Query Answering System for an Incomplete DKBS

Zbigniew W. Ras and Sucheta Joshi
University of North Carolina
Department of Computer Science
Charlotte, N.C. 28223, USA
* also Polish Academy of Sciences, Institute of Computer Science, 01-237 Warsaw, Poland
ras@uncc.edu or sucheta@informix.com

Abstract

A Cooperative Knowledge-Based System (CKBS) is a collection of autonomous knowledge-based systems called agents which are capable of interacting with each other. A query can be submitted to one agent or a group of agents. An agent when contacted by the user acts as a master agent, he sends requests to other agents which act as his slaves. Clearly, any agent in CKBS can be contacted by a user. So, any agent in CKBS can be a master agent. In this paper, an agent is represented by an information system (either complete or incomplete) and a collection of rules called a knowledge base. Rules we interpret as descriptions of some attribute values in terms of other attribute values. These descriptions are usually not precise and they only provide lower and upper approximations of attribute values. We say that an attribute value is reachable by an agent if either it belongs to the domain of one of the attributes in his information system or it is a decision part of one of the rules in his knowledge base. In the second case all attribute values from the classification part of a rule have to be reachable. Rules in our system are computed at one site of CKBS, sent to other sites of CKBS and stored in their knowledge bases, if needed. So, the set of reachable attribute values at any site of CKBS is constantly changing. Knowledge bases built that way might easily become inconsistent because rules they contain are created independently at different sites of CKBS. The problem of repairing inconsistent rules was investigated in [19]. In this paper, we propose a strategy for discovering rules in incomplete information systems and give a formal system for handling queries in CKBS where each site is represented by either an incomplete or a complete information system.

Key Words: incomplete information system, cooperative query answering, rough sets, multi-agent system, knowledge discovery.

1 Introduction

By a cooperative knowledge-based system ($CKBS$) we mean a collection of autonomous knowledge-based systems called agents (sites) which are capable of interacting with each other. Each agent is represented by an information system (either complete or an incomplete) and a collection of rules called a knowledge base. Any site of $CKBS$ can be a source of a local or a global query. By a local query for a site i (or i-reachable query) we mean a query entirely built from values of attributes local for i. Local queries need only to access an information system of the site where they were issued and they are completely processed on the system associated with that site. In order to resolve a global query for a site i (built from values of attributes not necessarily local for i) successfuly, we may need to access an information system at more than one site of $CKBS$ and search for rules describing values of attributes (used in a query) which are not local for the site i. So, by learning at site i we mean the process of creating rules, by neighbors of i, which describe values of attributes not reachable at the site i in terms of values of attributes reachable at i. These rules [[20], [21], [22]] are used to replace a query which is not reachable at site i by an approximate reachable query at site i. This approach to query evaluation also reduces the volume of data transfer between sites.

When a query is received by a site k, its Query Answering System (QAS) identifies all k-unreachable values of attributes in a query. Next, it transformes the query to its equivalent DNF form and asks other sites of $CKBS$ for descriptions of k-unreachable values of attributes, used in a query, in terms of k-reachable one. If j is a site contacted by QAS, then QAS identifies all attributes which are both k-reachable and j-reachable. In rough sets terminology, we should see them as classification attributes. At the same time, k-unreachable attributes are interpreted by site j as decision attributes. There is a number of strategies which allow us to find optimal rules describing decision attributes in terms of classification attributes. We should mention here such systems like $LERS$ (developed by J. Grzymala-Busse), $DQuest$ (developed by W. Ziarko), $AQ15$ (developed by R. Michalski) or rules discovery system based on discriminant functions proposed by A. Skowron (see [23]). Most of these strategies have been developed under the assumption that the database part of KBS is complete. Problem of inducing rules from attributes with incomplete values was discussed in [[9], [11]). M. Kryszkiewicz in [12] suggests a strategy for generating crisp (the certainty factor is equal to 1) rules from incomplete databases (with null values). Our strategy shows how to compute such rules with certainty factors not necessarily equal to 1.

2 Basic Definitions

In this section, we introduce the notion of an information system, distributed information system, a knowledge base, and $s(i)$-terms which are called local for a site i. We introduce the notion of a rule and show the process of building knowledge bases.

By an information system [12,13] we mean a structure $S = (X, A, V, f)$, where X is a finite set of objects, A is a finite set of attributes (or properties), V is the set-theoretical union of domains of attributes from A, and f is a classification function which describes objects in terms of their attribute values. We assume that:

- $V = \bigcup\{V_a : a \in A\}$ is finite,
- $V_a \cap V_b = \emptyset$ for any $a, b \in A$ such that $a \neq b$,
- $f : X \times A \longrightarrow 2^V \cup \{\star\}$ where $f(x, a) \in 2^{V_a} \cup \{\star\}$ for any $x \in X$, $a \in A$.

If $f(x, a) = \star$, then the value of the attribute a for the object x is unknown. We will call system S incomplete if there is $a \in A$, $x \in X$ such that $card(f(x, a)) \geq 2$ or $f(x, a) = \star$. Otherwise system S is complete. For simplicity reason any complete or incomplete information system will be just called an information system.

Example 1. Let us consider an information system $S_1 = (X_1, \{A, B, C, D, E\}, V_1, f_1)$ given below. We assume that $V_1 = \{a1, a2, a3, b1, b2, b3, c1, c2, c3, d1, d2, e1, e2, e3\}$. Clearly, S_1 is incomplete.

X_1	A	B	C	D	E
$x1$	$\{a1, a2\}$	$\{b1, b2\}$	$c1$	$d1$	$\{e1, e2\}$
$x2$	$\{a2, a3\}$	$\{b1, b2\}$		$d2$	$e1$
$x3$	$a1$		$\{c1, c3\}$		$e3$
$x4$	$a3$		$c2$	$d1$	$\{e1, e2\}$
$x5$	$\{a1, a2\}$	$b1$	$c2$		$e1$
$x6$	$a2$	$b2$	$c3$	$d2$	$\{e2, e3\}$
$x7$	$a2$	$\{b1, b3\}$	$\{c1, c2\}$	$d2$	$e2$
$x8$	$a3$	$b2$	$c1$	$d1$	$e3$

Table 1: Information System S_1

We assume here that $Dom(A) = \{a1, a2, a3\}$, $Dom(B) = \{b1, b2, b3\}$, $Dom(C) = \{c1, c2, c3\}$, $Dom(D) = \{d1, d2\}$, and $Dom(E) = \{e1, e2, e3\}$. Clearly, $f_1(x1, A) = \{a1, a2\}$, $f_1(x1, B) = \{b1\}$, etc. Often, we write a instead of a singleton set $\{a\}$.

Let $S_1 = (X_1, A_1, V_1, f_1)$, $S_2 = (X_2, A_2, V_2, f_2)$ be information systems.

- S_2 is a subsystem of S_1 if $X_2 \subseteq X_1$, $A_2 \subseteq A_1$, $V_2 \subseteq V_1$ and $(\forall x \in X_2)(\forall a \in A_2)[f_1(x, a) \subseteq f_2(x, a)]$.
- S_2, S_1 are consistent if $f_1(x, a) \subseteq f_2(x, a)$ or $f_2(x, a) \subseteq f_1(x, a)$ for any $a \in A_1 \cap A_2$, $x \in X_1 \cap X_2$.

X	A	B	C	D
$x1$	$\{a1, a2\}$	$\{b1, b2\}$		$d1$
$x2$	$\{a2, a3\}$			$d2$
$x3$	$a1$		$\{c1, c3\}$	
$x4$	$a3$		$c2$	$d1$
$x5$	$\{a1, a2\}$	$b1$	$c2$	
$x6$	$a2$	$b2$	$c3$	$d2$
$x7$	$a2$	$\{b1, b3\}$	$\{c1, c2\}$	$d2$

Table 2: Information System S

Example 2. It can be easily checked that an incomplete information system, represented by Table 2, is a subsystem of the system represented by Table 1.

Let us assume that $S = (X, A, V, f)$ is an information system, $a \in A$ and $V_a = \{v_1, v_2, .., v_k\}$. Let $X(a, v_i) = \{x \in X : v_i \in f(x, a)\}$. By a covering of X generated by an attribute a we mean the set $[a]^\star = \{X(a, v_i) : 1 \leq i \leq k\}$. If $B \subseteq A$, then by a covering of X generated by B we mean the set $[B]^\star = \bigcap\{[b]^\star : b \in B\}$. Now, if $a \in A$ and $B \subseteq A$, then B is called a covering of the attribute a in S (denoted $[B]^\star \leq [a]^\star$) if $(\forall Y \in [B]^\star)(\exists Z \in [a]^\star)(Y \subseteq Z)$.

For example, if S is an information system represented by Table 1, then
$[C]^\star = \{\{x1, x2, x3, x7, x8\}, \{x2, x4, x5, x7\}, \{x2, x3, x6\}\}$,
$[E]^\star = \{\{x1, x2, x4, x5\}, \{x1, x4, x6, x7\}, \{x3, x6, x8\}\}$, and
$[\{C, E\}]^\star = \{\{x1, x2\}, \{x1, x7\}, \{x3, x8\}, \{x4, x5\}, \{x4, x7\}, \{x2\}, \{x6\}, \{x3, x6\}\}$.
Clearly, $\{C, E\}$ is the covering of both C and E in S.

We will use the notation $[B]^\star \not\leq [a]^\star$ instead of $non([B]^\star \leq [a]^\star)$.

Now, assume that $S = (X, A, V, f)$, $a \in A$, and $Z \subseteq A - \{a\}$. Let P_k be the set of all subsets of the same cardinality k of the set Z. Below, we give an algorithm for finding the set of all coverings, included in Z, of the attribute $a \in A$. This algorithm is similar to the one proposed by Grzymala-Busse in [8].

Algorithm $Coverings(Z, a)$;
begin
$C := 0$;
compute covering $[a]^\star$;
$k := 1$;
while $k \leq card(Z)$ **do**
begin

for each set of attributes P in P_k **do**
compute covering $[P]^\star$ of X;
if (P is not a superset of any member of C)
and $[P]^\star \leq [a]^\star$ **then** add P to C;
$k := k + 1$
end
end
end

By an incomplete distributed information system [8] we mean a pair $DS = (\{S_i\}_{i \in I}, L)$ where:

- $S_i = (X_i, A_i, V_i, f_i)$ is an information system for any $i \in I$,
- L is a symmetric, binary relation on the set I,
- I is a set of sites,
- $(\exists i \in I)[S_i$ is *incomplete*$]$

Systems S_i, S_j (sites i, j) are called neighbors in a distributed information system DS if $(i, j) \in L$. The transitive closure of L in I is denoted by $L^\star$.

A distributed information system $DS = (\{S_i\}_{i \in I}, L)$ is consistent if:

$(\forall i)(\forall j)(\forall x \in X_i \cap X_j)(\forall a \in A_i \cap A_j)$
$[(x, a) \in Dom(f_i) \cap Dom(f_j) \longrightarrow f_i(x, a) \subseteq f_j(x, a)$ or $f_j(x, a) \subseteq f_i(x, a)]$.

The inclusion $f_i(x, a) \subseteq f_j(x, a)$ means that the system S_i has more precise information about the property a of an object x than the system S_j does. In order to show that this definition of consistency does not cause any problems, we introduce a set $I_{<x,a>} = \{i \in I : (x, a) \in Dom(f_i)\}$ for any $(x, a) \in \bigcup\{Dom(f_i) : i \in I\}$. It can be easily shown that $\bigcap\{f_j(x, a) : j \in I_{<x,a>}\} \neq \emptyset$ for any $(x, a) \in \bigcup\{Dom(f_i) : i \in I\}$ unless DS is inconsistent. To prove this proerty it is enough to observe that $\{f_i(x, a) : i \in I_{<x,a>}\}$ is a linearly ordered set with respect to inclusion. The set $\bigcap\{f_j(x, a) : j \in I_{<x,a>}\}$ can be interpreted as the consensus of all sites of DS on the property a of an object x.

Now, we plan to introduce the notion of a knowledge base D_{ki}, $(k, i) \in L^\star$, containing rules describing values of attributes from $A_k - A_i$ in terms of values of attributes from $A_k \cap A_i$ (see [20]). We begin with definitions of $s(i)$-terms, $s(i)$-formulas and their standard interpretation M_i in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$, where $S_j = (X_j, A_j, V_j, f_j)$ and $V_j = \bigcup\{V_{ja} : a \in A_j\}$, for any $j \in I$.

By a set of $s(i)$-terms we mean a least set T_i such that:

- $\mathbf{0}, \mathbf{1} \in T_i$,
- $(a, w) \in T_i$ for any $a \in A_i$ and $w \in V_{ia}$,
- if $t_1, t_2 \in T_i$, then $(t_1 + t_2), (t_1 * t_2), \sim t_1 \in T_i$.

We say that:

- $s(i)$-term t is *atomic* if it is of the form (a, w) or $\sim (a, w)$ where $a \in B_i \subseteq A_i$ and $w \in V_{ia}$
- $s(i)$-term t is *positive* if it is of the form $\prod\{(a, w) : a \in B_i \subseteq A_i \text{ and } w \in V_{ia}\}$
- $s(i)$-term t is *primitive* if it is of the form $\prod\{t_j : t_j \text{ is atomic }\}$
- $s(i)$-term is in *disjunctive normal form* (DNF) if $t = \sum\{t_j : j \in J\}$ where each t_j is primitive.

By a local query for a site i ($s(i)$-query) we mean any element in T_i which is in DNF.

Before we give the interpretation of $s(i)$-queries, we use Table 1 to outline a method for retrieving objects in the system S_1 having property $(A, a1)$. Clearly object $x3$ has a property $(A, a1)$. About objects $x1, x5$ we can only say that they might have property $(A, a1)$. In this case, our QAS returns a set $\{(x1, 1/2), (x3, 1), (x5, 1/2)\}$ as the answer for a local i-query $(A, a1)$. This set should be seen as a collection of three statements given below:

- object $x1$ has a property $(A, a1)$ with a confidence $1/2$,
- object $x3$ has a property $(A, a1)$ with a confidence 1, and
- object $x5$ has a property $(A, a1)$ with a confidence $1/2$.

Similarly, if we ask the system for all objects having property $(B, b1)$, we will get the set $\{(x1, 1), (x3, 1/2), (x5, 1), (x7, 1/2)\}$ as the answer. Now, in order to be more general, assume that $\{(x_i, p_i) : i \in N\}$ is the answer for a $s(i)$-query $(A, a1)$ and $\{(x_j, q_j) : j \in M\}$ is the answer for a $s(i)$-query $(B, b1)$ in an incomplete information system S. In order to find all objects in S having property $(a1 \star b1)$, we take the intersection of the above two sets defined as $\{(x_i, q_i * p_i) : i \in N \cap M\}$. In our example, $\{(x1, 1/2), (x3, 1/2), (x5, 1/2)\}$ is the answer for a $s(i)$-query $(a1 \star b1)$.

Let us be more formal. We assume that X is a set of objects. By an X-algebra we mean a sequence $(\mathbf{P}, \bigoplus, \bigotimes, \neg)$ where:

- $\mathbf{P} = \{P_i : i \in J\}$ where $P_i = \{(x, p_{<x,i>}) : p_{<x,i>} \in [0, 1] \ \& \ x \in X\}$,
- $P_i \bigotimes P_j = \{(x, p_{<x,i>} \cdot p_{<x,j>}) : x \in X\}$,

- $P_i \bigoplus P_j = \{(x, max(p_{<x,i>}, p_{<x,j>})) : x \in X\}$,
- $\neg P_i = \{(x, 1 - p_{<x,i>}) : x \in X\}$,
- $\mathbf{P}$ is closed under the above three operations.

Theorem 1 Let $P_i, P_j, P_k \in \mathbf{P}$. Then:

- $(P_i \otimes P_j) \otimes P_k = P_i \otimes (P_j \otimes P_k)$,
- $(P_i \bigoplus P_j) \bigoplus P_k = P_i \bigoplus (P_j \bigoplus P_k)$,
- $(P_i \bigoplus P_j) \otimes P_k = (P_i \otimes P_k) \bigoplus (Pj \otimes P_k)$,
- $P_i \otimes P_j = P_j \otimes P_i$,
- $P_i \bigoplus P_j = P_j \bigoplus P_i$,
- $P_i \bigoplus P_i = P_i$.

By a standard interpretation of $s(i)$-queries in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$ we mean a partial function M_i, from the set of $s(i)$-queries into X_i-algebra, defined as follows:

- $Dom(M_i) \subseteq \mathbf{T}_i$,
- $M_i((a, w)) = \{(x, p) : x \in X_i \ \& \ w \in f_i(x, a) \ \& \ p = 1/card(f_i(x, a))\}$ for any $w \in V_i$,
- $M_i(\sim (a, w)) = \neg M_i((a, w))$
- for any atomic term $t_1(a) \in \{(a, w), \sim (a, w)\}$ and any primitive term $t = \prod\{s(b) : (s(b) = (b, w_b)$ or $s(b) = \sim (b, w_b)) \ \& \ (b \in B_i \subset A_i) \ \& \ (w_b \in V_{ib})\}$ we have

$$M_i(t \star t_1(a)) = M_i(t) \otimes M_i(t_1) \text{ if } a \notin B_i$$
$$M_i(t \star t_1(a)) = \emptyset \text{ if } a \in B_i \text{ and } t_1(a) \neq s(a),$$
$$M_i(t \star t_1(a)) = M_i(t) \text{ if } a \in B_i \text{ and } t_1(a) = s(a).$$

- for any $s(i)$-terms t_1, t_2

$$M_i(t_1 + t_2) = M_i(t_1) \bigoplus M_i(t_2).$$

By (k, i)-rule in $DS = (\{S_j\}_{j \in I}, L)$, $k, i \in I$, we mean a pair (t, c) such that:

- $c \in V_k - V_i$
- t is a positive $s(k)$-term which belongs to $\mathbf{T}_k \cap \mathbf{T}_i$

- if $(x, p1) \in M_k(t)$ then $(\exists p2)[(x, p2) \in M_k(c)]$.

We say that (k,i)-rule (t,c) is in k-optimal form if there is no other subterm $t1 \in \mathbf{T}_k \cap \mathbf{T}_i$ of $s(k)$-term t, such that: if $(x, p1) \in M_k(t1)$, then $(\exists p2)[(x, p2) \in M_k(c)]$

An object x satisfies a rule $r = (t, c)$ with a certainty p at site k, if $p = p1 \cdot p2$, $(x, p1) \in M_k(t)$, and $(x, p2) \in M_k(c)$.

Let $X = \{x_i : 1 \leq i \leq n\}$ and x_i satisfies the rule $r = (t, c)$ with a certainty p_i at site k for any $i \in \{1, 2, ..., n\}$. We say that r has certainty p, if
$p = [\Sigma\{p_i : p_i \neq 0 \ \& \ 1 \leq i \leq n\}]/[card\{i : p_i \neq 0 \ \& \ 1 \leq i \leq n\}]$.

By a knowledge base D_{ki} we mean any set of (k,i)-rules satisfying the condition below:

$$\text{if } (t,c) \in D_{ki} \text{ then } (\exists t_1)(t_1, \sim c) \in D_{ki}.$$

We say that a knowledge base D_{ki} is in k-optimal form if all its rules are in k-optimal form.

Let us assume that an information system S_1 is represented by Table 1 and system S_2 by Table 3. Assume also that $N_k(t) = \{x \in X_k : (\exists p)[(x, p) \in M_k(t)\}$. We show how to construct a knowledge base D_{12} in 1-optimal form. The following coverings of X_1 can be computed directly from the information system S_1 applying the algorithm $Coverings(Z, a)$ described earlier:

$[C]^\star = \{N_1(c1), N_1(c2), N_1(c3)\} = \{\{x1, x2, x3, x7, x8\}, \{x2, x4, x5, x7\}, \{x2, x3, x6\}\}$,
$[E]^\star = \{N_1(e1), N_1(e2), N_1(e3)\} = \{\{x1, x2, x4, x5\}, \{x1, x4, x6, x7\}, \{x3, x6, x8\}\}$,
$[D]^\star = \{N_1(d1), N_1(d2)\} = \{\{x1, x3, x4, x5, x8\}, \{x2, x3, x5, x6, x7\}\}$,
$[B]^\star = \{N_1(b1), N_1(b2), N_1(b3)\} =$
$\{\{x1, x2, x3, x4, x5, x7\}, \{x1, x2, x3, x4, x6, x8\}, \{x3, x4, x7\}\}$,
$[C, E]^\star = \{N_1(c1 \star e1), N_1(c1 \star e2), N_1(c1 \star e3), N_1(c2 \star e1), N_1(c2 \star e2),$
$N_1(c2 \star e3), N_1(c3 \star e1), N_1(c3 \star e2), N_1(c3 \star e3)\} =$
$\{\{x1, x2\}, \{x1, x7\}, \{x3, x8\}, \{x2, x4, x5\}, \{x4, x7\}, \emptyset, \{x2\}, \{x6\}, \{x3, x6\}\}$,
$[C, D]^\star = \{N_1(c1 \star d1), N_1(c1 \star d2), N_1(c2 \star d1), N_1(c2 \star d2), N_1(c3 \star d1), N_1(c3 \star d2)\} =$
$\{\{x1, x3, x8\}, \{x2, x3, x7\}, \{x4, x5\}, \{x2, x5, x7\}, \{x3\}, \{x2, x3, x6\}\}$,
$[D, E]^\star = \{N_1(d1 \star e1), N_1(d1 \star e2), N_1(d1 \star e3), N_1(d2 \star e1), N_1(d2 \star e2), N_1(d2 \star e3)\} =$
$\{\{x1, x4, x5\}, \{x1, x4\}, \{x3, x8\}, \{x2, x5\}, \{x6, x7\}, \{x3, x6\}\}$.

It can be easily checked that:
$[C]^\star \not\leq [B]^\star$, $[D]^\star \not\leq [B]^\star$, $[E]^\star \not\leq [B]^\star$, $[\{D, E\}]^\star \not\leq [B]^\star$, $[\{C, E\}]^\star \leq [B]^\star$, $[\{C, D\}]^\star \leq [B]^\star$, $[\{C, D, E\}]^\star \leq [B]^\star$.

So, $\{C, E\}$, $\{C, D\}$ are minimal coverings of X_1.

In order to find the rules describing the attribute B in terms of attributes from $\{C, E\}$ we have to find all pairs $(N_1(t1), N_1(t2))$ such that $N_1(t1) \in [\{C, E\}]^\star$, $N_1(t2) \in [B]^\star$, and $N_1(t1) \subseteq N_1(t2)$. Each pair $(N_1(t1), N_1(t2))$ satisfying these conditions gives us a rule $(t1, t2)$.

$X2$	F	C	D	E	G
$a1$	$f1$	$c1$	$d1$	$e1$	$g1$
$a6$	$f2$	$c1$	$d2$	$\{e2, e3\}$	$g2$
$a8$		$c2$	$d1$	$e3$	$g1$
$a9$	$f2$	$c1$		$e3$	$g1$
$a10$	$f2$	$c2$	$d1$	$\{e1, e3\}$	$g1$
$a11$	$f1$	$c2$	$d1$	$e3$	$g2$
$a12$	$f1$	$c1$	$d2$	$\{e2, e3\}$	$g1$

Table 3: Information System S_2

So, in our example rules describing the attribute B in terms of attributes from $\{C, E\}$ are:

(c1*e1,b1), (c1*e2,b1), (c2*e1,b1), (c2*e2,b1), (c3*e1,b1),
(c1*e1,b2), (c1*e3,b2), (c3*e1,b2), (c3*e2,b2), (c3*e3,b2), and
(c2*e2,b3).

The corresponding rules in 1-optimal form (with certainty factors) are listed in Table 4.

$Rules$	$CertaintyFactor$
$(e1, b1)$	$[1/4 + 1/2 + 1/6 + 1]/4 = 23/48$
$(c1 * e2, b1)$	$[1/4 + 1/4]/2 = 1/4$
$(c2, b1)$	$[1/6 + 1/3 + 1 + 1/4]/4 = 7/16$
$(c1 * e1, b2)$	$[1/4 + 1/6]/2 = 5/24$
$(e3, b2)$	$[1/3 + 1/2 + 1]/3 = 11/18$
$(c3, b2)$	$[1/6 + 1/6 + 1]/3 = 8/18$
$(c2 * e2, b3)$	$[1/6 + 1/4]/2 = 5/24$

Table 4: Rules built at Site 1

Rules built at Site k can be sent to Site i of a distributed information system $DS = (\{S_j\}_{j \in I}, L)$, for any $k, i \in I$ (see [[14], [21], [22]]). They are stored in the knowledge base D_{ki} at Site i. So, the knowledge base D_{ki} represents beliefs of an agent k at site i. Clearly, if we store at Site i beliefs of several agents from DS, then the set of all beliefs stored at Site i might become inconsistent. In [19] we discussed the problem of repairing inconsistencies if DS is complete.

3 Distributed Knowledge-Based System

In this section, we define a Distributive Knowledge Based System ($DKBS$) and introduce the notion of its consistency. We also give an example of DKBS.

Let $\{D_{ki}\}_{k \in K_i}$, $K_i \subseteq I$, be a collection of knowledge bases where D_{ki} was created at site $k \in I$ for any $k \in K_i$ and $D_i = \bigcup\{D_{ki} : k \in K_i\} \cup R_i$. By R_i we mean a set of rules (t, c) created by an expert and stored at site i. Additionally, we assume here that t is an $s(i)$-term. System $(\{(S_i, D_i)\}_{i \in I}, L)$, introduced in [[20], [22]], is called a distributed knowledge-based system ($DKBS$).

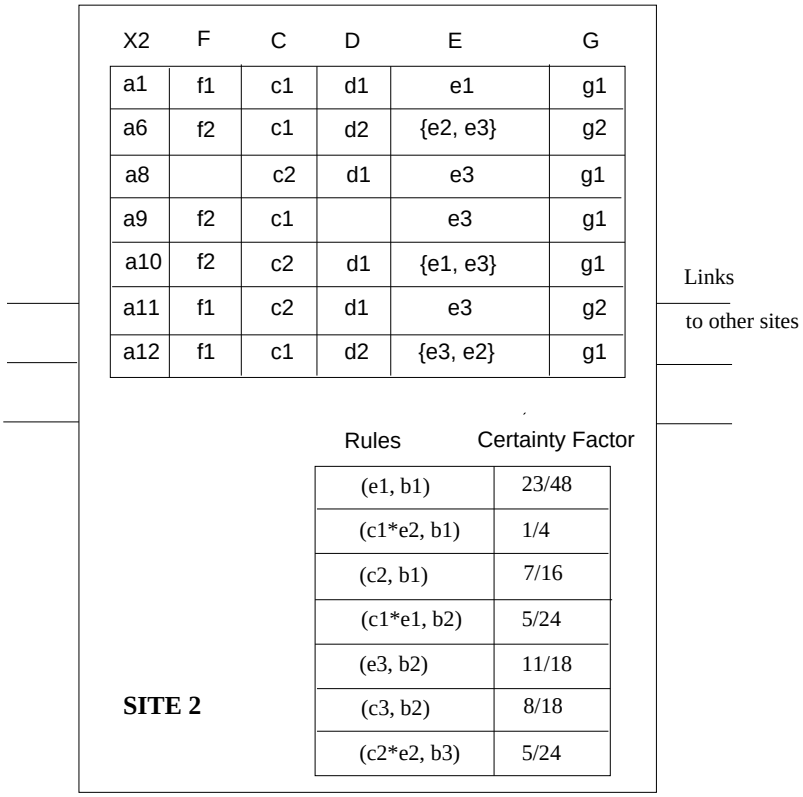

Figure 1: Site 2 of DKBS

Rules $(t1, w1) \in D_{ki}$, $(t2, w2) \in D_{ni}$ are consistent at Site i if $At(w1) \neq At(w2)$ or $w1 = w2$ or $M_i(t1 \star t2) = \emptyset$. Otherwise, we call them possibly inconsistent. We say that the knowledge base D_i is consistent at Site i if any two rules in D_i are consistent at Site i. Similarly, we say that the distributed knowledge based system $DS = (\{(S_i, D_i)\}_{i \in I}, L)$ is consistent if D_i is consistent at Site i for any $i \in I$.

To give an example of a $DKBS$ let us assume for simplicity reason that our system has only two sites ($Site1$ and $Site2$) represented by Table 1 and Table 3. Knowledge base D_{12}, represented by Table 4, is added to the $Site2$. Clearly, D_{12} is consistent at $Site2$. Figure 1 represents our $DKBS$.

4 Query Language and Its Interpretation.

In this section we introduce a query language and propose its optimistic interpretation in a $Site(i)$ of $DKBS$. We give a formal system for handling queries in $DKBS$. This system is sound and complete.

Standard interpretation M_i, introduced in Section 2, shows how to interpret $s(i)$-queries in a $Site(i)$ of $DKBS$. The question of interpreting DNF queries built from values of attributes belonging to a superset of V_i in $Site(i)$ remains open. Such queries are called global for a Site i. Their standard interpretation at Site i of a distributed knowlege based system $(\{(S_j, D_j)\}_{j\in I}, L)$, where $D_j = \bigcup\{D_{nj} : n \in K_j\}$, $S_i = (X_i, A_i, V_i, f_i)$ is proposed. To simplify our notation, we write S instead of S_i ,we write w instead of and atomic term (a, w) and assume that $V = V_i = \bigcup\{V_{ia} : a \in A_i\}$ and $C_S = \bigcup\{V_j : j \in I\} - V$. Elements in C_S are called concepts at site i.

By a query language $L(S, C_S)$ we mean a sequence (A, T, F), where A is an alphabet, T is a set of DNF terms (queries), and F is a set of atomic formulas.

The alphabet A of $L(S, C_S)$ contains:

- constants: w where $w \in V_i \cup C_S$
- constants: $\mathbf{0}, \mathbf{1}$
- functors: $+$, $\star$, $\sim$
- predicate: $=$
- auxiliary symbols: (,).

The set of terms T is a least set such that:

- constants $\mathbf{0}, \mathbf{1}$ are terms,
- if w is a constant, then $w, \sim w$ are terms,
- if t_1, t_2 are terms, then $t_1 \star t_2$ is a term.

The set of DNF terms is a least set such that:

- if t is a term, then t is a DNF term,
- if t_1, t_2 are DNF terms, then $t_1 + t_2$ is a DNF term.

Parentheses are used, if necessary, in the obvious way. As will turn out later, the order of a sum or product is immaterial. So, we will abbreviate finite sums and products as $\sum\{t_j : j \in J\}$ and $\prod\{t_j : j \in J\}$, respectively.

The set of atomic formulas F is a least set such that:

- if t_1, t_2 are DNF terms, then $(t_1 = t_2)$ is an atomic formula.

Let M_i be a standard interpretation of local $s(i)$-queries in $DS = (\{S_j)\}_{j\in I}, L)$. By a standard interpretation of DNF queries and atomic formulas from $L(S, C_S)$ in S-consistent distributed knowledge based system $(\{S_j, \{D_{kj}\}_{k\in K_j}\}_{j\in I}, L)$, where $S = (X_i, A_i, V_i, f_i)$ and $V_i = \bigcup\{V_{ia} : a \in A_i\}$, we mean a partial function N_i from the set of DNF queries into X_i-algebra $(\mathbf{P}, \bigoplus, \bigotimes, \neg)$ such that:

(1) for any $w \in V_{ia}$,
$N_i(w) = M_i(a, w)$,
$N_i(\sim w) = \neg N_i(w)$

(2) if $w \in C_S$,
$N_i(w) = max(\{(x, p) : x \in X_i \ \ (\exists n \in K_i)(\exists p > 0)(\exists t)[(t, w) \in D_{ni}\ \&\ (x, p) \in M_i(t)]\})$,
$N_i(\sim w) = max(\{(x, p) : x \in X_i \ \ (\exists n \in K_i)(\exists p > 0)(\exists t)[(t, \sim w) \in D_{ni}\ \&\ (x, p) \in M_i(t)]\})$
where $(x, p) \in max(D)$ iff $\sim (\exists q > p)((x, p) \in D\ \&\ (x, q) \in D)$

(3) $N_i(\mathbf{0}) = N_i(\sim \mathbf{1}) = \emptyset$,
$N_i(\mathbf{1}) = N_i(\sim \mathbf{0}) = X_i$

(4) for any terms t, w
$N_i(t \star w) = N_i(t) \bigotimes N_i(w)$
$N_i(t \star (\sim w)) = N_i(t) \bigotimes N_i(\sim w)$

(5) for any DNF terms t_1, t_2
$N_i(t_1 + t_2) = N_i(t_1) \cup N_i(t_2)$,

(6) for any DNF terms t_1, t_2
$N_i((t_1 = t_2)) =$ (if $N_i(t_1) = N_i(t2)$ then T else F)
(T stands for True and F for False)

From the point of view of site i, the interpretation N_i represents a pessimistic approach to query evaluation. If (x, p) belongs to the response of a query t, it means that x satisfies the query t with a confidence not less than p.

Let us adopt the following set A of Axiom Schemata:

- $A1$. Substitutions of the axioms of distributive lattices for terms and the axioms of equality
- $A2$. $\sim v + v = \mathbf{1}$ for any $v \in V_i$
- $A3$. for any term t,
$\sim \mathbf{0} = \mathbf{1}$, $\sim \mathbf{1} = \mathbf{0}$, $\mathbf{1} + t = \mathbf{1}$, $\mathbf{1} \star t = t$, $\mathbf{0} \star t = \mathbf{0}$, $\mathbf{0} + t = t$
- $A4$. for any $w \in C_S$
$w = \sum\{t : (t, w) \in D_{ki}\ \&\ k \in K_i\}$
- $A5$. for any $w \in C_S$
$\sim w = \sum\{t : (t, \sim w) \in D_{ki}\ \&\ k \in K_i\}$

The set R of rules of inference for our formal system is the following:

$R1$. from $(\alpha \Rightarrow \beta)$ and α we can deduce β for any formulas α, β

$R2$. from $t_1 = t_2$ we can deduce $t(t_1) = t(t_2)$,
where $t(t_1)$ is a term containing t_1 as a subterm and $t(t_2)$ comes from $t(t_1)$ by replacing some of the occurences of t_1 with t_2.

We write $A \vdash \alpha$ if there exists a derivation from a set A of formulas as premises to the formula α as the conclusion.

We write $A \models \alpha$ to denote the fact that A semantically implies α, that is, for any S-standard interpretation M_{i,K_i} of $L(S, C_S)$ in S-consistent cooperative knowledge-based system we have $M_{i,K_i}(\alpha) = T$.

Theorem 2 (Soundness). For any formula α, $A \vdash \alpha$ iff $A \models \alpha$.

5 Conclusion

We give a model of a distributed information system with sites being able to learn statements describing the data belonging to their nearest neighbors. These statements are stored as rules in the knowledge bases (knowledge bases) added to all sites of the distributed information system.

References

[1] Bazan, J., Skowron, A, Synak, P., "Dynamic reducts as atool for extracting laws from decision tables", in *Methodologies for Intelligent Systems, Proceedings of the 8th International Symposium* (ed. Z. Ras, M. Zemankova), Lecture Notes in Artificial Intelligence, Springer Verlag,No. 869, 1994, 346-355

[2] Bosc, P., Pivert, O., "Some approaches for relational databases flexible querying", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 355-382

[3] Chu, W.W., "Neighborhood and associative query answering", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 355-382

[4] 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

[5] Cuppers, F., Demolombe, R., "Cooperative answering: a methodology to provide intelligent access to databases", in *Proceedings 2nd International Conference on Expert Database Systems*, Virginia, USA, 1988

[6] Deen, S.M., "A general framework for coherence in a CKBS", in *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 2, 1993, 83-107

[7] Gaasterland, T., Godfrey, P., Minker, J., "An overview of cooperative answering", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 123-158

[8] Grzymala-Busse, J., *Managing uncertainty in expert systems*, Kluwer Academic Publishers, 1991

[9] Grzymala-Busse, J., *On the unknown attribute values in learning from examples*, Proceedings of ISMIS'91, LNCS/LNAI, Springer-Verlag, Vol. 542, 1991, 368-377

[10] Kacprzyk, J., "On measuring the specificity of if-then rules", in *International Journal of Approximate Reasoning*, Vol. 11, No. 1, 1994, 29-53

[11] Kodratoff, Y., Manago, M.V., Blythe, J."Generalization and noise", in *Int. Journal Man-Machine Studies*, Vol. 27, 1987, 181-204

[12] Kryszkiewicz, M., Rybinski, H., *Reducing information systems with uncertain attributes*, Proceedings of ISMIS'96, LNCS/LNAI, Springer-Verlag, Vol. 1079, 1996, 285-294

[13] 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

[14] Manago, M.V., Kodratoff, Y., "Noise and knowledge acquisition", in *Proceedings of the IJCAI 87, Int. Joint Conf. on AI*, 348-354

[15] Nakamura, A., "On rough logic based on incomplete knowledge systems", in *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, Society for Computer Simulation, 1994, 36-39

[16] Pawlak, Z., "Rough Sets - theoretical aspects of reasoning about data", Kluwer Academich Publishers, 1991

[17] 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

[18] Pawlak, Z., "Mathematical foundations of information retrieval", *CC PAS Reports*, No. 101, Warsaw, 1973

[19] 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

[20] Ras, Z.W., "Query processing in distributed information systems", in *Fundamenta Informaticae Journal*, Special Issue on Logics for Artificial Intelligence, IOS Press, Vol. XV, No. 3/4, 1991, 381-397

[21] Ras, Z., Chilumula, N., "Answering queries by cooperative knowledge based system", in *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, Society for Computer Simulation, 1994, 263-266

[22] Ras, Z.W., "Cooperative knowledge-based systems", in *Intelligent Automation and Soft Computing*, AutoSoft Press, Vol. 2, No. 2, 1996, 193-202

[23] Skowron, A., "Boolean reasoning for decision rules generation", in *Methodologies for Intelligent Systems, Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems*, (eds. J. Komorowski, Z. Ras), Lecture Notes in Artificial Intelligence, Springer Verlag, No. 689, 1993, 295-305