Answering Non-Standard Queries in Distributed Knowledge-Based Systems

Zbigniew W. Ras

University of North Carolina
Department of Comp. Science
Charlotte, N.C. 28223, USA
ras@uncc.edu

Abstract

In this paper we present a query answering system for solving non-standard queries in a distributed knowledge based system (DKBS). Our system is different from solving queries on a conventional distributed database or cooperative database in the sense that it discovers rules, if needed, and uses them to resolve unknown attributes. In [12], the rules used to resolve unknown attributes are discovered directly from the tables (relational databases) either locally or on remote sites. In this paper, the rule discovery process is dependent on descriptions of objects which will never be stored in our system (they either do not exist or we have no interest in storing them). Such descriptions are called either locally-negative (l-negative) or globally-negative (g-negative) terms. L-negative terms refer to the situation when only a local site of DKBS is taken into cosideration. If any site of DKBS is considered for storing the data, we use g-negative terms instead.

1 Introduction

By a distributed knowledge-based system (DKBS) we mean a collection of autonomous knowledge-based systems called agents which are capable of interacting with one another. Each agent is represented by an information system (collection of data) with structured attributes, a knowledge-based system (collection of rules and negative terms), and a query answering system based on Client/Server schema.

Each agent can be a source of a non-standard query. We will consider two types of queries:

- queries asking for objects in a local information system satisfying a given description (o-queries),
- queries asking for rules describing a local attribute value in terms of a group of local attributes (r-queries)

By a local query for a given agent we mean a query entirely built from values of attributes local for that agent. Otherwise, a query is called global (non-standard). To resolve a local o-query, we use a cooperative approach similar to the one proposed by Chu [1], Gaasterland [2], and others. In order to resolve a global o-query for a site i (called a client), information systems at other sites (called servers) have to be contacted. To be more precise, the client site will search for servers which can resolve unknown attribute values used in a global o-query. Such servers will try to discover approximate descriptions of these unknown attribute values, from their information systems, in a form of rules and if they succeed, they will send these descriptions to the client site. These sets of rules are sound at the sites they have been discovered (they can only overlap on g-negative terms) but clearly they do not have to be sound at the client site. If more than one server site sends these rules to the client site, then the new set of rules at the client site has to be checked for consistency. If the result is negative, then this set of rules has to be repaired. The repair algorithm is successful if condition parts of initially inconsistent rules overlap at the client site only on g-negative and l-negative terms.

The query answering system at the client site is using these newly discovered and repaired (if needed) rules to resolve a global o-query. In a case of a local r-query, we use a modified LERS system (the overlaps on both g-negative and l-negative condition parts of the rules are allowed).

Our system is different from solving queries on a conventional relational database or from solving queries in a cooperative information system ([1],[2]) in the sense that it uses rules discovered on remote servers to resolve unknown attributes.

2 Basic Definitions

In this section, we introduce the notion of an attribute tree, an information system which is a generalization of Pawlak's system [10], an information system with negative constraints (called nc-system), a distributed information system (DIS), and finally we give definitions of local and global queries for one of the sites of DIS.
To simplify some definitions, attributes and attribute values are called attributes in this paper.
By an attribute tree we mean a pair $(V, \leq)$ such that:

- $(V, \leq)$ is a partially ordered set of attributes,
- $(\forall a, b, c \in V)[(a \leq b \wedge c \leq b) => (a \leq c \vee c \leq a)]$,
- $(\forall a, b \in V)(\exists c \in V)(c \leq a \wedge c \leq b)$,
- $(\forall a)[$ a has minimum two children or a is a leaf$]$.

We say here that b is a child of a if $\sim (\exists c)[c \neq a \wedge c \neq b \wedge a \leq c \leq b]$.

Let $(V, \leq)$ and $(U, \leq)$ are attribute trees. We say that $(U, \leq)$ is a subtree of $(V, \leq)$ if $U \subseteq V$ and $(\forall a \in U)(\forall c \in V)(a \leq c \Rightarrow c \in U)$.

Information system S is defined as a sequence $(X, V, \leq, f)$, where X is a set of objects, V is a set of attributes and f is a classification function. We assume that:

- $V = \bigcup\{V_i : i \in I\}$ and $(V_i, \leq)$ is an attribute tree for any $i \in I$,
- $V_i \cap V_j = \emptyset$ for any $i, j \in I$,
- $f : X \times I \longrightarrow 2^V$ where $(f(x,i), \leq)$ is a subtree of $(V_i, \leq)$ for any $i \in I$.

Clearly card(I) is equal to the number of maximal subtrees in $(V, \leq)$. We interpret I as the set of attribute names in the system S. The root of the tree $(f(x,i), \leq)$ gives the value of the attribute i for an object x and the set $f(x,i)$ gives all possible values of the attribute i for x. If $(\exists x)[f(x,i) = V_i]$, then the value of the attribute i for an object x in S is unknown.

Example 1. Let us assume that the value $f(x, color)$ of the attribute x is represented by the Figure 1 given below.

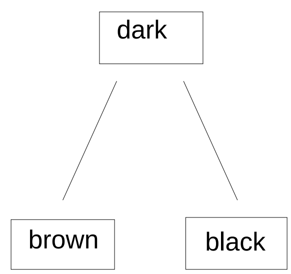

Fig. 1. Value of the attribute color for object x

In this case the color of x is dark and it can be either brown or black. □

With each information system $S = (X, V, \leq, f)$, we link a formal language $L(S)$ called a description language or query language (see [7]). If only the attributes of S are taken as the descriptors of $L(S)$, then $L(S)$ is called local for S (see [7], [12]). If descriptors of $L(S)$ contain some attributes which are not from S, then $L(S)$ is not local. In this paper, we mainly deal with query languages which are not local (we call them global for S).

Let us be more precise. By a set of S-terms for $S = (X, V, \leq, f)$, $V = \bigcup\{V_i : i \in I\}$ we mean a least set T_S such that:

- if $v \in V_i$ then $(i, v) \in T_S$, for any $i \in I$
- if $t_1, t_2 \in T_S$ then $(t_1 + t_2), (t_1 * t_2), \sim t_1 \in T_S$.

We say that:

- S-term t is *atomic* if it is of the form (i, w) or $\sim (i, w)$ where $w \in V_i$,
- S-term t is *positive* if it is of the form $\prod\{(i, w) : w \in V_i\}$,

- S-term t is *primitive* if it is of the form $\prod\{t_j : t_j$ is atomic $\}$,
- S-term is in *disjunctive normal form* (DNF) if $t = \sum\{t_j : j \in J\}$ where each t_j is primitive.

By a local o-query for S we mean any element in T_S which is in DNF. Informally, o-query $t \in T_S$ can be read as:

find all objects in X which descriptions are consistent with query t.

By a local r-query (called in this paper r-query) for S we mean either a pair $((i,w), I_1)$ or $(\sim (i,w), I_1)$, where $i \in I - I_1$ and $I_1 \subset I$. Correspondingly, we can read such r-queries as:

describe (i,w) in terms of attributes from I,
describe $\sim (i,w)$ in terms of attributes from I.

Before we give the semantics (interpretation J_S) of local o-queries for S and r-queries for S, where $S = (X, V, \leq, f)$, we introduce function $\bar{f}$. Let us assume that $S = (X, V, \leq, f)$ is an information system, where $V = \bigcup\{V_i : i \in I\}$. Then, function $\bar{f}$ is defined by two conditions below:

- $\bar{f} : X \times I \longrightarrow 2^V$,
- $\bar{f}(x,i)$ is a root of the tree $(f(x,i), \leq)$.

The set $\{\bar{f}(x,i) : i \in I\}$ contains values of attributes which conjunct gives the most specific description of x which is known by the agent represented by S.

By an S-rule we mean either a pair $[(i,w),t]$ or $[\sim (i,w),t]$, where t is an S_1-term in DNF and $S_1 = (X, \bigcup\{V_j : j \in I - \{i\}\}, \leq, f)$.

Now, let us assume that $S = (X, V, \leq, f)$, $V = \bigcup\{V_i : i \in I\}$ and $v \in V_i$. By $Ant(v,i)$ we mean the smallest subset of $Neg(v,i) = \{w \in V_i : \sim (w \leq v) \& \sim (v \leq w)\}$ such that:
if $w_1 \in Neg(v,i)$, then $(\exists w_2 \in Ant(v,i))(w_2 \leq w_1)$.

Example 2. Let us assume that $V_{color} = \{$*dark*, *bright*, *brown*, *black*, *gray*, *yellow*, *white*, *blue*$\}$ is a set of values of the attribute *color* both represented by Figure 2.
Then, $Ant(blue, color) = \{white,\ yellow,\ dark\}$. □

Terms t_1, t_2 are called contradictory if:

- there is (i, w_1) which is a subterm of t_1,
- there is (i, w_2) which is a subterm of t_2,
- the set $\{w_1, w_2\}$ is an antichain in $(V_i, \leq)$.

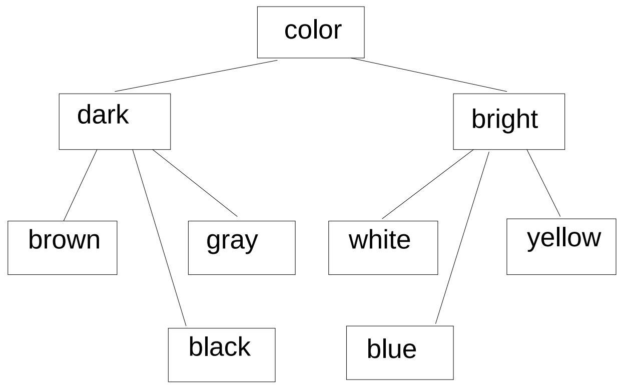

Fig. 2. Structured attribute *color*.

The interpretation J_S of local o-queries for S in $S = (X, V, \leq, f)$, $V = \bigcup\{V_i : i \in I\}$ is given below:

- $J_S((i, v)) = \{x \in X : v \leq \bar{f}(x, i)\}$,
- $J_S(\sim (i, v)) = \bigcup\{\{x \in X : w \leq \bar{f}(x, i)\} : w \in Ant(v, i)\}$,
- if t_1, t_2 are S-terms, then
  $J_S(t_1 + t_2) = J_S(t_1) \cup J_S(t_2)$,
  $J_S(t_1 * t_2) = J_S(t_1) \cap J_S(t_2)$.

Assume now that $r = [t_1, t_2]$, where $t_2 = \prod\{t_j : j \in J_1\}$, is an S-rule. We say that:

- r is valid in S if $J_S(t_2) \subset J_S(t_1)$,
- r is simple if t_2 is positive,
- r is optimal if r is valid and simple and there is no other valid and simple rule $[t_1, t_3]$ in S, such that
  $t_3 = \prod\{s_k : k \in J_2\}$,
  $(\forall k \in J_2)(\exists j \in J_1)(s_k \leq t_j)$,
  $(\forall j \in J_1)(\exists k \in J_2)(s_k \leq t_j)$.

By an information system with negative constraints (nc-system) we mean a pair (S, N), where $S = (X, V, \leq, f)$, is an information system and N is a set of primitive terms called negative constraints for S. A term t is a negative constraint for S if $J_S(t) = \emptyset$.

Let q be a local r-query for a nc-system (S, N). By nc-interpretation of local r-queries in (S, N) we mean any function J_S satisfying three conditions below:

- if $q = ((i, w), I_1)$, then $J_S(q)$ is a non-empty set of optimal S-rules describing (i, w) in terms of values of attributes from $\bigcup\{V_j : j \in I_1\}$,

- if $q = (\sim (i,w), I_1)$, then $J_S(q)$ is a non-empty set of optimal S-rules describing $\sim (i,w)$ in terms of values of attributes from $\bigcup\{V_j : j \in I_1\}$,
- if $((i,w_1),t_1) \in J_S(q_1)$, $((i,w_2),t_2) \in J_S(q_2)$, $w_1 \neq w_2$ then either $(\exists t \in N)(t$ is a subterm of $t_1 * t_2)$ or terms t_1, t_2 are contradictory.

We say that J_S is standard if:

- for any antichain $\{v_1, v_2\} \subset V_i$, $J_S(v_1) \cap J_S(v_2) = \emptyset$,
- $(\forall v \in V_i)[X - J_S(v) = \bigcup\{J_S(u) : u \in Ant(v,i)\}]$.

The class of standard nc-interpretations is the simplest class for which the results presented in [13], [14] (including completeness theorem) are naturally extended. In this paper we plan to outline the methodology for answering o-queries and r-queries in a distributed information system. We assume here that each system (agent) knows, according to his experience, both locally-negative terms and globally-negative terms. As we have mentioned earlier, a $locally-negative\ term$ refers to the situation when objects consistent with that term either do not exist or will never be stored at the client site. Similarly, a $globally-negative\ term$ refers to the situation when objects consistent with that term either do not exist or will never be stored at any site of our distributed information system.

We begin with the definition below:

By a distributed information system we mean a pair $DS = (\{(S_k, N_k)\}_{k \in K}, L)$ where:

- $S_k = (X_k, V_k, \leq, f_k)$ is an information system for any $k \in K$,
- $N_k = N_{l_k} \cup N_{g_k}$ is a set of negative constraints for S_k,
- N_{l_k} is a set of locally-negative constraints for S_k,
- N_{g_k} is a set of globally-negative constraints for S_k,
- $(\forall k1, k2 \in K)[N_{g_{k1}} = N_{g_{k2}}]$,
- L is a symmetric, binary relation on the set K,
- K is a set of sites.

We assume here that $V_k = \bigcup\{V_{\leq k,i>} : i \in I_k\}$.

Systems $(S_{k1}, N_{k1}), (S_{k2}, N_{k2})$ are called neighbors in a distributed information system DS if $(k1,k2) \in L$. The transitive closure of L in K is denoted by L^*.

Before we introduce o-queries and r-queries for a distributed information system DS, we generalize first the definition of S_k-terms. By a set of DS-terms for $DS = (\{(S_k, N_k)\}_{k \in K}, L)$ we mean a least set T_{DS} such that:

- if $v \in \bigcup\{V_k : k \in K\}$ then $v \in T_{DS}$,
- if $t_1, t_2 \in T_{DS}$ then $(t_1 + t_2), (t_1 * t_2), \sim t_1 \in T_{DS}$.

By o-query for DS we mean any element in T_{DS} which is in DNF.

By r-query for DS we mean either a pair $((i,w),I)$ or $(\sim (i,w),I)$, where $i \in \bigcup\{I_k : k \in K\} - I$ and $I \subset \bigcup\{I_k : k \in K\}$.

We say that r-query (either $((i,w),I)$ or $(\sim (i,w),I)$) for DS is k-local, if $i \in I_k$ and $I \subset I_k$. We say that r-query (either $((i,w),I)$ or $(\sim (i,w),I)$) for DS is k-global, if $I \subset I_k$. So, in a case of k-global queries the attribute i does not have to belong to I_k. In this paper we are only interested in r-queries which are either k-local or k-global.

Similarly, o-queries for DS built from elements in V_k are called k-local. All other o-queries for DS are called global. Global r-queries are initiated by agents only when they have to answer global o-queries. The interpretation J_{DS} of global o-queries at site k of DS was given for instance in [13] and [14]. In this paper we assume that our system DS is cooperative in the sense of Chu [1] or Gaasterland [2]. It means that if the interpretation of o-query at site k is giving us an empty set, we generalize first the local attribute values listed in o-query to answer it. If we still fail to answer the query at site k, then we contact servers at other sites of DS.

Assume now that $DS = (\{(S_k, N_k)\}_{k \in K}, L)$. Let q be r-query for DS which is k-local. An nc-interpretation J_{DS} of q in $S = S_k$ is defined below:

- if $q = ((i,w),I)$, then $J_{DS}(q)$ is a non-empty set of optimal S-rules describing (i,w) in terms of attributes from I,
- if $q = (\sim (i,w),I)$, then $J_{DS}(q)$ is a non-empty set of optimal S-rules describing $\sim (i,w)$ in terms of values of attributes from I,
- if $((i,w_1),t_1),((i,w_2),t_2) \in J_{DS}(q)$, $w_1 \neq w_2$ then either $(\exists t \in N_k)(t$ is a subterm of $t_1 * t_2)$ or terms t_1, t_2 are contradictory.

Assume now that q is r-query for DS which is k-global. It means that we can not resolve our r-query at site k or saying another words any nc-interpretation J_{DS} is not defined for q. In this case the client program at site k will search for servers which can resolve the query q. If such a server is found, the nc-interpretation J_{DS} at site k will be replaced by a new nc-interpretation linked with that server.

3 Distributed Knowledge-Based System

In this section, we show how to construct rules and dictionaries (knowledge-bases). Next, we show how to use them to improve nc-interpretations of o-queries for DS at site k.

Let us take an information system (S_k, N_k), where $(X_k, V_k, \leq, f_k)$, $X_k = \{a1, a3, a4, a6, a8, a9, a10, a11\}$, $V_k = \{H, h1, h2, E, e1, e2, F, f1, f2, f3, G, g1, g2, g3, K, k1, k2, L, l1, l2\}$, $I_k = \{i1, i2, i3, i4, i5, i6\}$, and f_k is defined by Table 1.

We assume here that: $H \leq h1$, $H \leq h2$, $E \leq e1$, $E \leq e2$, $F \leq f1$, $F \leq f2$, $F \leq f3$, $G \leq g1$, $G \leq g2$, $G \leq g3$, $K \leq k1$, $K \leq k2$, $L \leq l1$, $L \leq l2$. System S_k represents one of the sites of DS. A knowledge-base which is basically

X_k	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$
$a1$	$h1$	$e1$	$f2$	$g1$	$k1$	$l1$
$a3$	$h2$	$e1$	$f1$	$g1$	$k1$	$l1$
$a4$	$h1$	$e1$	$f2$	$g2$	$k1$	$l1$
$a6$	$h2$	$e2$	$f3$	$g3$	$k2$	$l2$
$a8$	$h2$	$e2$	$f2$	$g2$	$k2$	$l2$
$a9$	$h1$	$e1$	$f1$	$g1$	$k1$	$l2$
$a10$	$h2$	$e1$	$f2$	$g2$	$k2$	$l2$
$a11$	$h1$	$e1$	$f2$	$g1$	$k1$	$l1$

Table 1. Information System S_k

seen as a set of rules is added to each site of DS. A pair (information system, knowledge-base), is called a knowledge-based system. In [12], we proposed, so called, standard interpretation of rules and gave a strategy to construct rules which are optimal (not reducible).

Now, to recall our strategy, let us assume that information system represented by Table 1 is used to generate rules describing $e1, e2$ in terms of $\{f1,\ f2,\ f3,\ g1,\ g2,\ k1,\ k2\ \}$. Following Grzymala-Busse in [4], $f3 * g3 * k2 \rightarrow e2$ is a certain rule and $f2 * g2 * k2 \rightarrow e2$ is a possible one in S_k. Similarly, $f1 * g1 * k1 + f2 * g1 * k1 + f2 * g2 * k1 \rightarrow e1$ is a certain rule and $f2 * g2 * k2 \rightarrow e1$ is a possible rule in S_k. Now, assuming that S_k is not changing (we are not allowed to make any updates or add new tuples), we optimize the rules in S_k. As a result, we get two generalized certain rules: $f3 \rightarrow e2$ and $k1 \rightarrow e1$. The generalization process for possible rules is not trivial unless we want to generalize our rule $f2 * g2 * k2 \rightarrow e2$ to $\mathbf{1} \rightarrow e2$. We should also notice that the generalization process for certain rules allows us to create rules $f3 \rightarrow e2$ and $k1 \rightarrow e1$ which will become contradictory (no longer certain) if the term $f3 * k1$ does not belong to N_k. To prevent the last problem, we can change the optimization process for rules.

Let us assume that $\{u1, u2\}$ is an antichain in V_k such that $(\exists u \in V_k)(u \leq u1 \wedge u \leq u2)$ and $t1 \rightarrow u1$, $t2 \rightarrow u2$ are certain rules in (S_k, N_k), where $S_k = (X_k, V_k, \leq, f_k)$ and $N_k = N_{g_k} \cup N_{l_k}$. We say that these rules are k-locally sound if $J_S(t_1 * t_2) = \emptyset$ for any nc-system (S, N_k). We say that these rules are k-globally sound if $J_S(t_1 * t_2) = \emptyset$ for any nc-system (S, N_{g_k}).

Now, let us assume that $\{u1, u2\}$ is an antichain in V_k such that $(\exists u \in V_k)(u \leq u1 \wedge u \leq u2)$, $t1 \rightarrow u1$ is a certain rule, and $t2 \rightarrow u2$ is a possible rule in (S_k, N_k). We again say that these rules are k-locally sound if $J_S(t_1 * t_2) = \emptyset$ for any nc-system (S, N_k). We also say that these rules are k-globally sound if $J_S(t_1 * t_2) = \emptyset$ for any nc-system (S, N_{g_k}).

From this time on, we will allow only those generalizations which are preserving local soundness of rules on the client site and global soundness on the server sites. In [7] and [12], we described the process of building such rules when the set of negative constraints was empty. In both papers, we have used similar representation for certain and possible rules. Namely, we have defined them as triples $[u, t_1, t_2]$, where $t_1 \longrightarrow u$ represents a certain rule and $t_1{+}t_2 \longrightarrow u$ represents a possible one.

X_m	$i1$	$i2$	$i3$	$i4$	$i5$
$a1$	$f2$	$c1$	$d1$	$e1$	$g1$
$a6$	$f2$	$c1$	$d2$	$e3$	$g2$
$a7$	$f1$	$c2$	$d1$	$e3$	$g1$
$a11$	$f1$	$c1$	$d2$	$e3$	$g1$
$a13$	$f1$	$c2$	$d2$	$e3$	$g1$
$a14$	$f1$	$c2$	$d1$	$e3$	$g2$
$a15$	$f1$	$c1$	$d1$	$e3$	$g1$

Table 2. Information System S_m

Let us assume that we have information system (S_m, N_m), where $S_m = (X_m, V_m, \leq, f_m)$, $X_m = \{a1, a6, a7, a11, a13, a14, a15\}$, $V_m = \{C, c1, c2, E, e1, e2, e3, D, d1, d2, F, f1, f2, G, g1, g2, \}$, $I_m = \{i1, i2, i3, i4, i5\}$, and f_m is defined by Table 2.

We assume here that: $F \leq f1$, $F \leq f2$, $G \leq g1$, $G \leq g2$, $E \leq e1$, $E \leq e2$, $E \leq e3$, $C \leq c1$, $C \leq c2$, $D \leq d1$, $D \leq d2$. System (S_m, N_m) represents one of the sites of DS. Now, employing similar strategy to the one described in [12], we can generate two globally sound rules from (S_m, N_m): $[d1, e1, f1{*}e3]$ and $[d2, f2{*}e3, f1{*}e3]$. These rules can be added to the knowledge-base KB_k assigned to the site k of our distributed information system because $N_{g_m} = N_{g_k}$. If KB_k is empty, then (S_k, N_k) is extended to a knowledge-based system $((S_k, N_k), KB_k)$. If KB_k is not empty then the k-local soundness of any two rules in KB_k have to be checked. If the rules are not k-locally sound, then they have to be repaired following a strategy similar to the one described in [11].

Let us assume that $((S_k, N_k), KB_k)$ represents one of the sites of a distributed knowledge-based system DS, $S_k = (X_k, V_k, \leq, f_k)$ and J_{S_k} is the interpretation of queries from $L(S_k)$ in S_k. By a standard interpretation of global queries (elements of $L(DS)$) at site k, we mean function M_k such that:

- $M_k(\mathbf{0}) = \emptyset$, $M_k(\mathbf{1}) = X_k$,
- for any $w \in V_k$, $M_k(w) = J_{S_k}(w)$,
- for any $w \notin V_k$, $M_k(w) = \{x \in X_k : (\exists t, s \in L(S_k))([w, t, s] \in KB_k \wedge x \in J_{S_k}(t)\}$,
- for any $w \notin V_k$, $M_k(\sim w) = \{x \in X_k : (\exists t, s \in L(S_k))([w, t, s] \in KB_k \wedge x \notin J_{S_k}(s)\}$,
- for any global query t, $M_k(t) = J_{S_k}(t)$.

Let us go back to Table 2. Clearly, we can also generate the following rules from S_m:

$[g1, e1 + c1 * f1, c2 * f1 * e3]$,
$[g2, f2 * e3, c2 * f1 * e3]$.

These rules are globally sound and can be added to KB_k. If they are added to KB_k, they may change the local nc-interpretation M_k of global queries at site k. There is one problem, attributes $c1, c2$, listed in the descriptions of both rules, are not local for a site k. So, we can either interpret them as empty sets of objects or ask other sites of DS for k-global rules describing $c1$ and $c2$.

To conclude our discussion, assume that M_k is retrieving empty set when asking for a local nc-interpretation of a local attribute. In this case, we can go to a parent of this attribute (our attributes are represented as trees) and check if M_k retrieves any objects for that parent node. There is a possibility that the empty set will not be retrieved. Also, by generalizing queries we may retrieve some objects which are not interesting for the user. Clearly, it makes sense to give a chance to the user to make him decide if objects retrieved by the client system are useful or useless. If queries contain foreign attributes, then the client will search for server systems which can resolve these attributes. The use of negative constraints gives us the possibility to search for more compact representation of rules and improves the time complexity of the query answering system.

Conclusion

This paper presents a methodology and theoretical foundations of QRAS-NC (Query Rough Answering Systems with Negative Constraints) which first version is implemented at UNC-Charlotte on a cluster of SPARC workstations.

References

1. Chu, W.W., Chen, Q., Lee, R., "Cooperative query answering via type abstraction hierarchy", In *Cooperating Knowledge-based Systems* (ed. S.M. Deen), North Holland, 1991, 271-292
2. Gaasterland, T., Godfrey, P., Minker, J., "An overview of cooperative answering", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 123-158
3. Grice, H., "Logic and conversation", in *Syntax and Semantics* (eds. P. Cole and J. Morgan), Academic Press, 1975
4. Grzymala-Busse, J., *Managing uncertainty in expert systems*, Kluwer Academic Publishers, 1991
5. Kryszkiewicz, M., Ras, Z.W., "Query rough-answering system for CKBS", in *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, RSFD-96, Tokyo, Japan, November 6-8, 1996, pp. 162-167
6. Lipski, W., Marek, W., "On information storage and retrieval systems", in *Mathematical Foundations of Computer Science*, Banach Center Publications, Warsaw, Vol. 2, 1977, 215-259

7. Maitan, J., Ras, Z.W., Zemankova, M., "Query handling and learning in a distributed intelligent system", in *Methodologies for Intelligent Systems*, 4, (ed. Z.W. Ras), North Holland, 1989, 118-127
8. Michalski, R.S., "Patern recognition as rule-guided inductive inference", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, July 1980
9. Pawlak, Z., "Rough sets and decision tables", in *Proceedings of the Fifth Symposium on Computation Theory*, Springer Verlag, Lecture Notes in Computer Science, Vol. 208, 1985, 118-127
10. Pawlak, Z., "Mathematical foundations of information retrieval", *CC PAS Reports*, No. 101, Warsaw, 1973
11. Ras, Z.W., "Dictionaries in a distributed knowledge-based system", in *Proceedings of Concurrent Engineering: Research and Applications Conference*, Pittsburgh, August 29-31, 1994, Concurrent Technologies Corporation, 383-390
12. Ras, Z.W., "Collaboration control in distributed knowledge-based systems", in *Information Sciences Journal*, Elsevier, Vol. 96, No. 3/4, 1997, pp. 193-205
13. Ras, Z.W., "Cooperative knowledge-based systems", in *Intelligent Automation and Soft Computing Journal*, Vol. 2, No. 2, 1996, pp. 193-202
14. Ras, Z.W., "Resolving queries through cooperation in multi-agent systems", in *Rough Sets and Data Mining*, Eds. T.Y. Lin, N. Cercone, Kluwer Academic Publishers, 1997, pp. 239-258

This article was processed using the LaTeX macro package with LMAMULT style