Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 37(4), 2010, Pages 1–8
ISSN: 1223-6934

Factorization of an inheritance knowledge base (II)

Nicolae Ţăndăreanu and Claudiu-Ionuţ Popîrlan

Abstract. In [6] we start to develop the decomposition of inheritance knowledge bases ([5]) into disjoint components such that the answer mapping can be locally computed in some component(*factorization problem*). This paper includes several further results concerning these structures. The main results presented in this paper are the following: we demonstrate that every component is itself an inheritance knowledge base and we prove that an interrogation for the object $x \in Obj(K)$ can be equivalently accomplished in the component which contains the object x. The factorization of a knowledge base is a useful operation in the vision of an implementation on several work stations in a network architecture. Several open problems are discussed in the last section.

2010 Mathematics Subject Classification. Primary 68T30; Secondary 68T37.
Key words and phrases. inheritance knowledge base, equivalence relation, factorization, global/local interrogation.

1. Introduction

Artificial Intelligence combines type hierarchies with other declarative information and calls the result a semantic network. Type hierarchies always reflect both the nature of the data they classify and the operations performed upon it. Recent intelligent systems have encoded knowledge bases whose volume is orders of magnitude smaller than real-world data bases, but whose internal complexity, heterogeneity, and variety of operations performed upon the data are vastly greater. In [3] were discussed various types of inheritance mechanisms in type hierarchies, followed by an attempt to unify these into a coherent, inheritance inference method.

Research in artificial intelligence based on the inheritance mechanism (theoretical and practical aspects) were treated in the last decade:

- an extension of the disjunctive logic programming (with strong negation) by inheritance ([2]);
- the inheritance of business rules in the medical insurance domain was studied in [4];
- a natural model-theoretic semantics for inheritance in frame-based knowledge bases, which supports inference by inheritance as well as inference via rules was treated in [11];
- the use of the lattice theory to characterize the features of the answer mapping in knowledge systems based on inheritance were described in [7], [8] and [9];
- the use of the voice interfaces to interrogate an inheritance based knowledge system is given in [10].

Received May 11, 2010. Revision received September 17, 2010.
C.I. Popîrlan has been supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007–2013.

In the present paper we develop the idea introduced in [6], decomposition of inheritance knowledge bases into disjoint components and mathematical study of the factorization problem. This paper is organized as follows:

- Section 2 contains the basic notions and results obtained in [6];
- In Section 3 we demonstrate that every component resulted from factorization mechanism is itself an inheritance knowledge base;
- In Section 4 we prove that an interrogation for the object $x \in Obj(K)$ can be equivalently accomplished in the component which contains the object x;
- The last section contains conclusions and future work.

The purpose of this research is to apply these results to accomplish an implementation based on multi-agent technology ([1]). In a forthcoming paper we propose a master-slave structure of agents that implements the results presented in this paper. The structure includes one master agent and several client agents. Every component can be uploaded on a work station in a network architecture, where a local agent can perform the computations. The master agent receives the interrogation and communicates with the slave agents. On the other hand, due to the fact that the search operation is a basic one in any implementation, the factorization of a knowledge base improves the running time of the computations performed by the answer mapping.

2. Basic notions and notations

In this section we present several algebraic properties of the binary relations, which are used in the next sections. We recall the following properties:

- The usual product operation $\circ$ between binary relations is defined as follows:

$$\rho_1 \circ \rho_2 = \{(x, y) \in X \times X \mid \exists z \in X : (x, z) \in \rho_1, (z, y) \in \rho_2\}$$

- The product operation is an associative one:

$$(\rho_1 \circ \rho_2) \circ \rho_3 = \rho_1 \circ (\rho_2 \circ \rho_3)$$

- The powers of the relation ρ are defined recursively as follows:

$$\begin{cases} \rho^1 = \rho \\ \rho^{n+1} = \rho^n \circ \rho, \ n \geq 0 \end{cases}$$

Definition 2.1. *If $\rho \subseteq X \times X$ is a binary relation then we define the following binary relations on X:*

$\rho^{-1} = \{ \ (y, x) \mid (x, y) \in \rho \ \}$
$\overline{\rho} = \rho \cup \rho^{-1}$
$\widetilde{\rho} = \bigcup_{i \geq 1} \overline{\rho}^{\ i}$

Definition 2.2. ([6]) *An element $x \in X$ is an* ***isolated element*** *with respect to ρ if there is no $y \in X$ such that $(x, y) \in \overline{\rho}$.*

Intuitively, a knowledge base K which uses the inheritance mechanism is a finite set of objects and an interrogation of K is defined by a pair (f, a_1), where f is the name of an object of K and a_1 is an attribute. The value of a attribute is of the form (v_1, q_1), where v_1 is a direct value of a_1 or the name of a procedure which returns the value of a_1 and q_1 is a parameter specifying some feature of a_1.

We consider a subset $K_0 \subseteq L_{obj} \times 2^{L_{obj}} \times 2^{L_{attr} \times (V_{dir} \cup L_{proc}) \times Param}$. If $x = (m, P, Q) \in K_0$ is an object then m determines uniquely the object x and we denote by $N(x) = m$ the name of x. For this reason an object is denoted by $x = (N(x), P_x, Q_x)$.

Definition 2.3. *([5]) An* ***inheritance knowledge base*** *is a pair $K = (Obj(K), \rho_K)$, where*

(1) *$Obj(K) \subseteq L_{obj} \times 2^{L_{obj}} \times 2^{L_{attr} \times (V_{dir} \cup L_{proc}) \times Param}$ is a finite set of elements named the* ***objects*** *of K, such that if $x = (N(x), P_1, Q_1) \in Obj(K)$, $y = (N(y), P_2, Q_2) \in Obj(K)$ and $N(x) = N(y)$ then $P_1 = P_2$ and $Q_1 = Q_2$.*

(2) *$\rho_K \subseteq Obj(K) \times Obj(K)$ is the relation generated by K, which is defined as follows:*

$$(x, y) \in \rho_K \Longleftrightarrow N(x) \in P_y$$

(3) *$\rho^i_K \cap \rho^j_K = \emptyset$ for $i \neq j$*

Definition 2.4. *([5]) An inheritance knowledge base K is an* ***accepted knowledge base*** *if the following conditions are fulfilled:*

- *There is no isolated object in K with respect to ρ_K.*
- *K contains minimal-ρ_K elements.*
- *inh_K is a strict partial order.*

Definition 2.5. *([5]) The relation $inh_K = \bigcup_{n \geq 1} \rho^n_K$ is named the* ***inheritance relation*** *generated by K.*

Definition 2.6. *([6]) An* ***interrogation*** *of a knowledge base K is an element of the set $Obj(K) \times Attr(K)$. The* ***answer*** *of an interrogation (x, a_1) is the value of the attribute a_1 for $x \in Obj(K)$.*

Notation 2.1. *([6]) The equivalence class of the object $x \in Obj(K)$ with respect to $\widetilde{\rho}_K$ is denoted by $[x]_{\widetilde{\rho}_K}$.*

In the vision of the computer network programming we are interested to decompose a knowledge base K which uses the inheritance mechanism into several components D_1, ..., D_m such that the following conditions are satisfied:

- $Obj(K) = \bigcup_{i=1}^{m} Obj(D_i)$, where $Obj(K)$ and $Obj(D_i)$ denotes the objects of K and D_i respectively..
- Two arbitrary components are disjoint sets of objects: $Obj(D_i) \cap Obj(D_j) = \emptyset$ for $i \neq j$.
- Each component D_i is itself a knowledge base. As a consequence we have a "global" interrogation (f, a_1) for K and a "local" interrogation (f, a_1) for D_i if $f \in Obj(D_i)$. The answer should be the same both for global and for local interrogation.
- For every $i \in \{1, \ldots, m\}$ the component D_i is an "atomic" knowledge base: it can not be itself divided into several knowledge bases.

We can say that m is the greatest number of components such that each component is an independent knowledge base that uses the inheritance mechanism. The problem specified above can be named the *factorization problem* of an inheritance knowledge base. This name comes from universal algebra domain, where the factor set X/ρ of the set X with respect to the equivalence relation ρ is the set of all equivalence classes.

The main aspects connected by our research presented in [6] can be shortly described as follows:

(1) Based on the inheritance mechanism from the knowledge base K was defined an equivalence relation $\widetilde{\rho}_K$ on the set of objects.
(2) Was proved that the set of all components of K is the factor set $K/\widetilde{\rho}_K$ and thus a component is an equivalence class.
(3) Was defined a method to find the components of a knowledge base. This method was based on the fact that a component is the equivalence class generated by some free of parents objects.

(4) Was given a necessary and sufficient condition for the factorization of a knowledge base.

3. The factorization problem

Proposition 3.1. *The mapping* $\omega : Initial(K)/\sigma_K \longrightarrow Obj(K)/\widetilde{\rho}_K$ *defined by* $\omega(X) = X_{\widetilde{\rho}_K}$ *is bijective.*

Proof. Take $X, Y \in Initial(K)/\sigma_K$ such that $\omega(X) = \omega(Y)$. Consider an element $z \in \omega(X) = \omega(Y) = X_{\widetilde{\rho}_K} = Y_{\widetilde{\rho}_K}$. If $r_1 \in X$ and $r_2 \in Y$ then by Proposition 5.5 from [6] we obtain $(r_1, z) \in \widetilde{\rho}_K$ and $(r_2, z) \in \widetilde{\rho}_K$. But $\widetilde{\rho}_K$ is an equivalence relation therefore $(r_1, r_2) \in \widetilde{\rho}_K$. We observe that $r_1, r_2 \in Initial(K)$ and we proved that $(r_1, r_2) \in \widetilde{\rho}_K$. It follows that $(r_1, r_2) \in \sigma_K$. But X is a σ_K-equivalence class and $r_1 \in X$. It follows that $X = [r_1]_{\sigma_K}$. In a similar manner we have $Y = [r_2]_{\sigma_K}$. Taking into consideration the property $(r_1, r_2) \in \sigma_K$ we obtain $X = Y$. Thus the mapping ω is injective.
Let us prove that ω is a surjective mapping. Take $T \in Obj(K)/\widetilde{\rho}_K$. From Proposition 6.5 from [6] we know that T is a component of K. In virtue of Proposition 6.1 from [6] we have $T \cap Initial(K) \in Initial(K)/\sigma_K$. We use now Proposition 5.7 from [6] and deduce that $T = X_{\widetilde{\rho}_K}$. Thus $\omega(X) = T$ and the proposition is proved. □

Corollary 3.1. $Card(Initial(K)/\sigma_K) = Card(Obj(K)/\widetilde{\rho}_K)$

Propositions 6.5 from [6] and 3.1 show that the space $Obj(K)/\widetilde{\rho}_K$ gives the components of a knowledge base and this space is the set $\{X_{\widetilde{\rho}_K} \mid X \in Initial(K)/\sigma_K\}$.

Proposition 3.2. *A knowledge base K can be decomposed into several components if and only if* $Card(Initial(K)/\sigma_K) \geq 2$.

Proof. Immediate from the previous propositions. □

Remark 3.1. *Obviously a knowledge base K can not be decomposed if and only if* $Card(Initial(K)/\sigma_K) = 1$.

Suppose that $Obj(K)/\widetilde{\rho}_K = \{Obj(D_1), \ldots, Obj(D_n)\}$, where $n \geq 2$. For each $i \in \{1, \ldots, n\}$ we denote $D_i = (Obj(D_i), \rho_{D_i})$. We prove that each D_i is an accepted knowledge base. We observe first that $Obj(D_i) \subseteq Obj(K)$. If $x = (N(x), P_x, Q_x) \in D_i$ and $y = (N(y), P_y, Q_y) \in D_i$ then using Definition 2.3 we obtain

$$(x, y) \in \rho_{D_i} \Longrightarrow N(x) \in P_y \tag{1}$$

But $(N(x), P_x, Q_x) \in K$ and $(N(y), P_y, Q_y) \in K$. Applying the same definition we obtain

$$N(x) \in P_y \Longrightarrow (x, y) \in \rho_K \tag{2}$$

From (1) and (2) we deduce that ρ_{D_i} is a restriction of the relation ρ_K:

$$\rho_{D_i} \subseteq \rho_K \tag{3}$$

Let us consider the objects $x = (N(x), P_x, Q_x) \in Obj(K)$ and $y = (N(y), P_y, Q_y) \in Obj(K)$ such that $(x, y) \in \rho_K$. In other words we have $N(x) \in P_y$. But $Obj(K) = \bigcup_{m=1}^{n} Obj(D_m)$. It follows that there are $i, j \in \{1, \ldots, n\}$ such that $x \in Obj(D_i)$ and $y \in Obj(D_j)$. We have $\rho_K \subseteq \widetilde{\rho}_K$ and $(x, y) \in \rho_K$, therefore $(x, y) \in \widetilde{\rho}_K$. But $Obj(D_i)$ is a $\widetilde{\rho}_K$-equivalence class and $x \in Obj(D_i)$. It follows that $y \in Obj(D_i)$. By our assumption we have $y \in Obj(D_j)$. This implies that $j = i$. Thus we have

$x \in Obj(D_i)$, $y \in Obj(D_i)$ and $N(x) \in P_y$, therefore $(x, y) \in \rho_{D_i}$. In conclusion we have

$$\bigcup_{m=1}^{n} \rho_{D_m} = \rho_K$$

Proposition 3.3. *Every component of an accepted knowledge base is an accepted knowledge base.*

Proof. The conditions specified in Definition 2.4 are satisfied:

(1) There is no useless object in D_i because $Obj(D_i) \subseteq Obj(K)$ and K does not contain such objects.

(2) D_i is a component of K. This means that there is a minimal-ρ_K element in D_i. But a minimal-ρ_K element in D_i is a minimal-ρ_{D_i} element. Thus there is a minimal-ρ_{D_i} element in D_i.

(3) The relation $\rho_{D_i} \subseteq Obj(D_i) \times Obj(D_i)$ is a strict partial order:
- We have $\Delta_{D_i} \cap \rho_{D_i} \subseteq \Delta(K) \cap \rho_K = \emptyset$, therefore $\Delta_{D_i} \cap \rho_{D_i} = \emptyset$.
- Suppose that $x, y, z \in Obj(D_i)$, $(x, y) \in \rho_{D_i}$ and $(y, z) \in \rho_{D_i}$. Because $Obj(D_i) \subseteq Obj(K)$ and ρ_{D_i} is a restriction of ρ_K we have $(x, z) \in \rho_K$. It follows that $x \in P_z$, therefore $(x, z) \in \rho_{D_i}$.

The proposition is proved. □

4. Global and local interrogations

We consider an accepted knowledge base K, which can be decomposed into m components and denote by $D_1 = (Obj(D_1), \rho_{D_1})$, …, $D_m = (Obj(D_m), \rho_{D_m})$ its components. An interrogation (x, a) is:

- a *global interrogation* addressed to K; the answer is obtained by a sequence of computations in K;

- a *local interrogation* addressed to D_i if $x \in Obj(D_i)$ and in this case the answer is given by a sequence of computations in D_i.

If $y \in Obj(K)$ then we denote by $a_1 \hookleftarrow\!\square\; y$ the fact that there is a slot of y which contains the attribute a_1. For every $(x, a_1) \in Obj(K) \times Attr(K)$ and $p \geq 1$ we consider the set

$$Parent^p_K(x, a_1) = \{\ y \in Obj(K) \mid (y, x) \in \rho^p_K, a_1 \hookleftarrow\!\square\; y\ \} \tag{4}$$

of all parents of order p in K for x that contain the attribute a_1. In a similar manner for every $(x, a_1) \in Obj(D_i) \times Attr(D_i)$ and $p \geq 1$ we consider the set

$$Parent^p_{D_i}(x, a_1) = \{\ y \in Obj(D_i) \mid (y, x) \in \rho^p_{D_i}, a_1 \hookleftarrow\!\square\; y\ \} \tag{5}$$

of all parents of order p in D_i for x that contain the attribute a_1.

The next two propositions are used in the remainder of this section.

Proposition 4.1. *If $y, x \in Obj(D_i)$ and $(y, x) \in \rho^p_K$ then $(y, x) \in \rho^p_{D_i}$.*

Proof. If $(y, x) \in \rho^p_K$ then there is z_1, …, $z_{p+1} \in Obj(K)$ such that
$y = z_1$
$(z_r, z_{r+1}) \in \rho_K$ for every $r \in \{1, \ldots, p\}$
$z_{p+1} = x$

But D_i is an equivalence class with respect to $\widetilde{\rho}_K$, $\rho_K \subseteq \widetilde{\rho}_K$ and $y \in Obj(D_i)$. It follows that $z_2 \in Obj(D_i)$. It is easy to observe that if $z_i \in Obj(D_i)$ then $z_{i+1} \in Obj(D_i)$, therefore $z_{p+1} \in Obj(D_i)$. Now, from $z_1, \ldots, z_{p+1} \in Obj(D_i)$ and $(z_r, z_{r+1}) \in \rho_K$ for every $r \in \{1, \ldots, p\}$ we obtain $(z_r, z_{r+1}) \in \rho_{D_i}$. It follows that $(y, x) \in \rho^p_{D_i}$. □

Proposition 4.2. *For every $p \geq 1$ and $(x, a_1) \in Obj(D_i) \times Attr(D_i)$ we have $Parent^p_{D_i}(x, a_1) = Parent^p_K(x, a_1)$.*

Proof. Suppose that $y \in Parent^p_{D_i}(x, a_1)$. From (5) we obtain

$$y \in Obj(D_i), (y, x) \in \rho^p_{D_i}, a_1 \hookleftarrow\!\Box\; y \tag{6}$$

But $Obj(D_i) \subseteq Obj(K)$ and $\rho_{D_i} \subseteq \rho_K$, therefore

$$y \in Obj(K), (y, x) \in \rho^p_K, a_1 \hookleftarrow\!\Box\; y \tag{7}$$

From (7) and (4) we obtain $y \in Parent^p_K(x, a_1)$, therefore

$$Parent^p_{D_i}(x, a_1) \subseteq Parent^p_K(x, a_1) \tag{8}$$

Take now $y \in Parent^p_K(x, a_1)$. From (4) we obtain (7). But $(x, a_1) \in Obj(D_i) \times Attr(D_i)$ and D_i is a component of K. A component of K is an equivalence class with respect to $\widetilde{\rho}_K$, therefore from $(y, x) \in \rho^p_K$, $x \in Obj(D_i)$ and $\rho^p_K \subseteq \widetilde{\rho}_K$ we deduce that $y \in Obj(D_i)$. From Proposition 4.1 we obtain $(y, x) \in \rho^p_{D_i}$, therefore we have (6). It follows that $y \in Parent^p_{D_i}(x, a_1)$ and thus

$$Parent^p_K(x, a_1) \subseteq Parent^p_{D_i}(x, a_1) \tag{9}$$

From (8) and (9) we obtain $Parent^p_{D_i}(x, a_1) = Parent^p_K(x, a_1)$. □

We denote by Val^K_{attr} and $Val^{D_i}_{attr}$ the mappings which compute the value of an attribute in K and D_i respectively. In the next proposition we show that Val^K_{attr} extends the mappings $Val^{D_i}_{attr}$ for $i \in \{1, \ldots, m\}$.

Proposition 4.3. *$Val^K_{attr}(x, a_1) = Val^{D_i}_{attr}(x, a_1)$ for $(x, a_1) \in Obj(D_i) \times Attr(D_i)$.*

Proof. We define the *computing environment* for $Val^K_{attr}(x, a_1)$, which is denoted by $Env(Val^K_{attr}(x, a_1))$. This entity contains all objects from $Obj(K)$ which are used to compute $Val^K_{attr}(x, a_1)$. We define

$$Env(Val^K_{attr}(x, a_1)) = \bigcup_{n \geq 0} R^K_n$$

where the sequence $\{R^K_n\}_{n \geq 0}$ is defined as follows:

$$\begin{cases} R^K_0 = \{x\ \} \\ \\ R^K_{n+1} = \{y \in Obj(K) \mid \exists t \in R^K_n : (y, t) \in \rho_K\ \} \end{cases}$$

In a similar manner we define

$$Env(Val^{D_i}_{attr}(x, a_1)) = \bigcup_{n \geq 0} Q^{D_i}_n$$

where the sequence $\{Q^{D_i}_n\}_{n \geq 0}$ is obtained as follows:

$$\begin{cases} Q^{D_i}_0 = \{x\ \} \\ \\ Q^{D_i}_{n+1} = \{y \in Obj(D_i) \mid \exists t \in Q^K_n : (y, t) \in \rho_{D_i}\ \} \end{cases}$$

Let us take an element $x \in Obj(D_i)$. We prove that

$$Env(Val^K_{attr}(x, a_1)) = Env(Val^{D_i}_{attr}(x, a_1))$$

More precisely, we prove by induction on $n \geq 0$ that $R^K_n = Q^{D_i}_n$. For $n = 0$ this property is obviously true. Suppose that $R^K_n = Q^{D_i}_n$ and let us prove that $R^K_{n+1} = Q^{D_i}_{n+1}$. Because $Obj(D_i) \subseteq Obj(K)$ and ρ_{D_i} is a restriction of ρ_K we have $Q^{D_i}_{n+1} \subseteq R^K_{n+1}$.

It remains to prove that $R^K_{n+1} \subseteq Q^{D_i}_{n+1}$. Take an element $y \in R^K_{n+1}$. There is $t \in R^K_n$ such that $(y,t) \in \rho_K$. From the following facts

- $Obj(D_i)$ is a $\widetilde{\rho}_K$-equivalence class
- $Q^{D_i}_n \subseteq Obj(D_i)$
- $t \in R^K_n = Q^{D_i}_n$

we deduce that $y \in Obj(D_i)$. Now, from the following facts

- $t, y \in Obj(D_i)$
- $(y,t) \in \rho_K$
- ρ_{D_i} is the restriction of ρ_k to $Obj(D_i)$

we deduce that $(y,t) \in \rho_{D_i}$. Finally we observe that from

- $t \in Q^{D_i}_n$ and $y \in Obj(D_i)$
- $(y,t) \in \rho_{D_i}$

we deduce that $y \in Q^{D_i}_{n+1}$. Thus we proved that $R^K_{n+1} \subseteq Q^{D_i}_{n+1}$. Taking into account the converse inclusion we have $R^K_{n+1} = Q^{D_i}_{n+1}$. In conclusion we have $R^K_n = Q^{D_i}_n$ for every $n \geq 0$.

Now we observe that in order to compute the value $Val^K_{attr}(x, a_1)$ only the objects from $Env(Val^K_{attr}(x, a_1))$ are used. In the same manner, to compute $Val^{D_i}_{attr}(x, a_1)$ only the elements from $Env(Val^{D_i}_{attr}(x, a_1))$ are used. But we proved above that $Env(Val^K_{attr}(x, a_1)) = Env(Val^{D_i}_{attr}(x, a_1))$. □

5. Conclusions and future work

In this paper some demonstrations based on factorization problem ([6]) which prove that an interrogation for the object $x \in Obj(K)$ can be equivalently accomplished in the component which contains the object x were presented. The main problem studied in this paper was connected by the factorization of an inheritance knowledge base([5]).

The factorization is a useful operation for large knowledge bases. The factorization of a knowledge base allows to upload each component on a work station in a network architecture. The local computation on a component gives the same result as the global computation for the entire knowledge base. The replication of a component becomes a possible problem, as we proceed in the domain of databases. The computation presented in this paper is not a distributed one.

In the future we are interested to imply the mobile agents in a master-slave structure that implements the results presented in these papers. Every component can be uploaded on a work station in a network architecture, where a local agent can perform the computations. We intend to develop a research line concerning the modeling of the distributed knowledge by inheritance.

References

[1] J. Baumann, Mobile Agents: Control Algorithms, Lecture Notes in Computer Science, Springer-Berlin, 2000.

[2] F. Buccafurri, W. Faber and N. Leone, Disjunctive logic programs with inheritance, *Proceedings of ICLP-99*, MIT Press, (1999), 79–93.

[3] J.G. Carbonell, Default Reasoning and Inheritance Mechanisms on Type Hierarchies, *Computer Science Department* (1980), Paper 366. `http://repository.cmu.edu/compsci/366`

[4] L. Morgenstern, Inheritance comes of age: Applying non monotonic techniques to problems in industry, *Artificial Intelligence* **103** (1998), 237–271.

[5] C.-I. Popirlan and N. Ţăndăreanu, An Extension of Inheritance Knowledge Bases and Computational Properties of their Answer Functions, *Annals of the University of Craiova* **35** (2008), 149–170.

[6] N. Ţăndăreanu and C.-I. Popirlan, Factorization of an Inheritance Knowledge Base (I), *Annals of the University of Craiova, Mathematics and Computer Science Series* **37** (2010), no. 2, 62–74.

[7] N. Ţăndăreanu, Lattices of Labelled Ordered Trees (I), *Annals of the University of Craiova, Mathematics and Computer Science Series* **28** (2001), 29–39.

[8] N. Ţăndăreanu, Lattices of Labelled Ordered Trees (II), *Annals of the University of Craiova, Mathematics and Computer Science Series* **29** (2002), 137–144.

[9] N. Ţăndăreanu, Inheritance-based knowledge systems and their answer functions computation using lattice theory, *Romanian Journal of Information Science and Technology* **6** (2003), no. 1-2, 227–248.

[10] N. Ţăndăreanu, Communication by Voice to Interrogate an Inheritance Based Knowledge System, *Proceedings of the 7th International Conference on Artificial Intelligence and Digital Communications - AIDC 2007*, Craiova, Romania, Research Notes in Artificial Intelligence and Digital Communications **107**, Craiova (2007), 1–15.

[11] G. Yang and M. Kifer, Well-Founded Optimism: Inheritance in Frame-Based Knowledge Bases. In *Lecture Notes in Computer Science* **2519** (2008), ISBN 978-3-540-00106-5, 1013–1032.

(Nicolae Ţăndăreanu and Claudiu-Ionuţ Popîrlan) Faculty of Mathematics and Computer Science, University of Craiova, Al.I. Cuza Street, No. 13, Craiova RO-200585, Romania, Tel. & Fax: 40-251412673
E-mail address: `ntand@rdslink.ro, popirlan@inf.ucv.ro`