A Unified Interaction Scheme for Information Sources

YANNIS TZITZIKAS tzitzik@csd.uoc.gr
Computer Science Department, University of Crete, Greece

CARLO MEGHINI meghini@isti.cnr.it
Istituto di Scienza e Tecnologie dell' Informazione, CNR, Pisa, Italy

NICOLAS SPYRATOS spyratos@lri.fr
Laboratoire de Recherche en Informatique, Universite de Paris-Sud, France

Abstract. Commonly, for retrieving the desired information from an information source (knowledge base or information base), the user has to use the query language that is provided by the system. This is a big barrier for many ordinary users and the resulting interaction style is rather inflexible. In this paper we give the theoretical foundations of an interaction scheme that allows users to retrieve the objects of interest without having to be familiar with the conceptual schema of the source or with the supported query language. Specifically, we describe an *interaction manager* that provides a quite flexible interaction scheme by unifying several well-known interaction schemes. Furthermore, we show how this scheme can be applied to taxonomy-based sources by providing all needed algorithms and reporting their computational complexity.

1 Introduction

Information sources such as information retrieval systems (Baeza-Yates and Ribeiro-Neto, 1999), or databases and knowledge bases (Ullman, 1988), aim at organizing and storing information in a way that allows users to retrieve it in a flexible and efficient manner. Commonly, for retrieving the desired information from an information source, the user has to use the query language that is provided by the system. This not only poses a cognitive prerequisite, but also results in a quite inflexible interaction scheme as it is well known that most ordinary users are not aware of their precise information needs and that they find the existing query languages too limited (or crispy) for them.

We propose an interaction scheme whose objective is to make the desired objects easy to find for the user, even if the source has a query language which is *unknown* to the user. This scheme is actually a generalization of the interaction schemes that are currently used by information systems. In particular, we describe an *interaction manager* which supports in a uniform manner several kinds of interaction, including: query by example, index relaxation/contraction, query relaxation/restriction, answer enlargement/reduction, relevance feedback, and adaptation facilities.

In particular, we view the interaction of a user with the information source as a sequence of *transitions* between *contexts* where a context is a consistent "interaction state". The user has at his/her disposal several ways to express the desired transition. Then, it is the interaction manager that has to find (and drive the user to) the new context (see Figure 1). The traditional query-and-answer interaction scheme is only one kind of transition. Methods allowing the user to specify a transition relatively to the current context are also provided. Furthermore, we describe methods for restricting the set of transitions to those that can indeed lead to a context. As we shall see below, the unified interaction scheme that we introduce allows defining more complex interaction mechanisms than those that are supported by existing systems.

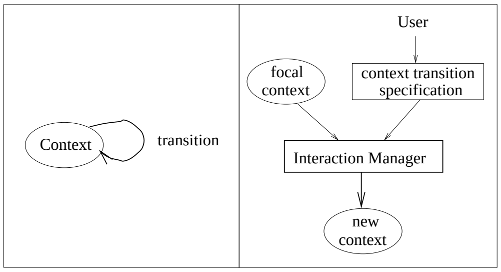

Fig. 1. A Context-based Interaction Scheme

For generality, we describe this scheme in terms of an abstract view of a source. Subsequently, we specialize it for taxonomy-based sources and show how the corresponding computational tasks can be performed and that they are indeed computationally tractable. This paper extends and analyzes in depth the ideas first sketched in (Tzitzikas et al., 2004b;c).

The paper is organized as follows. Section 2 describes the generalized interaction scheme. Section 3 introduces taxonomy-based sources and Section 4 describes how the generalized interaction scheme can be applied on them. Finally, Section 5 concludes the paper and identifies issues for further research.

2 The Generalized Interaction Scheme for Information Access

This scheme is described in terms of an abstract view of a source. Specifically, a source S is viewed as a function $S : Q \to \mathcal{A}$ where Q is the set of all queries that S can answer, and $\mathcal{A}$ is the set of all answers to those queries, i.e. $\mathcal{A}$=$\{ S(q) \mid q \in Q\}$. As we focus on retrieval queries, we assume that $\mathcal{A}$ is a subset of $\mathcal{P}(Obj)$, the powerset of Obj, where Obj is the set of all objects stored at the source.

Let $\mathcal{S}$ be the set of all sources that can be derived by "updating" the source S (e.g. for adapting it); for the moment let us suppose that $\mathcal{S}$ is the set of all functions from Q to $\mathcal{P}(Obj)$.

Let $\mathcal{U}$ denote the set of all triples in $\mathcal{S}\times Q \times \mathcal{A}$, i.e. $\mathcal{U} = \mathcal{S}\times Q \times \mathcal{A}$. A triple $c = (S, q, A) \in \mathcal{U}$ is called an interaction context, or *context* for short, if $S(q) = A$. Let $\mathcal{C}$ denote the set of all contexts, i.e. $\mathcal{C}$= $\{ (S, q, A) \in \mathcal{U} \mid S(q) = A\}$. Given a context $c = (S, q, A)$, S is called the *source view* of c, q is called the *query* of c and A is called the *answer* of c. The interaction between the user and the source is carried out by a software module called *Interaction Manager* (IM). The interaction is viewed as a sequence of *transitions* between contexts. At any given time, the user is in one context, the *focal context.* At the beginning of the interaction with the system the user starts from the initial context $(S, \epsilon, \emptyset)$ where S is the stored information source, ϵ is the empty query and $\emptyset$ is the empty answer[1]. However, any context of $\mathcal{U}$ could

[1] We assume that $S(\epsilon) = \emptyset$ therefore $(S, \epsilon, \emptyset)$ is a context.

serve as the initial context, e.g. the context $(S, \top, Obj)$ assuming that $\top$ denotes the top element of the query language. There are several methods the user can use for moving from one context to another, i.e. for changing the focal context. For example, in the traditional query-and-answer interaction scheme, the user actually "replaces" the current query q by formulating a new query q' and the "interaction manager" drives him to the new context (S, q', A') where $A' = S(q')$. However this is only one way of changing the focal context. Several other ways will be presented below.

2.1 Context Transition Specifications (CTSs) through Replacements

Suppose that the user can *replace* one component of the focal context (i.e. either S, q or A) by explicitly providing another component (resp. S', q' or A'). So we can have three kinds of replacements:
(a) query replacements, denoted by $[q \rightarrow q']$,
(b) answer replacements, denoted by $[A \rightarrow A']$, and
(c) source view replacements, denoted by $[S \rightarrow S']$.

As the user should always be in a context i.e. in a triple (S, q, A) where $S(q) = A$, after any of the above kinds of replacement, the IM should try to reach a context c' by changing one (or both) of the remaining two components of the focal context. Instead of leaving the IM to decide, it is the user that indicates to the IM the component(s) to be changed after a replacement. A replacement plus an indication of the above kind, is a *Context Transition Specification (CTS)*. Below we list the CTSs that we consider and discuss in brief the motivation beneath.

- $[q \rightarrow q'/A]$. Here the answer A must be changed. This is the classical query-and-answer interaction scheme: when the user replaces the current query q with a new query q', the user is given a new answer (i.e. an A' such that $A' = S(q')$). Thus we can write: $[q \rightarrow q'/A](S, q, A) = (S, q', S(q'))$.
- $[S \rightarrow S'/A]$. This is again a classical interaction scheme: whenever the source changes (e.g. after an update) the answers of queries change as well, i.e. here we have $A' = S'(q)$.
- $[A \rightarrow A'/q]$. Here the query must be changed. This interaction scheme may help the user to get acquainted with the query language of the source. It can be construed as an alternative query formulation process. The user selects a number of objects (i.e. the set A') and asks the IM to formulate the query that "describes" these objects. Subsequently the user can change the query q' in a way that reflects his/her information need. Roughly this resembles the Query By Example (QBE) process in relational databases. It also resembles the relevance feedback mechanisms in Information Retrieval systems. For example, the user selects a subset A' of the current answer A consisting of those elements of A which the user finds relevant to his/her information need. Subsequently, the IM has to change appropriately the query.
- $[S \rightarrow S'/q]$. Here the query q must be changed. This resembles the way that a relational database management system changes the query q that defines a relational view, after an update of the database, in order to preserve the contents (tuples) of the view.

- $[A \to A'/S]$. Here the source S has to be changed. In this case the user wants A' (instead of A) to be the answer of q. The IM should try to *adapt* to the desire of the user by changing the source from S to an S' such that $S'(q) = A'$. So this is a flexible and easy-to-use method that allows users to express their demand for source adaptation.
- $[q \to q'/S]$. Here again the source S must be changed. This means that the user replaces the current query q by a query q', because the user wants the current answer A to be the answer of q', not of q. The IM should try to *adapt* to the desire of the user by changing the source from S to an S' such that $S'(q') = A$. This is a way of attuning the query language to the user language.

The second column of Table 1 lists each kind of CTS that can be applied on a focal context $c = (S, q, A)$. Given a context c and a context transition specification R, the role of the IM is to find the desired target context $R(c)$, if one exists. The third column shows the target context after each kind of CTS where boldface is used to indicate the component that the IM has to compute in order to reach that context, assuming that only one of the remaining two components of the focal context can be changed. The notations $n_S(A')$ and $n_{S'}(A)$ denote queries that will be explained in detail in the next section.

Table 1. Context Transitions Specifications

	CTS	$R(c)$	evaluation of $R(c)$
(1)	$[q \to q'/A]$	$(S, q', \mathbf{S}(\mathbf{q'}))$	relies on query evaluation
(2)	$[S \to S'/A]$	$(S', q, \mathbf{S'}(\mathbf{q}))$	relies on query evaluation
(3)	$[A \to A'/q]$	$(S, \mathbf{n_S}(\mathbf{A'}), A')$	relies on naming functions
(4)	$[S \to S'/q]$	$(S', \mathbf{n_{S'}}(\mathbf{A}), A)$	relies on naming functions
(5)	$[A \to A'/S]$	$(\mathbf{S'}, q, A')$	relies on source adaptation
(6)	$[q \to q'/S]$	$(\mathbf{S'}, q', A)$	relies on source adaptation

2.2 Finding the Target Context

In CTSs (1) and (2) of Table 1, the IM has to change the answer in order to reach a context. These cases are relatively simple as the target context always exists and the desired answer A' can be derived using the query evaluation mechanism of the source. However, the cases in which the IM has to find either a new query (i.e. in CTSs (3) and (4)) or a new source (i.e. in (5) and (6)) are less straightforward as the target context does not always exist.

In CTS (3) and (4) the IM has to find a $q \in Q$ such that $S(q) = A$ for given S and A. Supporting these cases requires having a "naming service", i.e. a method for computing one or more queries that describe (name) a set of objects $A \subseteq Obj$. Ideally we would like a function $n : \mathcal{P}(Obj) \to Q$ such that for each $A \subseteq Obj$, $S(n(A)) = A$. Such a function would be called an "exact naming function", and the query $n(A)$ an exact name for the object set A, for all A. Note that if S is an *onto* function then the naming function n coincides with the inverse relation of S, i.e. with the relation $S^{-1} : \mathcal{P}(Obj) \to Q$. However, this is not always the case, as more often than not, S is not an onto function, i.e. $\mathcal{A} \subset \mathcal{P}(Obj)$. Furthermore, if S is onto and one-to-one, then

S^{-1} is indeed a function, thus there is always a unique q such that $S(q) = A$ for each $A \subseteq Obj$ [2]. As S is not always an onto function, "approximate" naming functions are introduced, specifically a *lower* naming function n^- and an *upper* naming function n^+, defined as follows:

$$n^-(A) = lub\{\ q \in Q \mid S(q) \subseteq A\}$$
$$n^+(A) = glb\{\ q \in Q \mid S(q) \supseteq A\}$$

where *lub* stands for least upper bound and *glb* for greatest lower bound with respect to the query containment ordering. If A is a subset of Obj for which both $n^-(A)$ and $n^+(A)$ are defined (i.e. the above *lub* and *glb* exist), then $S(n^-(A)) \subseteq A \subseteq S(n^+(A))$ and $n^-(A)$ and $n^+(A)$ are the best "approximations" of the exact name of A. Note that if $S(n^-(A)) = S(n^+(A))$ then both $n^-(A)$ and $n^+(A)$ are exact names of A.

Let us now return to CTS (3) and (4). If a naming function n_S is available for source S, then a single target context exists as shown in the third column of Table 1. If only approximate naming functions are available, then two "approximate" target contexts exist:

- $[A \rightarrow A'/q](S, q, A) = \begin{cases} (S, n_S^-(A'), S(n_S^-(A'))) \\ (S, n_S^+(A'), S(n_S^+(A'))) \end{cases}$
- $[S \rightarrow S'/q](S, q, A) = \begin{cases} (S', n_{S'}^-(A), S'(n_{S'}^-(A))) \\ (S', n_{S'}^+(A), S'(n_{S'}^-(A))) \end{cases}$

Notice that in the above cases, IM does not change only q, but also the answer. In the case $[A \rightarrow A'/q]$, the new context has an answer, say A'', which is the closest possible to the requested (by the user) answer A'. In the case $[S \rightarrow S'/q]$, the new context has an answer A' which is the closest possible to the current A.

In CTS (5) and (6) the IM is looking for a $S \in \mathcal{S}$ such that $S(q) = A$ for given q and A. We shall hereafter call *source adaptation* the process of finding the desired S. Clearly, the sought source S always exists if and only if $\mathcal{S}$ is the set of all functions from Q to $\mathcal{A}$. However, even if this is the case, there may be several sources that satisfy the equation $S(q) = A$. So we also need a criterion for choosing one of them. Undoubtedly, for doing this we should take into account the current source (as it is source adaptation). According to this view, it is reasonable to select as new source, the source that is as "close" as possible to the current source S_c. Of course, "closeness" or "distance" has to be defined formally. As we view sources as functions from Q to $P(Obj)$, i.e. as subsets of $Q \times P(Obj)$, it is quite natural to define the distance between two sources S, S' as the cardinality of their symmetric difference (in the classical set-theoretic sense), i.e. we may write:

$$dist(S, S') = |S \ominus S'| = |(S - S') \cup (S' - S)|$$

The process of finding the desired S cannot be further analyzed in our abstract framework. However one remark here is that the restricted relative replacements and

[2] Usually, sources are not one-to-one functions. For instance, the supported query language may allow formulating an infinite number of different queries, while the set of all different answers $\mathcal{A}$ is usually finite (e.g. $\mathcal{P}(Obj)$).

the source relaxation/contraction mechanisms that are introduced in the sequel, can be exploited for finding the target context even if source adaptation is not supported.

Table 2. Relative Context Transitions Specifications

	CTS	description
(1u)	$[q \rightarrow Up(q)/A]$	query relaxation resulting in answer enlargement
(1d)	$[q \rightarrow Down(q)/A]$	query contraction resulting in answer reduction
(2u)	$[S \rightarrow Up(S)/A]$	source relaxation resulting in answer enlargement
(2d)	$[S \rightarrow Down(S)/A]$	source contraction resulting in answer reduction
(3u)	$[A \rightarrow Up(A)/q]$	answer enlargement resulting in query relaxation
(3d)	$[A \rightarrow Down(A)/q]$	answer reduction resulting in query contraction
(4u)	$[S \rightarrow Up(S)/q]$	source relaxation resulting in query contraction
(4d)	$[S \rightarrow Down(S)/q]$	source contraction resulting in query relaxation
(5u)	$[A \rightarrow Up(A)/S]$	answer enlargement resulting in source relaxation
(5d)	$[A \rightarrow Down(A)/S]$	answer reduction resulting in source contraction
(6u)	$[q \rightarrow Up(q)/S]$	query relaxation resulting in source contraction
(6d)	$[q \rightarrow Down(q)/S]$	query contraction resulting in source relaxation

2.3 Relative Replacements and Relative CTSs

Relative replacements allow the user to specify the desired replacement without having to provide explicitly the new component, but by defining it relatively to the current one. We choose two quite generic relative replacements that are analogous to the way that humans explore an image: by zooming in and out at various points. These methods can be very helpful for the user during the interaction with the system. Relative replacements are based on the three partially ordered sets (posets) that are associated with the components of context, namely:

- The poset of answers $(\mathcal{P}(Obj), \subseteq)$.
- The poset of queries $(Q, \leq)$. Given two queries q and q' of Q, we write $q \leq q'$ iff $S(q) \subseteq S(q')$ in every possible source S in $\mathcal{S}$. We write $q \sim q'$ if both $q \leq q'$ and $q' \leq q$ hold. Let $Q_\sim$ denote the set of equivalence classes induced by $\sim$ over Q.
- The poset of sources $(\mathcal{S}, \sqsubseteq)$. Given two sources S and S' in $\mathcal{S}$, $S \sqsubseteq S'$ iff $S(q) \subseteq S'(q)$ in every query $q \in Q$.

For every element x of the above lattices, let $Br(x)$ denote the elements that are greater than or equal to x, $Nr(x)$ the elements that are less than or equal to x, x^+ a component that covers x, and x^- a component that is covered by x in the corresponding poset[3]. Note that there may not always exist a unique x^+ or a unique x^-. Specifically, we may have zero, one, or more x^+'s and x^-'s for a given x.

These partial orders can be exploited by the IM for moving to a component (answer, query, or source) that covers, or is covered by, the current component. Let $Up(x)$

[3] An element x is covered by y (or y covers x) if $x < y$ and there is no z such that $x < z < y$.

denote a component among those greater than x and that the IM can compute (ideally $Up(x) = x^+$), and let $Down(x)$ denote a component among those less than x and that the IM can compute (ideally $Down(x) = x^-$). These relative replacements can be used within context transition specifications. A CTS defined by a relative replacement will be called *relative CTS*. Table 2 lists all possible relative CTSs.

Relative CTSs can enhance flexibility during information access by the user. Also note that they can very easily be reflected at the user interface layer of the system. A classical query-answer interface extended with an indicative control panel that allows the specification of relative context transitions is sketched in Figure 2. For every component of the focal context there are two buttons "Up" and "Down", and an additional option control can allow the user to indicate to the IM the component (S, q, or A) that should be changed in order to reach the new context.

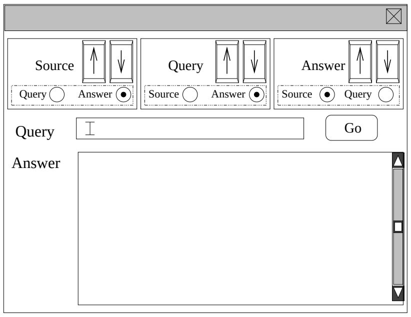

Fig. 2. A user interface for specifying context transitions through relative replacements

2.4 Restricting the Relative CTSs

The three partial orders mentioned earlier (and their interrelationships) can be exploited in order to restrict the set of relative CTSs to those for which the IM can indeed compute the target context.

In particular, we can restrict $Up/Down(A)$ so that to support CTS (3) even if naming functions are not available (or if they are available, but there is no exact name for A'). This can be achieved by defining:

$$Up(A) = S(Up(q)) \text{ and } Down(A) = S(Down(q))$$

We can restrict $Up/Down(A)$ so that to support CTS (5) even if source adaptation is not available. This can be achieved by defining $Up(A)$ and $Down(A)$ as follows:

$$Up(A) = Up(S)(q) \text{ and } Down(A) = Down(S)(q)$$

We can restrict $Up/Down(q)$ so that to support CTS (6) even if source adaptation is not available, but naming functions are available. This can be achieved by defining $Up(q)$ and $Down(q)$ as follows:

$$Up(q) = n_{Down(S)}(A) \text{ and } Down(q) = n_{Up(S)}(A)$$

3 Taxonomy-based Sources

Taxonomies is probably the oldest and most widely used conceptual modeling tool still widely used in Web directories (e.g. in Google and Yahoo!), Content Management (hierarchical structures are used to classify documents), Web Publishing (many authoring tools require to organize the contents of portals according to some hierarchical structure), Web Services (services are typically classified in a hierarchical form), Marketplaces (goods are classified in hierarchical catalogs), Personal File Systems, Personal Bookmarks for the Web, Libraries (e.g. Thesauri (International Organization For Standardization, 1986)), Semantic Web (e.g. see XFML (XFM)) and in very large collections of objects (e.g. see (Sacco, 2000)). Furthermore, the design of taxonomies can be done more systematically if done following a *faceted* approach (e.g. see (Prieto-Diaz, 1991; Ranganathan, 1965)). In addition, thanks to techniques that have emerged recently (Tzitzikas et al., 2004a), taxonomies of *compound terms* can be also defined in a flexible and systematic manner.

Although more sophisticated conceptual models (including concepts, attributes, relations and axioms) have emerged and are recently employed even for meta-tagging in the Web, almost all of them have a backbone consisting of a subsumption hierarchy, i.e. a taxonomy.

We view a taxonomy-based source S as a quadruple $S = \langle T, \preceq, I, Q \rangle$ where:

- T is terminology, i.e. a finite set of names called *terms*, e.g. `Canaries, Birds`.
- $\preceq$ is a reflexive and transitive binary relation over T (i.e. a pre-order) called *subsumption*, e.g. $\texttt{Canaries} \preceq \texttt{Birds}$,
- I is a function $I : T \rightarrow 2^{Obj}$ called *interpretation* where Obj is a finite set of objects called *domain*. For example $Obj = \{1, ..., 100\}$ and $I(\texttt{Canaries}) = \{1, 3, 4\}$, and
- Q is the set of all queries defined by the grammar $q ::= t \mid q \wedge q' \mid q \vee q' \mid \neg q \mid (q) \mid \epsilon$ where t is a term in T and ϵ the empty query.

The pair $(T, \preceq)$ is the taxonomy of S. We assume that every terminology T also contains two special terms, the *top term*, denoted by $\top$, and the *bottom term*, denoted by $\bot$. The top term subsumes every other term t, i.e. $t \preceq \top$. The bottom term is strictly subsumed by every other term t different than top and bottom, i.e. $\bot \preceq \bot$, $\bot \preceq \top$, and $\bot \prec t$, for every t such that $t \neq \top$ and $t \neq \bot$. We also assume that $I(\bot) = \emptyset$ in every interpretation I.

Query answering in a source S is based on the notion of model. An interpretation I is a *model* of a taxonomy $(T, \preceq)$ if for all t, t' in T, if $t \preceq t'$ then $I(t) \subseteq I(t')$. Given an interpretation I of T, the model of $(T, \preceq)$ *generated* by I, denoted $\bar{I}$, is given by: $\bar{I}(t) = \bigcup\{I(s) \mid s \preceq t)\}$.

An interpretation I can be extended to an interpretation of queries as follows: $I(q \wedge q') = I(q) \cap I(q')$, $I(q \vee q') = I(q) \cup I(q')$, and $I(\neg q) = Obj - I(q)$.

Given a source $S = \langle T, \preceq, I, Q \rangle$ and a query $q \in Q$, the answer of q is the set $\bar{I}(q)$, also denoted by $S(q)$.

Figure 3 illustrates one bibliographic taxonomy-based source. Objects (e.g. scientific papers) are represented by natural numbers, the membership of an object to the interpretation of a term is indicated by a dotted arrow from the object to that term, subsumption of terms is indicated by a continuous-line arrow from the subsumed term to the subsuming term. For better readability we do not show the top and bottom terms and we show only the transitive reduction of the subsumption relation.

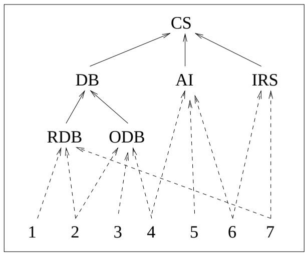

Fig. 3. One bibliographic taxonomy-based source

A remark here is that the taxonomy of a source does not necessarily has to be defined manually. For instance, it may be the concept lattice extracted by applying Formal Concept Analysis (Ganter and Wille, 1999), or the compound taxonomy defined by an expression of Compound Term Composition Algebra (Tzitzikas et al., 2004a). Furthermore, it may be the concept lattice of a Description Logics (DL) base (Donini et al., 1996). Specifically, we can view a DL-theory $\Sigma = (TBox, ABox)$ as a source $S = \langle T, \preceq, I, Q \rangle$ where

- T consists of all concepts that appear in $TBox$ and $ABox$,
- $t \leq t'$ iff $\Sigma \models t \sqsubseteq t'$,
- Obj is the alphabet of individuals, and
- I is defined by all the assertions of the $ABox$.

4 Application on Taxonomy-based Sources

For applying on taxonomy-based sources the interaction scheme just described, we have to find a method for computing the target context after every kind of CTS. The CTSs that require computing a new answer, can be supported using the query evaluation method already described in Section 3. So we only have to focus on the cases where given a CTS the IM has to find a new query or a new source. The first case relies on naming functions and the second on source adaptation. Below we show that naming functions and source adaptation are possible for taxonomy-based sources.

4.1 Naming functions

Given a source S and an answer A, we seek for a query q' such that $S(q') = A$. Finding q' requires defining the naming functions n^- and n^+ for taxonomy-based sources. Naming functions for taxonomy-based sources were first described in (Tzitzikas and Meghini, 2003) where they were exploited for creating inter-taxonomy mappings automatically. Before describing them, let us first introduce an auxiliary definition. Given an object $o \in Obj$, we shall use $D_I(o)$ to denote the query obtained by taking the conjunction of all terms t such that $o \in I(t)$, i.e. $D_I(o) = \bigwedge\{t \in T \mid o \in I(t)\}$. For keeping our notation simple, we shall also sometimes use $D_I(o)$ to denote the set

$\{t \in T \mid o \in I(t)\}$. It can be easily proved that if we exclude from our consideration the queries that contain negation, then the naming functions are defined as follows:

Proposition 1. $n^+(A) \sim \bigvee\{\ D_I(o) \mid o \in A\}$

Proposition 2. $n^-(A) \sim \bigvee\{\ D_I(o) \mid o \in A,\ S(D_I(o)) \subseteq A\}$

All proofs can be found in Appendix A. As an example, consider the source shown in Figure 4. Here we have: $n^-(\{1,2,3\}) = a$ and $n^+(\{1,2,3\}) = a \vee b$, so $\{1,2,3\}$ has only approximate names. However, the set $\{1,2,4\}$ has an exact name, i.e. $n^-(\{1,2,4\}) = n^+(\{1,2,4\}) = a \vee (b \wedge c)$.

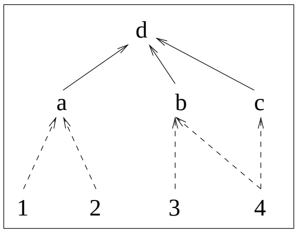

Fig. 4. One taxonomy-based source

It is important to note that the evaluation of the above formulas is a computationally tractable task.

Proposition 3. The time complexity for computing $n^+(A)$ is $O(|A||T|)$.

Proposition 4. The time complexity for computing $n^-(A)$ is $O(|A||Obj||T|^2)$.

4.2 Source Adaptation

Given a query q and an answer A, here we seek for a source S such that $S(q) = A$. We first need to define the set of sources $\mathcal{S}$ considered and their ordering. Let $S = \langle T, \preceq, I, Q\rangle$ be the original stored source. We assume that all sources in $\mathcal{S}$ have the *same* taxonomy, i.e. the taxonomy $(T, \preceq)$, and the same query language Q. In other words, we confine ourselves to the case where the taxonomy cannot be changed during source adaptation. The case where source adaptation involves taxonomy adaptation is an issue for further research.

So two sources S and S' of $\mathcal{S}$ may differ only in their interpretation. Let S_I denote the source with interpretation I. If $\mathcal{I}$ denotes the set of all interpretations of T over Obj, then we can define the set of all sources $\mathcal{S}$ as follows:

$$\mathcal{S} = \{\ S_I |\ I \in \mathcal{I}\} = \{\ \langle T, \preceq, I, Q\rangle \mid I \in \mathcal{I}\}$$

For defining the ordering over $\mathcal{S}$, let us first introduce an auxiliary definition. Given two interpretations I and I' of T we write $I \sqsubseteq I'$ if and only if $I(t) \subseteq I'(t)$ for every $t \in T$. One can easily see, that if S_I and $S_{I'}$ are sources of $\mathcal{S}$, then $S_I \sqsubseteq S_{I'}$ iff $I \sqsubseteq I'$.

Let us now return to the problem at hand. Specifically, let us consider CTS (5), i.e. $[A \rightarrow A'/S]$. Here we seek for a S' such that $S'(q) = A'$.

As we already mentioned for the general framework, there may be several sources S' that satisfy this equation. In our context, as all sources differ only in their interpretation, there may be several interpretations that satisfy the desired equation. Let I_{sol} be the set of all such interpretations, i.e.

$$I_{sol} = \{I \in \mathcal{I} \mid S_I(q) = A'\}$$

Again, we need a criterion for selecting one of them, and according to Section 2.2, it is reasonable to select the interpretation that is closest to the current interpretation I. We can define the distance between two interpretations as we did for sources, i.e.

$$dist(I, I') = |\bar{I} \ominus \bar{I}'|$$

We use $\bar{I} \ominus \bar{I}'$ instead of just $I \ominus I'$ so as to take into account the structure of the taxonomy. Moreover, the resulting metric is very close to the metric defined over sources (in the general framework) because $\bar{I}$ is actually the restriction of S on T, i.e. $S_{|T}$.

Now we can define the desired new interpretation $I_\circ$ as follows:

$$I_\circ = arg_{I'}\ min\{dist(I, I') \mid I' \in I_{sol}\}$$

Of course, note that in practice we need a method for *modifying* the existing I to the desired $I_\circ$ rather than generating the entire set I_{sol} and selecting the element that is closest to I . Below we describe an efficient method for reaching $I_\circ$ by modifying I.

Let $A^+ = A' - A$ and $A^- = A - A'$, so A^+ is the set of objects that have to added to the current answer, and A^- is the set of objects that have to be deleted from the current answer in order to reach A', i.e. we can write $A' = (A \cup A^+) - A^-$, or equivalently, $A' = (A - A^-) \cup A^+$. Below we explain how source adaptation can be achieved by considering each form that the query q may have.

Single-term queries

At first, suppose that $q = t$ where $t \in T$. For reaching the desired S' we have to update the interpretation I of S. Note that $\bar{I}(t)$ must no longer contain the set A^-, but it has to contain the set A^+. Below we will show how we can add each object of A^+ and how we can delete each object of A^-.

We shall use $Nr(t)$ and $Br(t)$ to denote the narrower and the broader terms of t, i.e. $Nr(t) = \{t' \mid t' \preceq t\}$ and $Br(t) = \{t' \mid t \preceq t'\}$.

Addition of an object o to $\bar{I}(t)$.

Precondition: $o \notin \bar{I}(t)$
Postcondition: $o \in \bar{I}'(t)$

In order to satisfy the postcondition we could define $I' := I \cup \{(o, t')\}$ where t' is any term such that $t' \leq t$. The distance between $\bar{I}$ and $\bar{I}'$ in this case becomes:

$$|\bar{I} \ominus \bar{I}'| = |\bar{I}' - \bar{I}| = |\{\ (o, t'') \mid t'' \in Br(t') - D_{\bar{I}}(o)\}|$$

It follows easily that for reaching an I' that has the minimum distance from I we have to select a term t' (among $Nr(t)$) that minimizes the quantity $|Br(t') - D_{\bar{I}}(o)|$.

Let call this quantity *prosthetic perturbation* and denote it by $pertAdd(o,t')$, i.e.

$$pertAdd(o,t') = |Br(t') - D_{\bar{I}}(o)|$$

Roughly, $pertAdd(o,t')$ is the number of terms that are broader than t and are not already applied to o according to $\bar{I}$, i.e. $o \notin \bar{I}(t)$. Note that if the precondition was not true, i.e. if o was already in $\bar{I}(t')$, then $pertAdd(o,t')$ would be 0.

If follows easily that in order to find the term t' that gives the minimal prosthetic perturbation we have to select the broader terms of $Nr(t)$, i.e. t. This is because:

$$t' \leq t \Leftrightarrow Br(t') \supseteq Br(t) \Leftrightarrow Br(t') - D_{\bar{I}}(o) \supseteq Br(t) - D_{\bar{I}}(o)$$

Thus in order to satisfy the postcondition and being as close to I as possible, we define I' as $I' := I \cup \{(o,t)\}$.

Note: If we also want our sources to have the smallest possible interpretations, then we must delete from I' the redundant pairs that may exist (they do exist if $Br(t) \cap D_I(o) \neq \emptyset$), i.e. the pairs (o,t'') where $t'' \geq t$. Examples of object additions are given at Figure 5.

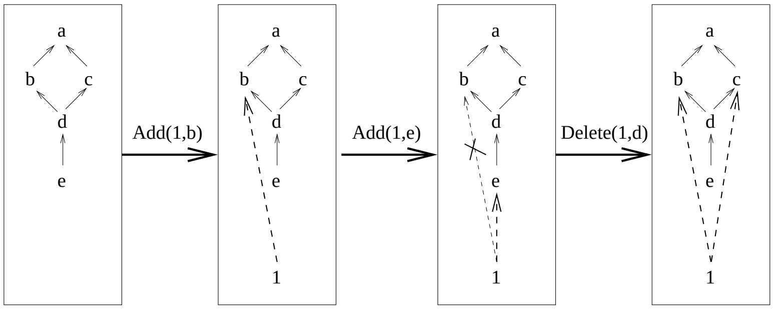

Fig. 5. Additions/Deletions of single objects from single term queries

Deletion of an object o from $\bar{I}(t)$.
Precondition: $o \in \bar{I}(t)$
Postcondition: $o \notin \bar{I}'(t)$

Recall that an object o may belong to $\bar{I}(t)$ either because $o \in I(t)$ or because $o \in I(t')$ where $t' \preceq t$. Clearly, in order to satisfy the postcondition we have to define an I' by deleting from I all pairs (o,t') such that $o \in I(t')$ and $t' \leq t$. Thus we could define I' as follows:

$$I' := I - \{\ (o,t') \mid t' \in D_I(o) \cap Nr(t)\}$$

Obviously, the distance between I and I', i.e. the quantity $|I \ominus I'|$, equals $|I - I'|$ which is equal to the cardinality of the right-hand side of the above formula. Let us now measure the distance between $\bar{I}$ and $\bar{I}'$. Note that each pair (o,t') that we delete from I causes the following difference in $\bar{I}$ and $\bar{I}'$:

$$\{\ (o,t'') \mid t'' \in Br(t') - D_{\bar{I}'}(o)\}$$

We could reduce the above quantity (and thus the overall distance) by adding some pairs to I'. Specifically, for each deleted (o, t') we could add the pairs (o, t_x) for each t_x that immediately subsumes t (i.e. $t_x \in Br^{(1)}(t)$) so that to maximize the quantity $D_{\bar{I}'}(o)$ and thus reduce the difference. One can easily see that by doing this, the overall distance becomes:

$$|\bar{I} \ominus \bar{I}'| = |\bar{I} - \bar{I}'| = |\{\ (o, t'')\ |\ t'' \in Nr(t) \cap D_{\bar{I}}(o)\}|$$

Clearly, the resulting interpretation satisfies the postcondition and it is the closest to I. An example of object deletion is given in Figure 5.

For reasons that will become evident below, we shall call the above difference *aphaeretic perturbation* and we shall denote it by $pertDel(o, t)$, i.e.

$$pertDel(o, t) = |Nr(t) \cap D_{\bar{I}}(o)|$$

Roughly, aphaeretic perturbation expresses how deep below t, o is indexed. Note that if the precondition was not true, i.e. if o was not in $\bar{I}(t)$, then $pertDel(o, t)$ would be 0.

Conjunctive queries

Now suppose that q is a conjunction $q = t_1 \wedge ... \wedge t_k$. One can easily see that the set A^+ must be added to each $\bar{I}(t_i)$ for $i = 1, ..., k$. According to the discussion of single-term queries, we just have to add each object of A^+ to the interpretation of each t_i, $i = 1, ..., k$ (specifically to the minimal elements of $\{t_1, ..., t_k\}$ if we prefer space minimal interpretations).

However notice that we don't necessarily have to delete the set A^- from each $\bar{I}(t_i)$, as it suffices to delete each object of A^- from only one $\bar{I}(t_i)$. In order to select this term t_i, as our objective is to minimize the distance with the original interpretation, we can exploit the aphaeretic perturbation that we introduced earlier. For example, consider the source shown in Figure 6. In this source we have $D_I(1) = \{e, c\}$ and $D_{\bar{I}}(1) = \{a, b, c, e\}$. Some examples follow:
$pertDel(1, c) = |Nr(c) \cap D_{\bar{I}}(1)| = |\{c, f, g\} \cap \{a, b, c, e\}| = |\{c\}| = 1$
$pertDel(1, b) = |Nr(b) \cap D_{\bar{I}}(1)| = |\{b, e\} \cap \{a, b, c, e\}| = |\{b, e\}| = 2$
$pertDel(1, a) = |Nr(a) \cap D_{\bar{I}}(1)| = |\{a, b, c, d, e, f, g\} \cap \{a, b, c, e\}| = |\{a, b, c, e\}| = 4$

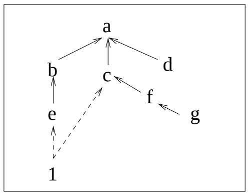

Fig. 6. One taxonomy-based source

Disjunctive queries

Now suppose that q is a *disjunction* of terms, i.e. $q = t_1 \vee ... \vee t_k$. It is evident that here the set A^- must be excluded from each $\bar{I}(t_i)$, $i = 1, ..., k$ and we can do it as described for single term queries.

On the other hand, we don't have to add the objects of A^+ to all $\bar{I}(t_i)$, as it suffices to add each object of A^+ to only one term of t. We can exploit the prosthetic perturbation as a criterion for selecting the appropriate term of q. Recall that $pertAdd(o,t) = |Br(t) - D_{\bar{I}}(o)|$. Some examples over the source of Figure 6 follow:
$pertAdd(1,d) = |Br(d) - D_{\bar{I}}(1)| = |\{a,d\} - \{a,b,c,e\}| = |\{d\}| = 1$
$pertAdd(1,f) = |Br(f) - D_{\bar{I}}(1)| = |\{a,c,f\} - \{a,b,c,e\}| = |\{f\}| = 1$
$pertAdd(1,g) = |Br(g) - D_{\bar{I}}(1)| = |\{a,c,f,g\} - \{a,b,c,e\}| = |\{f,g\}| = 2$
Recall that $|Br(t) - D_{\bar{I}}(o)|$ is the number of terms which are broader than t *and do not* already apply on o. So, it is the number of additional terms that will apply on o if we add o to the interpretation of t, $I(t)$.

CNF queries
Now the queries that contain all Boolean connectives can be handled analogously. Specifically, Table 3 outlines the algorithms for each kind of queries. The Table includes the case of queries in CNF (conjunctive normal form). Note that any query with logical connectives can be converted to CNF by using one of the existing algorithms (e.g. see (Galton, 1990)).

The above methods are efficient, as the following proposition proves.

Proposition 5. The time complexity of source adaptation is :

- $O(|A^+| + |A^-||T|^2)$ for *single term* queries,
- $O(|A^- \cup A^+|k|T|^2)$ for *disjunctive* queries $t_1 \vee ... \vee t_k$,
- $O(|A^-|k|T|^2 + |A^+|k)$ for *conjunctive* queries $t_1 \wedge ... \wedge t_k$.

4.3 Source Relaxation/Contraction

In order to define relative CTCs for taxonomy-based sources we have to show how source relaxation/contraction and query relaxation/contraction can be achieved. For relaxing and contracting a source S we shall use two operations: one for relaxing and one for contracting the interpretation of S. Hereafter with I, we shall denote a model of the taxonomy. For relaxing (resp. contracting) a model I, we will use an operation $\cdot^+$ (resp. $\cdot^-$) defined as follows:

$$I^+(t) = \bigcap \{I(t') | t \prec t'\}$$
$$I^-(t) = \bigcup \{I(t') | t' \prec t\}$$

The $\cdot^+$ operation was first introduced in (Meghini and Tzitzikas, 2003) and it is founded on *abduction* (Eiter and Gottlob, 1995). So, if $S = \langle T, \preceq, I, Q\rangle \in \mathcal{S}$, then $Up(S) = \langle T, \preceq, I^+, Q\rangle$ and $Down(S) = \langle T, \preceq, I^-, Q\rangle$.

Another remark, is that the above operators can give us an alternative solution to the source adaptation problem in the special case where A' is a subset or a superset of the current answer A. Specifically, if $A \subset A'$ then our search space is $Br(S)$. We can apply iteratively the operator $\cdot^+$ on S until reaching to a source S' such that $S'(q) \supseteq A'$ or until reaching the least fixed point of $\cdot^+$. If $S'(q) = A'$ then we have found the solution, while if $S'(q) \supset A'$ then there is no exact solution. In the last case

Algorithms for Adapting Taxonomy-based Sources	
query form	algorithm
$q = t$	(a) for each $o \in A^-$ do $\texttt{Delete}(o, t)$ (b) $I'(t) := I'(t) \cup A^+$
$q = \neg t$	(a) for each $o \in A^+$ do $\texttt{Delete}(o, t)$ (b) $I'(t) := I'(t) \cup A^-$
$q = t_1 \vee ... \vee t_k$	(a) for each $o \in A^-$ do for each $i = 1, ..., k$ do $\texttt{Delete}(o, t_i)$ (b) for each $o \in A^+$ do $m = \arg_i \texttt{min} \{pertAdd(o, t_i) \mid i = 1, .., k\}$; $I'(t_m) := I(t_m) \cup \{o\}$ end for
$q = t_1 \wedge ... \wedge t_k$	(a) for each $i = 1, ..., k$ do $I'(t_i) := I(t_i) \cup A^+$ (b) for each $o \in A^-$ do $m = \arg_i \texttt{min} \{pertDel(o, t_i) \mid i = 1, .., k\}$; $\texttt{Delete}(o, t_m)$ end for
$q = d_1 \wedge ... \wedge d_m$ where $d_j = t_{j1} \vee ... \vee t_{jn_j}$	(a) add each object of A^+ to each disjunction of q (b) delete each object of A^- from the disjunction that causes the less perturbation
	algorithm $\texttt{Delete}(o, t)$ $X := D_{\bar{I}}(o)$; for each $t' \in X$ such that $t' \preceq t$ do $I'(t') := I(t') - \{o\}$; $Y := X - Nr(t)$; for each $t' \in minimal(Y)$ do $I'(t') := I(t') \cup \{o\}$; end algorithm

Table 3. Source Adaptation

we can define only "approximate" solutions. Analogously, we can treat the case where $A \supset A'$. In particular, if $A \supset A'$ then the search space is $Nr(S)$ and we can search it by applying iteratively the operator $\cdot^-$ on S until reaching to a source S' such that $S'(q) \subseteq A'$ or until reaching the least fixed point of $\cdot^-$.

4.4 Query Relaxation/Contraction

Concerning relative query replacements, given a query $q \in Q$ we need to define $Up(q)$ and $Down(q)$. Note that $Up(q)$ and $Down(q)$ do not necessarily have to correspond to q^+ and q^- respectively. Due to lack of space we do not describe these functions in detail. Besides, mechanisms for query relaxation have already been proposed for several kinds of sources, including relational (Gaasterland, 1997), semi-structured (Dongwon, 2002), Description-Logics-based (Bidault et al., 2000; Mena et al., 1998) and Web sources (Li et al., 2002). We only note that the answer of the query $Up(q)$ (resp. $Down(q)$) should be larger (resp. smaller) than the current one, and that intentional query containment does not always implies extensional subsumption, e.g. if $I(a) = \{1\}$ and $I(b) = \{1\}$ then although a is intentionally contained by $a \vee b$, here we have $I(a) = I(a \vee b)$.

In conclusion, we have just showed that the generalized interaction scheme is feasible for taxonomy-based sources and that all associated computational tasks have polynomial complexity.

5 Concluding Remarks

We presented a unified generalized framework for information access that captures several kinds of interaction that are more flexible and more complex than those that are currently supported. We studied a set of context transition specifications which are elemental. Of course, they can be refined and adapted to the characteristics of specific kinds of sources or applications.

In addition, we specialized and demonstrated this framework for the case of taxonomy-based sources by describing all needed algorithms and by showing that these tasks are computationally tractable.

Further research includes specializing the generalized interaction scheme for sources with conceptual models that allow representing the relationships that may hold between the individual objects of the domain, notably Description Logics (DL) (Donini et al., 1996) knowledge bases, as DL is the knowledge representation language of the Semantic Web (Berners-Lee et al., 2001). We strongly suspect that this specialization is again feasible because we can view a DL source as a taxonomy-based source. Indeed, there are several approaches for constructing in polynomial time the taxonomy of concepts of a DL knowledge base (e.g. see (Sanner, 2003)).

Bibliography

"XFML: eXchangeable Faceted Metadata Language". http://www.xfml.org.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *"Modern Information Retrieval"*. ACM Press, Addison-Wesley, 1999.

Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web". *Scientific American*, May 2001.

Alain Bidault, Christine Froidevaux, and Brigitte Safar. "Repairing Queries in a Mediator Approach". In *Proceedings of the ECAI'00*, pages 406–410, Berlin, Germany, August 20-25 2000.

Lee Dongwon. *"Query Relaxation for XML Model"*. PhD thesis, University of California, 2002.

F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. "Reasoning in Description Logics". In Gerhard Brewka, editor, *Principles of Knowledge Representation*, chapter 1, pages 191–236. CSLI Publications, 1996.

T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42 (1):3–42, January 1995.

Terry Gaasterland. "Cooperative Answering through Controlled Query Relaxation". *IEEE Expert: Intelligent Systems and Their Applications*, 12(5), 1997.

Antony Galton. *"Logic for Information Technology"*. John Wiley & Sons, 1990.

Bernhard Ganter and Rudolf Wille. *"Formal Concept Analysis: Mathematical Foundations"*. Springer-Verlag, Heidelberg, 1999.

International Organization For Standardization. "Documentation - Guidelines for the establishment and development of monolingual thesauri", 1986. Ref. No ISO 2788-1986.

Wen-Syan Li, K. Seluk Candan, Quoc Vu, and Divyakant Agrawal. "Query Relaxation by Structure and Semantics for Retrieval of Logical Web Documents". *IEEE Transactions on Knowledge and Data Engineering*, 14(4), 2002.

Carlo Meghini and Yannis Tzitzikas. "An Abduction-based Method for Index Relaxation in Taxonomy-based Sources". In *Proceedings of MFCS 2003, 28th International Symposium on Mathematical Foundations of Computer Science*, pages 592–601, Bratislava, Slovak Republic, August 2003. Springer Verlag.

E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth. "Estimating Information Loss for Multi-ontology Based Query Processing". In *Proceedings of Second International and Interdisciplinary Workshop on Intelligent Information Integration*, Brighton Centre, Brighton, UK, August 1998.

Ruben Prieto-Diaz. "Implementing Faceted Classification for Software Reuse". *Communications of the ACM*, 34(5):88–97, 1991.

S. R. Ranganathan. "The Colon Classification". In Susan Artandi, editor, *Vol IV of the Rutgers Series on Systems for the Intellectual Organization of Information.* New Brunswick, NJ: Graduate School of Library Science, Rutgers University, 1965.

Giovanni M. Sacco. "Dynamic Taxonomies: A Model for Large Information Bases". *IEEE Transactions on Knowledge and Data Engineering*, 12(3), May 2000.

S. Sanner. "Towards practical taxonomic classification for description logics on the Semantic Web". Technical Report KSL-03-06, Stanford University, Knowledge Systems Lab, 2003.

Yannis Tzitzikas, Anastasia Analyti, Nicolas Spyratos, and Panos Constantopoulos. "An Algebraic Approach for Specifying Compound Terms in Faceted Taxonomies". In *Information Modelling and Knowledge Bases XV, 13th European-Japanese Conference on Information Modelling and Knowledge Bases, EJC'03*, pages 67–87. IOS Press, 2004a.

Yannis Tzitzikas and Carlo Meghini. “Ostensive Automatic Schema Mapping for Taxonomy-based Peer-to-Peer Systems”. In *Seventh International Workshop on Cooperative Information Agents, CIA-2003*, pages 78–92, Helsinki, Finland, August 2003. (Best Paper Award).

Yannis Tzitzikas, Carlo Meghini, and Nicolas Spyratos. “A Unifying Framework for Flexible Information Access in Taxonomy-based Sources”. In *Procs of the 6th Intern. Conference on Flexible Query Answering Systems ,FQAS'2004*, Lyon, France, June 2004b.

Yannis Tzitzikas, Carlo Meghini, and Nicolas Spyratos. “Towards a Generalized Interaction Scheme for Information Access”. In *Procs of the 3rd Intern. Symposium on Foundations of Information and Knowledge Systems,FoIKS'2004*, Vienna Austria, February 2004c.

Jeffrey D. Ullman. *“Principles of Database and Knowledge-Base Systems, Vol. I”*. Computer Science Press, 1988.

A Proofs

Prop. 1 $\bigvee_{o\in A} D_I(o) \sim n^+(A)$
Proof:
It suffices to prove the following two:
(a) The query $\bigvee_{o\in A} D_I(o)$ is a lower bound of $\{\ q \mid S(q) \supseteq A\}$.
(b) $\bigvee_{o\in A} D_I(o) \in \{\ q \mid S(q) \supseteq A\}$
(proof of (a))
Let x denote the query $\bigvee_{o\in A} D_I(o)$. We will prove that x is a lower bound of the set $\{\ q \mid S(q) \supseteq A\}$, i.e. we will prove $x \leq y$ for each $y \in \{\ q \mid S(q) \supseteq A\}$. Since $A \subseteq \bar{I}(y)$, for each $o \in A$, $o \in \bar{I}(y)$. Recall that each $o \in A$ is indexed under the set of terms $D_I(o)$. This implies that it must be $y \geq D_I(o)$ otherwise o would not be an element of $\bar{I}(y)$. Thus $y \geq \bigvee_{o\in A} D_I(o)$, i.e. $y \geq x$.
(proof of (b))
Each $o \in A$ is an element of $I(t)$ for each $t \in D_I(o)$. Thus o is an element of $I(D_I(o))$. This implies that $A \subseteq \bigcup\{I(D_I(o)) \mid o \in A\} = I(\bigvee_{o\in A} D_I(o))$. Since $I \sqsubseteq \bar{I}$, we infer that $I(\bigvee_{o\in A} D_I(o)) \subseteq \bar{I}(\bigvee_{o\in A} D_I(o))$, thus $\bigvee_{o\in A} D_I(o) \in \{\ q \mid S(q) \supseteq A\}$.

Since (according to (a)) the query $\bigvee_{o\in A} D_I(o)$ is a lower bound of $\{\ q \mid S(q) \supseteq A\}$ and (according to (b)) it is an element of $\{\ q \mid S(q) \supseteq A\}$, it follows that this query is the *glb* of $\{\ q \mid S(q) \supseteq A\}$.
$\diamond$

Prop. 2 $\bigvee\{\ D_I(o) \mid o \in A,\ S(D_I(o)) \subseteq A\} \sim n^-(A)$
Proof:
If suffices to prove the following two:
(a) The query $\bigvee\{\ D_I(o) \mid o \in A,\ S(D_I(o)) \subseteq A\}$ is an upper bound of $\{\ q \mid S(q) \subseteq A\}$.
(b) $\bigvee\{\ D_I(o) \mid o \in A,\ S(D_I(o)) \subseteq A\} \in \{\ q \mid S(q) \subseteq A\}$
(proof of (a))
Let x denote the query $\bigvee\{\ D_I(o) \mid o \in A,\ S(D_I(o)) \subseteq A\}$. We will prove that x is an upper bound of the set $\{\ q \mid S(q) \subseteq A\}$, i.e. we will prove $x \geq y$ for each $y \in \{\ q \mid S(q) \subseteq A\}$. Suppose that there is an object $o \in \bar{I}(y)$ such that $o \not\in \bar{I}(x)$. Since o is indexed under the set of terms $D_I(o)$, it must be $y \geq D_I(o)$ otherwise o would not be an element of $\bar{I}(y)$. If $S(D_I(o)) \subseteq A$ then certainly o would be an element of $\bar{I}(x)$ by the definition of x. So, let us suppose that $S(D_I(o)) \not\subseteq A$. In this case $y \geq D_I(o) \Leftrightarrow \bar{I}(y) \supseteq S(D_I(o))$. As $S(D_I(o)) \not\subseteq A$

we infer that $\bar{I}(y) \not\subseteq A$ which is a contradiction. Thus the hypothesis $o \notin \bar{I}(x)$ is not valid, hence x is an upper bound of $\{ \ q \mid S(q) \subseteq A\}$.

(proof of (b))
If $S(D_I(o)) \subseteq A$ then $\bigcup\{S(D_I(o)) \ |o \in A, \ S(D_I(o)) \subseteq A\} \subseteq A$. Thus $\bigvee\{ \ D_I(o) \mid o \in A, \ S(D_I(o)) \subseteq A\} \in \{ \ q \mid S(q) \subseteq A\}$.

Since (according to (a)) the query $\bigvee\{ \ D_I(o) \mid o \in A, \ S(D_I(o)) \subseteq A\}$ is an upper bound of $\{ \ q \mid S(q) \subseteq A\}$ and (according to (b)) it is an element of $\{ \ q \mid S(q) \subseteq A\}$, it follows that this query is the *lub* of $\{ \ q \mid S(q) \subseteq A\}$.
◇

Prop. 3 The time complexity for computing $n^+(A)$ is $O(|A||T|)$.
Proof:
Suppose that $(T, \leq)$ is stored as an adjacency matrix $\texttt{MT}[|\texttt{T}|, |\texttt{T}|]$ and I is stored as a matrix $\texttt{MI}[|\texttt{Obj}|, |\texttt{T}|]$ such that $\texttt{MI}[\texttt{o}, \texttt{t}] = \texttt{1}$ iff $o \in I(t)$.

The complexity of computing $D_I(o)$ for a given object $o \in Obj$ is $O(|T|)$ as we need to scan one row of the matrix MI. Consequently, the complexity of computing $D_I(o)$ for each o in a set $A \subseteq Obj$ is $|A|$ times the complexity of $D_I(o)$, thus it equals $O(|A||T|)$.
◇

Prop. 4 The time complexity for computing $n^-(A)$ is $O(|A||Obj||T|^2)$.
Proof:
Let us first measure the complexity of some basic tasks.

(a) The complexity of deciding whether $o \in I(t)$ is $O(1)$ as we have to check if $\texttt{MI}[\texttt{o}, \texttt{t}] = 1$.
(b) Let us now measure the complexity of deciding whether $o \in S(t)$. Clearly, $o \in S(t) \Leftrightarrow (o \in I(t) \vee (o \in I(t') \ \wedge \ t' \leq t)$. The right part of the above formula requires computing the terms that are narrower than t. We can find the narrower terms of a term in $O(|T|)$ as we only have to scan one row of MT (recall that $\leq$ is transitive). It follows (by also considering (a)) that we can decide whether $o \in S(t)$ is $O(|T|)$ time.
(c) The complexity of answering a single term query, i.e. computing $S(t)$, is $O(|Obj||T|)$ as we can do it by running decision problem (b) for each element of Obj.
(d) Let q be a conjunctive query $t_1 \wedge ... \wedge t_k$. Deciding whether $o \in S(q)$ requires $O(k|T|)$ time as we have to run k times decision problem (b).
(e) The complexity of answering a conjunctive query $t_1 \wedge ... \wedge t_k$, i.e. computing the set $S(t_1 \wedge ... \wedge t_k)$, is $O(|Obj|k|T|)$ time as we have to decide problem (d) for each object o of Obj.

Let us now measure the complexity of finding $n^-(A)$. For deciding whether $S(D_I(o)) \subseteq A$ we have to:
(1) compute $D_I(o)$,
(2) compute $S(D_I(o))$, and
(3) check if the latter is subset of A.

The computation of $D_I(o)$ for an object o is $O(|T|)$. The computation of $S(D_I(o))$ is $O(|Obj|k|T|)$ according to (e), i.e. $O(|Obj||T|^2)$. Task (2) is $O(Obj)$. So the tasks (1)-(3) can be performed in $O(|Obj||T|^2)$ time.

It follows that to perform the above tasks for every object of A requires $O(|A||Obj||T|^2)$ time.
◇

Prop. 5 The time complexity of source adaptation is:

- $O(|A^+| + |A^-||T|^2)$ for *single term* queries,
- $O(|A^- \cup A^+|k|T|^2)$ for *disjunctive* queries $t_1 \vee ... \vee t_k$,
- $O(|A^-|k|T|^2 + |A^+|k)$ for *conjunctive* queries $t_1 \wedge ... \wedge t_k$.

Proof:
Single-term queries.

(sa1) The complexity of $Add(o, t)$ is $O(1)$ as we only have to set `MI[o,t] = 1`.
(sa2) It follows from (sa1) that the complexity of adding all objects in A^+ is $O(|A^+|)$.
(sd1) Let us now measure the complexity of $Delete(o, t)$. The computation of $D_I(o)$ is $O(|T|)$ according to Prop. 3. However, the computation of $D_{\bar{I}}(o)$ is $O(|T|^2)$ as it requires computing $Br(t)$ for each term t in $D_I(o)$. Now as the computation of $Nr(t)$ is again $O(|T|)$ and the addition/deletetion of an object from the interpretation of one term is $O(1)$, we conclude that the complexity of $Delete(o, t)$ is $O(|T|^2)$.
(sd2) It follows from (sd1) that the complexity of deleting all objects in A^- is $O(|A^-||T|^2)$.

From (sa2) and (sd2) we infer that the complexity of source adaptation for single term queries is $O(|A^+| + |A^-||T|^2)$.

Disjunctive queries.

(dd1) It follows from (sd1) that the complexity of $Delete(o, t_1 \vee ... \vee t_k)$ is $O(k|T|^2)$.
(dd2) It follows from (dd1) that the complexity of $Delete(A^-, t_1 \vee ... \vee t_k)$ is $O(|A^-|k|T|^2)$.
(da1) Let us now measure the complexity of $Add(o, t_1 \vee ... \vee t_k)$. This involves computing k times the prosthetic perturbation *pertAdd*. As $Br(t)$ can be computed in $O(|T|)$ time, $D_{\bar{I}}(o)$ in $O(|T|^2)$ time, it follows the *pertAdd* can be computed in $O(|T|^2)$ time. Thus the complexity of $Add(o, t_1 \vee ... \vee t_k)$ is $O(k|T|^2)$.
(da2) It follows from (da1) that the complexity of $Add(A^+, t_1 \vee ... \vee t_k)$ is $O(|A^+|k|T|^2)$.

It follows from (dd2) and (da2) that the time complexity of source adaptation for disjunctive queries is $O(|A^-|k|T|^2) + O(|A^+|k|T|^2)$. Now as A^- and A^+ are disjoint, this complexity equals $O(|A^- \cup A^+|k|T|^2)$.

Conjunctive queries.

(ca1) The complexity of $Add(o, t_1 \wedge ... \wedge t_k)$ is $O(k)$.
(ca2) It follows from (ca1) that the complexity of $Add(A^+, t_1 \wedge ... \wedge t_k)$ is $O(|A^+|k)$.
(cd1) Let us now measure the complexity of $Delete(o, t_1 \wedge ... \wedge t_k)$. This involves computing k times the apheretic perturbation *pertDel*. *pertDel* can be computed in $O(|T|^2)$ time. Thus the complexity of $Delete(o, t_1 \wedge ... \wedge t_k)$ is $O(k|T|^2)$.
(cd2) It follows form (cd1) that the complexity of $Delete(A^-, t_1 \wedge ... \wedge t_k)$ is $O(|A^-|k|T|^2)$.

It follows from (ca2) and (cd2) that the time complexity of source adaptation for conjunctive queries is $O(|A^-|k|T|^2 + |A^+|k)$.
◇