HAL
open science

Query-Answering CG Knowledge Bases

Michel Leclère, Nicolas Moreau

To cite this version:

Michel Leclère, Nicolas Moreau. Query-Answering CG Knowledge Bases. ICCS: International Conference on Conceptual Structures, Jul 2008, Toulouse, France. pp.147-160, 10.1007/978-3-540-70596-3_10 . lirmm-00300160

HAL Id: lirmm-00300160

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00300160v1

Submitted on 21 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Query-answering CG Knowledge Bases*

Michel Leclère and Nicolas Moreau

LIRMM, Univ. Montpellier 2, CNRS
161, rue Ada
34392 Montpellier, France
{leclere,moreau}@lirmm.fr

Abstract. Conceptual graphs are a good choice for constructing and exploiting a knowledge base. In several of our projects (semantic portal for e-tourism, exploitation of digital object corpus, etc.), we have to query such bases. So it is natural to consider queries and bases as simple graphs and to compute the set of all projections from a query to a base. However, there is a problem of the return of this set of projections to the user. More generally, the main issue is about the definition of the notion of answers in an query-answering system made of knowledge bases formalized by graphs (Conceptual Graphs, RDF (Resource Description Framework) , Topic Maps, etc.). In this paper, we study several notions of answers and some of their characterizations. We distinguish between notions of answers by subgraphs of the base and answers by creation of result graphs. For the last type of answers, we define completeness, non-redundancy and minimality criteria of the answer sets and propose several notions of answers w.r.t these criteria.

1 Introduction

Many knowledge applications involve the elaboration and use of knowledge bases. Some examples are document management, digital object corpus management, enterprise knowledge repositories, construction of semantic portals, teaching aid management, the semantic web, etc. Two general contexts for using such bases can be noted, whereby use of these bases presupposes a query Q specifying the knowledge to be searched:

- Annotation context: resources are annotated by "descriptions" characterizing it; in this case, the exploitation is based on a search of resources whose annotations contain specific knowledge (e.g. [1–3]). This type of exploitation only requires a definition of a deduction notion allowing selection of descriptions D (and thus the resources R linked to these descriptions) which are deductions of the searched knowledge Q. The set of answers to a query Q on an annotation base B is $\{R \mid (R, D) \in B \ \wedge \ D \models Q\}$;
- Knowledge base context: some unstructured data that comply with a formal vocabulary defined by an ontology are stored in a base of assertions [4];

* This work is supported by the Eiffel project, funded by ANR-RNTL.

a querying system allows extraction of knowledge from the base (moreover, this type of exploitation could be used in the annotation context, considering descriptions of a subset of resources). This type of exploitation also needs a deduction notion to characterize the existence of an answer, but it also requires a definition of what should be returned to a query Q on such a knowledge base.

In this paper, we propose a preliminary answer to this question in the framework of knowledge bases formalized by conceptual graphs [5]. The long-term goal of this work is to define a query language for conceptual graphs and to implant it as a knowledge server over the CoGITaNT framework [6]. This first approach is a study of different notions of answers constructed from projections of a query graph Q to a graph base B.

In a knowledge base querying system with a formal semantic, answers are built upon "pre-answers" that are logical consequences of the base proving the existence of answers. A query is often composed of two parts: an head which specifies how answers are constructed from the "pre-answers", and a body which details how to select these "pre-answers". In the case of a base only made of conjunctive assertions, these "pre-answers" are the "smallest parts" of the base which has the body of the query as a logical consequence.

In this work, we only consider such simple bases (*i.e.* without rules) formalized by conceptual graphs [7], although our results could be directly applied to other labeled graph formalisms, particularly to RDF/S knowledge bases of the semantic web [8] (*cf.* [9] for an equivalence of the two formalisms).

Unlike relational databases, these formalisms allow the introduction of an order relation over relations permitting the representation of a simple ontology and, more importantly, the use of variables in the base. In the querying mechanism, this leads to the problem of "pre-answer" equivalence and thus the problem of the definition of the answer notion.

We consider queries whose body is a conceptual graph and focus on the definition of several "pre-answer" notions. The main goal of this preliminary work is to study redundancy problems of these "pre-answers" which arise regardless of following operations applied to these "pre-answers" (e.g. specifyingthe concepts to keep by a lambda, or constructing a new graph with these answers).

As far as we know, this topic has not yet been studied. Several proposals have been made on querying relational databases with conceptual graphs [10–12]. Many studies have been conducted in an annotation context (e.g. [13, 2]). An adaptation of relational algebra to the context of conceptual graph knowledge bases has been studied by S. Coulondre [14]. The relational bias limits answers to tuples of individuals over which several relational algebra operators are used (moreover, this author did not seem to consider variable-free knowledge bases). The most similar works were carried out by C. Gutierrez et al. [15], who studied querying of knowledge bases of the semantic web formalized by RDF/S, but the set of "pre-answers" is not fully detailed, particularly the problem of answer redundancies.

The following section briefly introduces the main notions of the conceptual graph formalism upon which our work is based. Section 3 defines the querying model. Section 4 proposes several notions of answers and answer criteria based on the answers redundancy problem. Finally, the conclusion proposes some ideas for other types of answers.

2 $\mathcal{SG}$ formalism

The CG formalism we use in this paper has been developed at LIRMM over the last 15 years [7]. The main difference with respect to the initial general model of Sowa [5] is that only representation primitives allowing graph-based reasoning are accepted. Several extensions that preserve this link with graph theory have been introduced (rules, constraints, conjunctive types, nested graphs, etc.), however, for the sake of clarity and presentation, we only present the simplest model in this paper.

Simple graphs (SGs) are built upon a *support*, which is a structure $S = (T_C,\ T_R,\ I,\ \sigma)$ where T_C is the set of concept types, T_R is the set of relations with any arity (arity is the number of arguments of the relation). T_C and T_R are partially ordered sets. The partial order represents a specialization relation ($t' \leq t$ is read as "t' is a specialization of t"). I is a set of individual markers. The mapping σ assigns a signature to each relation specifying its arity and the maximal type for each of its arguments.

SGs are labeled bipartite graphs denoted by $G = (C_G, R_G, E_G, l_G)$ where C_G and R_G are the concept and relation node sets respectively, E_G is the set of edges and l_G is the mapping labeling nodes and edges. Concept nodes are labeled by a couple $t : m$ where t is a concept type and m is a marker. If the node represents an unspecified entity, its marker is the generic marker, denoted $*$, and the node is called a *generic* node, otherwise its marker is an element of I, and the node is called an *individual* node. Relation nodes are labeled by a relation r and, if n is the arity of r, it is incidental to n totally ordered edges.

A graph $G = (C_G, R_G, E_G, l_G)$ is *consistent* w.r.t. a support $S = (T_C, T_R, I, \sigma)$ if :

- the labels of the concept nodes (resp. relation nodes) belong to $(T_C \times (I \cup \{*\}))$ (resp. T_R);
- the relation nodes satisfy their signatures defined by σ.
- for each individual marker i of G, types of concept nodes with this marker have a greatest lower bound. This condition can differ if one considers a conformity relation in the support, if one imposes a lattice structure to the ordered set of concept types, if banned types are considered (a disjointness type axiom), etc.

A specialization/generalization relation corresponding to a deduction notion is defined over SGs and can be easily characterized by a graph homomorphism called *projection*. When there is a projection π from G to H, H is considered to be more specialized than G, denoted $H \leq G$. More specifically, a projection π

from G to H is a mapping from C_G to C_H and from R_G to R_H, which preserves edges (if there is an edge numbered i between r and c in G then there is an edge numbered i between $\pi(r)$ and $\pi(c)$ in H) and may specialize labels (by observing type orders and allow substitution of a generic marker by an individual one).

Conceptual graphs are provided with a first-order-logic semantics, defined by a mapping denoted Φ.

The fundamental result of *projection soundness and completeness* establishes the equivalence between projection and deduction on formulas assigned to SGs: given two SGs G and H on a support S, there is a projection from G to H if and only if $\Phi(G)$ can be deduced from $\Phi(H)$ and $\Phi(S)$. Completeness is obtained up to a condition on H: H has to be in a normal form, so any individual marker must appear at most once in it (i.e. a specific entity cannot be represented by two nodes). An SG consistent w.r.t. a support can be easily normalized by joining concept nodes with a same individual marker. The normal form of a consistent graph G is denoted $norm(G)$.

Two notions of equivalence can be defined over SGs: a syntactic equivalence, and a semantic one. The syntactic equivalence is characterized by the existence of an isomorphism between two graphs (which is a bijective mapping from nodes of one of the graphs to nodes of the other preserving edges and without label specialization). The semantic equivalence is characterized by the existence of a projection from the first graph to the normal form of the second, and from the second to the normal form of the first and corresponds to a logic equivalence between formula associated with graphs: $\Phi(S) \models \Phi(G) \leftrightarrow \Phi(H)$ *iff there is a projection from* G *to* $norm(H)$ *and a projection from* H *to* $norm(G)$ (denoted $G \equiv H$).

This equivalence relation defines classes of equivalent SGs. SGs in figure 1 are from the same equivalence class. In each class, some graphs contain useless knowledge repetitions (redundancies) and there is a sole smallest graph with no redundancy, called the irredundant graph of the class (cf. [16]). A graph that is not in normal form contains redundancies (if two concept nodes have the same individual marker).

A subSG $H = (C_H, R_H, E_H, l_H)$ of an SG $G = (C_G, R_G, E_G, l_G)$ is an SG, where :

- $C_H \subseteq C_G$ and $R_H \subseteq R_G$
- E_H is a restriction of E_G to elements of $C_H \times R_H$
- l_H is a restriction of l_G to elements of H.

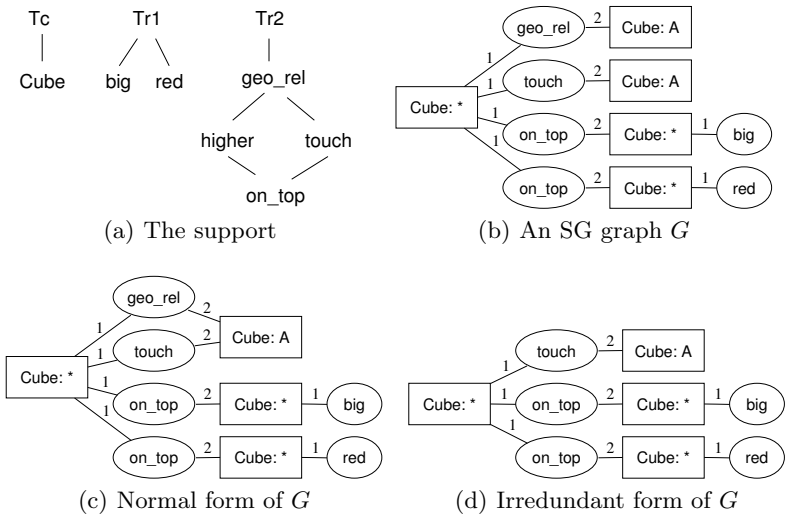
A strict sub-SG of G is a sub-SG with a strictly inferior number of nodes.

Characterization : An SG is said to be *redundant* if it is not in normal form or if it is equivalent to one of its strict sub-SGs. Otherwise, it is said to be *irredundant* [1].

[1] Note that our irredundant definition is stricter than that given in [7].

Property : [16] An equivalence class contains one and only one irredundant SG, which is the graph (single up to isomorphism) with the smallest set of nodes.

An algorithm to compute the irredundant form of an SG, whose complexity is polynomially related to the complexity of the projection algorithm, has been described in [17].



(a) The support (b) An SG graph G

(c) Normal form of G (d) Irredundant form of G

Fig. 1. Three equivalent SGs built on a support.

3 Studied querying model

The chosen context is a knowledge base composed of assertions of entity existences and relations over these entities, called *facts*, and stored in a single graph (not necessarily connected) consistent w.r.t. a given support. A support can be seen as a basic ontology. This framework does not put forward hypotheses on how facts have been collected and does not prohibit the existence of metadata (date, etc.) on facts composing the base (e.g. a selection mechanism of facts to be queried based on metadata). The only important hypothesis put forward is that all of the facts of the base to be queried are consistent relative to each other (with respect to individual markers) and consistent w.r.t. the support. Figure 1(a) presents the support used in the following examples of the paper.

The SG *base* is assumed to be normalized. The base can be redundant because the irredundancy of the base does not solve the problem of answer redundancies, and computation of the irredundant form is expensive as the base can be large. Computation of the normal form is linear in the size of the base. Moreover, it is easy to write an incremental algorithm (called for each addition of knowledge in the base and whose complexity depends only on the size of the addition) for

the normalization, whereas it seems difficult to find such an efficient algorithm for computation of the irredundant form.

From propositions of querying languages proposed for such knowledge base (e.g. SPARQL [18]), their definitions are clearly based on several mechanisms:

1. Adaptation of the base to the query, which consists of computing a base D' from a base D by application of a set of updating operations P (e.g. rule applications).
2. Selection of relevant parts of the base that respond to the query. This selection is made of "patterns" which specify selection criteria of base parts useful for contruction of the answer. These patterns have the same formalism as the base or indirectly of the same formalism because of the addition of variables. Thus these patterns are used like filters to select relevant base parts.
3. Verification of properties allowing to impose complementary selection criteria differing from a simple assertion of relations between entities : path existence, constraints (not present in the representation language) on the entity linked to a variable.
4. Construction of an answer from these parts, which is a specification of the type of answer to return (tuples of values associated with query variables, a graph built from all parts, the number of answers, etc.).

The second point is the core of the querying mechanism. Constructing a query boils down to making assertions with unknown values (variables) which is information to be retrieved. When the formalism allows the introduction of variables in the base, it is important to know what to do when these variables are linked with query variables.

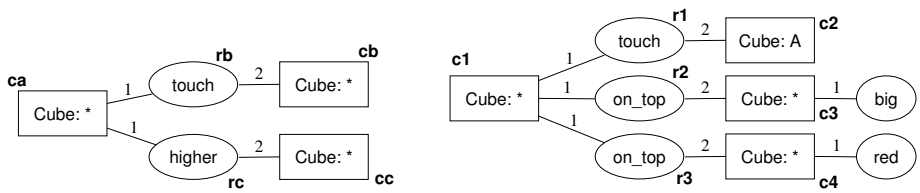
In our formalism, the selection criterion is a given SG Q, called the *query* SG. There is no constraint on the query (in terms of relevance, normalization or irredundancy). However, it seems natural to verify the consistency of the query w.r.t. the support of the base to avoid queries with no links to the base. One can consider to compute the irredundant form of the query as the size of the query is generally small. One can consider an unconnected query as several queries.

Therefore as one considers formalisms provided with a formal semantic, one can define "relevant parts" as "the smallest subgraphs" of the base whose query graph is a logical consequence; such subgraphs are called *pre-answers* in [15]. In conceptual graph formalism, the existence of an answer is directly linked with the existence of a projection and a pre-answer is just the query image of a projection.

Definition 1 (Proofs of answers). *Let B be an SG base and Q an SG query, a proof of answer is a projection π from Q to B. The set of proofs of answers from Q to B is denoted $\Pi(Q,B)$: $\Pi(Q,B) = \{\pi_i \mid \pi_i : Q \longrightarrow B$ is a projection$\}$.*

Definition 2 (Images of proofs). *An image of a proof (or pre-answer), denoted $\pi(Q)$, is a sub-SG of B, image of the proof of answer π from Q to B. The images of proofs sequence from a query Q to a base B, denoted $IP(Q,B)$, is $IP(Q,B) = \langle \pi_1(Q), ..., \pi_n(Q) \rangle$, where n is the size of $\Pi(Q,B)$.*

All examples of the paper are from base and query of figure 2. Base is the SG of the figure 1(d). There are six projections of the query to the base, and thus six images of proofs. We have named some vertices (c_1, r_2, etc.) of graphs to refer directly to one vertex or to distinguish different subgraphs (see figure 3).



(a) Query (b) Base

Fig. 2. Base and query (with named vertices).

Therefore answers are based on images of proofs, one may have to return the same subgraph of B several times. In the example of figure 2, projections $\pi_i = \{(c_a, c_1), (r_b, r_2), (c_b, c_3), (r_c, r_3), (c_c, c_4)\}$ and $\pi_j = \{(c_a, c_1), (r_b, r_3), (c_b, c_4), (r_c, r_2), (c_c, c_3)\}$ define the same subgraph (R_5 on figure 3). One can choose two ways to solve this problem:

- Considering it as the same answer (several proofs for the answer)
- Differentiating answers by representing the projection in answer graphs (e.g. by adding id of query vertices to their images vertices in answers).

We consider only the first case, as we want to express answers in the same language (structure and vocabulary) as the base and query.

4 Different notions of answers

In this section we study several kinds of answers. The first type of notion is to keep base subgraphs, similary to images of proofs.

4.1 Answering by base subgraphs

The most basic answer that can be returned is the set of images of proofs.

Definition 3 (Answer by image subgraphs). *The set of images of proofs of a query Q in a base B, noted $R_{IP}(Q, B)$, is $R_{IP}(Q, B) = \{\pi(Q) \mid \pi \in \Pi(Q, B)\}$.*

This answer notion can be used to select exploration start points of the base (by focusing a subgraph of B), to explore the base or to be a first step of base-updating queries. Figure 3 shows all of the five subgraphs answering the query.

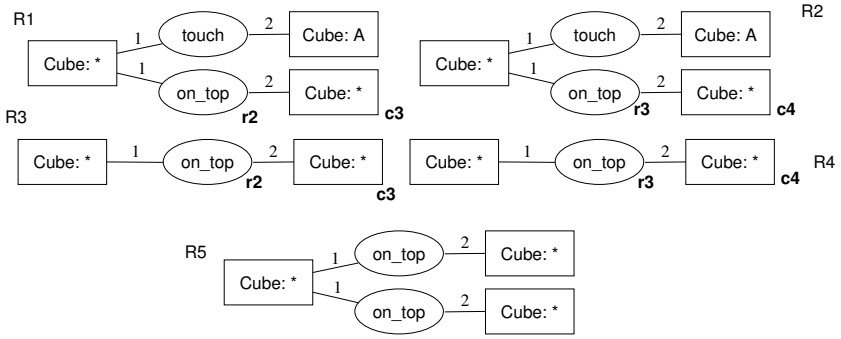


Fig. 3. The five subgraphs composing $R_{IP}(Q, B)$.

4.2 Answering by base-independent graphs

In many cases, the query language should allow knowledge extraction rather than "pinpointing" knowledge in the base. Answers are "copies" of base subgraphs. Thus, answers result from the construction of isomorphic graphs of images of proofs.

Since answer graphs are constructed up to an isomorphism, two isomorphic graphs should be considered equal. The set of answers is no longer a set of subgraphs of B, but rather a set of graphs isomorphic to images of proofs subgraphs of B. In such a set of answer graphs there are no two isomorphic graphs.

Definition 4 (Graph set (iso-set)). *A graph (iso-)set $\{G_1, ..., G_n\}$ is a set of graphs in which, for all i, j with $i \neq j$, G_i is not isomorphic to G_j.*

Hereafter, the term "set of graphs" is short for the preceding iso-set notion (a set has no more two isomorphic graphs)[2].

An iso-answer is an answer notion corresponding to computation of all isomorphic graphs to images of proofs. On the example, $R_{ISO}(Q, B)$ is equal to graphs in figure 4.

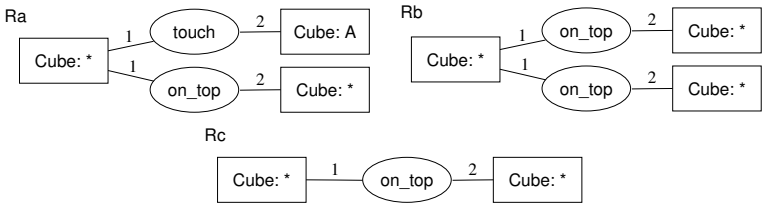


Fig. 4. Graphs composing $R_{ISO}(Q, B)$.

[2] The images of proofs sets can contain isomorphic subgraphs (if they are not the same subgraph of the base).

Definition 5 (Iso-answer). *An iso-answer from Q to B is an isomorphic graph corresponding to an image of a proof from Q to B. The set of iso-answers is $R_{ISO}(Q,B) = \{G \mid \text{where } G_i \in R_{IP}(Q,B) \text{ with } G_i \text{ isomorphic to } G\}$*

With this answer notion, the link with the base is lost (particularly when the answer has no individual marker) since it is unknown which nodes of the base corresponds to a generic node of an answer, or even how many images of proofs correspond to the same answer graph [3]. Only the proof of the existence of a particular knowledge in the base is preserved with this notion. Thus, one can question the relevance of only considering isomorphism (i.e. a syntactic equivalence) as an equivalence criterion. In our example, graphs R_b and R_c in figure 4 both state that "there is a cube on top of a cube".

It seems advisable to detect equivalent answers (in terms of semantics) and to only keep one graph of each equivalent class. We define a criterion of such a set of answers :

Definition 6 (Without equivalence criterion). *A set of answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is "without equivalence" iff $\forall G_i \in R(Q,B)$, $\nexists G_j \in R(Q,B)$ with $G_j \neq G_i$ with $G_i \equiv G_j$.*

Two types of equivalences arise. The first is from the implicit relation of equality of two nodes having the same individual marker. This type of equivalence is avoided by the base normalization; therefore all images of proofs are in normal form (even if the query is not normalized). The second comes from the intrinsic redundancies of the language, and cannot be handled previously, because even if the base and query are irredundant, proof subgraphs are not necessarily irredundant (in the example, image of a proof R_5 is redundant despite that base and query are irredundant).

In each equivalence class, there is a single graph (up to isomorphism) that is the smallest graph of his class: it is the irredundant form. It seems natural to choose this graph to represent one equivalence class of images of proofs. In the example, $R_{IRR}(Q,B)$ is equal to graphs R_a and R_c in figure 4.

Definition 7 (Irredundant answers). *The set of irredundant forms of images of proofs from Q to B is $R_{IRR}(Q,B) = \{Irr(G) \mid G \in R_{ISO}(Q,B)\}$.*

This notion of irredundant answers may seem strange because one may think that an answer could not be isomorphic to an image of a proof. The following property holds that the irredundant form of each image of a proof is itself an answer (up to isomorphism) and that the set $R_{IRR}(Q,B)$ does not have equivalent answers.

Proposition 1. *$R_{IRR}(Q,B) \subseteq R_{ISO}(Q,B)$ is "without equivalence".*

[3] The open world assumption of knowledge bases does not indicate whether two different but isomorphic images of proofs represent the same "situation" of the world, or two similar "situations".

Proof. For all $G \in R_{IRR}(Q,B)$, there is $G' \in R_{IP}(Q,B)$ such that G is isomorphic to $Irr(G')$. So there is a projection π' from Q to B such that $\pi'(Q) = G'$. There is also, by definition of the irredundant form, a projection π_r from G' to B such that $\pi_r(G') = Irr(G')$. Thus there is a projection $\pi = \pi_r \circ \pi'$ from Q to B such that $\pi(Q) = Irr(G')$ and so isomorphic copy of $Irr(G')$ is an answer of $R_{ISO}(Q,B)$. The unicity property of the irredundant form (up to isomorphism) in each equivalence class and our notion of graphs defined up to an isomorphism lead to the conclusion that $R_{IRR}(Q,B)$ is without equivalence.

The notion of answers without equivalence is not sufficient because, as one does not want to keep two equivalent answers, only answers that add new knowledge compared to other answers should be conserved. In the example, answer R_a states that "there is a cube that touches cube A and that is on top of a cube" and answer R_b states that "there is a cube on top of a cube", so knowledge of R_b is expressed by R_a. So we introduce a new incomparability criterion in the next section.

4.3 Sets of incomparable answers

Two answers are comparable if knowledge stated by the first is deducible [4] from the other. Once there are comparable answers in a set of answers, there is redundancy between them. The idea is to eliminate this redundancy by keeping only incomparable answers.

Definition 8 (Incomparability criterion). *A set of answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is "without redundancy" when all of its answers are incomparable: $\forall G_i, G_j \in R(Q,B)$ with $G_j \neq G_i$, $G_i \not\leq G_j$.*

Given a set of answers $R_{ISO}(Q,B)$ there is not a single subset "without redundancy" (e.g. $\{R_a\}$ or $\{R_b\}$). A natural constraint is to make such a subset to not remove too many answers, *i.e* it has to be maximal by inclusion.

Definition 9 (Maximality criterion). *A set of incomparable answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is "maximal" when no other answer can be added without adding redundancy: $\forall G \in R_{ISO}(Q,B) \setminus R(Q,B)$, $\exists G' \in R(Q,B)$ such that $G' \leq G$ or $G \leq G'$.*

This notion still does not ensure unicity of such a subset. One can want to constrain a bit more the notion of maximality by forcing the subset of answers R to be complete, as one wishes that all of the answers of $R_{ISO}(Q,B)$ can be deduced from answers of the subset. We thus extend the logical interpretation operator Φ from SGs to sets of SGs by taking the conjunction of logical formula associated with each graph of the set : so if $R = \{r_1, ...r_n\}$, $\Phi(R) = \Phi(r_1) \wedge ... \wedge \Phi(r_n)$.

[4] One could distinguish cases of a simple knowledge inclusion from cases of general deduction by using ontology knowledge, depending on the level of ontology knowledge appropriation by the user.

Definition 10 (Completeness criterion). *A set of answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is complete iff $\Phi(S), \Phi(R(Q,B)) \models \Phi(R_{ISO}(Q,B))$.*

We call the *normalized disjoint union* ($NormalizedDisjointUnion$) of a set of graphs E, the normal form of the graph resulting from the join of nodes and edges (and labeling functions) of graphs of E.

Proposition 2. *A set of answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is complete iff the normalized disjoint union of $R(Q,B)$ is more specialized than all of the answers of $R_{ISO}(Q,B)$, i.e. $\forall G \in R_{ISO}(Q,B)$, $NormalizedDisjointUnion(R(Q,B)) \leq G$.*

The proof is trivial, note that $\Phi(NormalizedDisjointUnion(R(Q,B)) \equiv \Phi(R(Q,B))$.

Corollary 1. *A set of complete incomparable answers is maximal.*

The completeness criterion still does not ensure unicity of a subset of answers (because of equivalent answers). A natural choice is to take the smallest set of answers. This leads to the definition of a minimality criterion based on a notion of answer size and equivalence of answers sets.

Definition 11 (Answer size). *The size of a set of answers $R(Q,B)$ is the sum of the number of nodes of all answers: $\Sigma_{g \in R(Q,B)} card(g)$.*

Definition 12 (Answers sets equivalence). *Two sets of answers $R(Q,B)$ and $R'(Q,B)$ of a query Q on base B and consistent w.r.t. a support S are equivalent iff $\Phi(S) \models \Phi(R(Q,B)) \leftrightarrow \Phi(R'(Q,B))$.*

Definition 13 (Minimality criterion). *A set of answers $R(Q,B) \subseteq R_{ISO}(Q,B)$ is minimal iff there is not an equivalent set of answers with a strictly smaller size.*

Proposition 3. *$R(Q,B)$ is minimal iff $R(Q,B) \subseteq R_{IRR}(Q,B)$.*

Corollary 2. *Each minimal set is unique (up to isomorphism).*

The previous proposal and its corollary can be easily deduced from irredundant graph properties.

Completeness and minimality constraints define a notion of answer that seems more appropriate when one searches to retrieve knowledge stored in a knowledge base.

Definition 14 (Most specific answers). *The set of the most specific irredundant answers is $R_{MIN}(Q,B) = \{G \in R_{IRR}(Q,B) \mid \nexists G' \neq G \in R_{IRR}(Q,B)$ with $G' \leq G\}$.*

On the example, $R_{MIN}(Q,B)$ is equal to graph R_a in figure 4.

Theorem 1. *$R_{MIN}(Q, B)$ is the only complete minimal and incomparable subset of answers of $R_{ISO}(Q, B)$.*

Proof. • Incomparability: $R_{MIN}(Q, B)$ is by definition composed of the most specific elements of $R_{IRR}(Q, B)$, and as two elements of $R_{IRR}(Q, B)$ are not equivalent, all of the answers of $R_{MIN}(Q, B)$ are incomparable. • Completeness: $R_{IRR}(Q, B)$ is complete because each answer of $R_{ISO}(Q, B)$ is equivalent to a graph of $R_{IRR}(Q, B)$. By the definition of $R_{MIN}(Q, B)$, one deletes in $R_{IRR}(Q, B)$ only G_i which is more general than another graph of $R_{IRR}(Q, B)$. Thus, $R_{MIN}(Q, B)$ is complete. • Minimality and unicity: $R_{MIN}(Q, B)$ is a subset of $R_{IRR}(Q, B)$, thus it is minimal and unique w.r.t. corollary 2. □

One can define an answer notion like that on the most specific element, but this time with the most general ones. This notion is like a "summary" of the set of answers: not all of the answers are returned, but a minimal subset generalizing all of the answers.

Definition 15 (Summary). *A set of answers $R(Q, B) \subseteq R_{ISO}(Q, B)$ is a summary iff it is minimal and it generalizes all of the answers of $R_{ISO}(Q, B)$, that is $\forall G \in R_{ISO}(Q, B)$, $\exists G' \in R(Q, B)$ such that $G' \geq G$.*

Definition 16 (Maximal answer). *The set of all of the most general irredundant answers is $R_{MAX}(Q, B) = \{G \in R_{IRR}(Q, B) \mid \nexists G' \neq G \in R_{IRR}(Q, B)$ with $G' \geq G\}$.*

Property 1. $R_{MAX}(Q, B)$ is the sole minimal and maximal subset of incomparable answers of $R_{ISO}(Q, B)$.

The proof is similar to the proof of property 1. In the example, $R_{MAX}(Q, B)$ is equal to graph R_c in figure 4.

4.4 Case of bases composed of (only) individual concepts

A special case is when a knowledge base does not contain any variables (corresponding to relational databases). In such bases, there is only a kind of redundancy from redundant relations between concepts (relations whose type is comparable and with the same ordered set of neighbors). Computation of the irredundant form is thus linear and incremental.

Proposition 4. *In an irredundant base whose concepts are all individual, all of the images of proofs of any query on this base are non-isomorphic and irredundant.*

Proof. A base in normal form and which only contains concepts that are individuals does not contain concept nodes with the same label (nor comparable concept nodes in terms of specialization/generalization). Moreover, as the base is irredundant, it does not contain redundant relations between concepts (all the more with the same label). So there is not any isomorphism (or projection) from

a subgraph of the base to another (except identity relation). Images of proofs are base subgraphs so they are non-isomorphic. They are irredundant because none of the base subgraph can be projected in one of its strict subgraphs (since there is no projection from a base subgraph in another, except identity). □

Corollary 3. *If B is irredundant, for any query Q there is a bijection from the set of images of proofs $R_{IP}(Q, B)$ to their copies $R_{ISO}(Q, B)$, and $R_{ISO}(Q, B) = R_{IRR}(Q, B)$.*

5 Conclusion

In this paper we define two main notions of answers to a query in the knowledge base querying framework: the first is composed of base subgraphs that can allow browsing in the knowledge base or that can be used as a first step to update queries; the second consists of graphs independent of the base. We define several good criteria for this last notion: the non-equivalence of answers, the incomparability of answers, the completeness of a set of answers, and the minimality (in terms of size) of a set of answers. The most interesting notion of answer w.r.t. these criteria is the set of the most specific irredundant answers. However, in one of our projects, we need the notion of answers of the set of most general irredundant answers, as these answers are used as the body of new queries in another knowledge base.

An another answer notion, not developed in this paper is the construction of a graph resulting from the disjoint sum of all answers. the definition of such a notion gives a closed querying system (query, knowledge base and answer are in the same formalism) and allows us to reuse the result of a query as a knowledge base (nested queries).

Finally, in this paper, we overcome redundancies between answers by deleting answers. Another possibility is to overcome redundancies (when it is not a redundancy of the knowledge base itself) by completing answers by the addition of knowledge from the base allowing to differentiate redundant answers. We are currently working on a definition of such an answer contextualization mechanism. This kind of mechanism seems relevant for such knowledge bases because, contrary to relational databases in which a hypothesis is put forward that the creator of the request knows the schema of the database, they are by definition weekly structured and the only reasonable hypothesis put forward is that the creator can verify that his query is correct w.r.t. the ontology. A contextualization mechanism thus allows to respond to queries of the type: *"What knowledge can I have on animals owned by Mary?"* with a set of such answers : { *"Mary owns a pedigree animal", "Mary owns a cat offered by her father", "The preferred animal that Mary owns is a cat"*} rather than the only answer *"Mary owns a cat"*.

References

1. Hollink, L., Schreiber, A., Wielemaker, J., Wielinga, B.: Semantic annotation of image collections. Knowledge Capture (2003) 41–48
2. Moreau, N., Leclère, M., Chein, M., Gutierrez, A.: Formal and graphical annotations for digital objects. In: SADPI '07: Proceedings of the 2007 international workshop on Semantically aware document processing and indexing, New York, NY, USA, ACM (2007) 69–78
3. Dieng-Kuntz, R., Corby, O.: Conceptual Graphs for Semantic Web Applications. Conceptual Structures: Common Semantics for Sharing Knowledge: 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005: Proceedings (2005)
4. Fensel, D., Decker, S., Erdmann, M., Studer, R.: Ontobroker: Or How to Enable Intelligent Access to the WWW. Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada (1998)
5. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley (1984)
6. Genest, D., Salvat, E.: A platform allowing typed nested graphes: How cogito became cogitant. In: Proceedings of the Sixth International Conference on Conceptual Structures (ICCS'98). Volume 1453 of LNCS., Springer (1998) 154–161
7. Chein, M., Mugnier, M.L.: Conceptual Graphs: Fundamental Notions. Revue d'Intelligence Artificielle **6**(4) (1992) 365–406
8. Hayes, P.: RDF Semantics. Technical report, W3C (2004)
9. Baget, J.F.: Rdf entailment as a graph homomorphism. (2005) 82–96
10. Sowa, J.F.: Conceptual graphs for a data base interface. IBM Journal of Research and Development **20**(4) (1976) 336–357
11. Boksenbaum, C., Carbonneill, B., Haemmerlé, O., Libourel, T.: Conceptual graphs for relational databases. In: Proceedings of the first International Conference on Conceptual Structures (ICCS'93). Number 699 in LNAI, Springer (1993) 142–161
12. Haemmerlé, O., Carbonneill, B.: Interfacing a relational databases using conceptual graphs. In: Proceedings of the Seventh International Conference and Workshop on Database and Expert Systems Applications (DEXA'96), IEEE-CS Press (1996) 499–505
13. Genest, D., Chein, M.: A content-search information retrieval process based on conceptual graphs. Knowl. Inf. Syst. **8**(3) (2005) 292–309
14. Coulondre, S.: Cg-sql: A front-end language for conceptual graph knowledge bases. Knowledge-Based Systems **12**(5-6) (1999) 205–325
15. Gutierrez, C., Hurtado, C., Mendelzon, A.O.: Foundations of semantic web databases. In: PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York, NY, USA, ACM Press (2004) 95–106
16. Mugnier, M.L., Chein, M.: Polynomial algorithms for projection and matching. In: Proceedings of the 7th Annual Workshop on Conceptual Structures: Theory and Implementation, London, UK, Springer-Verlag (1993) 239–251
17. Mugnier, M.: On generalization/specialization for conceptual graphs. Journal of Experimental & Theoretical Artificial Intelligence **7**(3) (1995) 325–344
18. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. Technical report, W3C (2007)