1 Introduction

In the fields of *conceptual classification* and *information retrieval*, a notion of *similarity* among objects plays a crucial role. The important point to be noted is that when we discuss similarities, we have to take its *dependence on a purpose or a context* into consideration. A standard approach to this issue is to use a *weighting schema* for attributes of objects or classes. In fact, the approach has been widely adopted in many studies on classification or information retrieval. In contrast to them, this paper proposes an approach from the viewpoint of *abstractions*.

As many researchers have reported, a notion of abstraction is well-known as a useful approach to improve efficiency of *problem solving*, e.g. *theorem proving* [3, 1] and *planning* [4, 2, 8]. In our previous works, the authors have claimed that the usefulness of abstractions should be *dependent on a goal to be proved or a problem to be solved*. According to the statement, we have proposed a framework of *Goal-Dependent Abstraction* (GDA) for the task of theorem proving under first-order Horn logic [7]. Then our framework has been extended to formalize a method of *analogical reasoning* under *order-sorted Horn logic* [13, 14]. In this extended framework [5], our GDA can work as the basis for similarities among primitive *sort concepts*, according to which some reasoning is performed *analogically*. In this paper, we try to further extend our GDA framework so that we can consider similarities among complex conceptual structures, represented by *concept graphs* (CG, for short), including primitive sort concepts as their special cases. The major reason why we perform such an extension in this paper is to establish a logical foundation of metaphoric search problem described as follows.

1.1 Metaphoric Search Problem and Its Abstraction Approach

We suppose a goal CG, representing a goal or a purpose, and an initial CG subsumed by the goal CG. The initial one is intuitively considered as a typical instance of the goal CG. Given such a pair of concept graphs, we suppose

> a task of searching for a concept graph satisfying the goal CG just like the initial CG.

The metaphoric search intends to restrict the search scope designated by the goal CG. Normally, most users are not aware of what they really want to search for at the initial stage of retrieval, while they are supposed to provide a search query corresponding to the goal CG. When no restriction is made for the query, the users will receive too many answers satisfying the goal. Then they must specify additional constraints to remove a lot of useless answers. However, this is a troublesome task to do. Instead of performing such a relevance-feedback process, we propose to apply a metaphoric search. Anyone who can specify a goal formula will give at least one typical instance, even when he/she cannot provide an exact or a precise formula realizing what he/she looks for. Then, our system should examine target objects, also represented by CGs, similar to the typical instance of the goal. The retrieved CGs are thus required not only to satisfy the goal but also to have a similarity to the typical instance.

The goal and its typical instance are, for instance, `dangerous_thing` and `car_in_a_park` both of which consist of various primitive sort concepts and relationships among them and are therefore represented by concept

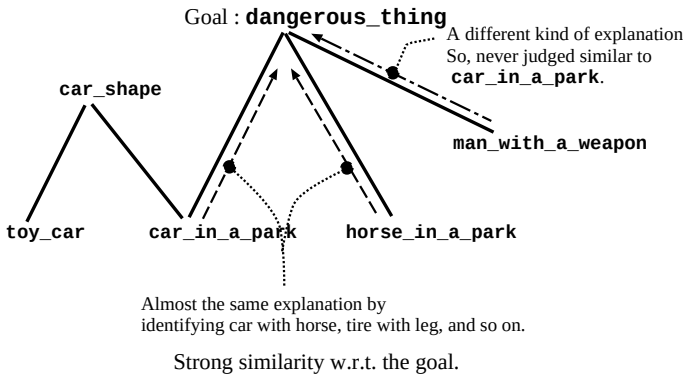


Figure 1: The Notion of Similarity under a Presence of Goal Concept

graphs. Then the problem is to determine what structured concepts in CG representation are dangerous just like `car_in_a_park`. When all the properties needed to explain why `car_in_a_park` is dangerous are very particular to `car`, then we have no way to interprets any other concept as a kind of `car`. However in this case, the properties used in the explanation will also hold for other things, `horse_in_a_park`, `tank_in_a_park` and so on. In addition, by neglecting the differences between `car`, `horse` and `tank`, we have an abstract sort concept, `car_or_horse_or_tank` defined as a disjunction of each concept, and a family of abstract properties of the abstract concept from which we can draw the dangerousness independently of particular cases. Such a group of primitive sort concepts, {`car`, `horse`, `tank`} in this example, is firstly computed from a chosen proof/explanation of the goal. The abstract property is precisely defined as a formula whose validness is independent of particular instantiations.

The concept graph consisting of such abstract properties denotes an abstract dangerous situation with all the above cases as its special instances. Then, we say that a CG is *potentially similar* to the initial situation `car_in_a_park` w.r.t. the goal `dangerous_thing`, if the CG is a specialization of some instance of the abstract CG.

It should be noted here that our notion of potential similarity is stronger than the standard similarity meaning that two or more things are similar whenever they possess some common properties. In the case of CG representation, the latter similarity notion is defined by the subsumption relationships between CGs. For instance, `man_with_a_weapon` is also dangerous, so the corresponding CG will be subsumed by the goal CG, `dangerous_thing`. However, `car_in_a_park` and `man_with_a_weapon` have each individual explanation of the dangerousness. In other words, `man_with_a_weapon` may not have some properties used in the `car` case. In fact, `car` can run fast, but `man` cannot. The former fact about `car` is used in the explanation. Therefore, as illustrated in Figure 1, `man_with_a_weapon` is never judged similar to `car_in_a_park`, and is not regarded as a possible instance of the abstract concept CG corresponding to `car_or_horse_or_tank_in_a_park`. So, any specialization of `man_with_a_weapon` cannot be an answer to our query, while `horse_in_a_park` and `tank_in_a_park` are possible answers. This completes a brief story which this paper is going to describe.

1.2 Comparison with Related Studies

The notion of concept graphs is similar to Ψ-*terms* [9] in logic programming, and can be regarded as a restricted form of concept definitions in *Description Logic* [15, 12, 10]. More precisely, a concept graph consists of typed variables connected by relational predicates, and just corresponds to a body part of Horn rule in logic programming. It has just one external variable called a root. All the other internal variables are existentially quantified.

As the goal CG is satisfied by the initial CG, the former should be more general than the latter. Such an ordering among concept graphs is formally defined as a concept graph subsumption under a domain theory, and is understood as a restricted form of generalized subsumption [16] when we translate order-sorted Horn logic to the standard Horn logic not involving type information syntactically.

One may understand that the system performs an operation like *explanation-based generalization* [19]. This is in some sense true because the present system is completely deductive under a domain theory except the following points.

EBG is used to deductively learn a universally quantified macro rule from a domain theory. For this purpose, some constant symbols are replaced with variables. On the other hand, according to our abstraction approach, a rule corresponding to a universal macro rule is restricted to some disjunction of primitive types, called sorts. In fact, the disjunction of sorts defines a subset of universe of discourse. It is not the whole interpretation domain. Thus a family of predicate symbols is replaced with more abstract predicate whose extension is wider than any predicate in that family.

Secondly, the system performs generalization to obtain more general abstractable concept graph, only when some adequate similarity cannot be found. As an initial concept graph is more specific, the similarities among them may be fewer. As a result, fewer situations become similar to the initial concept graph. In such a case, we apply the generalization operator to obtain a more general situation to which more concept graphs become similar.

Our GDA in this paper is closely related to some important studies with which the notion of similarities among concepts is deeply concerned. We especially mention here the relationships with *Purpose-Directed Analogy* (PDA) [20] and *Instance-Based Learning* (IBL) [21].

In the framework of PDA, for a goal concept and its purpose, a *base* example of the concept is tried to explain how it satisfies the purpose. Then, the explanation is used to justify a similarity between a *targe* example and the base one. If the target can be explained to satisfy the purpose in almost the same way, it can be considered similar to the base example. We can consider that our similarity is also based on this criterion. However, there is a significant difference between them. PDA assumes just one target, while our GDA can take all *potential targets* into account simultaneously that share the same explanation structure of the purpose. This is a great advantage of GDA to be emphasized. Based on such potential similarities, for example, we can formalize a flexible information retrieval method, called a *Metaphorical Information Retrieval*. The method is in fact under our investigation.

In IBL, similarities between concepts written in a *typed-language* play a very important role in learning problems including a classification problem. In this sense, we can find a close relationship between IBL and GDA. However, the similarity in IBL is argued without considering its dependence

on purpose or goal. From this point of view, the authors consider that the notion of similarities in IBL is not sufficient for our task. However, the efficiency of the learning method is very attractive. It works efficiently even for a large scale of data. In order to make our method more practical, therefore, it would be worth taking the computational advantage in our method.

This paper is organized as follows. In Section 2, we describe our underlying language, an order-sorted Horn logic, according to [13, 14], and a framework of sorted abstraction in the logic. In Section 3, a notion of concept graphs and their subsumption relationships are introduced. We present in Section 4 a framework of Goal-Dependent Abstraction (GDA) for order-sorted theory, which is a refined version of GDA stated in [5, 7], and discuss a context-dependent similarity among concept graphs. In Section 5, we show how our system behaves, using two examples. Some experimental results are also presented. In Section 6, we conclude this paper with discussion about some future work.

2 Preliminaries

Our domain theory is described in terms of *order-sorted logic* (See [13], for instance), so we first introduce it briefly. Secondly, we review a notion of *sorted abstraction*, presented in the literature [5], for grouping several primitive concepts in the domain theory and for constructing an abstract level theory.

2.1 Domain theory

A *domain theory* $\mathcal{KB}$ consists of two components, a *sort hierarchy* $\mathcal{H}$ and a *rule set* $\mathcal{T}$, and is denoted by $\langle \mathcal{H}, \mathcal{T} \rangle$.

2.1.1 Sort hierarchy

A *sort hierarchy* is given as a partially ordered set of sort symbols, $\mathcal{H} = (S, \preceq)$. Each sort symbol $s \in S$ denotes a concept and is interpreted as a non-empty set A_s of individual objects of s, given an interpretation. An ordering $s \preceq s'$ means that s is a subconcept of s', and is understood as a subset relationship $A_s \subseteq A_{s'}$ under the interpretation. We also say that s' *subsumes* s. Furthermore, we use no constant symbol denoting an individual, although Skolem constants are introduced for some technical reasons. Instead of constant symbols, some minimal sorts $s \in S$ may be understood as *individuals*. Formally, a sort s is said minimal if no $s' \in S$ except s satisfies $s' \preceq s$. Thus, unlike the standard formalism of sort hierarchy, we do not suppose the least sort $\bot$ meaning the contradiction or empty set.

2.1.2 Rule set

A *rule set* $\mathcal{T}$ is a set of function-free typed Horn clauses of the form

$$\forall X_1 : s_1 \cdots \forall X_n : s_n \quad A :- B_1, \cdots, B_k,$$

where A and B_i are positive literals and $X_j : s_j$ is a typed variable of sort $s_j \in S$, and $X_1, \cdots, X_n$ are all the variables appearing in it. X_j is supposed to have an instance of s_j as its possible value. When we simply omit

the universal quantifier, the first occurrence of X_j appears in the form of $X_j : s_j$. Other occurrences are simply written in its non-typed form. For instance, `married(X:person):- has_spouse(X,Y:person)` is an abbreviation of `∀X:person∀Y:person married(X):- has_spouse(X,Y)`.

2.2 Order-sorted inference:

The inference under $\mathcal{KB} = \langle \mathcal{H}, \mathcal{T} \rangle$ is the same as the standard first-order deduction except a point that every substitution used in our inference is assumed to be *order-sorted*. More precisely, every substitution $\theta = \{X_i : s_i / Y_i : u_i\}$, called an *order-sorted substitution*, should satisfy the type constraint $u_i \preceq s_i$. That is, the variable X_i of s_i must be instantiated to a variable Y_i of more specific sort u_i. Then, two terms $X_i : s_i$ and $Y_i : u_i$ are unifiable only if one of the following two conditions holds:

(a) Either $s_i \preceq u_i$ or $u_i \preceq s_i$.

(b) s_i and u_i has a common subsort v_i. That is, $v_i \preceq s_i$ and $v_i \preceq u_i$.

In the case (b), we introduce a new variable Z_i of a maximum v_i among common subsorts satisfying (b), and then make a unifier $\{X_i : s_i / Z_i : v_i, Y_i : u_i, / Z_i : v_i\}$.

When s_i and u_i have no common subsort, the unification fails. First order deduction equipped with the order-sorted unification is written as $\mathcal{T} \vdash_{\mathcal{H}} F$. In what follows, the term "order-sorted" may be dropped, for we assume it throughout this paper.

2.3 Sorted abstraction

We introduce here a notion of *sorted abstraction*. A non-sorted version of abstraction has been originally proposed by J.D.Tenenberg [1] for constructing a grouping of relational predicates. On the other hand, by the sorted abstraction, only monadic predicates denoting concepts are grouped together. Thus, our notion of abstraction is defined as a grouping of sorts in $\mathcal{H}$.

Definition 1 (Grouping of Sorts)
Let $\mathcal{H} = (S, \preceq)$ be a sort hierarchy. A grouping ψ of sort set S is simply a partition $\{g_1, \cdots, g_k\}$ of S. That is, $g_i \cap g_j = \phi$ $(i \neq j)$ and $S = \bigcup_{j=1}^{k} g_j$. Each g_j is called a group of sorts. Regarding g_j as an new abstract sort symbol, we identify ψ with a mapping given by $\psi(s) = g_j \in \psi$ if $s \in g_j$. ■

A grouping $\psi = \{g_1, \cdots, g_k\}$ can be easily extended to a mapping over a set of any expressions. For an expression E, $\psi(E)$ is defined as an expression obtained by rewriting any sort s in E to its abstract sort $\psi(s) = g$, where $s \in g$. $\psi(E)$ is simply called a ψ-*image* of E. Similarly, ψ-image of a sort subsumption $s_1 \preceq s_2$ is defined as $\psi(s_1) \preceq \psi(s_2)$.

As a grouping ψ classifies sorts in a sort set into several groups g_j, we can consider that ψ represents a hypothesis that s_1 and s_2 are similar whenever they belong to the same group g_j. In this sense, we say that s_1 is an *analogue* of s_2 or *similar* to s_2 w.r.t. ψ. The ψ-image $\psi(E)$ can be regarded as an expression at abstract level obtained by neglecting the differences between similar sorts in the same groups. By such a characteristic of ψ, $\psi(E)$ may not be valid even though the original E, either a rule or a sort

subsumption, is entailed by $\mathcal{KB} = (\mathcal{H}, \mathcal{T})$. So let us briefly examine the validness of abstract expression $\psi(E)$, according to the following definition.

Definition 2 (Meaning of Abstract Sorts)
For a grouping $\psi = \{g_1, \cdots, g_n\}$ of sort set S, g_j is treated as a new sort symbol defined by a disjunction $\bigvee_{s \in g_j} s$. So g_j is interpreted as the set given by $A_{g_j} = \bigcup_{s \in g_j} A_s$, where A_s is the denotation of s. ■

For instance, suppose we have a rule

$$E = E(\texttt{man}) = (\texttt{responsible_to_children(X:man):-parent(X)})$$

in $\mathcal{T}$ and a group g={`man, woman`} $\in \psi$. Then we have its ψ-image

$$\psi(E(\texttt{man})) = (\texttt{responsible_to_children(X:}g\texttt{):-parent(X)})$$

asserting that `responsible_to_children(X)` whenever `parent(X)`, where `X` is either `man` or `woman`. Therefore, $\psi(E(\texttt{man}))$ is valid if and only if $E(\texttt{woman}) \in \psi^{-1}(\psi(E(\texttt{man})))$, called an ψ-analogue of $E(\texttt{man})$, is also valid. Thus, $\psi(E(\texttt{man}))$ is a valid abstract rule not dependent to particular cases. More exactly, $\psi(E(\texttt{man}))$ is evaluated true for any interpretation satisfying $A_g = A_{\texttt{man}} \cup A_{\texttt{woman}}$. In such a case, we say that the original rule $E(\texttt{man})$ is abstractable w.r.t. the grouping ψ.

Since our domain theory involves the sort hierarchy $\mathcal{H} = (S, \preceq)$, the notion of abstractableness should be also defined for $\mathcal{H}$, based on Definition 2. Suppose we have a sort subsumption $\texttt{boy} \preceq \texttt{man}$ in $\mathcal{H}$ and a grouping $\psi = \{g_1 = \{\texttt{boy,girl}\}, g_2 = \{\texttt{man,woman,...}\}, ...\}$. Then, $\psi(\texttt{boy} \preceq \texttt{man}) = g_1 \preceq g_2$ is required to be true for every interpretation satisfying Definition 2, if and only if there exists a sort, `woman` in this case, such that $\texttt{girl} \preceq \texttt{woman}$ holds. If otherwise, we cannot guarantee $A_{\texttt{girl}} \subseteq A_{g_1} \subseteq A_{g_2} = \cup_{s \in g_2} A_s$. Thus, we require that, for any ψ-analogue u_1 of `boy`, there exists some ψ-analogue u_2 of the super sort `man` such that $u_1 \preceq u_2$.

Definition 3 (Abstractable Rule and Subsumption)
Let $\langle \mathcal{H} = (S, \preceq), \mathcal{T} \rangle$ be a domain theory and $\psi = \{g_1, \cdots, g_n\}$ be a grouping of S.

(a) A clause $C = (A :- B_1, \cdots, B_k)$ in $\mathcal{T}$ is said to be *abstractable w.r.t.* ψ to the corresponding ψ-image $\psi(C)$, if $\mathcal{T} \vdash_{\mathcal{H}} D$ holds for any $D \in \psi^{-1}(\psi(C))$, a ψ-analogue of C, where $\psi^{-1}(E_a) = \{E_c | \psi(E_c) = E_a\}$ (that is, the set of ψ-instances of E_a).

(b) A sort subsumption $s_1 \preceq s_2$ in $\mathcal{H}$ is said to be *abstractable w.r.t.* ψ to its ψ-image $\psi(s_1) \preceq \psi(s_2)$, if, for any ψ-analogue u_1 of s_1, there exists a ψ-analogue u_2 of s_2 such that $u_1 \preceq u_2$ holds in $\mathcal{H}$.

■

3 Concept Graphs and Their Subsumption Relationships

In order to describe a situation in inquiry, a notion of *concept graph* (CG) is introduced in terms of order-sorted Horn logic. After that, we consider in Section 4 the problem of similarities among the concept graphs, given

a goal situation. A concept graph consists of conceptual nodes labeled by sorts and relational predicates connecting them.

The concept graph can be considered as a variation of Ψ-terms [9] for representing complex record structures in logic programming. In fact, the syntax of Ψ-terms and the concept graphs are the same in case we treat only binary predicates. The differences are found in their semantics even in that case. In the Ψ-terms, links connecting conceptual nodes are regarded as attribute roles that are interpreted as functions, while the corresponding links in the concept graph in this paper are role relations interpreted as n-ary relations over an interpretation domain.

Similar notions are also used in the research area on *description logic* to depict their concept definitions in the form of graphs [12, 15]. Although the description logic can deal with various kinds of constraints for role relations, we are concerned with only existentially quantified constraints to perform our inference equipped with the concept graph formalism within the framework of order-sorted Horn logic.

Definition 4 (Concept Graph)
A *concept graph* $G(X : s)$ is simply defined as a conjunction of atomic formulas, where just one variable $X : s$ is designated as a root variable, where s is a sort in our sort hierarchy $\mathcal{H}$. Every variable except $X : s$ is assumed to be existentially quantified. When every predicate in $G(X : s)$ is unary or binary, then we depict it as a finite directed graph (E, R) with a set E of typed variables in $G(X : s)$ and a set R of predicate instances connecting them. ■

For instance, Figure 2 shows a concept graph

CG_1(X:person)= { has_a_child(X, Y:person), likes(Y, Z:dog), bite(Z,Y) }

As a formula, it represents

∃ Y:person ∃ Z:dog has_a_child(X:person,Y)∧ likes(Y,Z:dog)∧bite(Z,Y)

meaning "a person X who has a child bitten by a dog he/she likes".

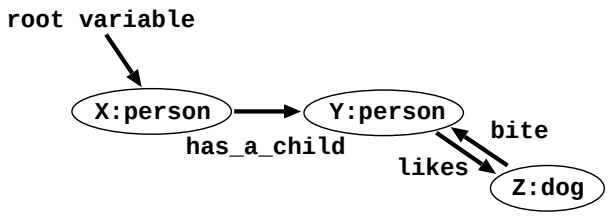


Figure 2: A Concept Graph

The denotation of a concept graph $CG(Y_0 : s_0) = CG(Y_0, Y_1 : s_1, ..., Y_k : s_k)$, given an interpretation $\mathcal{I} = \langle < A_s >_{s \in S}, \{p_{\mathcal{I}}\} \rangle$, is defined as the set of individual objects $b_0 \in A_{s_0}$ such that there exist individual objects $b_j \in A_{s_j}$ satisfying the property that every atom in $CG(Y_0 : s_0)$ becomes true under the variable assignment $\mathcal{I}(Y_j) = b_j$ $(0 \leq j \leq k)$.

Using our domain theory $\mathcal{T}$ as well as sort hierarchy $\mathcal{H}$, we can easily introduce an ordering between concept graphs. For instance, suppose we have a rule in $\mathcal{T}$

```
has_a_child(X:person,Y:male_person):-has_a_son(X,Y).
```

Then,

CG_2`(X:person)= { has_a_son(X, Y:male_person), likes(Y,Z:dog),`
`bite(Z,Y) }`

turns out to be a special case of CG_1. This is simply because

1. if `X:person` has a son `Y:male_person`, it follows from the rule that `X` has `Y` as his child and
2. `male_person` $\preceq$ `person`.

In such a case a more general case CG_1 is said to subsume a more special case CG_2. Formally, we have the following definition.

Definition 5 (Subsumption between Concept Graphs)
Let $CG_1(X_1 : s_1)$ and $CG_2(X_2 : s_2)$ be two concept graphs. We say that $CG_2(X_2 : s_2)$ subsumes $CG_1(X_1 : s_1)$ w.r.t. $\mathcal{KB} = \langle \mathcal{H}, \mathcal{T} \rangle$, denoted by $CG_1(X_1 : s_1) \preceq_{\mathcal{KB}} CG_2(X_2 : s_2)$, if

1. $s_1 \preceq s_2$ and
2. $\mathcal{T} \vdash_{\mathcal{H}} \forall X : s_1(\exists CG_1(X) \rightarrow \exists CG_2(X))$, where $\exists CG(X)$ stands for $\exists Y_1 ... \exists Y_m CG(X, Y_1, ..., Y_m)$.

■

It should be noted here that Definition 5 almost coincides with the notion of generalized subsumption [16], when we translate both rules in $\mathcal{T}$ and sort subsumptions $\mathcal{H}$ into non-sorted Horn rules. More precisely, $CG_1(X_1 : s_1) \preceq_{\mathcal{KB}} CG_2(X_2 : s_2)$ iff a clause $(p(X_2) : -s_2(X_2), CG_2^*)$ subsumes $(p(X_1) : -s_1(X_1), CG_1^*)$ under $\mathcal{KB}^*$, the first-order Horn theory translated from $\mathcal{KB}$, where $s(X)$ is a sortal predicate corresponding to sort s, and CG^* denotes CG followed by $s(Y)$ for every variable $Y : s$ in CG. Therefore, the following proposition can be viewed as a corollary of Theorem 4.2 in [16] after performing the translation. To save the space for describing the translation, we show a direct proof.

Proposition 1 *$CG_2(Z_2 : u_2) \preceq_{\mathcal{KB}} CG_1(Z_1 : u_1) \Leftrightarrow \mathcal{I}(CG_2) \subseteq \mathcal{I}(CG_1)$ holds for any model $\mathcal{I}$ of $\mathcal{T}$, where $\mathcal{I}(CG)$ is the denotation of CG, given an interpretation $\mathcal{I}$*

Proof. .
First, from the definition of subsumption, the left hand side directly implies the right hand side. Conversely, suppose an Herbrand interpretation $M(\mathcal{T})$ and let the variables in CG_2 be $Z_2 : u_2 = X_0 : s_0, X_1 : s_1, ..., X_m : s_m$. As we assume non-empty domain for each sort, we can introduce new constants a_j for each $X_j : s_j$ of the same sort, and make a ground atom $p(a_{j_1}, ..., a_{j_k})$ for each atom $p(X_{j_1}, ..., X_{j_k}) \in CG_2$ in addition to clauses in $\mathcal{T}$. This forms a logic program $P = \mathcal{T} \cup \{p(a_{j_1}, ..., a_{j_k}) | p(X_{j_1}, ..., X_{j_k}) \in CG_2\}$. From the definition of denotation of concept graph, we have $a_0 \in M(P)(CG_2)$, where $M(P)(CG_2)$ is the least Herbrand model of P. The model $M(P)$ is similarly constructed as in non-sorted case. That is, for

any ground atom α, $\alpha \in M(P)$ iff there exists a sequence $\beta_1, ..., \beta_\ell = \alpha$ such that, for any j, (1) β_j is either a fact in P or (2) there exists a set $\beta s \subseteq \{\beta_1, ..., \beta_{j-1}\}$ such that $\beta_j :- \beta s$ is a ground instance $C\theta$ of some rule $C \in P$. By the assumption $M(P)(CG_2) \subseteq M(P)(CG_1)$, a_0 is also a member of $M(P)(CG_1)$. As the root variable is typed by u_1, this occurs only when $u_2 \preceq u_1$. This establishes the first requirement of the definition of subsumption. Again by $M(P)(CG_2) \subseteq M(P)(CG_1)$, $\exists CG_1(a_0)$ is true under $M(P)$. This implies that there exists an SLD refutation of goal clause $:-CG_1(a_0)$ from P. This is equivalent to $\mathcal{T} \vdash \exists CG_2(a) \rightarrow \exists CG_1(a)$. Hence, by the theorem on constant, we have the conclusion. Q.E.D.

From the arguments in the proof, $CG_1(X_1 : s_1) \preceq_{\mathcal{KB}} CG_2(X_2 : s_2)$ holds if and only if the following two conditions (C1) and (C2) hold.

(C1) $s_1 \preceq s_2$.

(C2) Let τ be a Skolemization of variables in CG_1. Then,

$$\mathcal{T} \cup CG_1\tau \vdash_{\mathcal{H}} \exists Y_1 ... \exists Y_m \; CG_2(X_1\tau; Y_1 : s_1, ..., Y_m : s_m).$$

The condition (C2) is furthermore equivalent to (C3).

(C3) There exists a family $\mathcal{F}$ of ground instances $C\theta$ of rules $C \in \mathcal{T}$ such that

$\mathcal{F} \cup CG_1\tau \vdash CG_2(X_1\tau)\tau_2$, where τ_2 is a grounding substitution for variables Y_j in CG_2, and $\vdash$ is a derivation of propositional logic.

By the term proof of $CG_1 \preceq_{\mathcal{KB}} CG_2$, we refer to the deduction in the form of (C3), as it is the simplest form.

Let us exemplify the concept graph subsumption, using CG_{init}(`Ei:event`) given in Figure 6 and a concept graph

CG_{goal}(`Eg:event`) = {`dangerous(Eg), goa(Eg,Ps:public_space)`}.

CG_{init}(`Ei : event`)**:**
an event `Ei` in a park `P:park` such that there exist a car `C`, a human `H` and a tire `T` satisfying `obj(Ei:event, C:car), goa(Ei, P:park), in(H:human, P), in(C, P), has(C, T:tire)` and `maintained(T)`.

CG_{goal}(`Eg : event`)**:**
an dangerous event `Eg:event` in a public space `Ps:public_space`, i.e. `goa(Eg,Ps)`.

To check if CG_{init}(`Ei:event`) $\preceq_{\mathcal{KB}}$ CG_{goal}(`Eg:event`), the first task is to assign the same constant, `e` in this case, to the two root variables `Ei` and `Eg`. Then, each variable in CG_{init} except the root is replaced with the distinct (Skolem) constant of the same sort.

$$CG_{init}\tau_1 = \quad \texttt{obj(e, c:car), goa(e, p:park), in(h:human, p), in(c, p), has(c, t:tire), maintained(t).}$$

Then, the goal, $\exists$`Ps:public_space` `dangerous(e)`$\wedge$`goa(e,Ps)` will be proved by assigning the object `p` of the sort `park` to the variable `Ps` of more general sort `public_space`, using some rules and sort subsumptions, where we suppose `park` $\preceq$ `public_space` in addition to the hierarchy in Fig. 5. For example, the following rule in $\mathcal{T}$ coupled with its substitutions is applied:

$C7$: `mv(X:vehicle):-has(X,Y:rot_part), maintained(Y),`
θ_7 = {`X:vehicle/c:car,Y:rot_part/t:tire`},

The sort subsumptions car$\preceq$vehicle and tire$\preceq$rot_part are used to enable the atoms in $CG_{init}\tau_1$ to support mv(c).

Repeating similar processes, we get to have a family $\mathcal{F}$ of ground rules, including $C7\theta_7$, that satisfies

$$\mathcal{F} \cup CG_{init}\tau_1 \vdash CG_{goal}\tau_2 = (\texttt{dangerous(e)} \wedge \texttt{goa(e,p:park)}),$$

where $\tau_2 = \{\texttt{Ps:public_space/p:park}\}$.

Definition 6 (Subsumption Proof)
Given two CGs such that $CG_1(Z_0; Z_1, ..., Z_k) \preceq_{\mathcal{KB}} CG_2(W_0; W_1, ..., W_m)$, a proof of $CG_1 \preceq_{\mathcal{KB}} CG_2$ is a triple $< \tau_1, \tau_2, \mathcal{F} >$ such that

(1) τ_1 is a Skolemization for variables $Z_0, ..., Z_k$ in CG_1. Particularly, Z_j and $Z_j\tau_1$ have the same sort.

(2) τ_2 is a grounding substitution for variables $W_0, ..., W_m$ in CG_2.

(3) $Z_0\tau_1 = W_0\tau_2$. That is, the root variables are instantiated to the same constant. Note that, by the order-sortedness of τ_2 and (1), the condition (C1) is satisfied whenever we have $Z_0\tau_1 = W_0\tau_2$.

(4) $\mathcal{F}$ is a family of ground instances $C\theta$ of rules $C \in \mathcal{T}$.

(5) $\mathcal{F} \cup CG_1\tau_1 \vdash CG_2\tau_2$.

■

The subsumption proof is depicted by an acyclic directed graph, as in Fig. 6, where we draw an edge $(C, subs(\theta))$ whenever $C\theta \in \mathcal{F}$, where for $\theta = \{X_1 : s_1/a_1 : u_1, ..., X_m : s_m/a_m : u_m\}$, $subs(\theta) = \{u_j \preceq s_j | 1 \leq j \leq m\}$.

4 Abstractable Concept Graph

In this section, we introduce a notion of goal-dependent abstraction for concept graphs and their subsumptions, and define a class of abstractable concept graphs. In the previous studies [5, 7], we have already proposed a notion of *Goal-Dependent Abstraction* (GDA) for rules based on a simple intuition that

> given a goal formula to be proved, only an abstraction that preserves every necessary rule needed to prove the goal should be regarded,

where the preservingness actually means that a rule, $\psi(E)$, after the abstraction is still valid whenever the original E is valid. Accordingly, any abstraction not satisfying this criterion is judged irrelevant w.r.t. the goal. The notions of goals and their proofs in the original GDA are now extended to a goal concept graphs and its subsumption proofs, respectively. Therefore, the GDA for concept graphs is stated as follows:

> Given a goal concept graph subsuming more particular concept graph as a typical instance, only an abstraction, that makes every necessary rule and sort subsumption needed in the subsumption proof abstractable, should be considered.

As the subsumption among concept graphs is viewed as a kind of macro rule in EBG [19] under a domain theory, the abstractableness actually means a

substitutability of macro rule w.r.t. a replacement of sort symbols (Definition 7). Consequently, every concept graph thus obtained by the replacement shares the almost same macro rule connecting the goal to them. Thus we consider that those concept graphs represent potentially similar situations to the typical instance of the goal concept graph.

As an example, suppose we have a person looking for "a dangerous thing in a park just like a car". Normally, the property of dangerousness will be specified by using some attributes or predicates. In addition, it should be similar to car from the viewpoint of dangerousness. As we know various dangerous things because of their *own particular reasons*, we cannot consider that things are similar to car simply because they share the property of dangerousness. Similar things must be dangerous in *the same reason* why car is dangerous in a park. In other words, we regard the similarity is situationally valid only when we can explain dangerous aspect of things just as we can do for car in a park. In the definition below, two concept graphs CG_{goal} and CG_{init} are used to specify such a similarity situation. The former is called a goal and is used to specify what aspect or property of things we are concerned with. The latter shows a particular situation from which we can draw an explanation, the proof structure of goal from the situation. Then we try to find a goal-dependent abstraction, a grouping of sorts by which any used rules and sort subsumptions in the proof structure of a goal becomes abstractable.

4.1 Goal-Dependent Abstraction for Concept Graphs

We firstly define a notion of abstractableness of concept graph, given a goal and its subsumption proof.

Definition 7 (GDA in terms of Subsumption Proof)
A grouping ψ of sorts is said to be a GDA-grouping w.r.t. a subsumption proof $< \tau_1, \tau_2, \mathcal{F} >$ of $CG_{init}(Z_0 : t_0; Z_1 : t_1, ..., Z_k : t_k) \preceq_{\mathcal{KB}} CG_{goal}(W_0 : s_0; W_1, ..., W_m)$, if the following conditions hold:

(1) For any $C\theta \in \mathcal{F}$, both C and each sort subsumption in $subs(\theta)$ are abstractable w.r.t. ψ,
where $subs(\{X_1 : s_1/a_1 : u_1, ..., X_m : s_m/a_m : u_m\}) = \{u_j \preceq s_j | 1 \leq j \leq m\}$.

(2) Any sort subsumption in $subs(\tau_2)$ is abstractable w.r.t. ψ.

(3) For any sort s in CG_{goal} and its ψ-analogue s', a member of the same group as s, $s' \preceq s$ holds.

In this case, we also say that CG_{init} is *abstractable* w.r.t. $< \tau_1, \tau_2, \mathcal{F} >$ and ψ. Moreover, a ψ-analogue of CG_{init} is defined as a concept graph $CG_{ana} = CG_{init}(Y_0 : u_0, ..., Y_k : u_k)$, where $u_0 \preceq s_0$, and u_j is an ψ-analogue of t_j w.r.t. ψ for each j. ■

The conditions (1) and (2) require that both rules and subsumptions to prove $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$ should be abstractable. We need the condition (3) to guarantee for the analogue CG_{ana} of CG_{init} to be also subsumed by the same goal concept graph CG_{goal}, as we explain just later.

The following proposition is a basis for transforming an initial concept graph CG_{init} to its analogue, CG_{sim}.

Proposition 2 *Suppose we have a subsumption proof* $< \tau_1, \tau_2, \mathcal{F} >$, *a GDA grouping* ψ *w.r.t. the proof, and* $R\theta \in \mathcal{F}$. *Then, for any analogue* R'

of $var_abs(R\theta)$ w.r.t. ψ, we have $\mathcal{T} \vdash_{\mathcal{H}} R'$, where $var_abs(R\theta)$ is a rule obtained by replacing every constant $a_j : t_j$ in $R\theta$ by a new variable $Z_j : t_j$ of the same sort.

Proof. Suppose that $R = R(X_1 : s_1, ..., X_n : s_n)$ is instantiated to $R_1 = apply(R, subs(R)) = R(a_1 : t_1, ..., a_n : t_n)$ and that $\psi(t_j) = \psi(u_j)$ for each j. By the variable abstraction var_abs, we obtain $R_2 = var_abs(R_1) = R(Z_1 : t_1, ..., Z_n : t_n)$, and consider its arbitrary analogue $R' = R(Y_1 : u_1, ..., Y_n : u_n)$ of $var_abs(R_1)$. As $t_j \preceq s_j \in subs(R)$, $u_j \preceq v_j$ holds for some v_j such that $\psi(s_j) = \psi(v_j)$. Moreover the abstractableness of R implies that $\mathcal{T} \vdash_{\mathcal{H}} R(W_1 : v_1, ..., W_n : v_n)$. Hence, $\mathcal{T} \vdash_{\mathcal{H}} R(Y_1 : u_1, ..., Y_n : u_n)$ holds by the substitution $\{W_1/Y_1, ..., W_n/Y_n\}$. Q.E.D.

In the case of Fig.6, the proof structure involves ground rules in $\mathcal{F}$, for instance:

```
mv(c:car):-has(c,t:tire), maintained(t),
receive_damage_mv_obj(h:human,c):- mv(c), bigger_than(c,h) and
bigger_than(c,h).
```

From Proposition 2,

```
mv(T:tank):-has(T,V:caterpillar), maintained(V),
receive_damage_mv_obj(P:pet,T:tank):-mv(T), bigger_than(T,P)
and bigger_than(T:tank,P:pet)
```

are also valid rules under our domain theory w.r.t. a GDA grouping

ψ={{car, tank,...}, {tire,caterpillar,...}, {human, pet},...}.

So, we can consider various ψ-analogue, CG_{ana}, obtained by replacing a sort s of variable X with an analogue sort $s' \in \psi^{-1}(\psi(s))$. For example, in the case of CG_{init} in Fig.6, CG_{ana} for `tank-pet` case can contain the following facts:

$CG_{\texttt{tank-pet}}$(SE:event) =
  { obj(SE,T0:tank), has(T0,C0:caterpillar), maintained(C0),
  goal(SE,P:park),in(P,Pet:pet),...}.
τ_3 = {SE/se,T0/t0,C0/c0,Pet/p0...}, a Skolemization for
  $CG_{\texttt{tank-pet}}$.

Then `receive_damage_mv_obj(t0, p0)` can be supported by facts in $CG_{\texttt{tank-pet}}\tau_3$, using the above rules based on Proposition 2. Repeating similar process, we get to have $\mathcal{F}'$ such that $\mathcal{F}' \cup CG_{\texttt{tank-pet}}\tau_3 \vdash$ `dangerous(se)`, the same goal as CG_{init}.

We summarize the above argument in a form of theorem, and give a fundamental criterion for selecting similar CG, given a goal CG and its subsumption proof.

Theorem 1 *Suppose we have $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$ and a GDA grouping ψ w.r.t. its proof $< \tau_1, \tau_2, \mathcal{F} >$. Then, for any ψ-analogue CG_{ana} of CG_{init}, $CG_{ana} \preceq_{\mathcal{KB}} CG_{goal}$ holds.*

Proof. Let CG_{goal}, CG_{init} and its ψ-analogue CG_{ana} be $CG_{goal}(Z_0 : s_0; Z_1 : s_1, ..., Z_m : s_m)$ $CG_{init}(X_0 : u_0; X_1 : u_1, ..., X_k : u_k)$ and $CG_{ana} = CG_{init}(V_0 :$

$v_0; V_1 : v_1, ..., V_k : v_k)$, respectively. This means that $\psi(u_j) = \psi(v_j)$ for all j. From the assumption,

$$\mathcal{F} \cup CG_{init}(x : u_0; x_1 : u_1, ..., x_k : u_k) \vdash CG_{goal}(x, b_1 : w_1, ..., b_m : w_m)$$
$$u_0 \preceq s_0, \text{ and } w_j \preceq s_j \text{ for all } j.$$

Without loss of generality, we suppose that $b_1, ..., b_\ell$ correspond to some x_j, and $b_{\ell+1}, ..., b_m$ do not. Then, we have

$$\mathcal{F} \cup CG_{init}(x; x_1, ..., x_k) \vdash CG_{goal}(x, x_{i_1} : u_{i_1}, ..., x_{i_\ell} : u_{i_\ell}, b_{\ell+1}, ..., b_m)$$
$$u_0 \preceq s_0, u_{i_j} \preceq s_j \ (1 \leq j \leq \ell) \text{ and } w_j \preceq s_j \ (\ell \leq j \leq m).$$

According to Proposition 2, it is easily verified that there exists $\mathcal{F}'$ of ground instances of rules in $\mathcal{T}$ such that

$$\mathcal{F}' \cup CG_{init}(x' : v_0; x'_1 : v_1, ..., x'_k : v_k) \vdash$$
$$CG_{goal}(x', x'_{i_1} : v_{i_1}, ..., x'_{i_\ell} : v_{i_\ell}, b_{\ell+1}, ..., b_m).$$

As s_0 appears in CG_{goal}, $\psi(v_0) = \psi(u_0)$ implies $v_0 \preceq s_0$ by the condition (3) in Definition 7. Similarly, we have $v_{i_j} \preceq s_j \ \ (1 \leq j \leq \ell)$. Thus, letting

$$\tau_3 = \{V_0/x', V_1/x'_1, ..., V_k/x'_k\},$$
$$\tau_4 = \{Z_0/x', Z_1/x'_{i_1}, ..., Z_\ell/x'_{i_\ell}, Z_{\ell+1}/b_{\ell+1}, .., Z_m/b_m\},$$

τ_4 is an order-sorted substitution, and $\mathcal{F}' \cup CG_{ana}\tau_3 \vdash CG_{goal}\tau_4$. This completes the proof. Q.E.D.

Now we are ready to define a notion of potential instances of goal concept graph which are similar to initial concept graph.

Definition 8 (Potential Instance/ Similarity)
We say that a concept graph CG is a *potential instance* of CG_{goal} for a GDA grouping ψ w.r.t. a subsumption proof of $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$, if CG is subsumed by some ψ-analogue of CG_{init}. We also say that CG is similar to CG_{init}. ■

In a word, an analogous concept graph keeps almost similar proof, and the similar concept graph is a specialization of some analogue of the initial concept graph. Some concrete examples are shown in Section 5. Moreover, from the viewpoint of information retrieval, every concept graph similar to the initial graph CG_{init} w.r.t. some GDA ψ for the goal CG_{goal}, is understood as a relevant concept graph to the query specified by the pair $\langle CG_{goal}, CG_{init} \rangle$.

We sometime have only *trivial* abstraction as an GDA. Under the GDA, a concept graph similar to CG_{init} is only CG_{init} itself. Furthermore, we might often consider that possible GDAs are not interesting for us even if they are not trivial. In such cases, users would strongly desire to have more interesting ones. It is possible to find another GDAs by *generalizing* concept graphs.

4.2 Generalization

The generalization of concept graphs, given $\langle CG_{goal}, CG_{init} \rangle$, is simply defined as a "contraction process" from leaves to root in our proof structure. Another possible method is to go down the proof structure from its root to

leaves. However, we prefer a bottom up approach by some reason shown in the next section.

Suppose we have a proof structure $< \tau_1, \tau_2, \mathcal{F} >$ of $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$. we cut some ground atoms $\{\gamma_1, ..., \gamma_m\}$ in $CG_{init}\tau$ such that there exists a rule $C\theta = (\alpha :- \beta_1, ..., \beta_k) \in \mathcal{F}$ satisfying

(1) $\{\gamma_1, ..., \gamma_m\} \subseteq \{\beta_1, ..., \beta_k\}$,

(2) the remaining atom $\zeta \in \{\beta_1, ..., \beta_k\} - \{\gamma_1, ..., \gamma_m\}$ is an instance of a unit clause in $\mathcal{T}$.

Then, it is clear that $< \tau_1', \tau_2, \mathcal{F} - \{C\theta\} >$ is a proof structure of $(CG_{init} - \{var_abs(\gamma_j) | j = 1, ..., m\}$, where τ_1' might introduce X/x for some constant that newly appears in α. in case of our rules are not assumed range-restricted.

We show examples of generalization process in the next section. With the help of generalizations, we can obtain some interesting GDAs which can never be found without generalizing initial concept graphs.

5 Experimentation

In this section, we show our preliminary experimental results obtained by our current system [*] .

We have implemented the main module for finding GDA groupings among sort concepts. Let $\mathcal{KB}$ be a domain theory, and CG_{init} and CG_{goal} be an initial and goal concept graphs, respectively. Assume we have a proof structure of $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$. Given the proof structure, our system tries to find GDA ψ_i for CG_{goal} from CG_{init}, that is, CG_{init} is an abstractable concept graph for CG_{goal} w.r.t. ψ_i.

Before showing our experimental results, we present an overview of the computation process in our system.

5.1 Overview of Our System:

Given the proof structure as input, our system finds all *maximal* GDA ψ_i for CG_{goal} with a generate-and-test strategy. A candidate grouping is generated and then examined whether it is a GDA for or not. In order to systematically generate candidates, we introduce an ordering on candidate groupings defined as follows.

Definition 9 (Ordering on Groupings)
Let $\psi = \{g_1, \ldots g_k\}$ and $\psi' = \{g_1', \ldots g_l'\}$ be two groupings of a sort set S. Then ψ precedes ψ' (denoted by $\psi \preceq \psi'$) if for each g_i in ψ, there exists some g_j' in ψ' such that $g_i \subseteq g_j'$. ■

Given a set of sorts S, the possible candidate groupings of S forms a lattice under the ordering: the minimum (that is, trivial) grouping is placed at the bottom and the maximum grouping at the top. For example, Figure 3 shows the whole candidate lattice for a set of sorts, $S = \{\texttt{human}, \texttt{horse}, \texttt{car}, \texttt{tank}\}$.

For each candidate ψ_i, we examine whether all labels $(R_i, subs(R_i))$ in the proof structure are abstractable under ψ_i or not. If all of them are abstractable under ψ_i, the grouping is a GDA and is said to be *appropriate*.

[*] The system has been implemented in SICStus Prolog and run on a UltraSPARC Workstation(UltraSPARC II-i 440MHz).

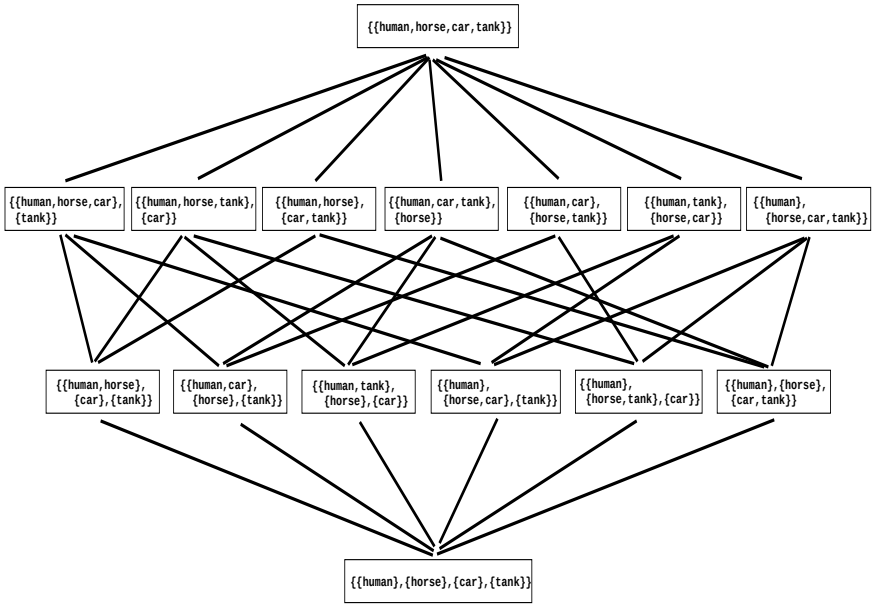


Figure 3: Candidate Grouping Lattice

Since we are interested in maximal GDAs, one may consider that candidate generation should be performed in a top-down manner. However, our current system generates candidate groupings in a bottom-up manner. There are two reasons why we do so. The first one is that we can *empirically* expect that appropriate groupings would be found, if anything, at lower positions in the candidate lattice. Since it is merely empirical, there might exist some cases in which top-down generations would be adequate for our purpose. However, we have not seen such a case yet. The second reason why we adopts a bottom-up method for candidate generation is that based on a property w.r.t. the appropriateness of candidate groupings, we can prune useless candidates in the upper direction in the lattice. More precisely speaking, the soundness of the pruning is ensured based on the following observation:

> For a candidate grouping ψ, if the rule R of a label $(R, subs(R))$ in the proof structure is not abstractable under ψ, then any grouping ψ' such that $\psi \preceq \psi'$ is not appropriate.

That is, once we found a grouping ψ is not appropriate, we do not have to generate any candidates succeeding to ϕ. Our bottom-up generation process is carried out with this pruning strategy. Our candidate pruning for the previous example is shown in Figure 4. If a candidate grouping $\{\{\texttt{human}, \texttt{car}\}, \{\texttt{horse}\}, \{\texttt{tank}\}\}$ with "*" is found to be not appropriate, then the shaded candidates are never generated.

Moreover, we can obtain further pruning from another viewpoint. The candidate lattice contains a large number of needless ones in the following sense. For an initial concept graph CG_{init}, we are interested in a GDA under which we can obtain some concept graphs that are analogous to CG_{init}. Therefore, any candidate grouping φ such that

> for each sort s occurring in CG_{init}, $\{s\} \in \varphi$

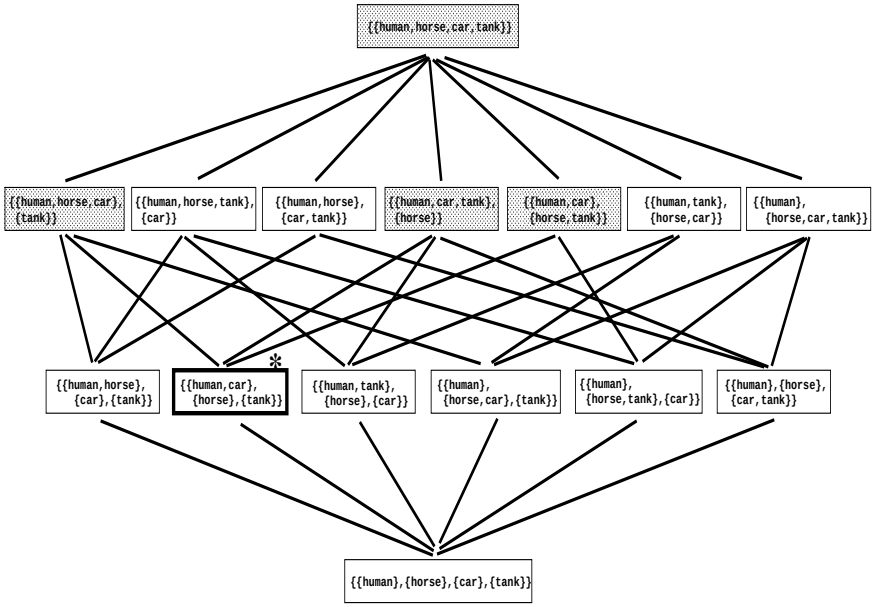


Figure 4: Candidate Pruning

can be considered *trivial*, since such a φ gives CG_{init} itself as only one concept graph that is analogue of CG_{init}. These needless candidates are never generated in our generation process. The number of candidates that our system would generate in the worst case is denoted by "Num. of Possible Cands." in the experimental results we will show later. Moreover, the numbers of candidates that are *actually* generated and examined are denoted by "Num. of Generated Cands." and "Num. of Examined Cands.", respectively.

Remarks:

It is expected that many sort concepts appear in a realistic domain theory. As stated above, since our whole search space for finding GDAs consists of the possible groupings of the set of sorts, further improvement of computational efficiency would be required in such a case. In addition to the above pruning methods, we can improve the efficiency with the following two constraints imposed on candidate groupings. These constraints are extracted based on a sort hierarchy and conceptual structure of sort we are concerned with.

Constraint based on Similarity Inheritance Condition (SIC) [5]:

Let us consider two concepts C and C' such that C is a subconcept of C'. In general, C inherits all characters or properties of C'. Therefore, if we consider a concept $\tilde{C}$ is similar to C', it would be quite reasonable to consider that $\tilde{C}$ is also similar to C. In this sense, a similarity among $\tilde{C}$, C and C' can be considered meaningful and we say that such a similarity satisfies *Similarity Inheritance Condition* (SIC).

Our candidate groupings to be generated can be restricted to ones satisfying SIC, keeping this kind of meaningfulness. For example, suppose we have a sort hierarchy showing that a sort `vehicle` sub-

sumes a sort `car`. If we consider a sort `horse` is similar to `vehicle`, then `horse` should be similar to `car` as well. Therefore, a grouping $\{\{\texttt{horse}, \texttt{vehicle}\}, \{\texttt{car}\}\}$ is excluded from candidates to be generated. Thus, by taking only candidates satisfying SIC into account, we can further reduce our computational cost of candidate generation.

Constraint based on Value Restriction Preservingness (VRP) [6]: As the literature on analogical reasoning (e.g. [22, 23]) has claimed, analogy should be performed based on *structural similarities* of concepts. By taking such a structural viewpoint into account, therefore, our similarities (that is, groupings of sorts) would become more meaningful. Furthermore, since it can work as a constraint on groupings to be considered, the efficiency of our candidate generation will be improved. Such a structural similarities can be discussed in terms of *value restrictions* on roles of sort concepts.

A value restriction on role is deeply concerned with structure of a sort concept, where a role is a binary relation between individuals. Let r be a binary relation that is intended to represent some role. We call r a *role predicate symbol* (or simply role). Furthermore, the second argument of r is called the *role-filler* for r, referred to as r-filler. A value restriction on r is used to restrict the range of r-filler to individuals of some sort. We denote a value restriction on r by $r : s \rightarrow s'$, where s and s' are sorts. Its meaning is that r-filler of s should be filled with an individual of s'. In other words, r-filler of the sort concept denoted by s should always be the concept denoted by s'. In this sense, the value restriction can be considered to define *structure of the concept* s. Therefore, according to the above claim of structural similarities, we can obtain a requirement, called *Value Restriction Preservingness* (VRP), that for value restrictions $r : s_1 \rightarrow s'_1$ and $r : s_2 \rightarrow s'_2$, if we consider s_1 and s_2 are similar, then s'_1 and s'_2 have to be also similar. It should be noted here that value restrictions taken into consideration should be relevant to our goal, since we try to find similarities depending on the goal [†] .

As candidate groupings in our search space, we can consider only candidates satisfying VRP. For example, suppose we have value restrictions

$\texttt{locomotive_part} : \texttt{horse} \rightarrow \texttt{leg}$ and
$\texttt{locomotive_part} : \texttt{vehicle} \rightarrow \texttt{rotational_part}$.

If `horse` and `vehicle` are similar, then their `locomotive_part`-fillers `leg` and `rotational_part` should be also similar. Therefore, a grouping $\{\{\texttt{horse}, \texttt{vehicle}\}, \{\texttt{leg}\}, \{\texttt{rotational_part}\}\}$ is excluded from candidates to be generated.

Although our current system does not yet adopt the pruning methods based on the constraints, it is highly expected that these pruning methods will drastically reduce the computational cost of finding GDAs. More detailed discussion about the constraints based on SIC and VRP can be found in the literature [5, 6], as well as some experimental results which show their effectiveness in efficiently finding similarities.

[†] For an initial concept graph CG_{init} and a goal CG_{goal}, if a role r appears in a proof structure of $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$, a value restriction on r is considered relevant to the goal. In this paper, although value restrictions are not included in our domain theory, we can introduce adequate restrictions into the theory without any difficulty.

5.2 Experimental Results

As discussed in the previous section, there often exists a case where the trivial (minimum) grouping is the only appropriate one. Furthermore, it might exist a case where obtained appropriate groupings might not be interesting to users. In such cases, we can try to generalize the initial concept graph to find another appropriate ones. Since we have some possible generalizations of the graph, a certain preferential order on possible ones would be necessary. In our system, we prefer a generalization which is obtained by removing less number of edges and for which a non-trivial grouping can be obtained as an appropriate one.

For our preliminary experimentations, two examples have been provided. We show the results for the examples and discuss them below.

Example1: Dangerous Event

Let us consider a domain theory $\mathcal{KB}_D$ and an initial concept graph $\mathtt{CG_{init1}}$ shown in Figure 5. In the figure, any variable with which no sort is associated is assumed to be typed with a sort `top` subsuming any sort. Since it is clear, any sort subsumption of the form $\mathtt{s_i} =< \mathtt{top}$ is omitted.

$\mathcal{KB}_D$ is a domain theory concerning a dangerous event. For example, a rule

$$\mathtt{mv(X : vehicle)} : -\mathtt{has(X, Y : rot_part), maintained(Y)}.$$

states that if a vehicle has rotational part that is maintained, then the vehicle is movable.

Suppose we have a goal

$$CG_{goal1}(\mathtt{E : event}) = \{\mathtt{dangerous(E)}\}.$$

It is easily verified that CG_{init1} is subsumed by $CG_{goal1}(\mathtt{E : event})$ under the domain theory. The proof structure of the subsumption is shown in Figure 6.

Given the set of labels $(\mathtt{C_i}, \mathit{subs}(\mathtt{C_i}))$ in the proof structure as input, our system tries to find maximal appropriate groupings among sort symbols, where $\mathtt{C_i}$ is a rule and $\mathit{subs}(\mathtt{C_i})$ is the set of sort subsumptions used to apply the rule in the proof. The obtained results are shown in Figure 7.

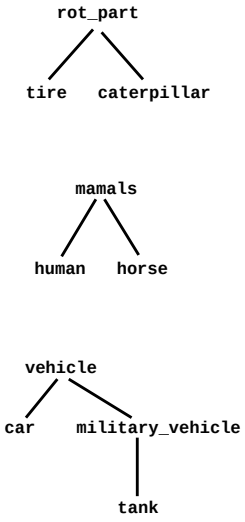
Since there are 14 sort symbols in the example, the number of possible candidates (including needless ones) is 190,899,322 ($\approx 2 \times 10^8$). As mentioned above, our system would generate approximately 4×10^7 of them in the worst case by rejecting needless candidates.

In order to find all maximal grouping, only 9628 candidates have actually been generated and 8921 of them examined their appropriateness. It should be emphasized here that 99.98% of possible candidates have been pruned with our pruning strategy for candidate generation. Therefore, it is considered that our strategy can work very well to improve efficiency of finding desired groupings.

There are 3 groups of similar sorts in the appropriate grouping:

$\{\mathtt{car}, \mathtt{tank}, \mathtt{military_vehicle}, \mathtt{vehicle}\}$,
$\{\mathtt{tire}, \mathtt{caterp}, \mathtt{rot_part}\}$ and $\{\mathtt{human}, \mathtt{pet}\}$.

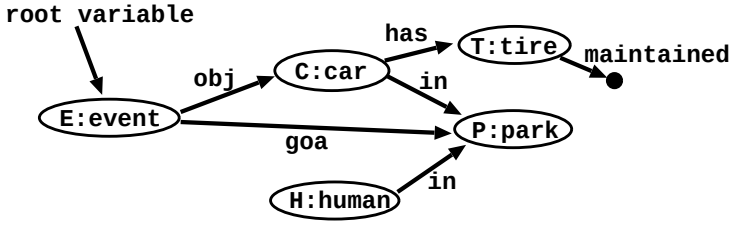
Note that the first two groups are implied from the sort subsumptions in the domain theory as well as from the result. Therefore, one might prefer more interesting groupings. In order to obtain such a grouping, we try to generalize the original concept graph CG_{init1} with the label



Sort Hierarchy

```
C1: receive_damage_mv_obj(X:pet,Y):-
       mv(Y), bigger_than(Y,X).

C2: receive_damage_mv_obj(X:human,Y):-
       mv(Y), bigger_than(Y,X).

C3: bigger_than(X:vehicle,Y:human).

C4: bigger_than(X:vehicle,Y:pet).

C5: bigger_than(X:horse,Y:human).

C6: bigger_than(X:horse,Y:pet).

C7: mv(X:vehicle):-
       has(X,Y:rot_part), maintained(Y).

C8: mv(X:horse):-
       has(X,Y:leg), maintained(Y).

C9: dangerous(E:event):-
       object(E,X), goa(E,Y:park),
       in(X,Y), bigger_than(X,Z), in(Z,Y),
       to_be_protected_against_mv_obj(Z,X)

C10: to_be_protected_against_mv_obj(X,Y):-
       mv(Y), receive_damage_mv_obj(X,Y).
```

Rule Set



Concept Graph

Figure 5: Domain Theory and Concept Graph Concerning Dangerous Event

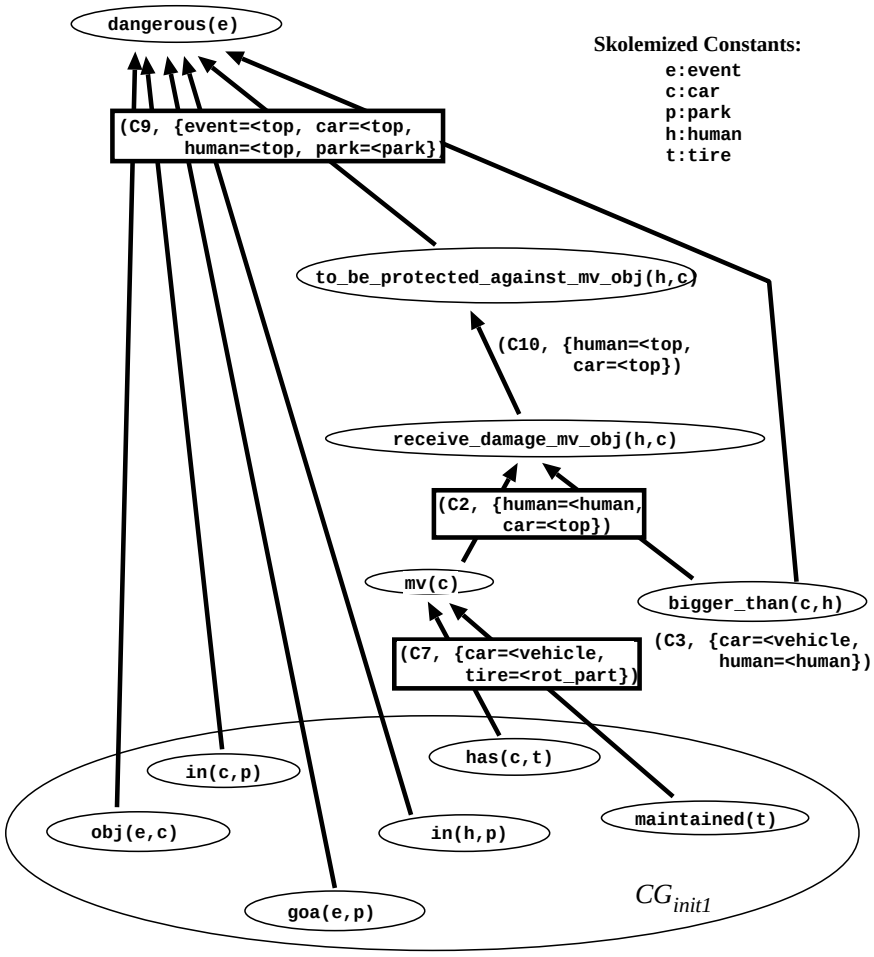


Figure 6: Proof Structure of $CG_{init1} \preceq CG_{goal}(\texttt{E} : \texttt{event})$

Num. of Possible Cands	36,458,303 ($\approx 4 \times 10^7$)
Num. of Generated Cands	9628
Num. of Examined Cands	8921
Execution Time	401 sec.
Maximal Appropriate Grouping	{{ car, tank, military_vehicle, vehicle }, { tire, caterp, rot_part }, { human, pet }, { park }, { event }}

Figure 7: Experimental Results

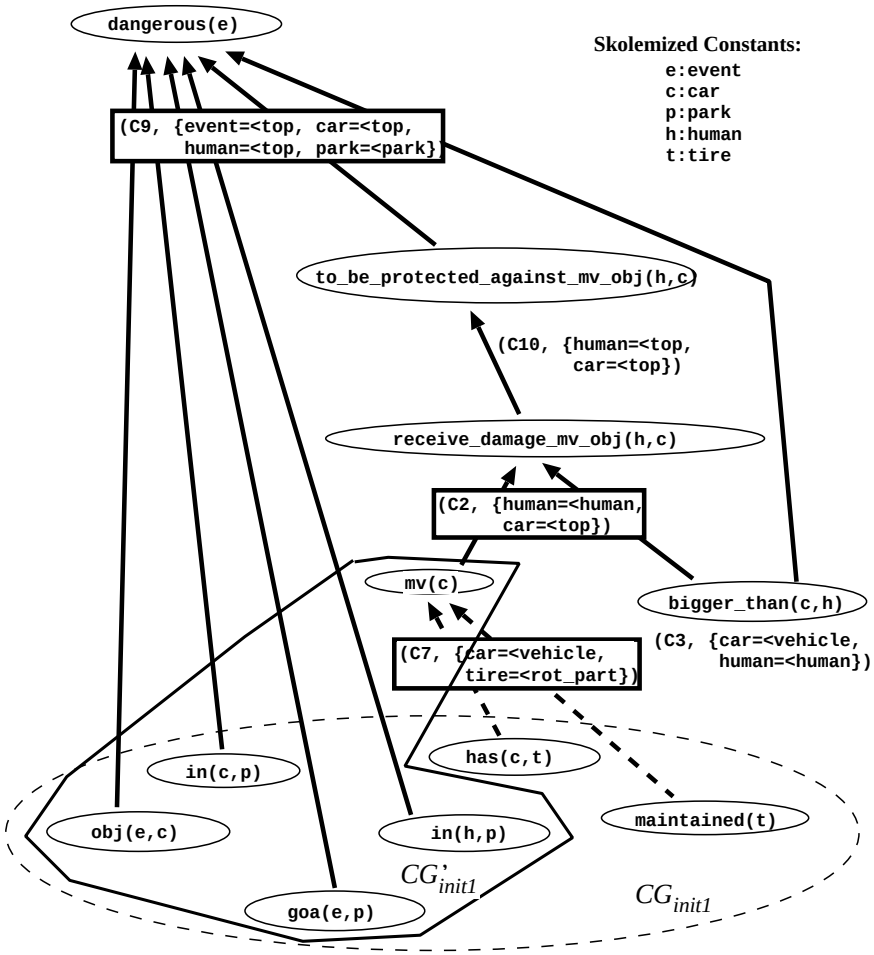


Figure 8: Proof Structure of $CG'_{init1} \preceq CG_{goal1}(\mathtt{E} : \mathtt{event})$

$$(\mathtt{C7} : \mathtt{mv}(\mathtt{X} : \mathtt{vehicle}) : -\mathtt{has}(\mathtt{X}, \mathtt{Y} : \mathtt{rot_part}), \mathtt{maintained}(\mathtt{Y}),$$
$$\{\mathtt{car} =< \mathtt{vehicle}, \mathtt{tire} =< \mathtt{rot_part}\})$$

in the proof structure. Figure 8 shows the generalized concept graph CG'_{init1} and the proof structure of $CG'_{init1} \preceq CG_{goal1}(\mathtt{E} : \mathtt{event})$

After the generalization, by taking the labels in the proof structure as input, our system can find a more interesting appropriate grouping shown in Figure 9.

The obtained grouping shows that we can consider `car`, `tank`, `military_vehicle`, `vehicle` and `horse` to be analogous. It should be emphasized that it is not implied from the given domain theory. Thus, we can find a more interesting grouping by taking a generalization of CG_{init1} into account. Furthermore our pruning strategy for candidate generation has still worked very well to reduce the computational cost.

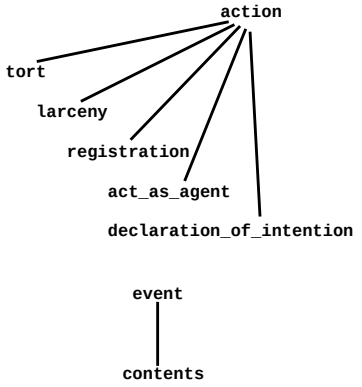
Example2: Priority on Legal Event

Let us consider a legal domain theory $\mathcal{KB}_P$ and a concept graph CG_{init2} shown in Figure 10.

$\mathcal{KB}_P$ is a domain theory concerning priority on legal events. For example, a rule $C2$ states that if a registration declares an event and the real

Num. of Possible Cands	16,471,448
Num. of Generated Cands	3104
Num. of Examined Cands	2649
Execution Time	104 sec.
Maximal Appropriate Grouping	{{ car, tank, military_vehicle, vehicle, horse }, { human, pet }, { park }, { event }}

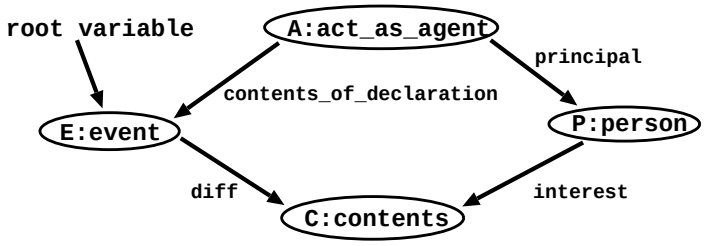
Figure 9: Experimental Results after Generalization



Sort Hierarchy

```
C1: prior(E1:event):-
       contents_of_declaration(D:decl_of_int,E1),
       real_intention(D,E2:event), diff(E1,E2).

C2: prior(E1:event):-
       contents_of_declaration(D:registration,E1)
       real_intention(D,E2:event), diff(E1,E2).

C3: prior(E1:event):-
       contents_of_declaration(D:act_as_agent,E1)
       real_intention(D,E2:event), diff(E1,E2).

C4: real_intention(A:act_as_agent,C:contents):-
       principal(A,P:person), interest(P,C).
```

Rule Set



Concept Graph

Figure 10: Domain Theory and Concept Graph Concerning Priority on Legal Event

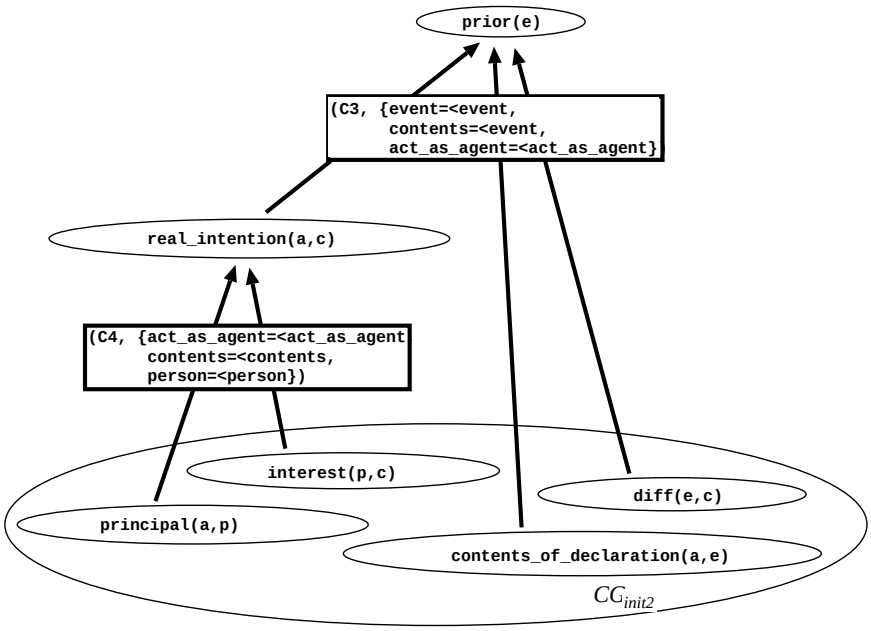


Figure 11: Proof Structure of $CG_{init2} \preceq CG_{goal2}(\mathtt{E} : \mathtt{event})$

intention of the registration is contrarily to declare a different event, then the registration should be prior to the real intention.

Suppose we have a goal

$$CG_{goal2}(\mathtt{E} : \mathtt{event}) = \{\mathtt{prior}(\mathtt{E})\}.$$

It is easily verified that CG_{init2} is subsumed by $CG_{goal2}(\mathtt{E} : \mathtt{event})$ under the domain theory. The proof structure of the subsumption is shown in Figure 11.

Given the set of labels $(C_i, subs(C_i))$ in the proof structure as input, our system can find only one maximal appropriate grouping. Unfortunately, it is a trivial grouping. We therefore try to generalize the original concept graph CG_{init2} with the label

$$(\mathtt{C4} : \mathtt{real_intention}(\mathtt{X} : \mathtt{act_as_agent}, \mathtt{Y} : \mathtt{contents}) : - \\ \mathtt{principal}(\mathtt{X}, \mathtt{Z} : \mathtt{person}), \mathtt{interest}(\mathtt{Z}, \mathtt{Y}), \\ \{\mathtt{act_as_agent} =< \mathtt{act_as_agent}, \\ \mathtt{contents} =< \mathtt{contents}, \mathtt{person} =< \mathtt{person}\}).$$

Figure 12 shows the generalized concept graph CG'_{init2} and the proof structure $CG'_{init2} \preceq CG_{goal2}(\mathtt{E} : \mathtt{event})$

After the generalization, our system can find another grouping as an appropriate one. The obtained grouping is shown in Figure 13.

As shown in the results, we can consider that `act_as_agent`, `registration` and `declaration_of_intention` to be similar, even though any two sorts are not similar without generalization. Thus an interesting grouping can be found with the help of generalization.

Furthermore, the number of actually generated candidates is only 13% of possible ones. It is, therefore, verified that our pruning strategy of candidate generation can still work very effectively in this example.

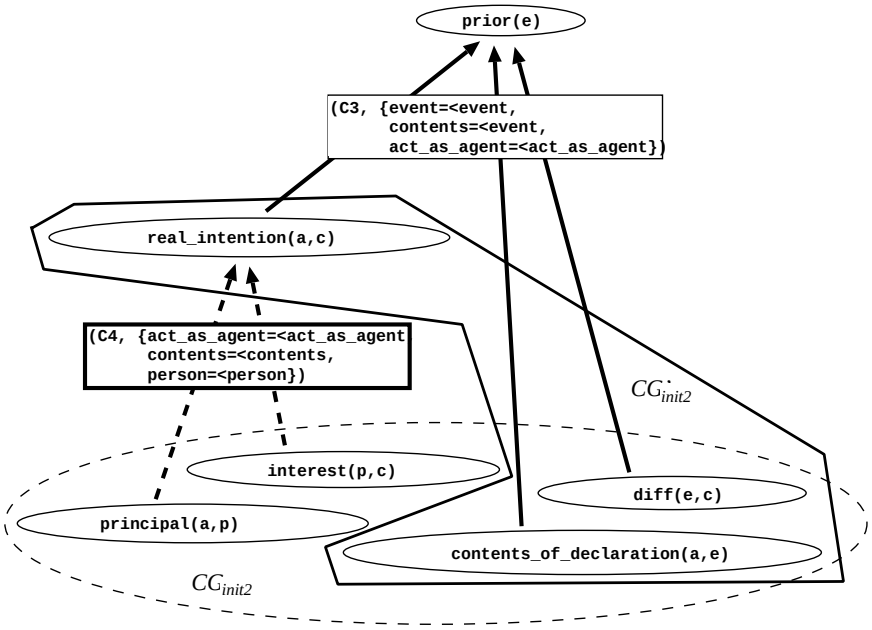


Figure 12: Proof Structure of $CG'_{init2} \preceq CG_{goal2}(\mathtt{E} : \mathtt{event})$

Num. of Possible Cands.	1785
Num. of Generated Cands.	232
Num. of Examined Cands.	160
Execution Time	593 msec.
Max. Appropriate Grouping	{{act_as_agent, registration, declaration_of_intention }, { contents, event }, { person }}

Figure 13: Experimental Results after Generalization

6 Discussion

According to the present version of this research, we do not use any negative information, as the underlying logic is order-sorted Horn logic. However, the use of negative information such as disjoint constraints [18] is useful to obtain richer and meaningful class of goal-dependent abstractions. For instance, suppose the following rule is used in a concept graph subsumption $CG_{init} \preceq_{\mathcal{KB}} CG_{goal}$:

```
movable(X:car):-has(X,Y:tire), maintained(Y).
```

When we examine a similarity between car and horse, the present method has to check the abstractableness for a grouping such as

$$\psi = \{\{\texttt{car, horse, ...}\}, \{\texttt{tire, leg, ...}\}, \texttt{...}\},$$

for instance. Then, an analogous rule

```
movable(X:horse):-has(X,Y:tire), maintained(Y)
```

is tested for its validness. However, it is not provable under a standard theory, so the grouping ψ is rejected. It is meaningless not to allow ψ because of such a reason. A possible way to resolve this kind of anomalies is to introduce some negative information or completion of constraint rules to prove that there does not exist any `X:horse` satisfying `has(X,tire)`. Then, the body part becomes false, so the analogous rule is evaluated as a logically valid rule, and is never used due to the constraint. This will enable us to take an adequate grouping ψ as a right GDA.

Another serious problem we have to consider is the incompleteness of our domain theory. The problem is troublesome, since it may be concerned with default knowledge we use. However, it seems that we can have at least two methods to cope with it. We suppose that a user presents an initial goal G, initial situation E_0 from which G is derived. In addition to the pair $\langle G, E_0 \rangle$ as a query to our system, suppose that a set of training instance situations some of which should be similar to E_0 w.r.t. G and some of which should not be. When our domain theory $\mathcal{T}$ is inadequate, then the system will return a false answer about the similarity opposed to user's intention. The methods to complete and to adjust $\mathcal{T}$ will depend on a reason why such an inadequateness is derived under the present $\mathcal{T}$.

Case 1: Some rule is missed in $\mathcal{T}$ and is never constructed from any rule in $\mathcal{T}$ by both specialization and generalization. In this case, we need to have a rule relating the goal with both E_0 and E_j. Abductive ILP might be successful for this purpose.

Case2: Some missing rule is constructed from a rule in $\mathcal{T}$ by generalizing or by specializing it. We have already tried to apply the specialization to adjust or to revise the existing rule so that wrong similarities are never concluded by the system under a new revised theory [11]. It will be interesting when we try the generalization so that it constructs a new revised theory under which only a right similarity between E_0 and E_j become observed by our system.

Finally, we must discuss how we provide a class of concept graphs. It is generally hard to answer this question. However, recent development of natural language processing for text corpus will be regarded. From a text, we can extract various kinds of relations between words in the form of case structure or of dependency triples. They can be pieces of our concept

graphs, where words correspond to our sorts. A concept graph would be extracted by gathering such information from a restricted small part of corpus. Moreover, some information theoretic measure is available to estimate an actual similarity between words appearing in the large text and to exclude non-relevant similarities. Using such a measure, it seems to complete the extracted concept graph and to design a heuristic control to find an appropriate grouping of sorts at the same time.

References

[1] J. D. Tenenberg: Abstracting first-order theories, *Change of Representation and Inductive Bias*, D. P. Benjamin ed., Kluwer Academic Publishers, pp. 67–79, 1989.

[2] Josh D. Tenenberg: Abstraction in Planning, *Reasoning about Plans*, J. F. Allen et al., Morgan Kaufmann Publishers, pp. 213–283, 1991.

[3] D. A. Plaisted: Theorem Proving with Abstraction, *Artificial Intelligence*, vol. 16, pp. 47–108, 1981.

[4] Craig A. Knoblock: A Theory of Abstraction for Hierarchical Planning, *Change of Representation and Inductive Bias*, D.Paul Benjamin, ed., Kluwer Academic Publishers, pp. 81–104, 1989.

[5] T. Kakuta, M. Haraguchi and Y. Okubo: A Goal-Dependent Abstraction for Legal Reasoning by Analogy, *International Journal of Artificial Intelligence and Law*, vol. 5, no. 1-2, pp. 97–118, 1997.

[6] T. Kakuta, M. Haraguchi and Y. Okubo: Legal Reasoning by Structural Analogy Based on Goal-Dependent Abstraciton, *Proc. of the 9th International Conference on Legal Knowledge-Based Systems – JURIX'96*, pp.111–121, 1996.

[7] Y. Okubo and M. Haraguchi: Constructing Predicate Mappings for Goal-Dependent Abstraction, *Annals of Mathematics and Artificial Intelligence*, vol. 23, no. 1–2, pp. 169–197, 1998.

[8] Y. Okubo and M. Haraguchi: Planning with Abstraction Based on Partial Predicate Mappings, *New Generation Computing*, vol.12, no. 4, pp. 409–437, 1994.

[9] H. Ait-Kaci and A. Podelski: Towards a meaning of LIFE, *Proc. of 3rd PLILP (Programming Language Implementation and Logic Programming*, Springer-Verlag, pp. 255–274, 1991.

[10] A. Y. Levy and M. C. Rousset: Combining Horn rules and description logics in CARIN, *Artificial Intelligence*, vol. 104, no. 1–2, pp. 165–209, 1998.

[11] N. Morita, M. Haraguchi and Y. Okubo: A method of similarity-driven knowledge revision for type specialization, *Proc. of the 10th International Conference on Algorithmic Learning Theory – ALT'99*, Springer LNAI 1720, pp. 194–205, 1999.

[12] W. W. Cohen and H. Hirsh: The Learnability of Description Logics with Equality Constraints, *Machine Learning*, vol. 17, no. 2–3, pp. 169–199, 1996.

[13] C. Beierle, U. Hedtstuck, U. Pletat, P. H. Schmitt and J. Siekmann: An order-sorted logic for knowledge representation systems, *Artificial Intelligence*, vol. 55, no. 2–3, pp. 149–91, 1992.

[14] C. Walter: Many-Sorted Unification, *JACM*, vol. 35, no. 1, pp. 1–17, 1988.

[15] R.J. Brachman, A. Borgida and D. L. McGuinness: Living with CLASSIC, *Principles of Semantic Networks*, J. Sowa ed., Morgan Kaufmann, pp. 401–456, 1991.

[16] W. Buntine: Generalized subsumption and its applications to induction and redundancy, *Artificial Intelligence*, vol. 36, no. 2, pp. 149–76, 1988.

[17] J. W. Lloyd: *Foundations of Logic Programming*, Springer, Berlin, 1984.

[18] B. Nebel: *Reasoning and Revision in Hybrid Representation Systems*, Springer LNAI 422, 270 pages, 1990.

[19] T. M. Mitchell, R. Keller and S. Kedar-Cabelli: Explanation-Based Generalization : An Unifying View, *Machine Learning*, vol.1, no.1, pp. 47–80, 1986.

[20] S. Kedar-Cabelli: Purpose-directed analogy, *Proc. of the 7th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 150–159, 1985.

[21] W. Emde and D. Wettschereck: Relational Instance-Based Learning, *Proc. of the 13th International Conference on Machine Learning*, pp. 122–130, 1996.

[22] D. Gentner: Structure Mapping: A Theoretical Framework for Analogy, *Cognitive Science*, vol. 7, pp. 155–170, 1983.

[23] B. Falkenhainer, K. D. Forbus and D. Gentner: The Structure Mapping Engine: Algorithm and Examples, *Artificial Intelligence*, vol. 41, no. 1, pp. 1–63, 1989.