



Aggrégations en graphes conceptuels

Nicolas Moreau

► **To cite this version:**

| Nicolas Moreau. Aggrégations en graphes conceptuels. RR-09021, 2009. <lirmm-00401788>

HAL Id: lirmm-00401788

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00401788>

Submitted on 6 Jul 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.

Aggrégations en graphes conceptuels

Nicolas Moreau <moreau@lirmm.fr>

6 juillet 2009

1 État de l’art

Contrairement au langage SQL, peu de langages d’interrogation de bases de connaissance de graphes offrent la possibilité d’utiliser des opérations d’aggrégation. Concernant les bases de connaissance en RDF, seul RQL [9] offre un tel mécanisme [7]¹. Des travaux [8] visent à ajouter les opérateurs d’aggrégation au langage de requête RDQL [12].

Les graphes conceptuels [5] ne définissent aucun mécanisme d’aggrégation. De plus, ils ne disposent pas de types concrets (*datatypes*) tels que des chaîne de caractères, des entiers, des dates, ce qui exclut l’utilisation des opérateurs d’aggrégation SUM, AVG, MIN et MAX. Les types abstraits ont été ajoutés au formalisme des graphes conceptuels dans [1], mais les problèmes posés par l’opérateur d’aggrégation étudié dans ce papier, COUNT, sont généralisables aux autres opérateurs ; le formalisme sans type concret est donc utilisé.

2 Problème des agrégateurs

Comme le souligne [6], poser la question du nombre d’objets ayant une certaine caractéristique (par exemple le nombre de voitures que possède Joe), ne trouve pas de réponse facile. Contrairement aux bases de données classiques, les bases de connaissances de graphes font l’hypothèse du monde ouvert et permettent de modéliser des individus non définis (blank nodes en RDF, concepts génériques en GC). Il en résulte que deux sommets différents peuvent potentiellement représenter le même individu du monde réel décrit par le graphe.

3 Framework

Nous définissons un framework basé sur une relation entre concepts de la base de connaissance, ainsi qu’une liste de contraintes que cette relation doit respecter. Une approche “boite noire” similaire est utilisée dans le mécanisme de résolution d’entités SWOOSH [3] qui a pour but de fusionner des entités qui correspondent à des tuples d’une base de données.

¹Les langages dont les mécanismes d’aggrégation portent sur la structure du graphe, comme par exemple sur la taille du chemin entre deux sommets, sont exclus

Notre relation de comparaison associe à un sous-ensemble de concepts une valeur booléenne. Cette valeur est positive si tous les concepts désignent la même entité, et négative s'ils ne représentent pas la même entité (ce qui ne veut pas dire qu'ils représentent tous des entités différentes, par exemple un sous-ensemble de concepts pourraient représenter la même entité).

Définition 1 (Relation de comparaison). *La relation de comparaison κ d'un graphe G associe à un sous ensemble S de concepts de G à une valeur booléenne, 1 si tous les sommets de S représentent la même entité, et 0 autrement. La relation doit respecter les propriétés suivantes :*

- *Reflexivité* : $\kappa(\{c\}) = 1$ pour tout $c \in C_G$
- *Décomposition* : si $\kappa(S) = 1$, alors pour tout $S' \subseteq S$ on a $\kappa(S') = 1$
- *Composition* : si $\kappa(S) = 1$, $\kappa(S') = 1$ et $S \cap S' \neq \emptyset$ alors $\kappa(S \cup S') = 1$
- *Transitivité* : si $\kappa(S) = 0$ alors pour tout $S' \supseteq S$ on a $\kappa(S') = 0$

Il faut noter que la relation κ permet d'exprimer qu'un ensemble d'éléments ne représentent pas la même entité, ce que ne permet généralement pas les modèles de résolution d'entité (par exemple [3] et [11]). Cela vient du fait que la relation qui lie des entités (ou des sommets du graphe) est souvent binaire, ce qui rend les sémantiques "ne représentent pas la même entité" et "représentent des entités toutes différentes" équivalentes. Dans ce modèle, on peut exprimer les deux sémantiques (la deuxième étant exprimée par un ensemble de relations binaires reliant tous les sommets deux à deux).

Des propriétés supplémentaires fondamentales sont ajoutées, basées sur le formalisme GC utilisé (incluant les types conjonctifs et bannis [4]) :

- Si c et c' partagent le même marqueur individuel, alors $\kappa(\{c, c'\}) = 1$
- Si c et c' ont deux marqueurs individuels différents, alors $\kappa(\{c, c'\}) = 0$
- Si la conjonction des types d'un sous-ensemble S de concepts est un type banni, alors $\kappa(S) = 0$

Les nouvelles contraintes définies dans κ peuvent amener des incohérences, celles ci se manifestent lorsque un sous-ensemble S de concepts ne représentent pas la même entité, et qu'un sous-ensemble de concepts contenant S représentent la même entité.

Définition 2 (Graphe valide). *Soit G un SG, G est valide s'il respecte la propriété :*

- *Intégrité* : Il n'existe pas de sous-ensemble de concepts S et S' tels que $S' \subseteq S$, $\kappa(S') = 0$ et $\kappa(S) = 1$

Le fait que des concepts différents représentent la même entité du monde réel n'est pas utilisé par la projection classique, ce qui peut amener des pertes de réponses. Une solution est de définir une forme κ -normale sur laquelle seront effectuées les projections. Cette forme correspond à la fusion des classes de concepts représentant une même entité.

Définition 3 (Forme κ -normale). *Soit G un SG, G est sous forme κ -normale s'il respecte la propriété :*

- *Unicité* : Il n'y a pas de sous-ensemble de concepts S tel que $|S| > 1$ et $\kappa(S) = 1$.

Pour mettre un graphe valide sous forme κ -normale, il suffit de fusionner les sous-ensembles de concepts S tels que $\kappa(S) = 1$ en un seul concept, dont le type est la conjonction des types des concepts de S , et dont le marqueur est générique si tous les concepts de S sont génériques, ou le seul marqueur

individuel partagé par les concepts de S (si plus d'un marqueur individuel était partagé, le graphe ne serait pas valide). Un SG valide peut néanmoins violer d'autres contraintes suivant le formalisme utilisé, lorsqu'il est mis sous forme κ -normale. Il pourrait par exemple violer des contraintes négatives.

3.1 Définir κ

La relation de comparaison κ définit un framework, qui doit être rempli par des règles permettant de dire si des concepts représentent la même entité ou non. La partie précédente présente certaines règles fondamentales qui devraient être appliquées quelques soient les autres règles choisies, car elles se basent sur les propriétés des marqueurs individuels, et des types bannis. Il est également possible d'utiliser des extensions du formalisme pour enrichir la relation κ , notamment en provenance d'OWL-DL et de SWRL [11].

Contraintes négatives

Les contraintes négatives sont des SG représentant des connaissances interdites [2]. Un SG dans lequel se projette une contrainte négative est dit inconsistant. Soient un SG G consistant par rapport à un jeu de contraintes négatives C^- , et un sous-ensemble S de concepts tel qu'il n'existe pas $S' \subseteq S$ avec $\kappa(S') = 0^2$, si le graphe G' résultant de la fusion des concepts de S en un seul concept n'est pas consistant par rapport à C^- , alors on en déduit que $\kappa(S) = 0$.

Relations (inversement) fonctionnelles

Un type de relation binaire est fonctionnel lorsque son premier paramètre n'admet qu'une unique entité comme deuxième paramètre. Par exemple la relation fonctionnelle *localisation(oeuvre, musee)* indique qu'une œuvre ne se trouve que dans un seul musée. Un type de relation est inversement fonctionnelle si son deuxième paramètre n'admet qu'une unique entité comme premier paramètre, comme par exemple la relation *pere_de(humain, humain)* (si deux concepts de types humains sont les pères d'un même concept, alors on en déduit que les deux concepts désignent la même entité). Une relation peut être à la fois fonctionnelle et inversement fonctionnelle, comme par exemple la relation *carte_identite(carte, humain)*.

Une relation fonctionnelle (resp. inversement fonctionnelle) r permet de déduire que l'ensemble C de concepts qui sont deuxièmes (resp. premiers) arguments par les relations r reliant un même concept en premier (reps. deuxième) argument désignent une même entité ($\kappa(C) = 1$).

Cardinalité minimale

On définit une cardinalité minimale par un triplet (C, P, m) définissant que le type de relation P est lié un concept de type C à au moins m concepts. Elle permet d'exprimer une connaissance comme "un véhicule possède au moins 2 roues".

²Car la propriété de transitivité nous dirait directement que $\kappa(S) = 0$.

Soit le concept c de type C qui est relié par des relations de type P à un ensemble de concepts V (de taille v). Si $v < m$, le graphe n'est pas valide. Sinon, il est impossible de fusionner un sous-ensemble de taille $(v - m + 2)$ de concepts de V . C'est à dire que pour chaque $U \subseteq V$ tel que $|U| \geq (v - m + 2)$ on a $\kappa(U) = 0$. Par exemple, si $v = m$, tous les concepts sont disjoints deux à deux (on ne peut pas fusionner 2 concepts). Si $v = 4$ et $m = 3$, deux concepts peuvent être fusionnés, mais pas trois, car sinon il n'y aurait que deux concepts associés à c par des relations de types P .

La contrainte *MaxCardinality* ne peut pas être exprimée dans notre modèle, car par exemple si la contrainte est $(C, P, 3)$ et qu'il y a quatre concepts liés à un concept de type C par des relations de types P , on peut en déduire qu'au moins deux concepts doivent être fusionnés, mais pas lesquels.

4 Compter le nombre de réponses

Pour compter le nombre de réponses, il faut pouvoir dire si une réponse, qui est un sous-graphe de la base de connaissance, représente une connaissance (au sens d'une portion du monde réel décrit) différente ou égale d'une autre réponse. Il faut donc étendre κ aux réponses.

Définition 4 (Ensemble de réponses). *L'ensemble de réponse d'une requête Q sur une base B , noté $Q(B)$ est l'ensemble des images engendrées par les projections de Q dans B . $Q(B) = \{\pi(Q) \mid \pi \in \Pi(Q, B)\}$.*

Pour ce faire, on introduit arbitrairement un ordre total sur les concepts de la requête. On en déduit un ordre sur les concepts des réponses, qui est celui de leur antécédent dans la requête par la projection qui a engendré la réponse (un concept peut avoir plusieurs rang, si plusieurs concepts de la requête se projette sur ce même concept). La comparaison des réponses se fait alors sur le plan sémantique (sont-elles équivalentes?) et sur le plan du monde réel (les concepts de même rang représentent-ils la même entité?).

Définition 5 (Comparaison de réponses). *Soient A, \dots, A^n un sous-ensemble de réponses de $Q(B)$,*

- $\kappa(A, \dots, A^n) = 1$, si $A \equiv \dots \equiv A^n$ et si pour tout rang i on a $\kappa(\{a_i, \dots, a_i^n\}) = 1$ ³
- $\kappa(A, \dots, A^n) = 0$ s'il existe A^i et A^j telles que $A^i \not\equiv A^j$ ou s'il existe un rang i tel que $\kappa(\{a_i, \dots, a_i^n\}) = 0$.

Comme nous supposons que la base de connaissance et la requête sont des SG valides et sous forme κ -normale, il ne peut pas y avoir de réponse A_i et A_j telles que $\kappa(A_i, A_j) = 1$. Tout sous-ensemble de réponse peut soit de manière certaine ne pas désigner une même portion du monde réel, soit avoir un statut inconnu. On sait que le nombre maximum de réponse est borné par le nombre de projections. Pour calculer le nombre minimal de réponses, on s'appuie sur l'hypergraphe de comparaison des réponses :

Définition 6 (Hypergraphe de comparaison). *L'hypergraphe de comparaison $G = (V, E)$ de $Q(B)$ est tel que V est l'ensemble de réponses $Q(B)$ et E représente l'ensemble des arêtes reliant les réponses A_i, \dots, A_j telles que $\kappa(A_i, \dots, A_j) = 0$.*

³Où a_j^i représente le concept de rang j de la réponse A^i .

L'hypergraphe de la figure 1(a) montre un graphe de comparaison de l'ensemble de réponses $Q(B) = \{A_1, \dots, A_5\}$ tel que les sous-ensembles de réponses $\{A_1, A_4\}$, $\{A_2, A_3\}$, $\{A_2, A_4\}$, $\{A_2, A_5\}$, $\{A_3, A_4\}$ et $\{A_1, A_3, A_5\}$ sont tels que les réponses de ces sous-ensemble ne représentent pas une même portion du monde réel.

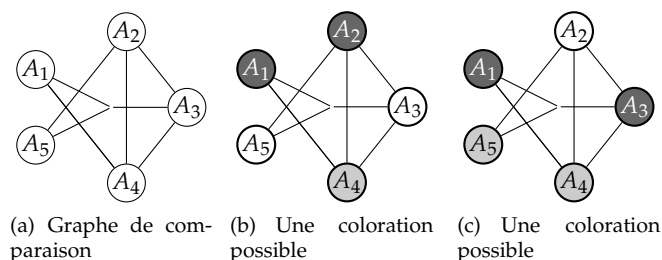


FIG. 1 – Graphe de comparaison, ainsi que deux solutions.

Pour calculer le nombre de résultat minimum, il faut voir qu'elles réponses peuvent être fusionnées ou non entre elles. Cela revient à chercher une coloration des sommet de l'hypergraphe de telle façon qu'il n'existe un ensemble de sommets de la même couleur relié par une multi-arête. Ceci correspond au problème de k -coloration faible (*weak k -coloration*) d'un hypergraphe. Le nombre minimal de réponses à la requête est donc le nombre chromatique $\chi(G)$ dans le problème de k -coloration faible (*weak k -coloration*) d'un multigraphe, qui est NP-complète pour $k \geq 2$ [10].

Dans notre exemple, $\chi(G) = 3$, comme le montre la coloration de la figure 1(b) ou 1(c).

5 Conclusion

L'hypothèse du monde ouvert ainsi que la possibilité de représenter des entités non définies (concepts génériques) rendent la formalisation de fonctions d'agrégation complexe. La première difficulté consiste à définir une fonction de comparaison de concepts, ce qui suppose d'enrichir le formalisme des graphes conceptuels. Même le cas le plus simple, celui de la définition du nombre de résultats à une requête (opérateur COUNT), comporte des incertitudes, traduites par un intervalle de valeurs possibles. La difficulté augmente encore lorsqu'il s'agit de renvoyer des résultats dans le même formalisme (réponses sous forme de graphe), puisqu'il y a plusieurs manières de grouper les réponses, pour un nombre de réponses donné (comme le montre les deux colorations des figures 1(b) ou 1(c)). Dans un formalisme GC plus évolué comprenant des types concrets et des requêtes de type "SELECT x WHERE y" (comme SPARQL), le problème étudié ici se répercuterait de manière plus profonde, puisque une somme ou une moyenne dépend fortement des regroupements de réponses possible (par exemple pour le calcul d'une somme, s'il y a deux réponses et qu'elles peuvent être regroupées, le résultat ne sera pas le même si l'on décide ou non de regrouper).

Références

- [1] J. Baget. A Datatype Extension for Simple Conceptual Graphs and Conceptual Graphs Rules. *Lecture Notes in Computer Science*, 4604 :83, 2007.
- [2] J.F. Baget and M.L. Mugnier. Extensions of Simple Conceptual Graphs : the Complexity of Rules and Constraints. *JAIR*, 16(12) :425–465, 2002.
- [3] O. Benjelloun, H. Garcia-Molina, Q. Su, and J. Widom. Swoosh : A generic approach to entity resolution. *VLDB Journal*, 2008.
- [4] M. Chein and M.L. Mugnier. Concept types and coreference in simple conceptual graphs. *Lecture notes in computer science*, pages 303–318, 2004.
- [5] Michel Chein and Marie-Laure Mugnier. *Graph-based Knowledge Representation : Computational Foundations of Conceptual Graphs*. Springer Publishing Company, Incorporated, 2008.
- [6] R. Fikes, P. Hayes, and I. Horrocks. OWL-QL, a language for deductive query answering on the Semantic Web. *Web Semantics : Science, Services and Agents on the World Wide Web*, 2(1) :19–29, 2004.
- [7] Peter Haase, Jeen Broekstra, Andreas Eberhart, and Raphael Volz. A comparison of RDF query languages. In *Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, 2004.*, NOV 2004.
- [8] E. Hung, Y. Deng, and VS Subrahmanian. RDF aggregate queries and views. *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 717–728, 2005.
- [9] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL : a declarative query language for RDF. In *WWW '02 : Proceedings of the 11th international conference on World Wide Web*, pages 592–603, New York, NY, USA, 2002. ACM.
- [10] L. Lovász. Coverings and colorings of hypergraphs. *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing*, pages 3–12, 1973.
- [11] Fatiha Sais, Nathalie Pernelle, and Marie-Christine Rousset. L2R : a Logical method for Reference Reconciliation . In *Twenty-second AAAI Conference on Artificial Intelligence (AAAI)*, pages 329–334, July 2007.
- [12] A. Seaborne. RDQL-A Query Language for RDF. *W3C Member Submission*, 9 :29–1, 2004.