for solving a problem by generating new concepts that have never appeared in existing knowledge. We propose Creative Problem Solving, which can derive a goal state by using a creative leap invoked by concept generation.

1 Introduction

Our work seeks to achieve creative knowledge processing. Although some studies have been conducted on creativity via computer, the intention of this paper is to formulate concept generation based on logic, and to investigate problem solving with concept generation. Only a few attempts have been made at such a study. There are studies Predicate Invention in ILP,[1, 2] but our focus was not only on induction, but rather on developing a general method of concept generation. This concept generation constitutes a new approach to problem solving, addressing problems that induction cannot solve. An early study described an AM[3] concept generation system cannot be considered a general concept generation.

This paper proposes a method for solving problems by generating new concepts that have never appeared in existing knowledge, and we confirm that such knowledge processing is implementable. We call this kind of processing Creative Problem Solving, and consider it a part of creative knowledge processing.

2 Preliminaries

2.1 Knowledge and Relation Structures

We are concerned with first-order logic as representing Knowledge. Any logical formulae can be transformed to the formulae include no functions and no individual constants. We may regard concept generation as predicate generation.

In this paper, knowledge is defined as a set of logical formulae with no functions and no individual constants. A relation structure is defined as a logical formula that has at least one predicate variable. A relation structure also has no functions and no individual constants.

For exmaple, Let $S_1 = \{\forall x, y : \neg P_1(x) \vee P_2(x, y)\}$, $S_2 = \{\forall x, y : \neg X_1(x) \vee P_2(x, y)\}$. When P_1, P_2 are predicate constants and X_1 is a predicate variable, S_1 is knowledge, and S_2 is a relation structure.

2.2 Simple Substitution

We define the substitution replace predicate variables with predicate constants.

Let P_{vn} be a universal set of free predicate variables of arity n, and let P_{cn} be a universal set of invariable predicate of arity n. If Θ satisfies $\Theta \subset P_{v1} \times P_{c1} \cup \cdots \cup P_{vi} \times P_{ci}$, and each variable and constant that occurs in Θ is distinct, then we say Θ is a *simple substitution*. The element of Θ is written in a manner similar to the style of the general substitution, v/c. Here, v is the variable and c is the constant.

For example, relation structure $S_1 = \{\forall x : X_1(x, y) \wedge X_2(x)\}$ and simple substitution $\Theta_1 = \{X_1/A, X_2/B\}$ are given, then $S_1\Theta_1 = \{\forall x : A(x, y) \wedge B(x)\}$.

3 Concept Generation

3.1 Predicate Generation

When knowledge Σ, relation structure RS, and a predicate variable X (which occurs in RS) are given, we define new knowledge S_{NEW} that holds a new predicate. If a simple substitution Θ has all predicate variables that occur in RS except X, then S_{NEW} is defined as follows : $S_{NEW} = RS(\Theta \cup \{X/NEW\})$.

The new Predicate NEW is obtained by generating new knowledge S_{NEW}. Here, the generated S_{NEW} is determined uniquely by Σ, RS, X, Θ, and new symbol NEW. Therefore we can regard predicate generation in terms of sets of these five values $(\Sigma, RS, X, \Theta, NEW)$.

3.2 Characteristics of New Knowledge

Novelty Novelty is a property representing whether new knowledge is obtained as a logical conclusion based on existing knowledge or not. If S_{NEW} satisfies the condition that : for all s such that $S_{NEW} \models s$ and s includes NEW and $\Sigma \not\models s$, then the predicate generation is said to possess novelty.

Consistency Consistency is the property by which new knowledge and existing knowledge are not in contradiction. With consistent predicate generation, we can take on $\Sigma \cup S_{NEW}$ as new knowledge instead of S_{NEW}.

Soundness To be sound means that $\Sigma \cup S_{NEW}$ is consistent with all logical formulae which are consistent with Σ and has no new predicates. If this condition is true, then we say that the predicate generation is sound or S_{NEW} is sound.

3.3 Example of Predicate Generation

Let Σ be knowledge and RS be a relation structure as follows:

$$\Sigma = \{\forall x : \neg bird(x) \vee ab(x) \vee fly(x)\}, RS = \left\{ \begin{array}{l} \forall x : \neg X_0(x) \vee X_1(x) \vee X_2(x) \\ \forall x : \neg X_3(x) \vee X_0(x) \\ \forall x : \neg X_3(x) \vee \neg X_2(x) \end{array} \right\}$$

knowledge Σ describes that birds fly if not ab. See predicate variable X_3, and consider two simple substitutions for predicate generation.

$\Theta_1 = \{X_0/bird, X_1/ab, X_2/fly\}$, $\Theta_2 = \{X_0/fly, X_1/bird, X_2/ab\}$

Predicate generations $(\Sigma, RS, X_3, \Theta_1, NEW_1)$, $(\Sigma, RS, X_3, \Theta_2, NEW_2)$ then generate new knowledge S_{N_1}, S_{N_2} as follows:

$$S_{N_1} = \left\{ \begin{array}{l} \forall x : \neg bird(x) \vee ab(x) \vee fly(x) \\ \forall x : \neg New_1(x) \vee bird(x) \\ \forall x : \neg New_1(x) \vee \neg fly(x) \end{array} \right\} S_{N_2} = \left\{ \begin{array}{l} \forall x : \neg fly(x) \vee bird(x) \vee ab(x) \\ \forall x : \neg New_2(x) \vee fly(x) \\ \forall x : \neg New_2(x) \vee \neg ab(x) \end{array} \right\}$$

NEW_1 is a new concept means such as "flightless bird", NEW_2 means "not ab and fly". S_{N_1} has novelty and are consistent and sound. S_{N_2} has novelty and is consistent but not sound. In fact, if $l = \neg(\forall x : \neg fly(x) \vee bird(x) \vee ab(x))$, then $\Sigma \wedge l$ is consistent, but $(\Sigma \cup S_{N_2}) \wedge l$ is not.

4 Expand Dimension

Even if the predicate generation is inconsistent, it can be useful in expanding the knowledge dimension. For example, consider adding a concept describing imaginary numbers to real-number knowledge. The rule "the square of any number greater than or equal to 0" is inconsistent with the new knowledge, but serves to expand the dimension of real numbers to complex numbers, making it possible to regard new knowledge as consistent knowledge. This is a method for expanding the knowledge dimension naturally by predicate generation

Now, if $C_\Sigma \subseteq \Sigma$ and $(\Sigma - C_\Sigma) \cup S_{NEW}$ is consistent and $C_\Sigma \cup S_{NEW}$ is inconsistent, consider C'_Σ as follows.

$$C'_\Sigma = \left\{ \begin{array}{l} \neg inD(x_{11}) \vee \cdots \vee \neg inD(x_{1l_1}) \vee s'_1 \\ \vdots \\ \neg inD(x_{n1}) \vee \cdots \vee \neg inD(x_{nl_n}) \vee s'_n \end{array} \right\}$$

inD is a new predicate prepared here, and $s_1, \cdots, s_n$ are elements of C_Σ. $x_{j1}, \cdots, x_{jl_j}$ are all bind variables occuring in s_j. $s'_1, \cdots, s'_n$ are formulae obtained by transforming $s_1, \cdots, s_n$ to prenex normal form and cutting out the head part. Each formula for C'_Σ has a head part $q_{j1}x_{j1} \cdots q_{jl_1}x_{jl_1}$ that is the head part of the corresponding s_j, although these are omitted here. q_{jk} is $\exists$ or $\forall$, equal to a qualifier of s_j. We can imagine that $inD(x)$ means x is in a dimension of former knowledge. $(\Sigma - C_\Sigma) \cup C'_\Sigma \cup S_{NEW}$ is then consistent.

Theorem 1. *For predicate generation $(\Sigma, RS, X, \Theta, NEW)$, if Σ and RS are consistent then $\Sigma' = (\Sigma - C_\Sigma) \cup C'_\Sigma \cup S_{NEW}$ is consistent.*

Proof. We assume that Σ' is inconsistent. According to the premise, Σ and RS are consistent, so $(\Sigma - C_\Sigma)$ and S_{NEW} are consistent. Also, C'_Σ is consistent because if $C_\Sigma \models C'_\Sigma$, $(\Sigma - C_\Sigma) \cup C'_\Sigma$ is consistent. Due to the definition of C_Σ, $(\Sigma - C_\Sigma) \cup S_{NEW}$ is consistent. For the reasons above, if we assume that Σ' is inconsistent, then $C'_\Sigma \cup S_{NEW}$ is inconsistent. Here, let φ be a set of logical formulae. We will describe the Skolem normal form of φ, $SNF(\varphi)$. If $C'_\Sigma \cup S_{NEW}$ is inconsistent, then the empty clause might be derived by resolution. Thus if predicate inD does not occur in $SNF(S_{NEW})$, there exists a set of clauses C has no inD derived by resolution, and $C \wedge SNF(S_{NEW})$ is inconsistent. All of the clauses in $SNF(C'_\Sigma)$ have negative inD literals and no positive inD literals, so it is impossible to obtain a clause with no inD from $SNF(C'_\Sigma)$ by resolution. Therefore this is inconsistent with the assumption, Σ' is consistent. □

5 Creative Leap

For predicate generation, it is important to consider whether the new knowledge represents a leap or not. The leap enable us to achieve our creative goal.

In defining a creative leap, it is important to consider whether new predicates occur in generated consequence from the conjunction of new knowledge and existing knowledge. Formulae that have new predicates, cannot belong to unknown facts, so they are not appropriate as leap conclusions, because a new predicate is made by the system on its own.

Therefore, we define a Creative Leap as the case where there exist logical formulae not having NEW, such that $\Sigma \cup S_{NEW} \models s$ and $\Sigma \not\models s$. We then, say the predicate generation leaps.

Theorem 2. *If predicate generation is not sound, then it leaps.*

Proof. Due to the definition of soundness, if predicate generation $(\Sigma, RS, X, \Theta, NEW)$ is not sound, then there exists logical formula l such that $\Sigma \wedge l$ is consistent and $(\Sigma \cup S_{NEW}) \wedge l$ is inconsistent. Because NEW does not occur in l, there exists logical formula l^- such that $\Sigma \cup S_{NEW} \models l^-$ and $l \wedge l^-$ is inconsistent. Thus $\Sigma \wedge l$ is consistent and $\Sigma \not\models l^-$. $\Sigma \not\models l^-$ and $\Sigma \cup S_{NEW} \models l^-$ and NEW not occurring in l^-, fulfill the definition of a creative leap. For the reasons above, if predicate generation is not sound, then leaps. □

6 Creative Problem Solving

When a goal state, which is not derived from existing knowledge and it's negation is not derived either, is given, predicate generation with a creative leap can lead to the goal state as a consequence. Even if existing knowledge derives the negation of the goal state, expanding of dimensions enable to lead to the goal state. We call such problem solving Creative Problem Solving.

For example, human kind had wanted to fly like a bird. And known that birds can fly and the wing enables them, so create new concept like a wing.

Example 1.

$$\Sigma_f = \left\{ \begin{array}{l} \forall x \exists y : Bird(x) \rightarrow Has(x,y) \wedge Wing(y) \\ \forall x : Human(x) \rightarrow \neg(\exists y : Has(x,y) \wedge Wing(y)) \\ \forall x,y : Has(x,y) \wedge WingBehavior(y) \rightarrow Fly(x) \\ \forall x : Wing(x) \rightarrow WingBehavior(x) \end{array} \right\}$$

is given, and goal $G_f = \forall x : Human(x) \rightarrow Fly(x)$. Then Σ_f cannot lead to G_f as a consequence. Then we consider RS as follows:

$$RS_f = \left\{ \begin{array}{l} \forall x \exists y : X_1(x) \rightarrow X_2(x,y) \wedge X_3(y) \\ \forall x,y : X_2(x,y) \wedge X_4(y) \rightarrow X_5(x) \\ \forall x : X_3(x) \rightarrow X_4(x) \end{array} \right\}$$

The simple substitution $\Theta_f = \{X_1/Human, X_2/Has, X_4/WingBehavior, X_5/Fly\}$. Predicate generation $(\Sigma_f, RS_f, X_3, \Theta_f, NEW_f)$ then generates S_{NEW_f} as follows :

$$S_{NEW_f} = \left\{ \begin{array}{l} \forall x \exists y : Human(x) \rightarrow Has(x,y) \wedge NEW_f(y) \\ \forall x,y : Has(x,y) \wedge WingBehavior(y) \rightarrow Fly(x) \\ \forall x : NEW_f(x) \rightarrow WingBehavior(x) \end{array} \right\}$$

S_{NEW_f} possesses novelty and is consistent but not sound.

S_{NEW_f} derives G_f. G_f was not obtained by Σ_f, therefore, a new predicate is generated to derive the goal state by using a creative leap. NEW_f was not generated randomly, and may fulfill $WingBehavior$ like a bird's wing.

6.1 Practical Method of Predicate Generation

When a goal state is given, what prepared for generation is Σ_f only. The important topic is how to prepare a relation structure and which simple substitutions and predicate variables to choose. The problem of what predicate should be generated and what relation structure should be prepared is differs for each purpose and each case. We show a general method of extracting relation structures from knowledge, and present one of the methods for preparing RS, Θ, and X.

The relation structure can be gained by generalizing knowledge. Concretely, replacing all or some predicate constants occurring in knowledge (or a subset) with free predicate variables.

Next, the point is which RS, Θ, and X are chosen. The destination is to derive G from $\Sigma \cup S_{NEW}$. We propose a method of extracting relation structure from an explanation structure inherent in existing knowledge.

Now, for a knowledge σ, let $RS(\sigma)$ be a relation structure obtained by replacing all of the predicate constants in σ with each independent predicate variables.

If there exists Σ^* such that $\Sigma^* \subset \Sigma : \exists \Theta : RS(\Sigma^*)\Theta \models RS(G)\Theta$, then we can consider predicate generation using $RS(\Sigma^*)$ as a relation structure. This $RS(\Sigma^*)$ is an explanation structure that derives a result with the same structure as G. New knowledge possossing the same explanation structure may be obtained by using another simple substitution to derive G. It can be thought this method is based on analogy as common structure.

Example 2. Only knowledge Σ_f and goal state G_f are given. Here, $RS(G_f) = \{\forall x : X_1(x) \rightarrow X_2(x)\}$, and Σ_f^* including $RS(G_f)$, can be taken :

$$\Sigma_f^* = \left\{ \begin{array}{l} \forall x \exists y : Bird(x) \rightarrow Has(x,y) \wedge Wing(y) \\ \forall x,y : Has(x,y) \wedge WingBehavior(y) \rightarrow Fly(x) \\ \forall x : Wing(x) \rightarrow WingBehavior(x) \end{array} \right\}$$

so we can obtain $RS(\Sigma_f^*)$ in concert with $RS(G_f)$ as : $RS(\Sigma_f^*) = RS_f$.

Let $\Theta = \{X_1/Bird, X_2/Fly, X_3/Has, X_4/Wing, X_5/WingBehavior\}$, then $RS(\Sigma_f^*)\Theta \models RS(G_f)\Theta$. Define simple substitution Θ' as follows : $\Theta' = \Theta_f$. Then, predicate generation $(\Sigma_f, RS(\Sigma_f^*), \Theta', X_4, NEW_f)$ produces $S'_{NEW} : S'_{NEW} = S_{NEW_f}$. A leap result with goal state G_f is derived from S'_{NEW}.

7 Conclusion

This paper proposed a method of concept generation, and an approach to creative problem solving. A creative leap shows the new predicate is not a mere paraphrase but achieves a new result.

We may note here that creative problem solving is similar to abduction in the sense that it is a method of deriving a goal by generating new knowledge. If the goal state is observed facts, we can regard creative problem solving as abduction accompanied by concept generation. However, creative problem solving is a more general method.

Our example of creative problem solving was the simulation of the invention of something like an airplane. Though not every invention can be described using this frame, it is very interesting that creative problem solving can simulate a part of human's creative knowledge processing and invention. However, many problems need to be solved to develop a practical system with creative problem solving. The problem of calculating of new-knowledge consistency, determining which structures and predicate variables are important, and selecting simple substitutions, among other factors, are important for practical systems. Some of these problems (perhaps most of them) are depend on specific systems.

We showed logically that concept generation may achieve creative knowledge processing. We expect to achieve a fully creative system in future work.

References

1. S.H. Muggleton., and Buntime, R.: Machine Invention of first-order Predicates by Inverting Resolution. the 5th Intl. Workshop on Machine Learning, ANN Arbor, MI. (1988) 339–352
2. S.H. Muggleton.: Predicate invention and utilisation. Journal of Experimental and Theoretical Artificial Intelligence, 6(1). (1994) 127–130
3. Davis, R., Lenat, D .B.: Knowledge-Based System in Artificial Intelligence. McGraw-Hill, NewYork (1982)