Un langage d'interrogation à la SPARQL pour les graphes conceptuels

Marc Legeay*, David Genest*, Stéphane Loiseau*

*LERIA - Université d'Angers
2, boulevard Lavoisier 49045 Angers Cedex 01
{prénom.nom}@univ-angers.fr

Résumé. Cet article propose un langage générique d'interrogation pour le modèle des graphes conceptuels. D'abord, nous introduisons les graphes d'interrogation. Un graphe d'interrogation est utilisé pour exprimer un « ou » entre deux sous-graphes, ainsi qu'une « option » sur un sous-graphe optionnel. Ensuite, nous proposons quatre types de requêtes (interrogation, sélection, description et construction) en utilisant les graphes d'interrogation. Enfin, les réponses à ces requêtes sont calculées à partir d'une opération basée sur l'homomorphisme de graphe.

1 Introduction

Le modèle des graphes conceptuels (Sowa, 1984; Chein et Mugnier, 2009) permet de représenter des connaissances sous la forme d'un graphe étiqueté. Le modèle des graphes conceptuels utilise une représentation graphique visuelle des connaissances afin de faciliter la compréhension pour les utilisateurs. La méthode d'interrogation du modèle est basée sur l'opération principale des graphes conceptuels, un homomorphisme de graphes appelé la *projection* : cette opération permet de déterminer si les connaissances exprimées dans un graphe conceptuel appelé *graphe requête* peuvent être déduites de celles exprimées dans la base de connaissances, représentée par un graphe conceptuel appelé *graphe fait*.

Les objectifs de ce modèle sont proches d'une partie de ceux des langages du *Web sémantique* tels que RDF, RDF-Schema ou OWL (Manola et al., 2004; Brickley et Guha, 2004; McGuinness et al., 2004) qui sont généralement interrogés en utilisant *SPARQL* (Garlik et al., 2013), une recommandation officielle du W3C disponible dans plusieurs outils. SPARQL offre plus de flexibilité par rapport aux graphes conceptuels dans l'interrogation d'une base de connaissances. D'une part, SPARQL permet d'exprimer une disjonction entre plusieurs parties d'une requête (SPARQL utilise le mot « union ») et d'identifier des parties comme obligatoires ou optionnelles. D'autre part, SPARQL permet d'interroger la base grâce à quatre *types de requêtes* : l'interrogation, la sélection, la description et la construction. La requête d'interrogation permet de savoir si la connaissance représentée par la requête est présente dans la base. La requête de sélection permet de trouver et extraire de la base des connaissances identifiées dans la requête comme importantes. La requête de description permet d'obtenir des informations sur des connaissances de la requête. La requête de construction permet de déduire de nouvelles connaissances à partir de celles contenues dans la base.

Cet article propose de combiner la simplicité de la représentation visuelle des graphes conceptuels avec la puissance du modèle d'interrogation du Web sémantique pour améliorer l'expression des requêtes des graphes conceptuels. La contribution de cet article est triple. D'abord, le *graphe d'interrogation* est introduit. La notion de graphe d'interrogation permet d'exprimer des conditions de disjonction – un « ou » entre deux de ses sous-graphes – et des conditions d'option – un sous-graphe est préféré mais non-nécessaire – dans un graphe conceptuel. Ensuite, le graphe d'interrogation nous permet de définir un langage d'interrogation pour les graphes conceptuels formé de quatre types de requêtes : requête d'interrogation, requête de sélection, requête de description et requête de construction. Une *requête d'interrogation* permet de savoir si le graphe de la requête est déductible du graphe fait. Une *requête de sélection* permet de trouver et extraire du graphe fait des sommets identifiés comme étant importants dans le graphe requête. Une *requête de description* permet d'obtenir des informations complémentaires du graphe fait liées à un sommet particulier du graphe requête. Une *requête de construction* permet de déduire de nouvelles connaissances en utilisant les connaissances extraites du graphe fait. Enfin, l'opération basique de calcul utilisée pour interroger et obtenir des *réponses* est introduite : la *projection d'un graphe d'interrogation* dans un graphe fait. Cette projection est définie en utilisant la projection classique d'un graphe conceptuel dans un graphe fait.

Aucun langage générique d'interrogation pour les graphes conceptuels n'a encore été proposé, mais différentes idées ont été mises en avant. De même que la requête de sélection, la possibilité d'identifier certains sommets dans un graphe requête pour facilement exploiter le résultat d'une projection a déjà été proposé dans Sowa (1984). Notre approche est différente de Sowa (1984) dans lequel les sommets sont simplement marqués puisque nous proposons de les nommer pour pouvoir facilement les identifier dans le résultat de la projection. Dans la communauté de SPARQL, Corby et Faron-Zucker (2007) propose une implémentation de la recherche de motifs de graphes de SPARQL en utilisant la projection classique des graphes conceptuels. Notre approche, à l'inverse, est de transposer les idées de SPARQL dans le modèle des graphes conceptuels.

L'article est organisé comme suit. La section 2 rappelle les bases du modèle des graphes conceptuels et de la projection. La section 3 présente le modèle des graphes d'interrogation. La section 4 présente la projection d'un graphe d'interrogation dans un graphe conceptuel fait ainsi que la requête d'interrogation, La section 5 présente la requête de sélection, La section 6 présente la requête de description, La section 7 présente la requête de construction. Enfin, la section 8 présente notre implémentation et donne quelques éléments pour comparer notre langage d'interrogation des graphes conceptuels avec SPARQL.

2 Modèle des graphes conceptuels

Le modèle des graphes conceptuels (Chein et Mugnier, 2009) représente les connaissances par un graphe défini sur un vocabulaire. Le *vocabulaire* définit des ensembles ordonnés de types de concepts et de types de relations. Chaque ensemble est ordonné par une relation de spécialisation. Un concept peut être spécifié par un *individu*, s'il ne l'est pas alors il est appelé *générique*. Chaque type de relations possède une *signature* qui renseigne l'arité et le type de concepts de chaque argument. Le vocabulaire rassemble l'ensemble des types de concepts,

l'ensemble des types de relations et leur signature, et l'ensemble des individus. Le vocabulaire est utilisé pour étiqueter les sommets du graphe.

Définition 1. Un *vocabulaire* est un quadruplet (T_C, T_R, I, σ), où :
- T_C, l'ensemble des types de concepts, est partiellement ordonné par une relation $\leq$ et possède un plus grand élément.
- T_R, l'ensemble des types de relations, est partiellement ordonné par une relation $\leq$ et est partitionné en sous-ensembles $T_R^1, \ldots, T_R^k$ de types de relations respectivement d'arité $1, \ldots, k$. Deux types de relations avec une arité différente sont incomparables.
- T_C et T_R sont deux ensembles finis disjoints.
- I est l'ensemble des marqueurs individuels, qui est disjoint de T_C et T_R. $*$ est appelé le marqueur générique, $I \cup \{*\}$ représente l'ensemble des marqueurs et est ordonné comme suit : $*$ est plus grand que tous les éléments de I et les éléments de I sont incomparables deux à deux.
- σ est une application qui associe à chaque type de relations $r \in T_R^j, 1 \leq j \leq k$ une *signature* $\sigma(r) \in (T_C)^j$.

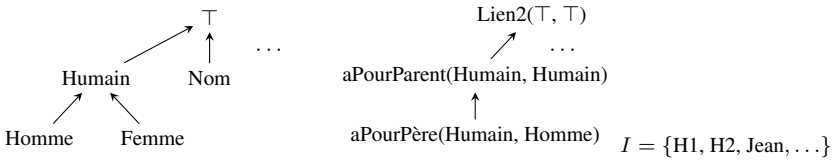

FIG. 1 – *Extrait d'un vocabulaire (T_C, T_R, I, σ) avec T_C partiellement représenté par l'arbre de gauche, T_R partiellement représenté par l'arbre de droite, et la signature σ de chaque relation est précisée à coté de chaque type de relation.*

Exemple. La figure 1 montre un extrait du vocabulaire utilisé dans les exemples suivants.

Un graphe conceptuel est un multigraphe biparti défini sur un vocabulaire. Un des ensembles de sommets est appelé l'ensemble des sommets *concepts*, et l'autre ensemble est appelé l'ensemble des sommets *relations*, représentant les liens entre les concepts. Chaque sommet est étiqueté. Un sommet relation est étiqueté par un type de relations et un sommet concept est étiqueté par un couple formé d'un type de concepts et d'un marqueur. Si un concept c est le i-ème argument d'une relation r, alors il y a une arête entre c et r étiquetée par i, et le type de concepts de c doit respecter les contraintes de la signature de r. Un sommet concept individuel est référencé par un *marqueur individuel* de $\mathcal{I}$. Un sommet concept générique est référencé par le *marqueur générique* $*$. Nous considérons les graphes sous forme normale : deux sommets concepts différents ne peuvent pas être étiquetés par le même marqueur individuel.

Définition 2. Soit un vocabulaire $\mathcal{V} = (T_C, T_R, I, \sigma)$.

Un *graphe conceptuel* défini sur $\mathcal{V}$ est un quadruplet $G = (C, R, E, l)$ qui satisfait les conditions suivantes :
- (C, R, E) est un multigraphe fini, non-orienté et biparti. C est l'ensemble des sommets concepts, R est l'ensemble des sommets relations, E est l'ensemble des arêtes [1].

1. Une arête entre un sommet concept c et un sommet relation r est notée rc.

– l est la fonction d'étiquetage des sommets et des arêtes de (C, R, E) qui satisfait :
  1. Un sommet concept c est étiqueté par une paire $l(c) = (type(c), marqueur(c))$, où $type(c) \in T_C$ et $marqueur(c) \in I \cup \{*\}$,
  2. Deux sommets concepts différents ne peuvent pas être étiquetés par le même individu,
  3. Un sommet relation r est étiqueté par $l(r) \in T_R$,
  4. Le degré d'un sommet relation r est égal à l'arité de $l(r)$,
  5. Les arêtes incidentes à un sommet relation r sont totalement ordonnées et étiquetées de 1 à l'arité de $l(r)$.

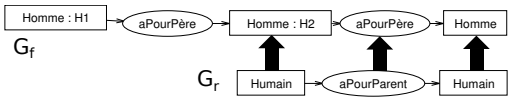

FIG. 2 – *Projection de G_r dans G_f.*

La *projection* est l'opération d'interrogation du modèle des graphes conceptuels. Soient un graphe requête G_r et un graphe fait G_f définis sur le même vocabulaire, G_r se projette dans G_f si les informations représentées par G_r se déduisent de celles représentées par G_f.

Définition 3. Soient $G_r = (C_r, R_r, E_r, l_r)$ et $G_f = (C_f, R_f, E_f, l_f)$ deux graphes conceptuels définis sur le même vocabulaire.

Une *projection* π de G_r dans G_f est une application de $C_r \cup R_r$ vers $C_f \cup R_f$ telle que :
– $\forall c \in C_r, \forall r \in R_r$, si $rc \in E_r$ alors $\pi(r)\pi(c) \in E_f$ et $l_r(rc) = l_f(\pi(r)\pi(c))$,
– $\forall s \in C_r \cup R_r, l_f(\pi(s)) \leq l_r(s)$.

Les sommets de G_f qui correspondent aux sommets de G_r sont appelés les *images* des sommets de G_r par π.

Exemple. Deux graphes conceptuels G_r et G_f sont représentés dans la figure 2. Par souci de lisibilité des graphes présentés ici, nous employons la simplification visuelle des relations binaires : les arêtes étiquetées 1 (resp. 2) sont représentées par un arc du concept (resp. de la relation) vers la relation (resp. le concept). Une projection de G_r dans G_f est représentée via les flèches en gras.

3 Graphes d'interrogation

Alors que dans le modèle classique des graphes conceptuels le graphe requête est un graphe conceptuel ordinaire comme le graphe fait, cet article propose de construire des requêtes plus complexes en s'appuyant sur un modèle de requête plus riche qu'un graphe conceptuel ordinaire, ce modèle est basé sur les graphes d'interrogation. Un graphe d'interrogation permet de préciser des conditions de disjonction ou d'option sur un graphe conceptuel. Il est construit en une structuration hiérarchique de *blocs* qui décrivent un graphe conceptuel : un bloc contient une partie du graphe conceptuel et sert à décrire une disjonction, une option ou simplement une partie standard du graphe conceptuel.

Un bloc est défini comme un ensemble de sommets du graphe associé à un type de bloc. Tous les sommets concepts de l'ensemble doivent avoir leurs sommets relations voisins dans l'ensemble : ceci permet à un sommet concept du bloc d'être caractérisé par les relations auxquelles il est lié. Différents *types de blocs* permettent d'identifier la condition d'un bloc : l'`Option` (le bloc est dit *optionnel*), la `Disjonction` (le bloc est dit *de disjonction*) et le `Standard` (le bloc est dit *standard*).

Un bloc optionnel permet d'exprimer que la partie du graphe qu'il contient est facultative : on recherche dans le graphe fait un graphe avec la partie optionnelle, mais si elle n'est pas trouvée, un graphe privé de la partie optionnelle conviendra. Un bloc de disjonction est composé de blocs fils et exprime une disjonction entre ses blocs fils : on recherche dans le graphe fait un graphe avec seulement un des fils.

Définition 4. Soit un graphe conceptuel $G = (C, R, E, l)$.

Un *bloc du graphe* G est un couple $b = (S, T)$ où

– $S \subseteq C \cup R$ est l'*ensemble des sommets* du bloc tel que :

$$\forall c \in S \cap C, \forall r \in R, \text{ si } rc \in E \text{ alors } r \in S$$

– $T \in \{\texttt{Standard}, \texttt{Option}, \texttt{Disjonction}\}$ est le *type* du bloc.

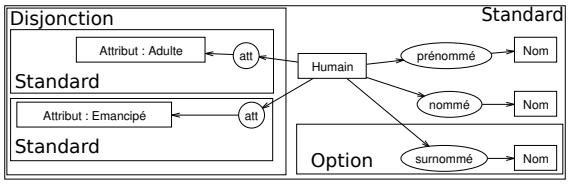

FIG. 3 – *Graphe d'interrogation.*

Une arborescence est utilisée afin de structurer les blocs d'un graphe requête. Les blocs sont les sommets de l'arborescence. Les notions de *bloc racine*, *bloc père* et *blocs fils* sont définies grâce au vocabulaire lié aux arborescences. La structuration est hiérarchique, elle impose que la relation père-fils entre blocs vérifie l'inclusion des sommets du bloc fils dans ceux du bloc père et que si deux blocs ont un sommet en commun, alors un de ces blocs est un ancêtre de l'autre.

Définition 5. Soit un graphe conceptuel $G = (C, R, E, l)$.

Une *structuration* $\mathcal{H}$ de G est une arborescence (B, A, r) où la racine r est le bloc $(C \cup R, \texttt{Standard})$, B étant l'ensemble des blocs de G qui forme l'ensemble des sommets de l'arborescence, et A l'ensemble des arcs. La structuration est telle que :

– Soit $b = (S_b, T_b) \in B \setminus \{r\}, pere(b) = (S_p, T_p)$ alors $S_b \subseteq S_p$,
– Soient $b = (S_b, T_b), b' = (S_{b'}, T_{b'}) \in B$ si $S_b \cap S_{b'} \neq \emptyset$ alors $S_b \subseteq S_{b'}$ ou $S_{b'} \subseteq S_b$,
– Soit $b = (S_b, \texttt{Disjonction})$ alors $|fils(b)| \geq 2$ et $S_b = \bigcup_{(S_f, T_f) \in fils(b)} S_f$.

Définition 6. Un *graphe d'interrogation* $G_i = (G, \mathcal{H})$ est un couple où G est un graphe conceptuel et $\mathcal{H}$ est une structuration de G.

Exemple. La figure 3 représente un graphe d'interrogation qui possède cinq blocs de graphe représentés par des cadres étiquetés.

4 Projection et requête d'interrogation

Les graphes d'interrogation sont utilisés comme base de construction pour chacun des quatre types de requêtes.

Un graphe d'interrogation peut être développé en un ensemble de graphes conceptuels en suivant les conditions d'option et de disjonction. Cet ensemble de graphes conceptuels, appelé l'*ensemble des graphes développés* du graphe d'interrogation, représente l'ensemble des graphes conceptuels dont on cherche à savoir s'ils sont déductibles du graphe fait. Cet ensemble permet de définir une projection d'un graphe d'interrogation dans un graphe fait comme étant une projection d'un graphe développé dans le graphe fait. Pour obtenir un graphe développé à partir d'un graphe d'interrogation, des opérations sont appliquées sur le graphe d'interrogation pour aboutir à un graphe ne contenant que des blocs standards. Les blocs optionnels seront retirés ou leur type sera changé en `Standard`. Le type des blocs de disjonction sera changé en `Standard` et seulement un et un seul de ses fils sera conservé.

Définition 7. Soit $G_i = (G, \mathcal{H})$, un graphe d'interrogation avec $\mathcal{H} = (B, A, r)$.

Soit $b \in B$, un bloc optionnel.

Le graphe G_i dans lequel le type de b est changé en `Standard` est un *développement* de G_i.

Le graphe G_i dans lequel b est retiré est un *développement* de G_i.

Soit $b \in B$, un bloc de disjonction.

Le graphe G_i dans lequel le type de b est changé en `Standard` et dans lequel tous les fils de b sauf un sont retirés est un *développement* de G_i.

Définition 8. Soit $G_i = (G, \mathcal{H})$, un graphe d'interrogation.

L'*ensemble des graphes développés* GD de G_i est l'ensemble de tous les graphes conceptuels associés aux graphes d'interrogation ne contenant que des blocs standards obtenus grâce à une suite d'applications de l'opération de développement à partir de G_i.

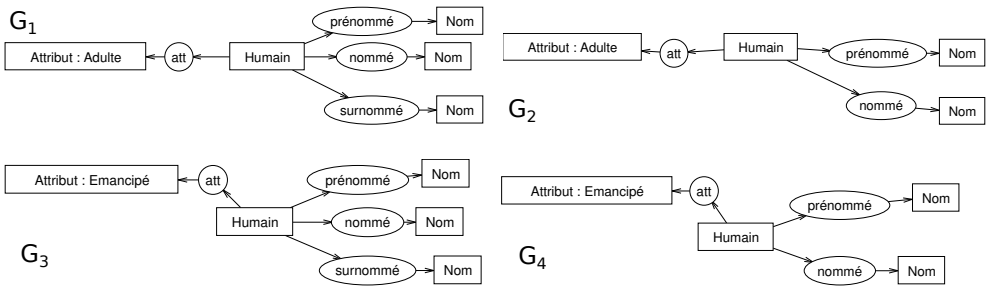

FIG. 4 – *Graphes développés.*

Exemple. Les graphes conceptuels de la figure 4 forment l'ensemble des graphes développés $GD = \{G_1, G_2, G_3, G_4\}$ du graphe d'interrogation G_i (figure 3).

Pour définir la projection d'un graphe d'interrogation dans un graphe fait, on construit les projections des graphes développés dans le graphe fait. La projection d'un graphe développé est généralement une projection du graphe d'interrogation. L'exception est due à la sémantique

de l'option, en effet, la partie optionnelle d'un graphe doit être présente dans la réponse si elle est présente dans le graphe fait : dans ce cas les projections qui ne contiendraient pas le bloc optionnel ne sont pas des projections du graphe d'interrogation. Notons que ces dernières projections sont prolongées [2] par la projection contenant les informations optionnelles. On définit donc l'ensemble des projections d'un graphe d'interrogation dans un graphe conceptuel comme l'ensemble des projections des graphes développés du graphe d'interrogation privé des projections prolongées.

Définition 9. Soient un graphe d'interrogation $G_i = (G, \mathcal{H})$, un graphe conceptuel G_f défini sur le même vocabulaire $\mathcal{V}$ que G, et l'ensemble des projections Π_{GD} des graphes développés GD de G_i dans G_f.

L'ensemble des *projections du graphe d'interrogation G_i dans G_f* est :

$$\Pi = \{\pi \in \Pi_{GD} | \nexists \pi' \in \Pi_{GD}, \pi \subset \pi'\}$$

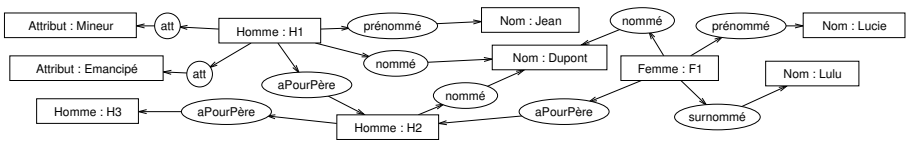

FIG. 5 – *Graphe conceptuel G_f.*

Une *requête d'interrogation* permet de vérifier si une information est déductible de la base de connaissances. Ce type de requête correspond à l'interrogation « classique » du modèle des graphes conceptuels en l'enrichissant ici de disjonctions et d'options. Une requête d'interrogation n'est donc composée que d'un graphe d'interrogation. Une réponse à une requête d'interrogation est simplement une projection du graphe d'interrogation de la requête dans le graphe fait.

Définition 10. Soit un graphe conceptuel G_f défini sur un vocabulaire $\mathcal{V}$.

Une *requête d'interrogation* est un graphe d'interrogation $G_i = (G, \mathcal{H})$, G étant défini sur $\mathcal{V}$.

Une *réponse à une requête d'interrogation G_i* est une projection π de G_i dans G_f.

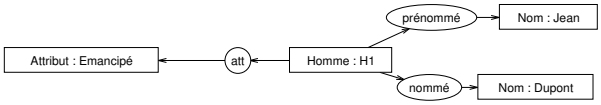

FIG. 6 – *Image d'une projection de G_i dans G_f.*

Exemple. Le graphe d'interrogation G_i de la figure 3 peut être utilisé comme requête d'interrogation. Il n'existe qu'une projection de G_i dans G_f (figure 5). La figure 6 montre l'image de la réponse à la requête d'interrogation.

2. Une fonction $f : D \to I$ prolonge une fonction $f' : D' \to I'$, noté $f' \subset f$, si $D' \subset D$, $I' \subseteq I$ et $\forall d \in D', f(d) = f'(d)$.

5 Requête de sélection

Dans une requête d'interrogation, l'image de certains sommets peut être plus importante que l'image des autres sommets de la requête. Cet article propose un type de requête dans lequel on peut identifier les sommets jugés importants et dont on veut retenir les images par la projection du graphe d'interrogation : la requête de sélection.

Une requête de sélection est donc un couple composé d'un graphe d'interrogation et d'une fonction de nommage qui associe à des noms à sélectionner, des sommets du graphe.

Définition 11. Soit un ensemble de noms N.

Une *requête de sélection* est un couple $R_s = [G_i, select]$, où $G_i = (G, \mathcal{H})$ est un graphe d'interrogation avec $G = (C, R, E, l)$, et où $select$ est une fonction de nommage définie comme suit : $select : N \to C \cup R$.

Une réponse à une requête de sélection est une fonction qui associe à chaque nom de l'ensemble des noms à sélectionner de la requête, un sommet du graphe fait. Si le graphe d'interrogation se projette dans le graphe fait, les noms à sélectionner seront associés avec l'image dans le graphe fait, si elle existe, du sommet du graphe d'interrogation auxquels ils sont liés. Un sommet peut ne pas avoir d'image par la projection, dans ce cas, le nom n'est associé à aucun sommet.

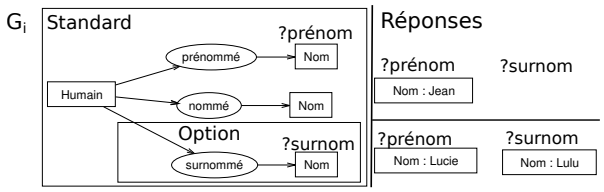

FIG. 7 – *Requête de sélection et ses deux réponses.*

Définition 12. Soient un graphe conceptuel $G_f = (C_f, R_f, E_f, l_f)$, une requête de sélection $R_s = [G_i, select]$ avec $select$ définie sur l'ensemble de noms N, et une projection π de G_i dans G_f.

Une *réponse à la requête de sélection* R_s est une fonction partielle de N dans $C_f \cup R_f$ associant à chaque élément n de N le sommet $\pi(select(n))$ de G_f s'il existe.

Exemple. La figure 7 montre une requête de sélection $R_s = [G_i, select]$, où la fonction de nommage $select$, définie sur $N = \{\texttt{?prénom}, \texttt{?surnom}\}$, est représentée sur G_i. La requête permet de connaître le prénom, et s'il existe le surnom, de tous les humains du graphe fait. Les deux réponses à l'exécution de R_s dans G_f (figure 5) sont présentées en partie droite de la figure 7.

6 Requête de description

La requête de sélection permet d'obtenir les images des sommets qui intéressent l'utilisateur, la requête de description permet, elle, d'obtenir non seulement les images des sommets

qui intéressent l'utilisateur, mais aussi leur description. La description d'un sommet est formée par les sommets voisins qui apportent des informations sur ce sommet. Une requête de description est un couple composé d'un graphe d'interrogation et d'une fonction de nommage qui à un nom de l'ensemble des noms à décrire, associe un sommet du graphe. Étant donné que seuls les concepts peuvent être décrits, l'ensemble à décrire ne doit contenir que des noms se référant à des sommets concepts.

Définition 13. Soit un ensemble de noms N.

Une *requête de description* est un couple $R_d = [G_i, desc]$, où $G_i = (G, \mathcal{H})$ est un graphe d'interrogation avec $G = (C, R, E, l)$, et où $desc$ est une fonction de nommage définie comme suit : $desc : N \rightarrow C$.

Une réponse à une requête de description est un sous-graphe du graphe fait dans lequel figurent les images des concepts à décrire, les relations qui sont liées à ces concepts par une arête étiquetée par « 1 », et les concepts liés à ces relations. Seules les relations reliées par une arête étiquetée par « 1 » sont sélectionnées puisque se sont les relations dont le concept à décrire est le sujet.

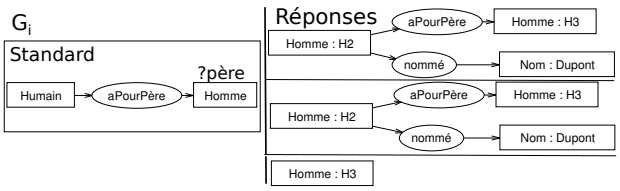

FIG. 8 – *Requête de description et ses trois réponses.*

Définition 14. Soient un graphe conceptuel $G_f = (C_f, R_f, E_f, l_f)$, une requête de description $R_d = [G_i, desc]$ avec $desc$ définie sur l'ensemble de noms N, une projection π de G_i dans G_f, et l'ensemble des sommets S_π du graphe conceptuel de G_i qui ont une image dans G_f par π.

Une *réponse à la requête de description* R_d est un sous-graphe $G_d = (C_d, R_d, E_d, l_d)$ de G_f tel que :

1. $C_0 = \{\pi(desc(d))|d \in N, desc(d) \in S_\pi\}$
2. $R_d = \{r \in R_f|rc \in E_f, c \in C_0, l_f(rc) = 1\}$
3. $C_d = C_0 \cup \{c \in C_f|rc \in E_f, r \in R_d\}$
4. $E_d = \{nn' \in E_f|n, n' \in C_d \cup R_d\}$
5. $\forall n \in C_d \cup R_d \cup E_d, l_d(n) = l_f(n)$

Exemple. La figure 8 montre une requête de description $R_d = [G_i, desc]$, où la fonction de nommage $desc$, définie sur $N = \{\texttt{?père}\}$, est représentée sur G_i. La requête permet de décrire tous les humains qui sont pères. Les trois réponses à l'exécution de R_d dans G_f (figure 5) sont présentées en partie droite de la figure 8 : `H2` et `H3` sont pères, mais `H2` est père à la fois de `H1` et de `F1`.

7 Requête de construction

Une requête de construction est composée de deux graphes : un graphe d'interrogation, appelé *graphe condition*, et un graphe conceptuel, appelé *graphe modèle*,. Le principe d'une requête de construction est d'une part d'extraire des informations du graphe fait à l'aide du graphe condition, qui doit se projeter dans le graphe fait, et d'autre part d'exploiter ces informations afin de construire un nouveau graphe conceptuel, appelé *graphe réponse*, à partir du graphe modèle. Une requête de construction peut être vue comme une règle (Salvat, 1998) dont la partie condition est définie par le graphe condition et la partie conclusion est obtenue par le graphe modèle. Pour lier les sommets du graphe condition à ceux du graphe modèle, on utilise une fonction $lien$ qui associe à un sommet du graphe condition, un sommet du graphe modèle. Un sommet du graphe condition qui est lié à un sommet du graphe modèle doit avoir la même étiquette que ce dernier. Notons qu'en utilisant l'extension des types conjonctifs (Chein et Mugnier, 2004), il serait possible que deux sommets liés ne possèdent pas la même étiquette.

Définition 15. Une *requête de construction* est un triplet $R_c = [G_i, G_m, lien]$, où $G_i = (G, \mathcal{H})$, avec $G = (C, R, E, l)$, est un graphe d'interrogation appelé *graphe condition*, $G_m = (C_m, R_m, E_m, l_m)$ est un graphe conceptuel appelé *graphe modèle*, tels que G et G_m sont définis sur le même vocabulaire, $lien$ est une fonction partielle définie comme suit : $lien$: $C \cup R \rightarrow C_m \cup R_m$, et tout sommet s de G lié au sommet $lien(s) = s'$ de G_m possède la propriété suivante : $l(s) = l_m(s')$.

Une réponse à une requête de construction est un graphe réponse construit sur le modèle du graphe modèle comme suit. Les sommets du graphe modèle liés aux sommets du graphe condition sont étiquetés par les images des sommets du graphe condition dans le graphe fait. Les sommets qui ne sont pas liés dans le graphe modèle gardent leurs étiquettes. Il se peut que des sommets du graphe condition ne possèdent pas d'image par la projection, les sommets liés au graphe modèle sont alors retirés du graphe réponse, ainsi que toutes les relations éventuellement connectées à ces sommets. Ce cas se présente lorsqu'un sommet du graphe modèle est lié à un sommet du graphe condition qui appartient à un bloc option, ou de disjonction.

Définition 16. Soient un graphe conceptuel $G_f = (C_f, R_f, E_f, l_f)$, une requête de construction $R_c = [G_i, G_m, lien]$ avec $G_i = (G, \mathcal{H})$ où $G = (C, R, E, l)$ et avec $G_m = (C_m, R_m, E_m, l_m)$, une projection π de G_i dans G_f, et l'ensemble S_π des sommets de G qui ont une image dans G_f par π.

Une *réponse à la requête de construction* R_c est un graphe conceptuel appelé *graphe réponse* $G_r = (C_r, R_r, E_r, l_r)$, tel que :

- $C_r = C_m \setminus \{c = lien(c') \in C_m | c' \in C, c' \notin S_\pi\}$
- $R_r = (R_m \setminus \{r = lien(r') \in R_m | r' \in R, r' \notin S_\pi\}) \setminus \{r \in R_m | rc \in E_m, c \notin C_r\}$
- $E_r = \{nn' \in E_m | n, n' \in C_r \cup R_r\}$
- l_r est définie de la façon suivante $\forall s \in C_r \cup R_r \cup E_r$:
  si $\exists s' \in C \cup R$ tel que $s = lien(s')$, $l_r(s) = l_f(\pi(s'))$
  sinon, $l_r(s) = l_m(s)$

Exemple. La figure 9 montre la requête $R_c = [G_i, G_m, lien]$ où la fonction $lien$ est représentée directement sur le graphe via les pointillés. Cette requête peut s'apparenter à une règle qui dit que si un humain a pour parent un autre humain, alors ce dernier est un parent du premier, mais

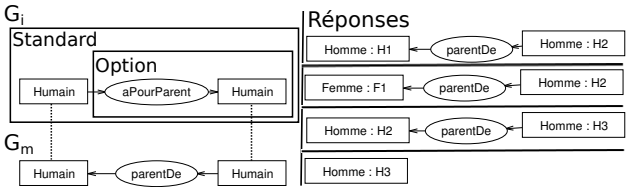

FIG. 9 – *Requête de construction et ses quatre réponses.*

si cet humain n'a pas de parent, personne n'est son parent. Les quatre réponses à l'exécution de R_c dans G_f (figure 5) sont présentées en partie droite de la figure 9.

8 Conclusion

Cet article présente les graphes d'interrogation utilisés pour interroger un graphe conceptuel, représentant une base de connaissances, grâce à la projection d'un graphe d'interrogation dans un graphe conceptuel. Les graphes d'interrogation sont utilisés dans quatre types de requêtes, fournissant ainsi au modèle des graphes conceptuels un langage d'interrogation riche. L'aspect visuel du modèle d'interrogation est un choix important qui a été privilégié pour rester proche des caractéristiques du modèle des graphes conceptuels. Un prototype implémente ce langage d'interrogation en utilisant la bibliothèque Cogitant (Genest, 2014).

Deux extensions au modèle des graphes conceptuels peuvent être considérés dans notre approche. La première extension ajoute les datatypes aux types de concepts (Raimbault et al., 2009). Les datatypes permettent d'étiqueter les concepts avec des valeurs comme `string`, `int` ou `date` et d'utiliser les opérateurs ($<$, $\neg$, $\leq$, ...) pendant la projection. La seconde extension permet à un sommet d'être étiqueté par des types conjonctifs (Chein et Mugnier, 2004). Brièvement, l'adaptation de l'extension des datatypes à notre langage requiert d'associer une expression avec un sommet datatype à l'aide de noms dans les blocs. Les noms seraient obtenus grâce à une fonction de nommage associée directement avec le graphe d'interrogation qui nomme ainsi les sommets datatypes. L'expression permet de filtrer les valeurs que le concept associé peut prendre lors de la projection, grâce aux opérateurs. Brièvement, l'adaptation de l'extension des types conjonctifs à notre langage permet de ne pas obliger que deux sommets liés dans une requête de construction soient étiquetés de la même manière. Ainsi, un sommet d'une réponse à une requête de construction peut être étiqueté par la conjonction des types de concept du graphe modèle et du graphe fait.

Puisque notre langage d'interrogation pour les graphes conceptuels s'inspire de SPARQL, une comparaison de ces deux langages s'impose. Cela requiert d'étudier les différences et similarités entre le modèle des graphes conceptuels et RDF, ainsi que de comparer les requêtes SPARQL avec les requêtes graphes conceptuels. Baget et al. (2010) définit RDFS-, un sous-ensemble de RDF tel que la transformation entre RDFS- et les graphes conceptuels est naturelle. Des travaux futurs s'intéresseront à la définition de SPARQL-, un sous-ensemble de SPARQL pour interroger une base de connaissances RDFS-. Une requête SPARQL- (resp. requête graphes conceptuels) se transformerait en requête graphes conceptuels (resp. requête SPARQL-) et la réponse d'une requête graphes conceptuels (resp. requête SPARQL-) contien-

drait les mêmes connaissances que la réponse à la requête SPARQL- (resp. requête graphes conceptuels). L'interrogation des graphes conceptuels pourrait en conséquence servir d'interface visuelle à SPARQL- et simplifier l'interface des requêtes SPARQL.

Références

Baget, J.-F., M. Croitoru, M. Leclère, et M.-L. Mugnier (2010). Translations between RDF(S) and Conceptual Graphs. In *ICCS'10 : 18th Int. Conf. on Conceptual Structures*, Volume 6208 of *LNAI*, pp. 28–41. Springer.

Brickley, D. et R. V. Guha (2004). RDF Vocabulary Description Language 1.0 : RDF Schema.

Chein, M. et M.-L. Mugnier (2004). Concept types and coreference in simple conceptual graphs. In *ICCS'04 : 12th Int. Conf. on Conceptual Structures*, Volume 3127 of *LNAI*, pp. 303–318. Springer.

Chein, M. et M.-L. Mugnier (2009). *Graph-based knowledge representation : computational foundations of conceptual graphs*. Springer.

Corby, O. et C. Faron-Zucker (2007). Implementation of SPARQL query language based on graph homomorphism. In *ICCS'07 : 15th Int. Conf. on Conceptual Structures*, Volume 4604 of *LNAI*, pp. 472–475. Springer.

Garlik, S. H., A. Seaborne, et E. Prud'hommeaux (2013). SPARQL 1.1 query language. *World Wide Web Consortium*.

Genest, D. (2014). Cogitant 5.3.0 - reference manual. `http://cogitant.sourceforge.net/files/cogitant.pdf`.

Manola, F., E. Miller, et B. McBride (2004). RDF primer. *W3C recommendation*.

McGuinness, D. L., F. Van Harmelen, et al. (2004). OWL web ontology language overview. *W3C recommendation*.

Raimbault, T., D. Genest, et S. Loiseau (2009). Conceptual Graphs and Datatypes. In *ICCS'09 : 17th Int. Conf. on Conceptual Structures*, Volume 5662 of *LNAI*, pp. 270–283. Springer.

Salvat, E. (1998). Theorem Proving Using Graph Operations in the Conceptual Graph Formalism. In *ECAI*, pp. 356–360.

Sowa, J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley.

Summary

This paper proposes a general query language for conceptual graphs. First, we introduce kernel query graphs. A kernel query graph can be used to express an "or" between two subgraphs, as well as an "option" on an optional sub-graph. Second, we propose a way to express four kinds of queries (ask, select, describe and construct) using kernel query graphs. Third, the answers of queries are computed by an operation based on graph homomorphism.