HAL open science

Homomorphismes d'hypergraphes pour la subsomption en RDF/RDFS

Jean-François Baget

To cite this version:

Jean-François Baget. Homomorphismes d'hypergraphes pour la subsomption en RDF/RDFS. Actes 10e conférence sur langages et modèles à objets (LMO), Mar 2004, Lille, France. pp.203-216. hal-00906633

HAL Id: hal-00906633

https://inria.hal.science/hal-00906633

Submitted on 20 Nov 2013

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.

Homomorphismes d'hypergraphes pour la subsomption en RDF/RDFS

Jean-François Baget

INRIA Rhône-Alpes
655 avenue de l'Europe
F–38334 Saint Ismier
jean-francois.baget@inrialpes.fr

RÉSUMÉ. *RDFS est un language de représentation de connaissances développé dans le cadre du Web Sémantique par le World Wide Web Consortium (W3C). Les objets de ce language (des graphes étiquetés) sont munis d'une sémantique formelle en théorie des modèles, ce qui permet de définir une relation de subsomption entre les documents RDFS. Dans cet article, nous présentons une reformulation de la subsomption en une sorte d'homomorphisme d'hypergraphes étiquetés, que nous appelons projection. La grande similarité entre cette projection et celle définie pour les graphes conceptuels nous permet d'importer un grand nombre de résultats théoriques et d'algorithmes.*

ABSTRACT. *RDFS is a knowledge representation language developped for the Semantic Web by the World Wide Web Consortium (W3C). Objects of this language (labelled graphs) are given formal model-theoretic semantics, allowing to define the subsumption relation between RDFS documents. In this paper, we reformulate subsumption into a kind of labelled hypergraphs homomorphism called projection. The huge similarities between this projection and the one defined for conceptual graphs allow us to translate many theoretical results and algorithms.*

MOTS-CLÉS : *web sémantique, RDF, RDFS, inférences, graphes conceptuels.*

KEYWORDS: *semantic web, RDF, RDFS, inferences, conceptual graphs.*

1. Introduction

RDF (Resource Description Framework) et son extension RDFS (RDF Schema) sont deux langages de représentation de connaissances développés par le World Wide Web Consortium (W3C) pour le web sémantique. Dans sa syntaxe abstraite [KLY 03], un document RDF ou RDFS (nous noterons RDF/S) est un ensemble de triplets représenté par un graphe orienté, étiqueté par des urirefs et des littéraux. RDF/S est muni d'une sémantique en théorie des modèles [HAY 03].

Nous proposons ici une alternative au mécanisme d'inférence suggéré par l'*interpolation lemma* [HAY 03]. Nous transformons dans un premier temps les graphes RDF/S en hypergraphes étiquetés, tout en préservant la propriété suivante : G subsume G' si et seulement si le transformé de G se projette dans le transformé de G'. La projection étant une sorte d'homomorphisme de graphes similaire à celle utilisée dans les graphes conceptuels [MUG 00], il s'avère ainsi possible de bénéficier des résultats théoriques et des algorithmes développés dans cette communauté.

Dans la section 2, nous présentons un sous-language de RDF que nous appelons RDFL. Nous rappelons sa syntaxe abstraite et sa sémantique, avant de présenter nos transformations et notre résultat d'adéquation et de complétude. Le même plan est respecté dans les sections 3 et 4, respectivement consacrées aux langages RDF puis RDFS. Enfin, avant de conclure, nous présentons les bénéfices de ce travail.

2. RDFL : RDF Light

RDFL (RDF Light) est un sous-langage de RDF ne comportant pas le vocabulaire pourvu d'une sémantique particulière. Nous présentons sa syntaxe (qui restera valable pour RDF et DRFS), sa sémantique, et proposons un mécanisme de subsomption.

2.1. *Syntaxe de RDFL*

Définition 1 (Document RDF/S) *Un* document RDF/S *est un ensemble de triplets* $\langle s, p, o \rangle$ *dont le* sujet s *peut être une* uriref *ou une* variable, *la* propriété p *est une* uriref, *et l'*objet o *peut être une* uriref, *une* variable, *ou un* littéral.

L'ensemble des urirefs et des littéraux apparaissant dans un document forme son *vocabulaire*, celui des urirefs, des littéraux et des variables forme ses *termes*. Les littéraux, les urirefs et les variables forment trois ensembles deux à deux disjoints.

– Les *urirefs* (URI references) sont un moyen adapté au web de nommer des ressources. Elles sont composées d'une URI (Uniform Resource Identifier), dont le cas particulier des URL est bien connu, ainsi que d'un fragment idetificateur, optionnel.

– Les *variables* (*blanks*) affirment l'existence d'une "chose" sans la spécifier. Notons dès à présent que la portée du nom d'une variable est limitée au document.

– Les *littéraux* sont une représentation lexicale pour une *valeur*. Ils peuvent être *simples* ou *typés*. La valeur associée à un littéral simple est sa représentation lexicale, celle associée à un littéral typé est également déterminée par son type. Tout type de données DT compatible RDF/S (défini en XML Schema [FAL 01]) encode un ensemble de représentations lexicales DT_L, un ensemble de valeurs DT_V, et une application VAL: $DT_L \rightarrow DT_V$. Si la représentation lexicale d'un littéral typé n'est pas dans DT_L, alors le littéral, comme le document dans lequel il apparaît, est *syntaxiquement inconsistant*.

2.2. *Sémantique en théorie des modèles [HAY 03]*

2.2.1. *Le vocabulaire et son interprétation*

Le *vocabulaire* est l'ensemble composé de toutes les urirefs et de tous les littéraux. Le *vocabulaire d'un ensemble de documents* est l'union de leur vocabulaire. Si $\mathcal{V}$ est un vocabulaire, nous notons $\mathcal{V}_U$ l'ensemble de ses urirefs et $\mathcal{V}_L$ celui de ses littéraux.

Définition 2 (Interprétation d'un vocabulaire) *Une* interprétation *d'un vocabulaire* $\mathcal{V}$ *est une paire* $\langle D, I\rangle$ *composée d'un* domaine $D = \langle I_R, I_P, LV, I_{EXT}\rangle$ *et d'une* fonction d'interprétation $I = \langle I_S, I_L\rangle$.

– I_R *est un ensemble non vide de resources et* I_P *est un ensemble de propriétés ;*

– $LV \subseteq I_R$ *est l'image par* VAL *des littéraux de* $\mathcal{V}_L$.

– $I_{EXT} : I_P \rightarrow 2^{I_R \times I_R}$ *associe à une propriété son extension ;*

– $I_S : \mathcal{V}_u :\rightarrow I_R \cup I_P$ *interprète chaque uriref par une resource ou une propriété ;*

– $I_L : \mathcal{V}_l :\rightarrow I_R$ *interprète chaque littéral syntaxiquement consistant* l *par sa valeur* VAL(l). *Si un littéral typé est syntaxiquement inconsistant (i.e. sa valeur par* VAL *n'est pas définie),* I_L *associe à ce littéral un élément de* I_R *qui n'est pas dans* LV[1].

Pour des raisons de simplicité, nous noterons $I : \mathcal{V} \rightarrow D$ la fonction d'interprétation qui associe à chaque élément du vocabulaire son image par I_S ou I_L.

2.2.2. *Modèles en RDFL*

Définition 3 (Modèle d'un document) *Une interprétation* $\langle D, I\rangle$ *d'un vocabulaire est un* modèle *d'un document* D *s'il existe une extension* I' *de* I *à l'ensemble des termes de* D *telle que :*

– *si* x *est une variable,* $I'(x) \in I_R$ *;*

– *pour chaque triplet* $\langle s, p, o\rangle$, $\langle I'(s), I'(o)\rangle \in I_{EXT}(I'(p))$

1. Ceci peut sembler curieux : un document RDFL peut être syntaxiquement inconsistant et sémantiquement valide. Ceci s'explique par le fait que RDF est un language dédié au web : si la fonction VAL n'est pas accessible, on veut tout de même pouvoir travailler sur le document.

Nous appelons I' la *preuve* que l'interprétation $\langle D, I\rangle$ est un modèle du document.

Par extension, $\langle D, I\rangle$ est un modèle pour un ensemble de documents $\{G_1, \ldots, G_k\}$ ssi $\langle D, I\rangle$ est un modèle pour chacun des G_i. La fusion de documents permet de ramener cette définition étendue au cas élémentaire, via la Prop. 1 [HAY 03].

Définition 4 (Fusion) *Nous appelons fusion des documents $\{G_1, \ldots, G_k\}$ et notons $G = G_1 + \cdots + G_k$ le document obtenu de la façon suivante :*

– les documents $G'_1, \ldots, G'_k$ sont obtenus respectivement à partir de $G_1, \ldots, G_k$ en renommant leurs variables (ce qui ne change pas leur sémantique) de façon à ce que les ensembles de variables intervenant dans chaque G'_i soient deux à deux disjoints ;

– le document G est l'union des documents G'_i.

Propriété 1 *Une interprétation est un modèle pour un ensemble de documents RDFL $\{G_1, \ldots, G_k\}$ ssi cette interprétation est un modèle de $G = G_1 + \cdots + G_k$.*

Nous allons maintenant pouvoir définir la subsomption entre deux documents. G subsume G' veut dire que toute l'information présente dans G est déja dans G'.

Définition 5 (Subsomption) *Soient G et G' deux documents. On dit que G subsume G' si tous les modèles de G' sont des modèles de G.*

2.3. *Subsomption et homomorphismes*

Nous reformulons la subsomption en un homomorphisme d'hypergraphes étiquetés que nous appelons *projection*[2].

2.3.1. *Hypergraphe d'une interprétation*

Codons une interprétation $\langle D, I\rangle$ par un hypergraphe orienté, étiqueté $\mathcal{H}(\langle D, I\rangle)$. Ses sommets sont les éléments de I_R et de I_P, dont l'étiquette $\epsilon(x)$ est la partie du vocabulaire interprétée par cet élément. Les hyperarcs (généralisation des arcs binaires à une dimension superieure) encodent la relation I_{EXT}.

– à chaque élément x de $I_R \cup I_P$ nous associons un sommet $h(x)$, étiqueté par $\{\}$.

– si u est une uriref de $\mathcal{V}$ et $I_S(u) = x$, alors on rajoute u à l'étiquette de $h(x)$;

– si x est un élément de LV (x est la valeur d'un littéral l), alors $\epsilon(h(x)) = l$;

– si l est un littéral syntaxiquement inconsistant et $I_L(l) = x$, ajouter l à $\epsilon(x)$;

– pour chaque élément p de I_p, pour chaque paire $\langle s, o\rangle \in I_{EXT}(p)$, le triplet $\langle h(s), h(p), h(o)\rangle$ est un hyperarc de l'hypergraphe.

2. Puisque les hypergraphes que nous obtenons sont en fait des graphes conceptuels [SOW 84]).

2.3.2. *Hypergraphe d'un document*

A tout document RDFL G, nous associons un hypergraphe orienté, étiqueté $\mathcal{H}(G)$.

– si u est une uriref de G, alors $h(u)$ est un nouveau sommet étiqueté par $\{u\}$;

– si l est un littéral du document :

  – syntaxiquement inconsistant, $h(l)$ est un nouveau sommet d'étiquette $\{l\}$;
  – si $\text{VAL}(l)$ est une valeur étiquetant déja un sommet s de $\mathcal{H}(G)$, alors $h(l) = s$;
  – sinon, $h(l)$ est un nouveau sommet étiqueté par $\{\text{VAL}(l)\}$

– si v est une variable du document, $h(v)$ est un nouveau sommet, étiqueté par $\{\}$;

– pour chaque triplet $\langle s, p, o\rangle$, nous obtenons un hyperarc $\langle h(s), h(p), h(o)\rangle$.

2.3.3. *La fonction d'interprétation vue comme un homomorphisme*

Nous définissons tout d'abord la projection entre nos hypergraphes. Il s'agit d'une sorte d'homomorphisme (conservation des hyperarcs) préservant les étiquettes.

Définition 6 (Projection) *Soient G et G' deux hypergraphes étiquetés. Une* projection *de G dans G' est une application Π des sommets de G dans ceux de G' tq :*

– pour chaque sommet x de G, pour chaque élément e de $\epsilon(x)$, $e \in \epsilon(\Pi(x))$;

– pour chaque hyperarc $\langle s, p, o\rangle$ de G, $\langle \Pi(s), \Pi(p), \Pi(o)\rangle$ est un hyperarc de G'.

Propriété 2 *Une interprétation $\langle D, I\rangle$ est un modèle pour un document RDFL G si et seulement si il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$.*

Preuve: La preuve est immédiate, puisqu'il s'agit d'une reformulation des définitions 2 et 3. Elle montre que les preuves qu'une interprétation est un modèle sont des projections.

($\Rightarrow$) Soit $\langle D, I\rangle$ une interprétation qui est un modèle du document RDFL G (il en existe donc une preuve I'). Nous allons prouver que l'application Π définie de la façon suivante est une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$.

  – Soit x un sommet de $\mathcal{H}(G)$ dont l'étiquette $\{e\}$ contient une uriref ou un littéral. Alors $\Pi(x) = h(I(e))$.
  – Soit x un sommet de $\mathcal{H}(G)$ d'étiquette vide, obtenu à partir d'une variable v. Alors $\Pi(x) = h(I'(v))$.

Nous devons maintenant prouver que Π est une projection, *i.e.* qu'elle préserve *(i)* les étiquettes et *(ii)*la relation I_{EXT}.

(i) Soit x un sommet de $\mathcal{H}(G)$. Prouvons que tous les éléments de son étiquette apparaissent dans celle de $\Pi(x)$. Si x a été obtenu à partir d'une variable, alors son étiquette est vide et la propriété est vraie. $I(e)$ est donc un élément de I_R, et, par construction, $h(I(e)) = \Pi(x)$ contient l'étiquette e.

(ii) Soit $\langle x, y, z\rangle$ un hyperarc de $\mathcal{H}(G)$. Par construction de $\mathcal{H}(G)$, il existe donc un triplet $\langle s, p, o\rangle$ de G tel que $h(s) = x$, $h(p) = y$ et $h(o) = z$. Comme I' est une preuve, $\langle I'(s), I'(o)\rangle \in I_{EXT}(I(p))$. Par construction, $\langle h(I'(s)), h(I(p)), h(I'(o))\rangle$ est donc un hyperarc de $\mathcal{H}(\langle D, I\rangle)$. Nous concluons en remarquant que $h(I'(s)) = \Pi(x)$, que $h(I(o)) = \Pi(y)$, et que $h(I'(o)) = \Pi(z)$.

($\Leftarrow$) Soit maintenant Π une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$. Nous allons maintenant exhiber une preuve I' que $\langle D, I\rangle$ est un modèle de G. Cette preuve est définie de la façon suivante :

  – si x est une uriref ou un littéral, alors $I'(x) = I(x)$;
  – si x est une variable, alors soit $I'(x) = y$, avec $h(y) = \Pi(h(x))$.

Il ne reste plus qu'à voir que, pour chaque triplet $\langle s, p, o\rangle$ de G, $\langle I'(s), I'(o)\rangle \in I_{EXT}(I(p))$, ce qui est immédiat par construction, le triplet $\langle \Pi(h(s)), \Pi(h(p)), \Pi(h(o))\rangle$ étant un hyperarc de $\mathcal{H}(\langle D, I\rangle)$. $\square$

2.3.4. *Théorème d'adéquation et de complétude*

La preuve de la propriété suivante va nous permettre d'introduire la notion de *modèle isomorphe d'un document RDFL*.

Propriété 3 *Tout document RDFL admet un modèle.*

Preuve: Soit G un document RDFL, et $\mathcal{H}(G)$ son hypergraphe associé. $\mathcal{H}(G)$ est aussi associé à une interprétation du vocabulaire de G. Or il existe une projection (un isomorphisme) de $\mathcal{H}(G)$ dans lui-même. Donc cette interprétation est un modèle de G (Prop. 2). Nous l'appelons le modèle isomorphe de G. Soit maintenant une interprétation $\langle I, D\rangle$ dont l'hypergraphe associé $\mathcal{H}(\langle I, D\rangle)$ admet comme sous-graphe partiel $\mathcal{H}(G)$. Alors il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle I, D\rangle)$ et $\langle I, D\rangle$ est donc un modèle de G (Prop. 2). On dit que $\langle I, D\rangle$ *contient le modèle isomorphe* de G. □

Théorème 1 (Adéquation et complétude) *Un document RDFL G subsume un document RDFL G' si et seulement si il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(G')$.*

Preuve: Nous prouverons successivement les deux sens de l'équivalence.

($\Rightarrow$) Soient G et G' deux documents RDFL tels que G subsume G'. Donc tous les modèles de G' sont des modèles de G. En particulier, tout modèle de G' contenant le modèle isomorphe de G' est un modèle de G. Parmi tous ces modèles, nous en choisissons un, $\langle D, I\rangle$ (il existe), qui ne connait pas d'autres propriétés que celles du modèle isomorphe de G' (*i.e.* $\langle x, y\rangle \in I_{EXT}(z)$ dans $\langle D, I\rangle$ si et seulement si x, y et z sont les interprétations de termes de G' et $\langle x, y\rangle \in I_{EXT}(z)$ dans le modèle isomorphe de G').

Comme $\langle D, I\rangle$ est aussi un modèle de G, alors (Prop. 2) il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$. Voir que cette projection ne peut se faire que dans la partie de $\mathcal{H}(\langle D, I\rangle)$ correspondant à $\mathcal{H}(G')$.

($\Leftarrow$) Soit Π une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(G')$. Soit $\langle D, I\rangle$ un modèle quelconque de G'. Prouvons que $\langle D, I\rangle$ est un modèle de G. Si $\langle D, I\rangle$ est un modèle de G', alors (Prop. 2) il existe une projection Π' de $\mathcal{H}(G')$ dans $\mathcal{H}(\langle D, I\rangle)$. La composition de deux projections étant une projection, $\Pi \circ \Pi'$ est une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$. Nous concluons, grâce à la Prop. 2, que $\langle D, I\rangle$ est un modèle de G. □

3. RDF = RDFL + vocabulaire RDF

RDF est construit à partir de RDFL, en ajoutant un vocabulaire spécifique (le vocabulaire RDF) doté d'une sémantique particulière.

`rdf:type`, `rdf:Property`, `rdf:XMLLiteral`, `rdf:nil`, `rdf:List`, `rdf:Statement`, `rdf:subject`, `rdf:predicate`, `rdf:object`, `rdf:first`, `rdf:rest`, `rdf:Seq`, `rdf:Bag`, `rdf:Alt`, `rdf:_1`, `rdf:_2`, ..., `rdf:_10000`, ..., `rdf:value`.

3.1. *Interprétation du vocabulaire RDF*

3.1.1. *Contraintes supplémentaires pour une interprétation*

– Toutes les propriétés ont pour type `rdf:Property`. Pour chaque ressource x dans I_P, nous rajoutons $\langle x, I(\texttt{rdf:Property}\rangle$ dans $I_{EXT}(I(\texttt{rdf:type})$.

– `rdf:XMLLiteral` est le seul type de données intégré à RDF. Si l est un littéral typé de type `rdf:XMLLiteral` syntaxiquement consistant, alors nous devons avoir $\langle I_L(l), I(\texttt{rdf:XMLLiteral})\rangle \in I_{EXT}(\texttt{rdf:type})$. Cette paire n'appartient pas à l'extension de l'interprétation de `rdf:type` si l est syntaxiquement inconsistant.

– Toute interprétation est un modèle pour les triplets axiomatiques suivants :

```
rdf:type rdf:type rdf:Property.
rdf:subject rdf:type rdf:Property.
rdf:predicate rdf:type rdf:Property.
rdf:object rdf:type rdf:Property.
rdf:first rdf:type rdf:Property.
rdf:rest rdf:type rdf:Property.
rdf:value rdf:type rdf:Property.
rdf:_1 rdf:type rdf:Property.
rdf:_2 rdf:type rdf:Property.
...
rdf:nil rdf:type rdf:List.
```

3.1.2. *Satisfiabilité et subsomption*

Les définitions de modèle et de subsomption ne changent pas. Certaines propriétés de RDFL sont invalidées. Par exemple, en RDFL, le document vide n'est subsumé que par lui même. En RDF, le document vide est subsumé par tout document qui subsume les triplets axiomatiques. De même, tout document n'admet pas un modèle :

Propriété 4 *Un document RDF G admet un modèle ssi il ne contient pas de triplet $\langle x, \texttt{rdf:type}, \texttt{rdf:XMLLiteral}\rangle$ où x n'est ni une variable, ni un littéral typé par* `rdf:XMLLiteral` *syntaxiquement valide. Un tel triplet dénote une* erreur de typage.

Preuve: La partie ($\Rightarrow$) est immédiate. Pour la réciproque, nous montrerons plus loin (Th. 2) comment construire le modèle isomorphe d'un document sans erreur de typage. □

3.2. *Mise à jour du mécanisme d'inférence*

Généralisons à RDF le résultat d'adéquation et de complétude énoncé pour RDFL. Les choses ne sont plus si simples. En effet, on peut déduire du document vide une infinité de triplets (par exemple les triplets $\langle$ `rdf:_i rdf:type rdf:Property`$\rangle$). Afin d'obtenir un résultat similaire au précédent, il faudrait construire un graphe $\mathcal{H}(G')$ permettant de projeter l'hypergraphe associé à cette infinité de triplets, *i.e.* un hypergraphe infini. Cependant, nous n'avons besoin que des triplets suffisants pour projeter $\mathcal{H}(G)$, qui est un graphe fini. Nous utilisons donc un sous-graphe fini $\mathcal{H}'(G', G)$ qui n'encode que l'information nécessaire.

3.2.1. *Hypergraphe associé à la subsomption entre deux documents RDF*

Soient G et G' deux documents RDF, tels que l'on se demande si G subsume G'. Nous construisons l'hypergraphe $\mathcal{H}'(G', G)$ de la façon suivante :

– soit k le maximum des entiers i tels que `rdf:_i` est un terme de G ou de G' ;

– soit A_k le document composé de tous les triplets axiomatiques de RDF auxquels on a ôté ceux utilisant `rdf:_i` avec $i > k$;

– soit X l'ensemble des triplets $\langle l\texttt{rdf:type rdf:XMLLiteral}\rangle$ où l est un littéral typé par `rdf:XMLLiteral` apparaissant dans le vocabulaire de G ou de G' ;

– nous construisons l'hypergraphe $H = \mathcal{H}(G + A_k + X)$;

– $\mathcal{H}'(G', G)$ est obtenu à partir de H en rajoutant, pour chaque hyperarc $\langle s, p, o\rangle$ de H, un hyperarc $\langle p, t, P\rangle$ où t est l'unique sommet de H ayant pour étiquette `rdf:type` et P est l'unique sommet de H ayant pour étiquette `rdf:Property`.

3.2.2. *Subsomption et projection*

Propriété 5 *Soient G et G' avec G' un document sans erreur de typage, et $\langle D, I\rangle$ une interprétation (au sens RDF). Alors les assertions suivantes sont équivalentes :*

(i) *$\langle D, I\rangle$ est un modèle de G' ;*

(ii) *il existe une projection de $\mathcal{H}'(G', G)$ dans $\mathcal{H}(\langle D, I\rangle)$;*

(iii) *il existe une projection de $\mathcal{H}(G')$ dans $\mathcal{H}(\langle D, I\rangle)$;*

Preuve: L'équivalence (*(i)* $\Rightarrow$ *(iii)*) vient directement de la Prop. 2. Le fait que $\langle D, I\rangle$ soit une interprétation "plus riche" que dans la version RDFL ne change rien.

(*(ii)* $\Rightarrow$ *(iii)*) Voir que $\mathcal{H}(G')$ est un sous-graphe partiel de $\mathcal{H}'(G', G)$. Or la restriction d'une projection à un sous-graphe partiel est une projection. Donc si $\mathcal{H}'(G', G)$ se projette dans $\mathcal{H}(\langle D, I\rangle)$, $\mathcal{H}(G')$ se projette dans $\mathcal{H}(\langle D, I\rangle)$.

(*(ii)* $\Leftarrow$ *(iii)*) Supposons maintenant une projection Π de $\mathcal{H}(G')$ dans $\mathcal{H}(\langle D, I\rangle)$. Nous allons étendre cette projection à une projection Π' de $\mathcal{H}'(G', G)$ dans $\mathcal{H}(\langle D, I\rangle)$.

Identifions dans un premier temps les sommets et les hyperarcs de $\mathcal{H}'(G', G)$ qui ne sont pas dans $\mathcal{H}(G')$. Ce sont ceux dont il faudra tenir compte pour étendre la projection. Ce sont nécessairement : *a)* des sommets et des hyperarcs issus des triplets axiomatiques de RDF ; *b)* des sommets et des hyperarcs issus de l'ensemble X de triplets ; ou *c)* les hyperarcs $\langle p, t, P\rangle$ issus des hyperarcs $\langle s, p, o\rangle$ de $\mathcal{H}(G')$ (avec t étiqueté par `{rdf:type}` et P étiqueté par `{rdf:Property}`). Les deux premiers cas ne posent aucun problème, ce sont des axiomes ne comportant aucune variable, qui ont été encodés dans $\langle D, I\rangle$ et donc dans $\mathcal{H}(\langle D, I\rangle)$. On n'a donc pas le choix pour étendre Π à Π' : chacun de ces sommets est envoyé vers l'unique sommet x de $\mathcal{H}(\langle D, I\rangle)$ contenant son étiquette.

Il ne reste plus qu'à vérifier que pour chacun des hyperarcs $r = \langle p, t, P\rangle$, $\langle \Pi'(p), \Pi'(t), \Pi'(P)\rangle$ est un hyperarc de $\mathcal{H}(\langle D, I\rangle)$. Par construction, l'existence de r indique que p était le deuxième argument d'un hyperarc de $\mathcal{H}(G')$, et $\Pi'(p) = \Pi(p)$ est nécessairement le deuxième argument d'un hyperarc de $\mathcal{H}(\langle D, I\rangle)$ (par définition de la projection). Donc, par construction de $\mathcal{H}(\langle D, I\rangle)$, il existe un hyperarc $\langle \Pi(p), t', P'\rangle$ dans $\mathcal{H}(\langle D, I\rangle)$ où t' est l'unique sommet dont l'étiquette contient `rdf:type` et P' est l'unique sommet dont l'étiquette contient `rdf:Property`. Nous pouvons maintenant conclure puisque nous avons défini Π' de la façon suivante : $\Pi'(t) = t'$ et $\Pi'(P) = P'$. □

Théorème 2 *Soient G et G' deux documents RDF. Alors G subsume G' ssi il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}'(G', G)$ ou si G' contient une erreur de typage.*

Preuve: Eliminons tout d'abord le cas où G' contient une erreur de typage. Dans ce cas (Prop. 4), G' n'admet pas de modèle et, par vacuité, tous les modèles de G' sont des modèles de G. Sinon :

($\Rightarrow$) Si G subsume G', alors tous les modèles de G' sont des modèles de G. Il existe parmi ceux-ci un modèle $\langle D, I\rangle$ tel que $\mathcal{H}(\langle D, I\rangle)$ admet comme sous-graphe $\mathcal{H}'(G, G')$, et ne possède pas d'autres hyperarcs que ceux issus des triplets axiomatiques que nous avons rejetés de la construction de $\mathcal{H}'(G, G')$, et ceux issus des littéraux typés rdf :XMLLitteral qui n'appartiennent ni au vocabulaire de G, ni à celui de G'. Ce modèle existe, puisqu'il n'y a pas d'erreur de typage. Comme $\langle D, I\rangle$ est aussi un modèle de G, alors il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$ (Prop. 5). Voir que cette projection ne peut se faire que dans le sous-graphe $\mathcal{H}'(G, G')$ de $\mathcal{H}(\langle D, I\rangle)$.

($\Leftarrow$) Soit Π une projection de $\mathcal{H}(G)$ dans $\mathcal{H}'(G', G)$. Soit $\langle D, I\rangle$ un modèle quelconque de G'. Prouvons que $\langle D, I\rangle$ est un modèle de G. Si $\langle D, I\rangle$ est un modèle de G', alors (Prop. 5) il existe une projection Π' de $\mathcal{H}(G', G)$ dans $\mathcal{H}(\langle D, I\rangle)$. Donc $\Pi \circ \Pi'$ est une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$. D'où (Prop. 5) $\langle D, I\rangle$ est un modèle de G. □

4. RDFS = RDF + vocabulaire RDFS

RDFS étend le langage RDF principalement par l'introduction de la notion de classe, de sous-classe et la mise en place d'un métamodèle pour le langage. Il introduit le vocabulaire réservé suivant :

`rdfs:domain`, `rdfs:range`, `rdfs:Resource`, `rdfs:Literal`, `rdfs:Datatype`, `rdfs:Class`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:member`, `rdfs:Container`, `rdfs:ContainerMembershipProperty`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`, `rdfs:label`.

4.1. *Interprétation d'un document RDFS*

La structure permettant d'interpréter un document RDFS est plus riche que pour RDF. En effet, une interprétation RDFS contiendra un ensemble I_C de classes et une application IC_{EXT} qui associe à chaque classe son extension. Cependant, cet enrichissement de la structure n'est qu'apparent : la contrainte *4)* dans la définition suivante montre que la relation I_{EXT} suffit à encoder la relation IC_{EXT}.

Définition 7 *Soit $\mathcal{V}$ un vocabulaire. Une* interprétation RDFS *de ce vocabulaire est une paire $\langle D, I\rangle$ composée d'un* domaine $D = \langle I_R, I_P, I_C, LV, I_{EXT}, IC_{EXT}\rangle$ *et d'une* fonction d'interprétation $I = \langle I_S, I_L\rangle$*, telle que :*

1) $\langle D', I\rangle$ *où* $D' = \langle I_R, I_P, LV, I_{EXT}\rangle$ *est une interprétation de $\mathcal{V}$ (au sens RDF) ;*

2) $I_C \subset I_R$ *est un ensemble de classes ;*

3) $IC_{EXT} : I_C \rightarrow 2^{I_R}$ *associe à chaque classe son extension ;*

4) $x \in IC_{EXT}(y)$ *ssi* $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdf:type}))$ *(assimilation types et classes) ;*

5) $I_C = IC_{EXT}(I(\texttt{rdfs:Class}))$ *;*

6) $I_R = IC_{EXT}(I(\texttt{rdfs:Resource}))$ *;*

7) $L_V = IC_{EXT}(I(\texttt{rdfs:Literal}))$ *;*

8) *Si* $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdfs:domain}))$ *et* $\langle u, v\rangle \in I_{EXT}(x)$ *alors* $u \in IC_{EXT}(y)$ *;*

9) *Si* $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdfs:range}))$ *et* $\langle u, v\rangle \in I_{EXT}(x)$ *alors* $v \in IC_{EXT}(y)$ *;*

10) *La relation* $I_{EXT}(I(\texttt{rdfs:subPropertyOf}))$ *est réflexive et transitive sur* I_P *;*

11) *Si* $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdfs:subPropertyOf}))$ *alors* x *et* $y \in I_P$ *et* $I_{EXT}(x) \subseteq I_{EXT}(y)$ *;*

12) *Si* $x \in I_C$ *alors* $\langle x, I_R\rangle \in I_{EXT}(I(\texttt{rdfs:subClassOf}))$ *;*

13) *Si* $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdfs:subClassOf}))$ *alors* x *et* $y \in I_C$ *et* $IC_{EXT}(x) \subseteq IC_{EXT}(y)$ *;*

14) *La relation* $I_{EXT}(I(\texttt{rdfs:subClassOf}))$ *est réflexive et transitive sur* I_C *;*

15) *Si* $x \in IC_{EXT}(I(\texttt{rdfs:ContainerMembershipProperty}))$ *alors* $\langle x, I(\texttt{rdfs:member})\rangle \in I_{EXT}(I(\texttt{rdfs:subPropertyOf}))$ *;*

16) *Si* $x \in IC_{EXT}(I(\texttt{rdfs:Datatype}))$ *et* $y \in IC_{EXT}(x)$ *alors* $\langle y, I(\texttt{rdfs:Literal})\rangle \in I_{EXT}(I(\texttt{rdf:type}))$ *;*

Toute interprétation RDFS devra être un modèle (au sens RDF) pour les triplets axiomatiques de RDFS (voir liste complète dans [HAY 03]) tels que :

```
rdf:type rdfs:domain rdfs:Resource.
rdfs:domain rdfs:domain rdf:Property.
rdfs:range rdfs:domain rdf:Property.
rdfs:subPropertyOf rdfs:domain rdf:Property.
rdfs:subClassOf rdfs:domain rdfs:Class.
rdf:subject rdfs:domain rdf:Statement.
rdf:predicate rdfs:domain rdf:Statement.
rdf:object rdfs:domain rdf:Statement.
rdfs:member rdfs:domain rdfs:Resource.
rdf:first rdfs:domain rdf:List.
//Triplets similaires pour rdfs:range
rdf:Alt rdfs:subClassOf rdfs:Container.
rdf:XMLLiteral rdf:type rdfs:Datatype.
rdf:XMLLiteral rdfs:subClassOf rdfs:Literal.
```

Modèles et subsomption

La définition de modèle ne change pas. En effet, l'assimilation entre types et classes ($x \in IC_{EXT}(y)$ ssi $\langle x, y\rangle \in I_{EXT}(I(\texttt{rdf :type}))$) permet de ne pas tenir compte de IC_{EXT} pour définir à quelles conditions une interprétation RDFS est un modèle pour un document RDFS.

4.2. *Mise à jour du mécanisme d'inférence*

4.2.1. *Complétion de l'hypergraphe associé à un document*

Soient G et G' deux documents RDF, tels que l'on se demande si G subsume G'. Nous construisons l'hypergraphe $\mathcal{H}''(G', G)$ (où les hyperarcs ternaires traduisent la relation I_{EXT} et les arcs – temporaires – binaires la relation IC_{EXT}) :

– soit k le maximum des entiers i tels que `rdf:_i` est un terme de G ou de G' ;
– soit B_k le document composé de tous les triplets axiomatiques de RDFS auxquels on a ôté ceux utilisant `rdf:_i` avec $i > k$;
– nous construisons l'hypergraphe $\mathcal{H}'(G', G)$ (voir RDF) ;
– nous construisons l'hypergraphe $H = \mathcal{H}'(G' + B_k, G)$;
– Nous appliquons à H les règles de complétion suivantes (traduisant les contraintes sur les interprétations RDFS) jusqu'à saturation, ce qui se fait en un nombre quadratique d'application de règles. Nous noterons "SI H_1, ..., H_i ALORS $C_1, \ldots C_j$" la règle exprimant que si les hyperarcs H_k sont présents dans le graphe, alors il faut rajouter les hyperarcs C_p.

4) SI $\langle x, \texttt{rdf:type}, y\rangle$ ALORS $\langle x, y\rangle$, et réciproquement ;
6) si x est un sommet qui ne peut s'interpréter que par un élément de I_R (variable, littéral, ou extension d'une classe), alors $\langle x, \texttt{rdfs:Resource}\rangle$ est un arc du graphe ;
8) SI $\langle x, \texttt{rdfs:domain}, y\rangle$, $\langle u, x, v\rangle$ ALORS $\langle u, y\rangle$;
9) SI $\langle x, \texttt{rdfs:range}, y\rangle$, $\langle u, x, v\rangle$ ALORS $\langle v, y\rangle$;
10) SI $\langle x, \texttt{rdf:type}, \texttt{rdf:Property}\rangle$ ALORS $\langle x, \texttt{rdfs:subPropertyOf}, x\rangle$;
10′) SI $\langle x, \texttt{rdfs:subPropertyOf}, y\rangle$, $\langle y, \texttt{rdfs:subPropertyOf}, z\rangle$ ALORS $\langle x, \texttt{rdfs:subPropertyOf}, z\rangle$;
11) SI $\langle x, \texttt{rdfs:subPropertyOf}, y\rangle$ ALORS $\langle x, \texttt{rdf:type}, \texttt{rdf:Property}\rangle$, $\langle y, \texttt{rdf:type}, \texttt{rdf:Property}\rangle$;
11′) SI $\langle x, \texttt{rdfs:subPropertyOf}, y\rangle$, $\langle u, x, v\rangle$ ALORS $\langle u, y, v\rangle$;
12) rajouter au graphe le sommet $\top$ étiqueté par $\{\}$;
12′) SI $\langle x, \texttt{rdfs:Class}\rangle$ ALORS $\langle x, \texttt{rdfs:subClassOf}, \top\rangle$;
12″) SI $\langle x, \texttt{rdfs:Resource}\rangle$ ALORS $\langle x, \top\rangle$;
13) SI $\langle x, \texttt{rdfs:subClassOf}, y\rangle$ ALORS $\langle x, \texttt{rdfs:Class}\rangle$, $\langle y, \texttt{rdfs:Class}\rangle$;
13′) SI $\langle x, \texttt{rdfs:subClassOf}, y\rangle$, $\langle u, x\rangle$ ALORS $\langle u, y\rangle$;
14) SI $\langle x, \texttt{rdfs:Class}\rangle$ ALORS $\langle x, \texttt{rdfs:subClassOf}, x\rangle$;
14′) SI $\langle x, \texttt{rdfs:subClassOf}, y\rangle$, $\langle y, \texttt{rdfs:subClassOf}, z\rangle$ ALORS $\langle x, \texttt{rdfs:subClassOf}, z\rangle$;
15) SI $\langle x, \texttt{rdfs:ContainerMembershipProperty}))\rangle$ ALORS $\langle x, \texttt{rdfs:subPropertyOf}, \texttt{rdfs:member}\rangle$;
16) SI $\langle x, \texttt{rdfs:Datatype}))\rangle$, $\langle x, y\rangle$ ALORS $\langle y, \texttt{rdf:type}, \texttt{rdfs:Literal}\rangle$;

– enfin, nous ôtons du graphe obtenu tous les arcs binaires (la relation IC_{EXT} étant redondante par rapport à la relation I_{EXT}).

4.2.2. *Théorème d'adéquation et de complétude*

Contrairement à RDF, les erreurs de typage ne se lisent pas directement dans le document, elles peuvent se déduire par le mécanisme de saturation exposé.

Propriété 6 *Un document RDFS G admet un modèle si et seulement si l'hypergraphe $\mathcal{H}''(G, G)$ ne contient pas d'hyperarcs tels que :*
– $\langle x, \texttt{rdf:type}, \texttt{rdf:XMLLiteral}\rangle$ *où x est le sommet associé à un littéral typé par* `rdf:XMLLiteral` *syntaxiquement inconsistant ;*
– $\langle x, \texttt{rdf:type}, \texttt{rdfs:Literal}\rangle$ *où x est un sommet qui n'est pas associé à un littéral typé syntaxiquement consistant.*

Remarquons que si on utilise d'autres types que `rdf:XMLLiteral` (les types de données XSD) alors se posent de nouveaux problèmes liés à l'insatisfiabilité. En effet, un ensemble de triplets suffit à exprimer une conjonction de types. Il faut alors savoir si il existe une valeur commune à ces types, ou le document n'a pas de modèle. Malheureusement, aucun mécanisme standard ne permet de décider de cette question...

Théorème 3 *Soient G et G' deux documents RDFS. Alors G subsume G' ssi il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}''(G', G)$ ou si G' n'admet pas de modèle.*

Nous ne prouverons ici ni cette propriété ni ce théorème. Ces deux démonstrations reposent sur l'assimilation des hypergraphes à des graphes conceptuels (q.v.), et des règles de saturation à des règles de graphes conceptuels [SAL 96] (et même des règles "range restricted" [BAG 02], ce qui assure la polynomialité de la saturation). Par manque de place, nous ne pouvons présenter ici les outils nécessaires.

5. Bénéfices de notre reformulation

Nous avons reformulé la subsomption entre documents RDF(S) en une projection d'hypergraphes étiquetés. Examinons maintenant les bénéfices de ce travail.

5.1. *Complexité*

Nous nous intéressons maintenant au problème de décision suivant :

RDF(S) SUBSOMPTION
Données : Deux documents RDF(S) G et G'.
Questions : Est-ce-que G subsume G', *i.e.* est-ce-que $\mathcal{H}(G)$ se projette dans $\mathcal{H}'(G', G)$ où est-ce-que G' est insatisfiable ?

Propriété 7 RDF SUBSOMPTION *est un problème NP-complet.*

Preuve: L'insatisfiabilité de G' se vérifie de façon polynomiale (lecture des triplets dans le cas de RDF, saturation quadratique préalable dans le cas de RDFS). Attachons-nous maintenant au cas où G' admet un modèle.

Nous montrons tout d'abord que RDF SUBSOMPTION admet un certificat polynomial : la projection de $\mathcal{H}(G)$ dans $\mathcal{H}'(G', G)$. La projection est linéaire dans la taille de $\mathcal{H}(G)$, et les hypergraphes sont de taille linéaire dans la taille des documents RDF. RDF SUBSOMPTION est donc bien dans NP.

Ensuite, un cas particulier de RDF subsomption est quand les sommets de G et G' sont tous des variables, que tous les arcs sont étiquetés par la même relation, et que tous les arcs ont un symmétrique. Nous avons ainsi comme sous-problème le problème HOMOMORPHISME DE GRAPHES, prouvé NP-complet par [LEV 73]. □

5.2. *RDF/S et graphes conceptuels*

Nous montrons ici que nos hypergraphes sont des graphes conceptuels (GCs), en renvoyant le lecteur à [MUG 00] pour un exposé synthétique. A chaque hypergraphe H (obtenus par la transformation $\mathcal{H}$ ou $\mathcal{H}'$ à partir de documents RDF/S) nous associons un GC très similaire $GC(H)$ (il a la même structure). Bien que [MUG 00] définisse les Graphes Conceptuels comme des graphes bipartis, il s'agit en fait des mêmes objets (c'est le biparti d'incidence de l'hypergraphe utilisé dans [BAG 03]).

Le support sur lequel nos graphes conceptuels sont définis est un support trivial :
– l'ensemble des *marqueurs individuels* est le vocabulaire RDFS ;
– l'ensemble des *types de concepts* ne contient que le type $\top$;
– l'ensemble des *types de relations* ne contient que le type I_{EXT}.

Nous n'avons plus qu'à changer légèrement les étiquettes de H :
– les étiquettes $\{\}$ (variabless) sont remplacées par $\top : *$ (sommets génériques) ;
– les étiquettes $\{e\}$ sont rempacées par $\top : e$ (ce sont des sommets individuels) ;
– les hyperarcs sont tous étiquetés par I_{EXT}.

Un document RDF/S G ayant la même sémantique que le GC $\mathcal{G}(G)$ obtenu à partir de son hypergraphe associé par cette dernière transformation, il s'ensuit la propriété suivante, qui précise la relation entre GC et RDF (évoquée, par exemple, dans [BER 01]), et permet de calculer la subsomption en RDF/S en utilisant un moteur d'inférence de graphes conceptuels comme CoGITaNT [GEN 98].

Propriété 8 *Soient G et G' deux documents RDF(S), tels que G' admet un modèle. Alors G subsume G' ssi il existe une GC-projection de $\mathcal{G}(G)$ dans $\mathcal{G}(G', G)$.*

5.3. *Une sémantique logique pour RDF/S en FOL*$(\wedge, \exists)$

Une première conséquence du corollaire 8 est de faire hériter RDF d'une sémantique en logique positive, conjonctive, existentielle du premier ordre sans symbole fonctionnel (notée FOL$(\wedge, \exists)$).

En effet, la transformation Φ [SOW 84] associe à tout support $\mathcal{S}$ une formule logique $\Phi(\mathcal{S})$ et à tout graphe conceptuel H une formule logique $\Phi(G)$, et la GC-projection se ramène à la déduction entre ces formules logiques associées, à une condition de normalité près [CHE 95]. Les GC que nous associons à un document RDF étant sous forme normale, et Φ associant un ensemble vide de formules à un support trivial, il s'ensuit que :

Propriété 9 *Soient G et G' deux documents RDF/S, G' admettant un modèle. Alors G subsume G' ssi $\Phi(G) = \Phi(\mathcal{G}(G))$ est conséquence de $\Phi(G', G) = \Phi(\mathcal{G}(G', G))$.*

Il est donc possible via cette transformation de calculer la subsomption en RDF(S) en utilisant un moteur d'inférence prévu pour FOL$(\wedge, \exists)$.

5.4. *RDF(S) et CSP*

Un réseau de contraintes (CSP pour Constraint Satisfaction Problem) [MON 74] est un hypergraphe étiqueté dont les sommets sont étiquetés par un domaine et les hyperarcs représentent des contraintes. Un CSP est satisfiable si il existe une instanciation du CSP qui respecte toutes les contraintes.

[MUG 96] proposent une transformation $\mathcal{C}$, associant un CSP à une paire de GCs et à un support, qui assimile la projection des GC à la satisfiabilité du CSP obtenu. La propriété suivante nous permet de calculer la subsomption en RDF/S en utilisant un solver de contraintes (Choco, ILOG Solver, ...)

Propriété 10 *Soient G et G' deux documents RDF, tels que G' admet un modèle. Alors G subsume G' ssi $\mathcal{C}(G, G') = \mathcal{C}(\mathcal{G}(G), \mathcal{G}'(G', G), \emptyset)$ est satisfiable.*

5.5. *Cas polynomiaux et algorithmes*

[GOT 99] ont proposé un grand nombre de cas polynomiaux, basés sur la structure de l'hypergraphe, pour le problème de satisfiabilité d'un réseau de contraintes. Or la transformation $\mathcal{C}$ de [MUG 96] préserve la structure du graphe G (G et $\mathcal{C}(G, G', \mathcal{S})$ ont même structure). De même, notre transformation GC associant un graphe conceptuel à l'hypergraphe $\mathcal{H}(G)$ d'un document RDF G préserve la structure de cet hypergraphe. Les conditions de polynomialité sur la structure du CSP sont donc directement traduisibles en conditions portant sur la structure du document RDF/S requête.

De plus, la communauté CSP a obtenu d'importants résultats portant sur l'optimisation du mécanisme de BackTrack permettant de décider de la satisfiabilité d'un réseau de contraintes (ordonnancement de variables, filtrage, backjump, ...). Grâce aux propriétés structurelles de la transformation $\mathcal{C}$, il est possible d'importer la plupart de ces optimisations à la GC-projection [BAG 03].

6. Conclusion et perspectives : RDF + règles

Nous avons présenté dans cet article un codage de la subsomption en RDF/S par la projection de graphes conceptuels. Ceci nous permet d'obtenir de façon immédiate des transformations permettant de décider de la subsomption RDF/S en utilisant un moteur de graphes conceptuels, de logique du premier ordre ou un solver de contraintes. Nous avons également évoqué des pistes permettant de définir des cas polynomiaux pour le problème RDF(S)-SUBSOMPTION et de développer des algorithmes efficaces.

Ce travail pourrait bénéficier du mécanisme de règles proposé comme une extension des graphes conceptuels [SAL 96]. Le rapport établi entre la subsomption RDF/S et la projection de GC permet d'adapter ce mécanisme, afin d'en faire une extension de RDF/S. Cette approche est intéressante puisque de nombreux auteurs ont fait part de la nécessité d'avoir un language de règles pour le Web Sémantique.

7. Bibliographie

[BAG 02] BAGET J.-F., MUGNIER M.-L., « Extensions of Simple Conceptual Graphs : the Complexity of Rules and Constraints », *Journal of Artificial Intelligence Research*, vol. 16, 2002, p. 425 – 465, `http://www.cs.washington.edu/research/jair/contents/v16.html`.

[BAG 03] BAGET J.-F., « Simple Conceptual Graphs Revisited : Hypergraphs and Conjunctive Tupes for Efficient Projection Algorithms », vol. 2746 de *Lecture Notes in Artificial Intelligence*, Springer, juillet 2003, p. 229 – 242.

[BER 01] BERNERS-LEE T., « Conceptual Graphs and the Semantic Web », 2001, `http://www.w3.org/DesignIssues/CG.html`.

[CHE 95] CHEIN M., MUGNIER M.-L., « Conceptual Graphs are also Graphs », *Quatrièmes journées du Laboratoire d'Informatique de Paris-Nord*, 1995, p. 81 – 97, RR-LIRMM 95-003.

[FAL 01] FALLSIDE D. C., « XML Schema Part 0 : Primer W3C Recommendation 2 May 2001 », rapport, mai 2001, World Wide Web Consortium (W3C), `http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/`.

[GEN 98] GENEST D., ÉRIC SALVAT, « A Platform Allowing Typed Nested Graphs : How CoGITo Became CoGITaNT », vol. 1453 de *Lecture Notes in Computer Science*, Springer, août 1998, p. 154 – 161.

[GOT 99] GOTTLOB G., LEONE N., SCARCELLO F., « A Comparison of Structural CSP Decomposition Methods », Morgan Kaufmann, juillet 1999, p. 394 – 399 (Vol. 1).

[HAY 03] HAYES P., « RDF Semantics W3C Proposed Recommendation 15 December 2003 », rapport, décembre 2003, World Wide Web Consortium (W3C), `http://www.w3.org/TR/2003/PR-rdf-mt-20031215/`.

[KLY 03] KLYNE G., CARROLL J. J., « Resource Description Framework (RDF) : Concepts and Abstract Syntax W3C Proposed Recommendation 15 December 2003 », rapport, décembre 2003, World Wide Web Consortium (W3C), `http://www.w3.org/TR/2003/PR-rdf-concepts-20031215/`.

[LEV 73] LEVIN L. A., « Universal sorting problems », *Problemy Peredaci Informacii*, vol. 9, 1973, p. 115 – 116, English translation in *Problems of Information Transmission*, 9, pp 265 – 266.

[MON 74] MONTANARI U., « Networks of Constraints : Fundamental Properties and Application to Picture Processing », *Information Sciences*, vol. 7, n° 2, 1974, p. 95 – 132.

[MUG 96] MUGNIER M.-L., CHEIN M., « Représenter des connaissances et raisonner avec des graphes », *Revue d'Intelligence Artificielle (numéro spécial "Graphes Conceptuels")*, vol. 10, n° 1, 1996.

[MUG 00] MUGNIER M.-L., « Knowledge Representation and Reasonings Based on Graph Homomorphism », vol. 1867 de *Lecture Notes in Computer Science*, Springer, août 2000, p. 172 – 192.

[SAL 96] ÉRIC SALVAT, MUGNIER M.-L., « Sound and Complete Forward and Backward Chainings of Graphs Rules », vol. 1115 de *Lecture Notes in Computer Science*, Springer, août 1996, p. 248 – 262.

[SOW 84] SOWA J. F., *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984.