Cent. Eur. J. Comp. Sci. • 2(3) • 2012 • 245-260
DOI: 10.2478/s13537-012-0023-7

Central European Journal of **Computer Science**

Modeling generalization and specialization with Extended Conceptual Graphs

Research Article

Erika Baksa-Varga[*], László Kovács[†]

Department of Information Technology, University of Miskolc
Miskolc-Egyetemváros, H-3515 Miskolc, Hungary

Received 02 February 2012; accepted 17 August 2012

Abstract: The final goal of our research is to show that the performance of statistical rule induction can be improved by augmenting training data with semantic information. In order to prove this hypothesis, a statistical grammar induction system is to be created the knowledge base of which is represented by Extended Conceptual Graphs (ECGs). Since generalization and specialization are the basic operations of induction, they are of great significance in machine learning. As a consequence, the paper aims at investigating the least common generalization and the greatest common specialization of two ECG graphs. These operations should be traced back to the examination of ECG graph element instances. For this reason, a domain-specific ECG element instance type lattice $(T', \prec)$ has been generated for the given test environment. Our final conclusion is that the least common generalization and the greatest common specialization of two ECG graphs always exist and can be computed. Therefore, the definition of the $\prec$ relation on element instances can be extended to a partial relation $\preceq$ on ECG diagram graphs, according to which $\Gamma_1 \preceq \Gamma_2$ if graph Γ_1 is more *specialized* than Γ_2.

Keywords: semantic modeling • knowledge representation • machine learning • conceptualization

1. Introduction

The main motivation for the research is to develop a new general rule learning methodology that alloys statistics with semantics. The actual learning problem is chosen to be grammar induction, because symbolic languages have a fairly complex systems of rules (grammars), so they must be considered when developing a *general* methodology. Also, grammar induction has many application areas, such as computational linguistics, chemistry or pattern matching.

The first phase of the research has covered the specification of an appropriate semantic knowledge representation model (called Extended Conceptual Graph, ECG [2]) optimized for grammar induction, which is used for representing the knowledge base of the agent examined. The capabilities of the grammar learning agent are fixed in advance, which are

- pattern recognition: the ability to recognize the objects of its direct environment and their relations;

[*] *E-mail: vargae@iit.uni-miskolc.hu (Corresponding author)*
[†] *E-mail: kovacs@iit.uni-miskolc.hu*

- association: the ability of incorporating new information items into its existing knowledge base; and
- generalization.

Generalization using ECG graphs can be interpreted at three levels. At the first level higher-level concepts can be revealed based on the common characteristics of existing concepts. At the second level the substitutability of objects can be learned on the basis of the semantic role relations defined by the predicate. At the third level the frequent predicate schemas can be explored. Amongst these the first level *concept generalization* has been implemented and defined as the process by which new abstract ECG concepts are derived and introduced from lower-level concepts by extracting their common characteristics.

The operations of association and generalization define the process of conceptualization within the grammar learning agent at the end of which stands the generalized knowledge an agent can obtain from the samples observed. This can be given as a generalized accumulated ECG diagram graph built up from a set of primary-level ECG diagram graphs through several generalization stages. On the other hand, the specialization of two general ECG diagram graphs is formulated as their greatest common specialization, which can be defined as the maximal common restriction of the two graphs.

The paper is aimed at demonstrating both processes of generalization and specialization. Accordingly, the paper is organized as follows. Section 2 introduces some related papers that gave the guidelines for the present investigations. Section 3 gives a brief introduction to the ECG semantic model. Section 4 gives the details of the process of conceptualization using ECG graphs, involving the operations of graph matching and association (4.2), and generalization (4.3). This section also specifies the ECG element type lattices (4.1) required for modeling conceptualization. Section 5 defines the process of specialization. After these, the paper exposes a test environment in Section 6 and a theoretical context in Section 7 in which the processes of generalization and specialization are demonstrated through examples. Finally, the paper ends up with a conclusion (Section 8).

2. Related works

In the literature, ECG is used for denoting the collection of models developed on the basis of Sowa's Conceptual Graph (CG) theory [21]. It is a logical formalism that includes classes, relations, individuals and quantifiers. This formalism is based on semantic networks, and possesses both a graphical representation, called display format and a textual representation, called linear format. In its graphical notation, a conceptual graph is a bipartite directed graph where instances of concept types are displayed as rectangles and conceptual relations are displayed as ellipses (the set of which corresponds to thematic roles [9] in linguistics). Directed edges then link these vertices and denote the existence and orientation of the relation. From a linguistic point of view, ”conceptual relations link the concept of a verb to the concepts of the participants in the occurrent expressed by the verb” [22].
From the 1990s onward several extensions of the CG model were being born serving special research purposes. The first trials include [6, 16, 17]. In the first two papers an extension to Sowa's approach is proposed in which temporal and nontemporal knowledge are differentiated which enables the representation of tenses as well as the aspectual properties of natural language sentences. In the third paper the authors suggest marking conceptual relations with cardinalities for specifying constraints. The extended conceptual structure notation developed at the University of Minnesota [25] – among other extensions – can represent generalized counting quantifiers without using conditionals. [14, 15] report on the development of CG-based modeling languages introducing several extensions to the standard CG theory. In 1997, [8] proposes to extend the CG formalism in order to allow the representation of default taxonomic knowledge. Conceptual programming graphs [23] are also extended forms of Sowa's conceptual graphs which introduce a representation for expressing procedural and constraint knowledge (called *overlays*) through extended features to actors. These features are both syntactic and semantic, and make the actors perform more like ”functional relations”. [1] presents a family of extensions of the CG model based on rules and constraints. In [18] the semantic content of photos is represented by extended conceptual graphs for effective photo retrieval. In [20] business process diagrams are transformed into an extended conceptual graph notation where CGs are extended with AND/OR trees [19] to express the explicit dynamic properties or domain concepts. Lately, fuzzy conceptual graphs [5] have also been developed that extend conceptual graphs with general quantifiers.

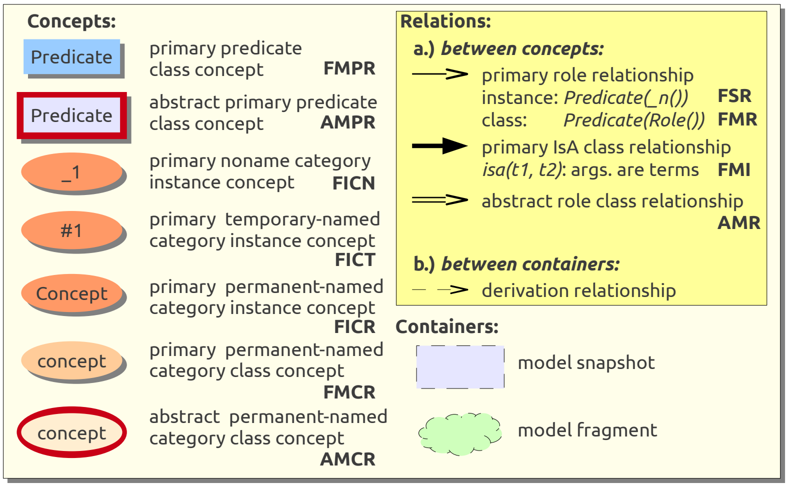

Concepts:
Predicate
primary predicate class concept FMPR
Predicate
abstract primary predicate class concept AMPR
_1
primary noname category instance concept FICN
#1
primary temporary-named category instance concept FICT
Concept
primary permanent-named category instance concept FICR
concept
primary permanent-named category class concept FMCR
concept
abstract permanent-named category class concept AMCR
Relations:
a.) between concepts:
primary role relationship
instance: Predicate(_n()) FSR
class: Predicate(Role()) FMR
primary IsA class relationship
isa(t1, t2): args. are terms FMI
abstract role class relationship AMR
b.) between containers:
derivation relationship
Containers:
model snapshot
model fragment

Figure 1. Graphical components of ECG diagram graphs.

Our approach got the name Extended Conceptual Graph because the model is a graph-based semantic network of concepts. These conceptual graphs, however, differ from Sowa's graphs not only in their graphical notation but also in the underlying theory. The most important difference is that our ECG model (described in Section 3) is a pure semantic model which allows a representation independent of the syntactic level of language. This means that, in contrast with Sowa's CG theory, two semantically equivalent statements always yield identical ECG graphs independent of their surface (syntactic) differences.

The guiding paper of the present article is [7] that outlines the principles of a Prolog-like resolution method which allows for expressing a large amount of background knowledge in terms of Sowa's conceptual graphs and performing deduction on very large linguistic and semantic domains. The interpreter developed is similar to a Prolog interpreter in which the terms are any conceptual graphs and in which the unification algorithm is replaced by a specialized algorithm for conceptual graphs. The paper introduces two algorithms: one for generating the greatest common specialized graph of two conceptual graphs (*maximal join of two graphs*), and one for generating the least common generalized graph that can be obtained from two conceptual graphs (*generalization of two graphs*). This CG processor is the main component of the KALIPSOS general system for knowledge acquisition from texts that is developed at IBM Paris Scientific Center. The success of this project has motivated our effort of simulating the process of generalization and specialization using ECG diagram graphs.

3. Modeling with Extended Conceptual Graphs

The Extended Conceptual Graph (ECG) model is a semantic modeling language which can be given in two equivalent forms: in an adequately extended higher-order predicate logic based textual format (ECG-HOPL [3]) and in a graphical representation called ECG diagram [2]. Its components are shown in Figure 1 and an example is illustrated in Figure 2. It is computationally tractable while highly expressive, i.e. it covers a wide range of linguistic phenomena. The model is designed for knowledge representation in learning agents and is specifically optimized for grammar induction. The capabilities of the agents are fixed in advance, and they are defined so that they are able to detect objects in the environment, their attributes, and the relationships between them; where the set of recognizable attributes and relationships are pre-defined. The main characteristics of the model can be summarized as follows.

Main building blocks of the model

The main building blocks of the model are concepts, relationships, and containers which serve for structuring the model. The *world* is built up of interconnected ECG model fragments representing separate observations, containing exactly

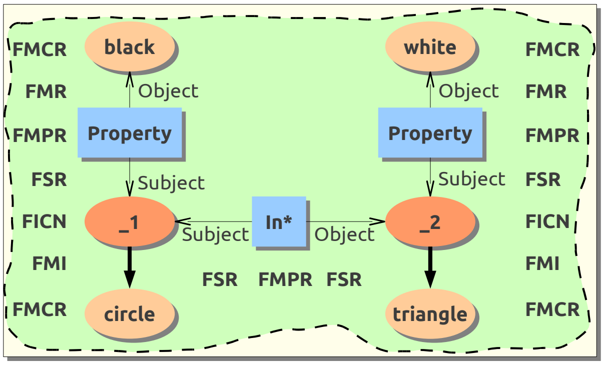


Figure 2. ECG diagram graph for ”A black circle is in a white triangle”.

one kernel predicate and having *true* truth value. Since the model bears the features of ontologies, it can be considered as an *ontology modeling language*.

Predicate-centeredness

In contrast with other existing semantic models – which have been deeply examined in view of their expressive power in [13] – the ECG model is predicate-centered. That is predicate concepts, which are distinguished from non-predicate concepts, are the kernels of atomic propositions. In the center of an ECG model fragment stays the kernel predicate, and each basic ECG graphical structure is organized around a predicate.

Multiple conceptualization levels

In the ECG model, the process of conceptualization occurs at two levels. The primary level of the ECG model serves for the direct mapping of environment objects and relationships into primary-level knowledge items. At the abstract level temporal and other complex relationship types can also be managed; and this level serves for modeling the process of conceptualization.

Distinction between apriori and learned elements

ECG differentiates between several categories of concept and relationship types both at the primary and at the abstract levels.

Flexibility

The ECG model is able to grasp the semantic content of situations. The elements of the environment can be represented by the relatively small, fixed set of ECG model elements. This means that several environment elements are mapped to the same ECG model element, which has therefore flexible semantic assignment.

Extendibility

The ECG model is a recursive, compositional system. That is infinitely many statements can be constructed from the small finite set of model elements.

4. The process of conceptualization

In terms of machine learning, concept formation is the process by which an agent learns to sort specific experiences into general rules or classes. In order to make learning feasible in complex domains, abstraction and generalization operators are often applied to make the problem tractable. In [24], abstraction is given as a technique to reduce the complexity of a problem by filtering out irrelevant properties while preserving all the important ones necessary to still be able to solve a given problem. At the same time, generalization is defined as a technique to apply knowledge previously acquired to

unseen circumstances or extend that knowledge beyond the scope of the original problem.
In the present approach, the learning agent builds up its knowledge base by

- incorporating and relating the information elements observed – which are instance-level ontologies represented by ECG diagram graphs – to its existing (and continuously evolving) knowledge base (*association*);
- introducing new (not observed) higher-level concepts into the knowledge base, thereby reducing its complexity (*generalization*).

The higher-level concepts to be introduced are pre-defined in a domain-specific concept lattice which is used for representing concept generalization structures [12]. Its generation from the given domain is called *abstraction*. A widely accepted formalism for conceptualization is the theory of Formal Concept Analysis (FCA) [10].
In FCA, there are two main variants of concept set building algorithms. The methods of the first group work in batch mode, assuming that every element of the context table is already present before starting the concept lattice building. The other group of proposals uses an incremental lattice building method [11].

Our concept lattice building algorithm belongs to the first group. It uses the information present in the samples, and also the abstract element types defined by the ECG model. There are two abstract concept types which can be used in generalization, that is

- AMCR: abstract category concept, for generalizing concepts in T_{cc};
- AMPR: abstract predicate concept, for generalizing concepts in T_{pc}.

4.1. ECG element type lattices

In the ECG model, concepts and relations (elements) are given by two attributes: a type and a caption, in the form of $type : caption$. Formally, each $ec \in EC$ element category in the ECG model is given as $ec = type_c : caption$, where $type_c \in T$ and $caption$ is a string representing the name of the corresponding element category. The T set of ECG element category types is the union of the subsets listed in Table 1.

Table 1. Classification of ECG element category types.

Subset of T	Element category types
T_{cc}: category concept types	FICN, FICT, FICR, FMCR, AMCR
T_{pc}: predicate concept types	FMPR, AMPR
T_{rr}: semantic role relation types	FSR, FMR, AMR
T_{sr}: specialization relation type	FMI

ECG element category types can be merged in a lattice $(T, <)$, whose partial ordering relation $<$ can be interpreted as a categorical generalization relation. The top and the bottom element categories of the $(T, <)$ lattice are UNIV (the suprenum element) and NIL (the infinum element), respectively. The generated lattice is displayed in Figure 3. On the basis of this lattice, we say that $ec_1 < ec_2$ if $type_{c_1} < type_{c_2}$. Also, it is possible to exhibit the least common generalization lcg, and the greatest common specialization gcs of two element category types ec_1 and ec_2:

$$lcg(ec_1, ec_2) = \min\{type_c \mid type_{c_1} <= type_c \ \wedge \ type_{c_2} <= type_c\}; \qquad (1)$$

$$gcs(ec_1, ec_2) = \max\{type_c \mid type_c <= type_{c_1} \ \wedge \ type_c <= type_{c_2}\}. \qquad (2)$$

Analogously, in an instance-level ECG diagram graph each $ei \in EI$ element instance is given as $ei = type_i : caption$, where $type_i \in T'$ and $caption$ is a string representing the name of the corresponding element instance. The members of T' are constructed from the members of T augmented with a number. Thus, an element instance type has the form $type_i = type_c\_n$, where $type_c$ is the corresponding element category type and n is a numeric code, so that $type_i = type_c\_n$ is a unique identifier within the problem domain. The element category type part of an element instance type is denoted by $[type_i] = type_c$, while the numeric code of an element instance type can be obtained as $\{type_i\} = n$.

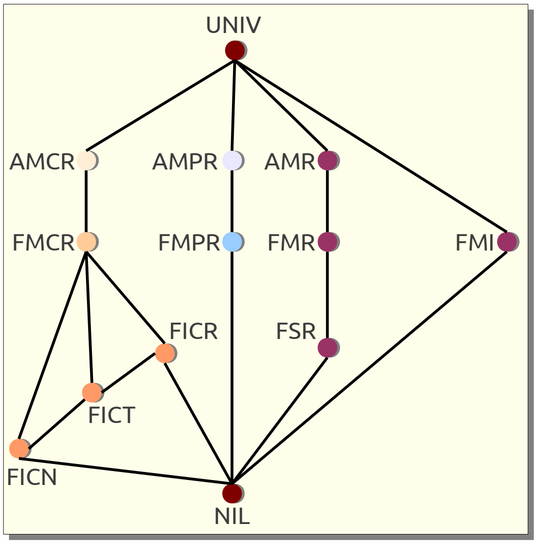


Figure 3. ECG element category type lattice.

Definition 4.1.
An ECG diagram graph can be defined as $\Gamma = \langle V, A, R \rangle$. V is the set of vertices containing e_i element instances where $[type_i] \in T_{cc} \cup T_{pc}$. A is the set of arrows (directed edges) containing e_i element instances where $[type_i] \in T_{rr} \cup T_{sr}$. R is the set of semantic roles with which the arrows in A are labeled. Thus, the f incidence function assigns an ordered pair of v vertices in V and a r semantic role in R to each a arrow in A, i.e. $f(a_i) = (v_i, v_j, r_k)$.

Two element instances ei_1 and ei_2 are said to be equivalent if $type_{i_1} = type_{i_2}$. ECG element instance types can be merged in a domain-specific lattice $(T', \prec)$ (see Figure 6 for an example), whose partial ordering relation $\prec$ can be interpreted as a categorical generalization relation. The top and the bottom elements of the $(T', \prec)$ lattice are UNIV (the suprenum element) and NIL (the infinum element), respectively. On the basis of this lattice we say that $ei_1 \prec ei_2$ if $type_{i_1} \prec type_{i_2}$. Also, it is possible to exhibit the least common generalization lcg, and the greatest common specialization gcs of two differing element instances ei_1 and ei_2 ($ei_1 \neq ei_2$):

$$lcg(ei_1, ei_2) = \min\{type_i \mid type_{i_1} \prec= type_i \ \wedge \ type_{i_2} \prec= type_i\}; \quad (3)$$

$$gcs(ei_1, ei_2) = \max\{type_i \mid type_i \prec= type_{i_1} \ \wedge \ type_i \prec= type_{i_2}\}. \quad (4)$$

4.2. Graph matching and association

Generally speaking, the graph matching problem (for unlabeled undirected graphs) can be stated as follows. Given two graphs $G_1 = \langle V_1, E_1 \rangle$ and $G_2 = \langle V_2, E_2 \rangle$, the problem is to find a bijective mapping $f : V_1 \to V_2$ so that $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$. When such a mapping exists, this is called an isomorphism, and G_1 is said to be isomorphic to G_2. If G_1 is isomorphic to G_2 we write $G_1 \cong G_2$. On the other hand, G_1 is subgraph isomorphic to G_2 if $G_1' \subset G_1$ exists where $G_1' \cong G_2$. For labeled graphs, the following requirements also need to be met:

- the label of vertex u must be the same as that of $f(u)$ for all $u \in V_1$;
- the label of edge (u, v) must be the same as that of $(f(u), f(v))$ for all $(u, v) \in E_1$.

According to Definition 4.1 ECG diagrams are labeled directed graphs, and the matching operation determines an alignment (i.e. a set of mapping elements M) for a pair of ECG graphs. A mapping element $m \in M$ is defined as a triplet $\langle ei_1, ei_2, \varphi \rangle$, where

- $ei_1 \in \Gamma_1$ and $ei_2 \in \Gamma_2$ are the aligned element instances of the two ECG graphs, and

- φ is the relation between ei_1 and ei_2.

In order to obtain an unambigous (bijective) mapping only the mapping elements where $\varphi \in \{=\}$ are included in the alignment. For describing the result of the matching, an alignment measure is introduced. Let l denote the number of mapping elements in an alignment M. For a given pair of aligned ECG diagram graphs Γ_1 and Γ_2, the fitness value μ is calculated as:

$$\mu(\Gamma_1, \Gamma_2) = \frac{l}{\frac{j+k}{2}}, \tag{5}$$

where j is the number of element instances in graph Γ_1, while k is the number of element instances in graph Γ_2. In this way, the fitness value falls into the $[0, 1]$ interval.

Association is the process by which ECG diagram graphs are gradually inserted into an initially empty knowledge base, which is itself another (accumulated) ECG diagram graph. In this operation, the ECG diagram graph to be inserted (Γ_2) should first be matched to the knowledge base (Γ_1) according to the following algorithm.

1. The M mapping (alignment) of the two ECG diagram graphs should be performed resulting in μ.

2. If $\mu(\Gamma_1, \Gamma_2) = 1$ then $\Gamma_1 \equiv \Gamma_2$, i.e. $V_1 = V_2 \ \wedge \ A_1 = A_2$. In this case Γ_2 does not need to be inserted into the knowledge base (it is already in the knowledge base).

3. If $\mu(\Gamma_1, \Gamma_2) = 0$ then $V_1 \cap V_2 = \emptyset$. In this case Γ_2 is inserted into the knowledge base in a disjunctive way.

4. If $0 < \mu(\Gamma_1, \Gamma_2) < 1$ then $V_1 \cap V_2 \neq \emptyset$. In this case, if Γ_2 is not a subgraph of Γ_1 then $\forall v_{2i}, a_{2i} \in \Gamma_2 \mid v_{2i}, a_{2i} \notin \Gamma_1$ are inserted into the knowledge base in a conjunctive way.

An ECG diagram graph Γ_2 is a subgraph of Γ_1 ($\Gamma_2 \subseteq \Gamma_1$):

- if Γ_1 contains a subgraph Γ_1' which is identical to Γ_2, i.e. $\Gamma_1' \equiv \Gamma_2$, i.e. $V_1' = V_2 \ \wedge \ A_1' = A_2$; or

- if Γ_1 contains a subgraph Γ_1' which is isomorphic to Γ_2 ($\Gamma_1' \simeq \Gamma_2$).

Definition 4.2.
Two ECG graphs are said to be isomorphic ($\Gamma_1' \simeq \Gamma_2$), if one ($\Gamma_2$) can be obtained from the other (Γ_1') by restricting some of the element instances of the latter (Γ_1') based on the $(T', \prec)$ element instance type lattice.

4.3. Generalization

The association operation does not involve generalization. It covers only the accumulation of incoming information. However, by the increase of the amount of incoming data the knowledge base would be subtle and computationally intractable without the use of generalization. The generalization algorithm (Algorithm 1) gives as result the least common generalized graph that can be obtained from two ECG graphs. This can be achieved formally by finding frequent knowledge patterns (ECG subgraphs) the context of which is similar.

Definition 4.3.
Two ECG diagram subgraphs $\gamma_1 \in \Gamma_1$ and $\gamma_2 \in \Gamma_2$ are said to be similar subgraphs if

- $\gamma_1 \equiv \gamma_2$ or $\gamma_1 \simeq \gamma_2$, and

- they are connected to differing but semantically comparable ECG concept nodes.

Two similar subgraphs are considered as maximal similar subgraphs if they cannot be extended further without violating the criterion of similarity.

Algorithm 1 The generalization algorithm.

```
Input: $\Gamma_1$, $\Gamma_2$
  $\mu$ = Match($\Gamma_2$, $\Gamma_1$)
  if $\mu = 1$ then
    Return
  end if
  if $\mu = 0$ then
    Insert $\Gamma_2$ into $\Gamma_1$
  end if
  if $0 < \mu < 1$ then
    if $\Gamma_2 \subseteq \Gamma_1$ then
      Return
    else
      Search for maximal similar subgraphs in $\Gamma_1$,$\Gamma_2$
      for all $(\gamma_1^*, \gamma_2^*)$ do
        if $ei_1 >< ei_2$ then
          if $lcg(ei_1, ei_2) \neq UNIV$ then
            if $lcg(ei_1, ei_2) \notin \Gamma_1$ then
              Insert $lcg(ei_1, ei_2)$ into $\Gamma_1$
            end if
            if $lcg(ei_1, ei_2) \neq ei_1$ then
              Connect $ei_1$ to $lcg(ei_1, ei_2)$ by $FMI$
            end if
            Update relations of $ei_1$ in $\Gamma_1$
            if $ei_2 \notin \Gamma_1$ then
              Insert $ei_2$ into $\Gamma_1$
            end if
            if $lcg(ei_1, ei_2) \neq ei_2$ then
              Connect $ei_2$ to $lcg(ei_1, ei_2)$ by $FMI$
            end if
          end if
        end if
      end for
      for all $ei \in \Gamma_2$ do
        if $ei \notin \Gamma_1$ then
          Insert $ei$ into $\Gamma_1$
        end if
      end for
      Update relations of $ei_2$ in $\Gamma_1$
    end if
  end if
  return $\Gamma_1$
```

Thus, the maximal similar subgraphs are searched for in Γ_1 and Γ_2. For this, the operation of ECG graph intersection should be introduced. The intersection of two ECG graphs Γ_1 and Γ_2 is the set of identical or isomorphic connected subgraphs. Formally,

$$\begin{aligned} \Gamma_1 \cap \Gamma_2 &= \{\gamma_1, \gamma_2, \ldots, \gamma_k\} \text{ where} \\ \forall \gamma_i \ : \ \gamma_i \in \Gamma_1 \ \wedge \ \gamma_i \in \Gamma_2 \quad \text{or} & \quad \forall \gamma_i \ : \ \gamma_i \in \Gamma_1 \ \wedge \ \gamma_i' \in \Gamma_2 \text{ where } \gamma_i \simeq \gamma_i'. \end{aligned} \tag{6}$$

The extension of the ECG graph intersection operation results in the pairs of similar subgraphs of Γ_1 and Γ_2. Formally,

$$\begin{aligned} \Gamma_1 \cap^* \Gamma_2 &= \{(\gamma_{11}^*, \gamma_{12}^*), (\gamma_{21}^*, \gamma_{22}^*), \ldots, (\gamma_{k1}^*, \gamma_{k2}^*)\} \text{ where} \\ \forall (\gamma_{i1}^*, \gamma_{i2}^*) \ : \ \gamma_{i1}^* \cup \gamma_{i2}^* &= \gamma_i \cup \{ei_1, ei_2\} \mid \gamma_{i1}^*, ei_1 \in \Gamma_1 \ \wedge \ \gamma_{i2}^*, ei_2 \in \Gamma_2 \ \wedge \ \gamma_i \in \Gamma_1 \cap \Gamma_2. \end{aligned} \tag{7}$$

Maximal similar subgraphs are then obtained from merging all similar subgraphs with the same root nodes, i.e. similar subgraphs having the same pair of differing concepts.

$$\{\max(\gamma_{i1}^*, \gamma_{i2}^*)\} = \{\cup(\gamma_{i1}^*, \gamma_{i2}^*)\} \mid \forall(\gamma_{i1}^* \cup \gamma_{i2}^*) = \gamma_{i_j} \cup \{ei_1, ei_2\}. \tag{8}$$

For the differing concepts semantic comparability ($ei_1 >< ei_2$) should be checked, where $ei_1 \in \Gamma_1$ and $ei_2 \in \Gamma_2$. Two element instances are said to be semantically comparable if $lcg([type_{e_{i_1}}],[type_{e_{i_2}}]) \neq UNIV$ on the basis of the element category type lattice (Figure 3). If this is the case, instead of the differing concepts a new concept is introduced from the element instance type lattice determined as $lcg(ei_1, ei_2)$, if it is not the $UNIV$ top element. It is possible, that $lcg(ei_1, ei_2)$ results in one of its arguments. In this case actually no insertion occurs. The differing concepts are connected to the new concept via specialization relationships and the other relationships originally in connection with the differing concepts should also be updated.

At the end of the generalization process (Figure 4) stands the generalized knowledge an agent can obtain from the samples observed. This can be formulated as recursively determining the least common generalization of the previous abstract-level ECG graph and the next primary-level ECG graph, that is

$$\Gamma_{a_i} = \mathrm{lcg}(\Gamma_{a_{i-1}}, \Gamma_{p_{i+1}}), \text{ where } \Gamma_{a_0} = \Gamma_{p_1}. \tag{9}$$

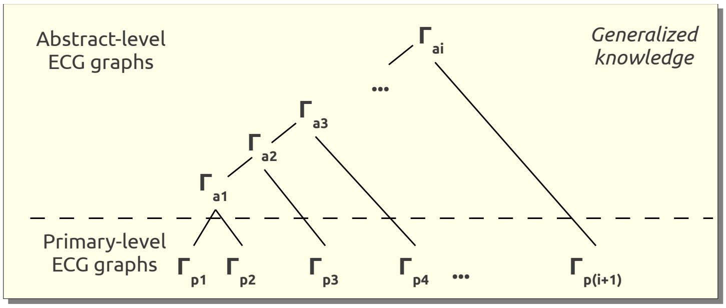


Figure 4. The process of generalization.

5. Specialization

Similarly to the formulation of the least common generalization of two ECG graphs, it is possible to give the greatest common specialization of two ECG graphs. The latter can be defined as the maximal common restriction of the two graphs, i.e. the union of the maximal similar subgraphs of the two graphs (see Algorithm 2). Formally,

$$gcs(\Gamma_1, \Gamma_2) = \bigcup(\{\max(\gamma_{i1}^*, \gamma_{i2}^*)\}). \tag{10}$$

Algorithm 2 The specialization algorithm.

```
Input: $\Gamma_1$, $\Gamma_2$
  Search for maximal similar subgraphs in $\Gamma_1$,$\Gamma_2$
  for all ($\gamma_1^*$, $\gamma_2^*$) do
    if $ei_1 \in \Gamma_2$ AND $ei_1 \notin \Gamma_3$ then
      Insert $ei_1$ into $\Gamma_3$
    end if
    if $ei_2 \in \Gamma_1$ AND $ei_2 \notin \Gamma_3$ then
      Insert $ei_2$ into $\Gamma_3$
    end if
  end for
  return $\Gamma_3$
```

6. Test environment

Let us assume that the environment of the learning agent is a microworld of 2D shapes and each observation includes two objects in a binary relation. The base set for modeling the process of conceptualization includes approximately 300 thousand ECG diagram graphs generated by the semantic annotation framework [4] which has been implemented in NetBeans IDE 6.9 integrated development environment using Java 1.6.0_20 version Java HotSpot(TM) 64-Bit Server VM 16.3-b01. The operational model of the system is shown in Figure 5.

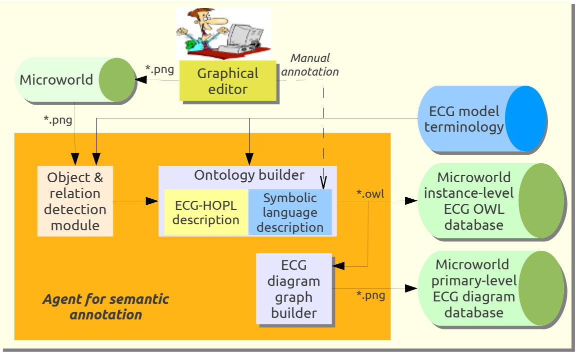

Microworld
*.png
Graphical editor
Manual annotation
ECG model terminology
*.png
Object & relation detection module
Ontology builder
ECG-HOPL description
Symbolic language description
*.owl
Microworld instance-level ECG OWL database
ECG diagram graph builder
*.png
Microworld primary-level ECG diagram database
Agent for semantic annotation

Figure 5. Operational model of the semantic annotation framework.

The system represents an agent for semantic annotation. Its environment is the graphical microworld created by a graphical editor. The environment snapshots (static observations) are processed by the sensor of the agent, which is the object and relation detection module. Its task is to recognize the objects of the environment and their relations, and to map them to the internal semantic representation of the agent using the terminological database. Next, the ontology builder generates the instance-level ontologies (assertions on the environment instances describing the semantics of the observations) in OWL DL textual format. Finally, the ECG diagram graph builder creates the graphical diagrams for the OWL descriptions.

The relevant segment of the element instance type lattice generated from the microworld is shown in Figure 6. In this example, abstract concepts can not be generated automatically from the samples. They are added manually to the lattice.

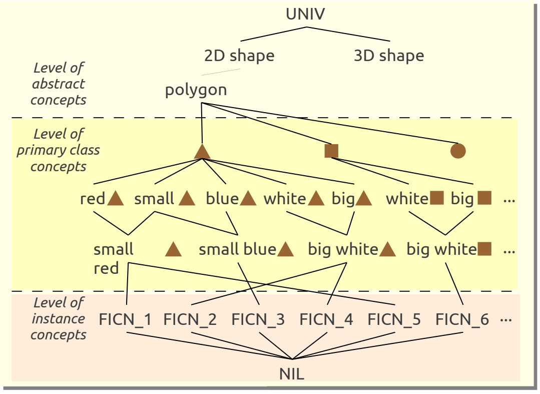


Figure 6. A segment of the element instance type lattice.

The generalization algorithm searches for maximal similar subgraphs in the two graphs to be joined, which differ in only one semantically comparable concept node. Instead of the differing concepts a new concept is introduced determined as the least common generalization of the differing concepts in the element instance type lattice.

In order to demonstrate the process of generalization, consider the example illustrated in Figures 7 and 8. Let Γ_1 denote the knowledge base of the agent already containing one observation and Γ_2 denote the observation to be inserted into the knowledge base. Two similar subgraphs are found marked by identical lines. The new concepts inserted from the element instance type lattice (see Figure 6) are indicated by dashed line ellipses. In the next stage Γ_3 denotes the current state of the knowledge base and Γ_4 is the new observation to be inserted. Again, two similar subgraphs are identified and two new concepts are inserted. As a result, after processing three observations Γ_5 represents the actual state of the knowledge base.

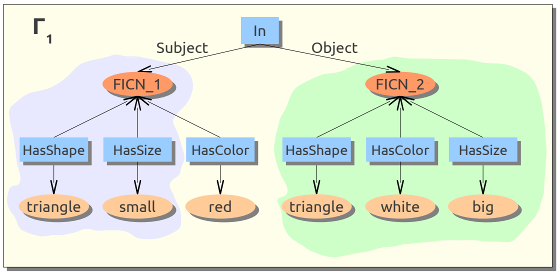
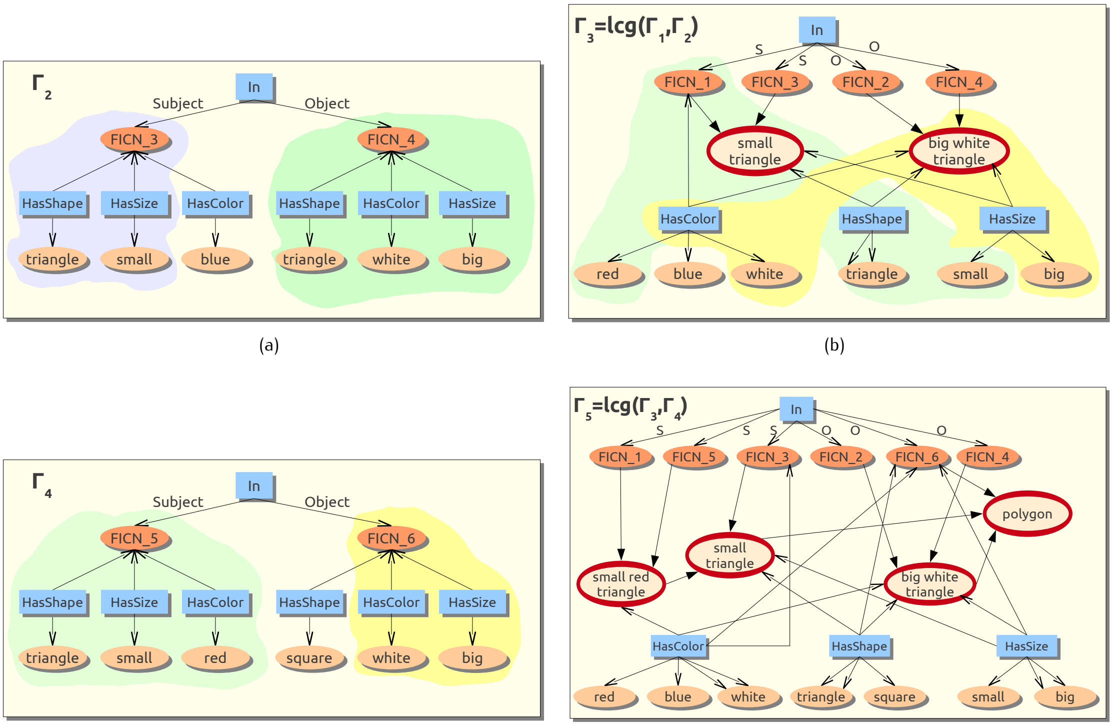


Figure 7. Initial state of the knowledge base containing one observation.



(a)

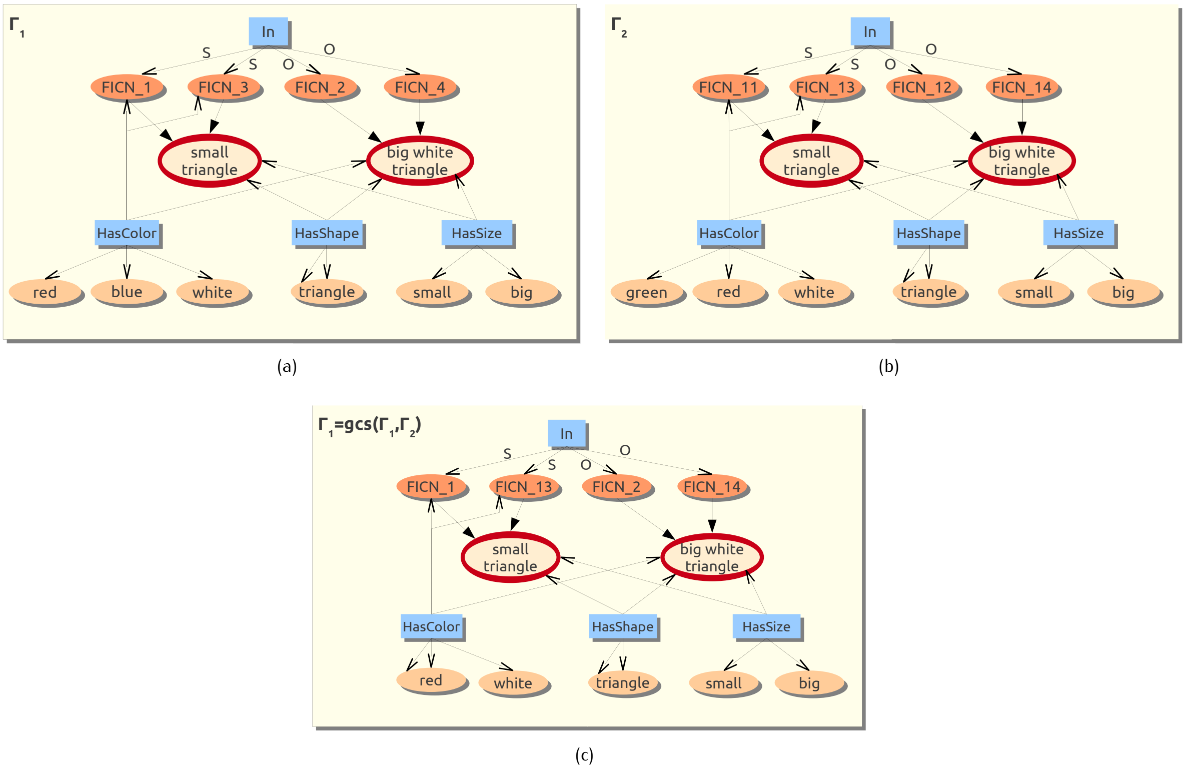
(b)

(c)

(d)

Figure 8. Demonstration of generalization.

For the demonstration of specialization, see the next example in Figure 9. Given two ECG diagram graphs Γ_1 and Γ_2, their greatest common specialization is shown by Γ_3.



(a)

(b)

(c)

Figure 9. Demonstration of specialization.

7. Theoretical example

In this section the algorithms of generalization and specialization are demonstrated on the concept lattice of planets in the Solar System as a classic example in formal concept analysis. The corresponding element instance type lattice is shown in Figure 10.

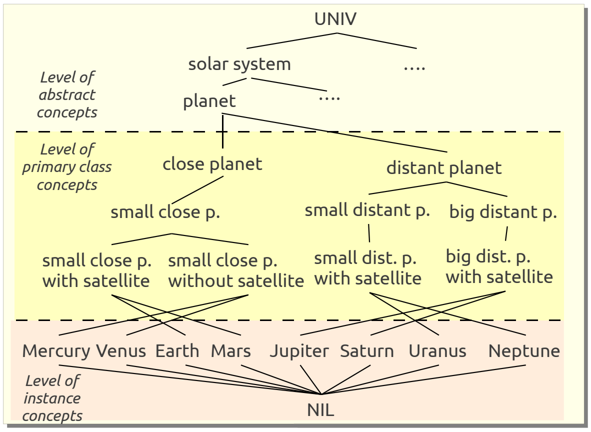
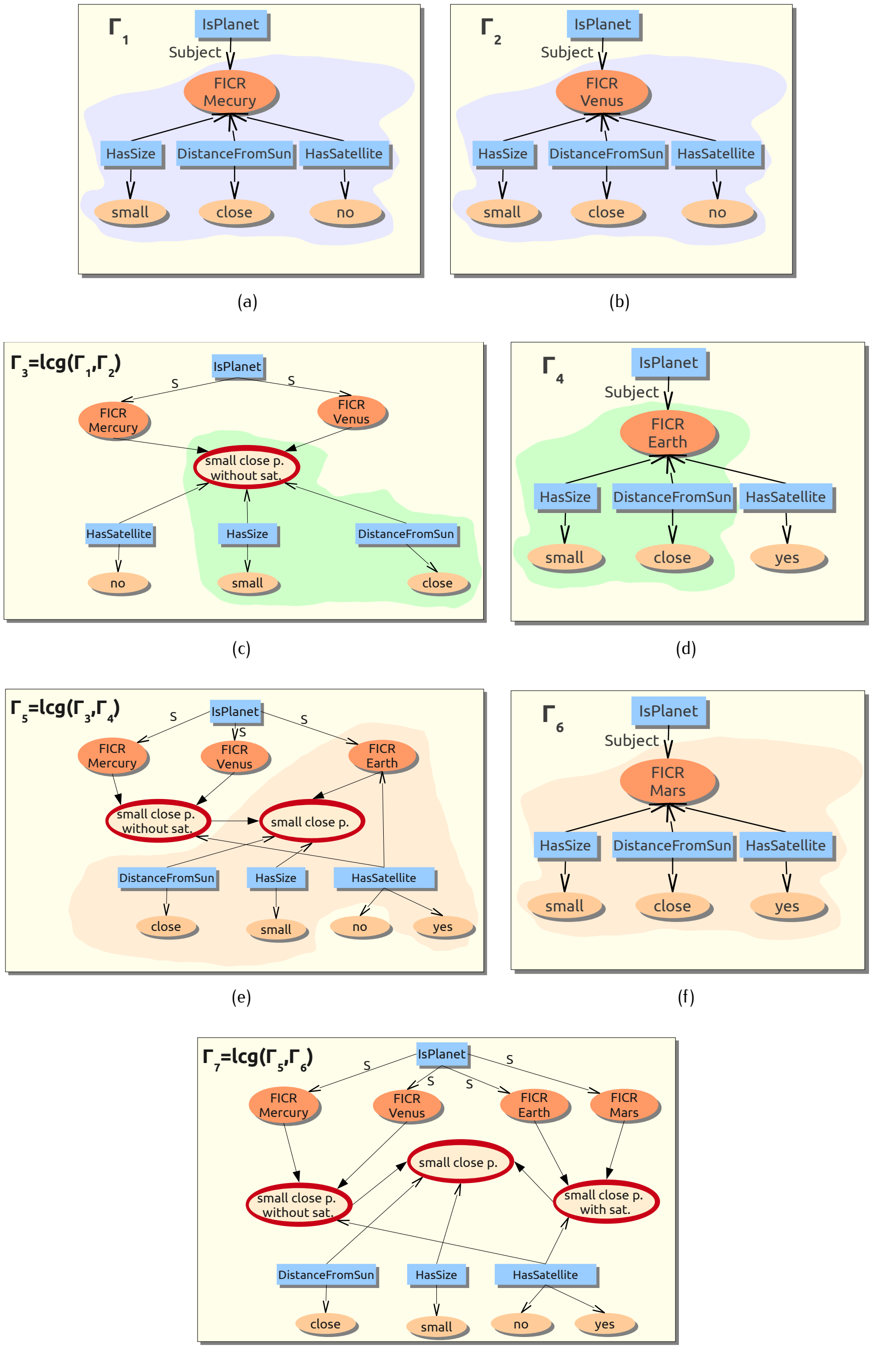

UNIV
solar system
....
Level of abstract concepts
planet
....
Level of primary class concepts
close planet
distant planet
small close p.
small distant p.
big distant p.
small close p. with satellite
small close p. without satellite
small dist. p. with satellite
big dist. p. with satellite
Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune
Level of instance concepts
NIL

Figure 10. Element instance type lattice of planets.

For the illustration of our generalization method, consider the planets that are close to the Sun. In Figure 11 ECG diagram graph Γ_1 represents Mercury while Γ_2 represents Venus. The least common generalization of the two graphs results in Γ_3 that introduces a new concept from the corresponding element instance type lattice. Γ_4 describes Earth, and the least common generalization of Γ_3 and Γ_4 yields Γ_5 that brings in another new concept. Γ_6 describes Mars and the least common generalization of Γ_5 and Γ_6 derives Γ_7. From this final ECG diagram graph we can see that close planets differ only in the attribute of having satellite. This implies that all close planets are small.



(a)

(b)

(c)

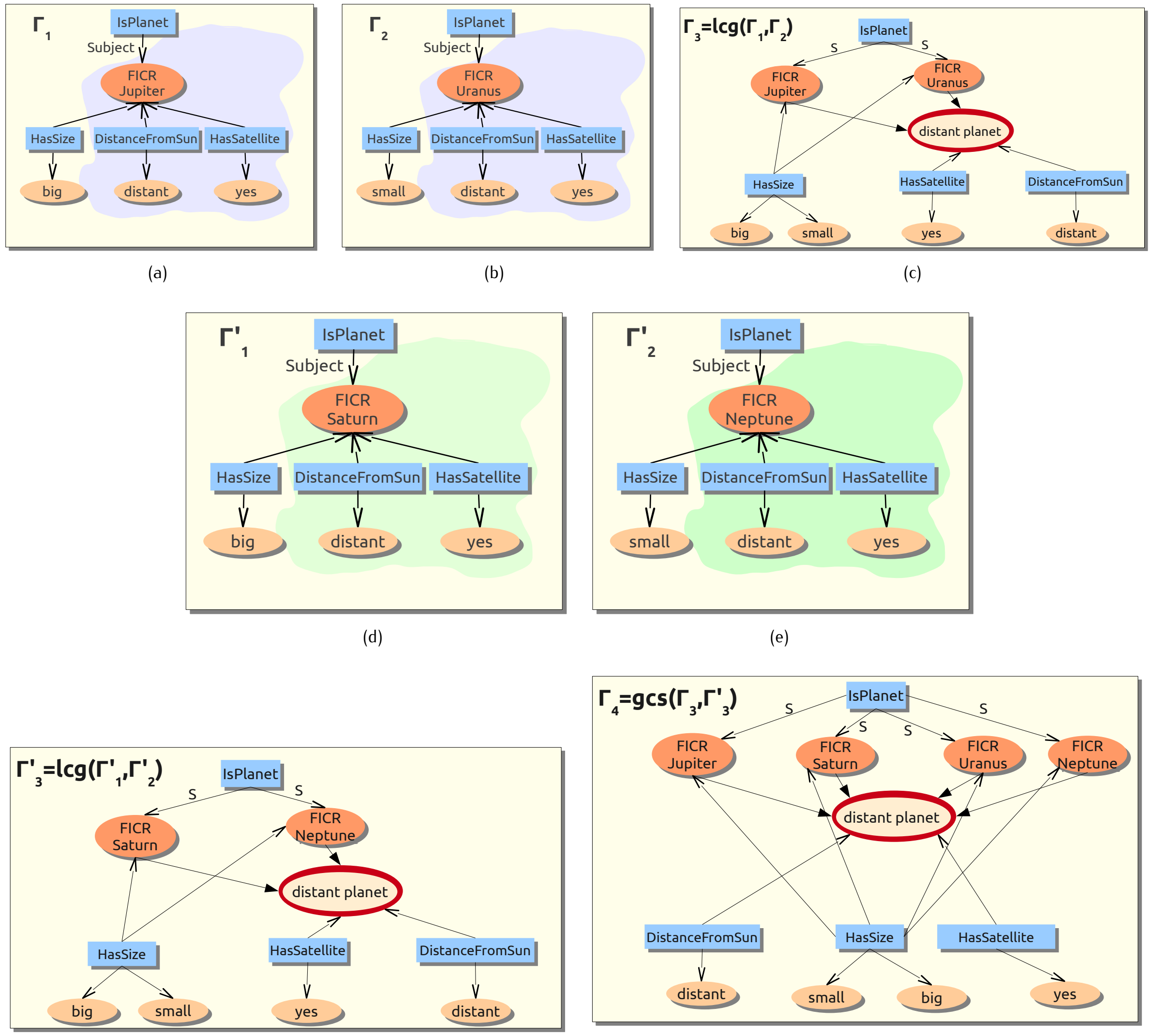
(d)

(e)

(f)

(g)

Figure 11. Demonstration of generalization in the context of close planets.



(a) (b) (c)

(d) (e)

(f) (g)

Figure 12. Demonstration of specialization in the context of distant planets.

The specialization algorithm is illustrated on distant planets in Figure 12. Γ_1 represents Jupiter and Γ_2 stands for Saturn. The least common generalization of the two ECG diagram graphs yields Γ_3. Γ_1' describes Uranus and Γ_2' stands for Neptune. The least common generalization of the two ECG diagram graphs yields Γ_3'. The greatest common specialization of Γ_3 and Γ_3', i.e. the union of the maximal similar subgraphs of the two graphs is shown by Γ_4. From this final ECG diagram graph we can see that distant planets differ only in their size attribute. This implies that all distant planets have satellites.

8. Conclusion

The paper has shown how the ECG graph-based knowledge base of the grammar learning agent examined is built up from primary-level ECG graphs. The operation of association – i.e. matching and connecting ECG diagram graphs – can always be accomplished, but generalization – i.e. the introduction of higher-level concepts – does not necessarily occur in each step. Also, it is possible that the greatest common specialization of two ECG graphs results in an empty graph. Nevertheless, the least common generalization and the greatest common specialization of two ECG graphs always exist and can be computed. Therefore, the definition of the $\prec$ relation on element instances can be extended to a partial relation $\preceq$ on ECG diagram graphs and the term *restriction of* can be used to describe this relation. Accordingly, an ECG diagram graph $\mathit{\Gamma}_2$ is a restriction of ECG diagram graph $\mathit{\Gamma}_1$, i.e. $\mathit{\Gamma}_2 \preceq \mathit{\Gamma}_1$ if graph $\mathit{\Gamma}_2$ is more *specialized* than graph $\mathit{\Gamma}_1$.

Let $\mathbf{\Gamma}$ denote the set of primary-level ECG diagram graphs (representing environment snapshots) that are matched to and incorporated in the knowledge base of the agent, and $\mathbf{\Gamma}(\mathbf{A})$ denote the set of accumulated ECG diagram graphs resulting from the conceptualization (association and generalization) steps executed.

Corollary 8.1.
The $\preceq$ relation effects a lattice structure on the union of $\mathbf{\Gamma}$ and $\mathbf{\Gamma}(\mathbf{A})$.

The top element of the lattice $(\mathbf{\Gamma} \cup \mathbf{\Gamma}(\mathbf{A}), \preceq)$ symbolizes the accumulated knowledge of the agent at the end of the conceptualization process. The bottom element is NIL, the infinum element.

Acknowledgment

This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

References

[1] Baget J.F., Mugnier M.L., Extensions of simple Conceptual Graphs: the complexity of rules and constraints, J. Art. Intel. Res., 16, 425-465, 2002

[2] Baksa-Varga E., Kovács L., Knowledge base representation in a grammar induction system with Extended Conceptual Graph, Scientific Bulletin of "Politehnica" University of Timisoara, 53, 107-114, 2008

[3] Baksa-Varga E., Kovács L., Semantic representation of natural language with Extended Conceptual Graph, J. Prod. Syst. Inf. Eng., 5, 19-39, 2009

[4] Baksáné Varga E., Ontology-based Semantic Annotation and Knowledge Representation in a Grammar Induction System, PhD Thesis, University of Miskolc, Hungary, 2011

[5] Cao T.H., Conceptual Graphs and Fuzzy Logic: A Fusion for Representing and Reasoning with Linguistic Information, Stud. Comput. Intel., 306, 2010

[6] Creasy P., Moulin B., Extending the Conceptual Graph approach for data conceptual modelling, Data Knowl. Eng., 8, 223-248, 1992

[7] Fargues J., Landau M.C., Dugourd A., Catach L., Conceptual graphs for semantics and knowledge processing, IBM J. Res. Develop., 30, 1986

[8] Faron C., Ganascia J.G., Representation of defaults and exceptions in conceptual graphs formalism, Conceptual Structures: Fulfilling Peirce's Dreams, Lecture Notes Comp. Sc., 1257, 153-167, 1997

[9] Fillmore C.J., The case for case, In: Universals in linguistic theory, Bach E., Harms R. (New York, 1968)

[10] Ganter B., Wille R., Formal Concept Analysis, Mathematical Foundations (Springer Verlag, 1999)

[11] Godin R., Missaoui R., Alaoui H., Incremental concept formation algorithms based on Galois lattices, Comput. Intel., 11, 246-267, 1995

[12] Kovács L., Concept lattice structure with attribute lattices, J. Prod. Syst. Inf. Syst., 4, 65-81, 2006

[13] Kovács L., Baksa-Varga E., Logical representation and assessment of semantic models for knowledge base representation in a grammar induction system, J. Comp. Sc. Contr. Sys., University of Oradea, Romania, 48-53, 2008

[14] Lukose D., Executable Conceptual Structures, Conceptual Graphs for Knowledge Representation, Lect. Notes Artif. Int., 699, Springer-Verlag, Berlin, Germany, 1993

[15] Möller J.U., Willems M., CG-DESIRE: Formal specification using Conceptual Graphs, Proceedings of the 9th Banff Knowledge Acquisition For Knowledge-Based Systems Workshop, Banff Conference Centre, Banff, Alberta, Canada, 1995

[16] Moulin B., Côté D., Refining Sowa's conceptual graph theory for text generation, In: Proceedings of the 3rd international conference on Industrial and engineering applications of artificial intelligence and expert systems, 1, 528-537, 1990

[17] Moulin B., Côté D., Representing temporal knowledge in conceptual graphs, Knowledge-Based Systems, 4, 197-208, 1991

[18] Mulhem P., Lim J.H., Home Photo Retrieval: Time Matters, Image and Video Retrieval, Lect. Notes Comp. Sc., 2728, 321-330, 2003

[19] Soshnikov D., Data and knowledge representation models of distributed frame systems, In: Proceedings of Pre-Conference Workshop of VLDB-2003 Emerging Database Research in Eastern Europe, Brandenburg University of Technology at Cottbus, 123-127, 2003

[20] Soshnikov D., Dubovik S., Structured functional decomposition aApproach to knowledge-based business process modeling, In: Proceedings of 6th Joint Conference in Knowledge-Based Software Engineering, Frontiers in Artificial Intelligence and Applications, IOS Press, 2004

[21] Sowa J.F., Conceptual Structures: Information Processing in Mind and Machine (Addison-Wesley, Reading, MA, USA, 1984)

[22] Sowa J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations (Brooks Cole Publishing Co., Pacific Grove, CA, 2000)

[23] Pfeiffer H.D., Hartley R.T., The Conceptual Programming Environment, CP: Time, Space and Heuristic Constraints, In: Proceedings of the Sixth Annual Workshop on Conceptual Graphs, Binghamton, 1991

[24] Ponsen M., Taylor M.E., Tuyls K., Abstraction and generalization in reinforcement learning: A summary and framework, Lect. Notes Artif. Int., Adaptive and Learning Agents Workshop, 2010

[25] Tjan B.S., Gardiner D.A., Slagle J.R., Direct inference rules for Conceptual Graphs with extended notation, Technical Report 90-28 (University of Minnesota, 1990)