Chapter 5

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 consideration. 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,
- $(\forall a, b, c \in V)[(a \leq b \land c \leq b) \Rightarrow (a \leq c \land c \leq a)],$
- $(\forall a, b \in V)(\exists c \in V)(c \leq a \land c \leq b),$
- $(\forall a)[a$ has minimum two children or $a$ is a leaf].

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

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: