Factorization of an inheritance knowledge base (II)

NICOLAE ȚĂNDĂREANU AND CLAUDIU-IONUȚ POPĂRLAN

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. Popărlan 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 \( x \in \text{Obj}(K) \) 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 \( \rho \) 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 \( X \):

- \( \rho^{-1} = \{ (y,x) \mid (x,y) \in \rho \} \)
- \( \bar{\rho} = \rho \cup \rho^{-1} \)
- \( \tilde{\rho} = \bigcup_{i \geq 1} \tilde{\rho}^i \)

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

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 \( K_0 \subseteq \text{Lobj} \times 2^{\text{Lobj}} \times 2^{\text{Lattr} \times \text{Vdir} \cup \text{Lproc} \times \text{Param}}. \) 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 \( (\text{Obj}(K), \rho_K) \), where:

1. \( \text{Obj}(K) \subseteq L_{\text{obj}} \times 2^{L_{\text{obj}}} \times 2^{L_{\text{attr}}} \times \text{Var} \times \text{Func} \times \text{Param} \) is a finite set of elements named the *objects* of \( K \), such that if \( x = (N(x), P_1, Q_1) \in \text{Obj}(K), y = (N(y), P_2, Q_2) \in \text{Obj}(K) \) and \( N(x) = N(y) \) then \( P_1 = P_2 \) and \( Q_1 = Q_2 \).
2. \( \rho_K \subseteq \text{Obj}(K) \times \text{Obj}(K) \) is the relation generated by \( K \), which is defined as follows:
   \[
   (x, y) \in \rho_K \iff N(x) \in P_y
   \]
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 \( \rho_K \).
- \( K \) contains minimal-\( \rho_K \) elements.
- \( \text{inh}_K \) is a strict partial order.

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

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

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 \( D_1, \ldots, D_m \) such that the following conditions are satisfied:

- \( \text{Obj}(K) = \bigcup_{i=1}^{m} \text{Obj}(D_i) \), where \( \text{Obj}(K) \) and \( \text{Obj}(D_i) \) denotes the objects of \( K \) and \( D_i \) respectively.
- Two arbitrary components are disjoint sets of objects: \( \text{Obj}(D_i) \cap \text{Obj}(D_j) = \emptyset \) 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 \( f \in \text{Obj}(D_i) \). 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 \( X/\rho \) of the set \( X \) with respect to the equivalence relation \( \rho \) 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 \( \tilde{\rho}_K \) on the set of objects.
2. Was proved that the set of all components of \( K \) is the factor set \( K/\tilde{\rho}_K \) 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.
3. The factorization problem

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

**Proof.** Take \( X, Y \in \text{Initial}(K)/\sigma_K \) such that \( \omega(X) = \omega(Y) \). Consider an element \( z \in \omega(X) = \omega(Y) = X_{\tilde{\rho}_K} = Y_{\tilde{\rho}_K} \). If \( r_1 \in X \) and \( r_2 \in Y \) then by Proposition 5.5 from [6] we obtain \( (r_1, z) \in \tilde{\rho}_K \) and \( (r_2, z) \in \tilde{\rho}_K \). But \( \tilde{\rho}_K \) is an equivalence relation therefore \( (r_1, r_2) \in \tilde{\rho}_K \). We observe that \( r_1, r_2 \in \text{Initial}(K) \) and we proved that \( (r_1, r_2) \in \tilde{\rho}_K \). It follows that \( (r_1, r_2) \in \sigma_K \). But \( X \) is a \( \sigma_K \)-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 \( \omega \) is injective.

Let us prove that \( \omega \) is a surjective mapping. Take \( T \in \text{Obj}(K)/\tilde{\rho}_K \). 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 \( T \cap \text{Initial}(K) \in \text{Initial}(K)/\sigma_K \). We use now Proposition 5.7 from [6] and deduce that \( T = X_{\tilde{\rho}_K} \). Thus \( \omega(X) = T \) and the proposition is proved. \( \square \)

**Corollary 3.1.** \( \text{Card}(\text{Initial}(K)/\sigma_K) = \text{Card}(\text{Obj}(K)/\tilde{\rho}_K) \)

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

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

**Proof.** Immediate from the previous propositions. \( \square \)

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

Suppose that \( \text{Obj}(K)/\tilde{\rho}_K = \{\text{Obj}(D_1), \ldots, \text{Obj}(D_n)\} \), where \( n \geq 2 \). For each \( i \in \{1, \ldots, n\} \) we denote \( D_i = (\text{Obj}(D_i), \rho_{D_i}) \). We prove that each \( D_i \) is an accepted knowledge base. We observe first that \( \text{Obj}(D_i) \subseteq \text{Obj}(K) \). 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

\[
(x, y) \in \rho_{D_i} \implies N(x) \in P_y \tag{1}
\]

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

\[
N(x) \in P_y \implies (x, y) \in \rho_K \tag{2}
\]

From (1) and (2) we deduce that \( \rho_{D_i} \) is a restriction of the relation \( \rho_K \):

\[
\rho_{D_i} \subseteq \rho_K \tag{3}
\]

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

**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 $\text{Obj}(D_i) \subseteq \text{Obj}(K)$ and $K$ does not contain such objects.
2. $D_i$ is a component of $K$. This means that there is a minimal-$\rho_K$ element in $D_i$. But a minimal-$\rho_K$ element in $D_i$ is a minimal-$\rho_K$ element. Thus there is a minimal-$\rho_K$ element in $D_i$.
3. The relation $\rho_D \subseteq \text{Obj}(D_i) \times \text{Obj}(D_i)$ is a strict partial order:
   - We have $\Delta_D \cap \rho_D \subseteq \Delta(K) \cap \rho_K = \emptyset$, therefore $\Delta_D \cap \rho_D = \emptyset$.
   - Suppose that $x, y, z \in \text{Obj}(D_i)$, $(x, y) \in \rho_D$, and $(y, z) \in \rho_D$. Because $\text{Obj}(D_i) \subseteq \text{Obj}(K)$ and $\rho_D$ is a restriction of $\rho_K$ we have $(x, z) \in \rho_K$. It follows that $x \in P_z$, therefore $(x, z) \in \rho_D$.

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 $D_1 = (\text{Obj}(D_1), \rho_{D_1}), \ldots, D_m = (\text{Obj}(D_m), \rho_{D_m})$ 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 $x \in \text{Obj}(D_i)$ and in this case the answer is given by a sequence of computations in $D_i$.

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

$$\text{Parent}^p_K(x, a_1) = \{ y \in \text{Obj}(K) \mid (y, x) \in \rho^p_K, a_1 \leftarrow \square y \} \quad (4)$$

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

$$\text{Parent}^p_{D_i}(x, a_1) = \{ y \in \text{Obj}(D_i) \mid (y, x) \in \rho^p_{D_i}, a_1 \leftarrow \square y \} \quad (5)$$

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 $(y, x) \in \rho^p_K$ then $(y, x) \in \rho^p_{D_i}$.

**Proof.** If $(y, x) \in \rho^p_K$ then there is $z_1, \ldots, z_{p+1} \in \text{Obj}(K)$ such that $y = z_1$

$$(z_r, z_{r+1}) \in \rho_K \text{ for every } r \in \{1, \ldots, p\}$$

$z_{p+1} = x$

But $D_i$ is an equivalence class with respect to $\rho_K$, $\rho_K \subseteq \rho_D$ and $y \in \text{Obj}(D_i)$. It follows that $z_2 \in \text{Obj}(D_i)$. It is easy to observe that if $z_i \in \text{Obj}(D_i)$ then $z_{r+1} \in \text{Obj}(D_i)$, therefore $z_{p+1} \in \text{Obj}(D_i)$. Now, from $z_1, \ldots, z_{p+1} \in \text{Obj}(D_i)$ 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 $(y, x) \in \rho_{D_i}$. □
For every $p \geq 1$ and $(x, a_1) \in \text{Obj}(D_i) \times \text{Attr}(D_i)$ we have $\text{Parent}^p_{K}(x, a_1) = \text{Parent}^p_{D_i}(x, a_1)$.

Proof. Suppose that $y \in \text{Parent}^p_{D_i}(x, a_1)$. From (5) we obtain
\[ y \in \text{Obj}(D_i), (y, x) \in \rho_{D_i}^p, a_1 \leftrightarrow y \] (6)
But $\text{Obj}(D_i) \subseteq \text{Obj}(K)$ and $\rho_{D_i} \subseteq \rho_K$, therefore
\[ y \in \text{Obj}(K), (y, x) \in \rho_K^p, a_1 \leftrightarrow y \] (7)
From (7) and (4) we obtain $y \in \text{Parent}^p_K(x, a_1)$, therefore
\[ \text{Parent}^p_{D_i}(x, a_1) \subseteq \text{Parent}^p_K(x, a_1) \] (8)
Take now $y \in \text{Parent}^p_K(x, a_1)$. From (4) we obtain (7). But $(x, a_1) \in \text{Obj}(D_i) \times \text{Attr}(D_i)$ and $D_i$ is a component of $K$. A component of $K$ is an equivalence class with respect to $\rho_K$, therefore from $(y, x) \in \rho_K^p, x \in \text{Obj}(D_i)$ and $\rho_{D_i}^p \subseteq \rho_K$ we deduce that $y \in \text{Obj}(D_i)$. From Proposition 4.1 we obtain $(y, x) \in \rho_{D_i}^p$, therefore we have (6). It follows that $y \in \text{Parent}^p_{D_i}(x, a_1)$ and thus
\[ \text{Parent}^p_K(x, a_1) \subseteq \text{Parent}^p_{D_i}(x, a_1) \] (9)
From (8) and (9) we obtain $\text{Parent}^p_{D_i}(x, a_1) = \text{Parent}^p_K(x, a_1)$. □

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

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

Proof. We define the computing environment for $\text{Val}^K_{\text{attr}}(x, a_1)$, which is denoted by $\text{Env}(\text{Val}^K_{\text{attr}}(x, a_1))$. This entity contains all objects from $\text{Obj}(K)$ which are used to compute $\text{Val}^K_{\text{attr}}(x, a_1)$. We define
\[ \text{Env}(\text{Val}^K_{\text{attr}}(x, a_1)) = \bigcup_{n \geq 0} R^n_K \]
where the sequence $\{R^n_K\}_{n \geq 0}$ is defined as follows:
\[
\begin{cases}
R^0_K = \{x\} \\
R^n_K = \{y \in \text{Obj}(K) | \exists t \in R^n_K : (y, t) \in \rho_K\}
\end{cases}
\]
In a similar manner we define
\[ \text{Env}(\text{Val}^{D_i}_{\text{attr}}(x, a_1)) = \bigcup_{n \geq 0} Q^n_{D_i} \]
where the sequence $\{Q^n_{D_i}\}_{n \geq 0}$ is obtained as follows:
\[
\begin{cases}
Q^0_{D_i} = \{x\} \\
Q^n_{D_i} = \{y \in \text{Obj}(D_i) | \exists t \in Q^n_{D_i} : (y, t) \in \rho_{D_i}\}
\end{cases}
\]
Let us take an element $x \in \text{Obj}(D_i)$. We prove that
\[ \text{Env}(\text{Val}^K_{\text{attr}}(x, a_1)) = \text{Env}(\text{Val}^{D_i}_{\text{attr}}(x, a_1)) \]
More precisely, we prove by induction on $n \geq 0$ that $R^n_K = Q^n_{D_i}$. For $n = 0$ this property is obviously true. Suppose that $R^n_K = Q^n_{D_i}$ and let us prove that $R^{n+1}_K = Q^{n+1}_{D_i}$. Because $\text{Obj}(D_i) \subseteq \text{Obj}(K)$ and $\rho_{D_i}$ is a restriction of $\rho_K$ we have $Q^{n+1}_{D_i} \subseteq R^{n+1}_K$. 
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
- $\text{Obj}(D_i)$ is a $\tilde{\rho}_K$-equivalence class
- $Q^{D_i}_n \subseteq \text{Obj}(D_i)$
- $t \in R^K_n = Q^{D_i}_n$
we deduce that $y \in \text{Obj}(D_i)$. Now, from the following facts
- $t, y \in \text{Obj}(D_i)$
- $(y, t) \in \rho_K$
- $\rho_{D_i}$ is the restriction of $\rho_K$ to $\text{Obj}(D_i)$
we deduce that $(y, t) \in \rho_{D_i}$. Finally we observe that from
- $t \in Q^{D_i}_n$ and $y \in \text{Obj}(D_i)$
- $(y, t) \in \rho_{D_i}$
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 $\text{Val}^{K\text{attr}}(x, a_1)$ only the objects from $\text{Env}(\text{Val}^{K\text{attr}}(x, a_1))$ are used. In the same manner, to compute $\text{Val}^{D_i\text{attr}}(x, a_1)$ only the elements from $\text{Env}(\text{Val}^{D_i\text{attr}}(x, a_1))$ are used. But we proved above that $\text{Env}(\text{Val}^{K\text{attr}}(x, a_1)) = \text{Env}(\text{Val}^{D_i\text{attr}}(x, a_1))$. □

5. Conclusions and future work

In this paper some demonstrations based on factorization problem ([6]) which prove that an interrogation for the object $x \in \text{Obj}(K)$ 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

N. TĂNDĂREANU AND C.-I. POPĂRLAN


(Nicolae Tăndăreanu and Claudiu-Ionuţ Popărlan) 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