Homomorphismes d'hypergraphes pour la subsomption en RDF

Jean-François Baget

Homomorphismes d'hypergraphes pour la subsomption en RDF

Jean-François Baget
INRIA Rhône-Alpes
655 avenue de l'Europe
38334 Saint Ismier, France
Jean-Francois.Baget@inrialpes.fr

Résumé *RDF est un language de représentation de connaissances développé pour le Web Sémantique par le World Wide Web Consortium (W3C). Ses objets (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 RDF.*

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.

1 Introduction

RDF (Resource Description Framework) est un language de représentation de connaissances en cours de développement par un groupe de travail du World Wide Web Consortium (W3C). Ce language a pour ambition d'être le language de base pour les languages dédiés au Web Sémantique. Ses spécifications sont données par un ensemble de documents, dont un est une recommandation officielle du W3C ; les autres documents en sont encore à une version de travail.

Dans sa syntaxe abstraite [13], un document RDF peut être représenté par un graphe orienté, étiqueté par des urirefs et des littéraux (ceci étant la base de l'adéquation de RDF au Web). Ce document sera cependant stocké sous la forme d'un document RDF/XML [14]. RDF est muni d'une sémantique formelle en théorie des modèles [10], ce qui permet de définir la notion de subsomption : un document G subsume un document G' si toute l'information représentée par G est déja présente dans G'. Pour fonder un langage de représentation de connaissances, il faut également un *mécanisme d'inférence* [12] adéquat et complet (il ne calcule que des inférences valides, et il les calcule toutes).

Fournir un mécanisme d'inférence efficace est donc d'une importance capitale si RDF connait le succès espéré par ses créateurs. Nous proposons ici une alternative au mécanisme d'inférence suggéré par l'*interpolation lemma* [10].

En effet, nous transformons dans un premier temps les graphes RDF en hypergraphes étiquetés, transformation dont la complexité est linéaire dans la taille des graphes. Cette transformation préserve 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 [17], il s'avère ainsi possible de bénéficier des résultats théoriques et des algorithmes développés dans cette communauté.

Dans la section 2, nous introduisons un sous-language de RDF appelé RDFL. Il s'agit du language RDF privé du vocabulaire réservé ayant une sémantique particulière. Nous rappelons et discutons sa syntaxe abstraite et sa sémantique, avant d'énoncer notre résultat d'adéquation et de complétude. Nous étendons ce travail dans la section 3, consacrée au language RDF dans sa totalité. Enfin, avant de conclure, nous présentons dans la section 4 les bénéfices (théoriques et algorithmiques) qui peuvent être retirés de ce travail.

2 RDFL

Le language RDFL (RDF Light) est une version simplifiée de RDF, ne comportant pas de vocabulaire spécifique. Nous présentons sa syntaxe, puis sa sémantique, et proposons enfin un mécanisme permettant de calculer la subsomption en RDFL, basé sur des homomorphismes de graphes.

2.1 Syntaxe

Dans sa syntaxe abstraite de RDFL, un document RDFL peut être vu comme un ensemble de triplets ou comme un graphe orienté (le graphe RDFL). Nous discutons des pièges que peut poser cette identification.

2.1.1 Le document RDFL : un ensemble de triplets

Définition 1 (Document RDFL) *Un* document RDFL *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 les triplets d'un document RDFL forment le *vocabulaire* de ce document. L'ensemble des urirefs, des littéraux et des variables forment les *termes* de ce document. Les littéraux, les urirefs et les variables forment trois ensembles deux à deux disjoints.

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

Variables : les *variables* (*blanks*) affirment l'existence d'une "chose" sans la spécifier. La portée du nom d'une variable est limitée au document RDFL.

Littéraux : les *littéraux* sont une représentation lexicale pour une *valeur*. Ces littéraux peuvent être *simples* (uniquement une représentation lexicale) ou *typés* (une représentation lexicale et un type de données). La valeur associée à un littéral simple est sa représentation lexicale elle-même, celle associée à un littéral typé est également déterminée par son type. Tout type de données DT compatible avec RDF doit encoder 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'appartient pas à celles définies dans son type, alors le littéral est dit *syntaxiquement inconsistant*. Un document RDFL dont un terme est syntaxiquement inconsistant sera, lui aussi, syntaxiquement inconsistant. Nous notons par extension VAL l'application qui associe sa valeur à tout littéral syntaxiquement consistant. Notons que RDF laisse à XML Schema [8] le soin de définir ces types de données.

2.1.2 Le document RDFL vu comme un graphe étiqueté

Graphe d'un document RDFL : A tout document RDF (ou RDFL) G, nous associons un graphe orienté, étiqueté $\mathcal{G}(G) = (V, U, \epsilon)$. A chaque terme x de G apparaissant comme sujet ou objet d'un triplet, nous associons un sommet distinct $g(x)$ dans V. Si x est une uriref ou un littéral, l'étiquette $\epsilon(g(x))$ de ce sommet est le terme x lui-même. Si x est une variable, le sommet $g(x)$ n'a pas d'étiquette. A chaque triplet $t = \langle s, p, o \rangle$ de G, nous associons un arc $g(t) = \langle g(s), g(o) \rangle$, étiqueté par $\epsilon(g(t)) = p$. Graphiquement, les sommets associés à des littéraux sont représentés par des rectangles, les autres par des ovales. L'étiquette, si elle existe, est inscrite à l'intérieur. Un arc $g(\langle s, p, o \rangle) = \langle g(s), g(o) \rangle$ est représenté par une flèche allant de $g(s)$ vers $g(o)$, à côté de laquelle on inscrit son étiquette $\epsilon(g(t))$.

Remarques Le choix fait en RDF est de confondre le document avec le graphe qui lui est associé. La confusion entre sommets et termes (étiquettes), ainsi que l'absence de véritables sommets (ils n'ont d'existence que relativement à un arc) peuvent être la source d'incompréhensions :

Un sous-graphe, en RDF, est un sous-ensemble de triplets. Nous l'appelons plutôt un *sous-document*, car cette notion ne correspond pas à celles de la théorie des graphes. Ce n'est ni un sous-graphe, ni un graphe partiel, ni un sous-graphe partiel.

Une *instance* d'un document RDF est obtenue en remplaçant une variable par un autre terme. Or il ne s'agit pas simplement du remplacement d'une étiquette : changer l'étiquette d'un sommet peut changer la structure du graphe (si l'étiquette est déja présente, il faut fusionner les sommets).

2.2 Sémantique en théorie des modèles

Nous rappelons ici la sémantique de RDFL, en théorie des modèles [10], et définissons interprétation, modèle, et subsomption.

2.2.1 Le vocabulaire et son interprétation

Le *vocabulaire RDFL* est l'ensemble de toutes les urirefs et de tous les littéraux. Le *vocabulaire d'un document RDFL* est sa restriction aux éléments apparaissant dans un des triplets du document. Le *vocabulaire d'un ensemble de documents RDFL* est l'union de leurs vocabulaires. 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) *Soit $\mathcal{V}$ un vocabulaire. Une* interprétation *de $\mathcal{V}$ est une paire 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 ;*
- $I_P \subseteq I_R$ *est un ensemble de propriétés ;*
- $LV \subseteq I_R$ *est composé de toutes les valeurs obtenues en appliquant* VAL *aux littéraux de $\mathcal{V}_L$.*
- $I_{EXT} : I_P \to 2^{I_R \times I_R}$ *associe à chaque propriété un ensemble de paires de resources ;*
- $I_S : \mathcal{V}_u :\to I_R$ *interprète chaque uriref de $\mathcal{V}$ par un élément de I_R ;*
- $I_L : \mathcal{V}_l :\to I_R$ *interprète chaque littéral syntaxiquement consistant par sa valeur (obtenue par* VAL*). Notons que si un littéral typé est syntaxiquement inconsistant (i.e. sa valeur par* VAL *n'est pas définie), I_L associe tout de même à ce littéral un élément de I_R qui n'est pas dans LV*[1].

Nous noterons $I : \mathcal{V} \to D$ 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

Les définitions suivantes indiquent comment prouver qu'une interprétation est un modèle d'un document ou d'un ensemble de documents RDFL.

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

- *si x est une variable, $I'(x) \in I_R$;*
- *pour chaque triplet $\langle s, p, o \rangle$, $\langle I'(s), I'(o) \rangle \in I_{EXT}(I'(p))$*

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

Définition 4 (Modèle d'un ensemble de documents RDFL) *Soit un ensemble $\{G_1, \ldots, G_k\}$ de documents RDFL. Une interprétation $\langle D, I \rangle$ d'un vocabulaire RDFL est un modèle pour cet ensemble de documents si et seulement si $\langle D, I \rangle$ est un modèle pour chacun des G_i.*

[1] Un document RDFL peut donc être syntaxiquement inconsistant tout en étant sémantiquement valide. Ceci s'explique par le fait que RDF est prévu pour le web : si la fonction VAL n'est pas accessible, on souhaite tout de même pouvoir travailler.

Cette définition se ramène à la précédente, via la propriété 1.

Définition 5 (Fusion d'un ensemble de documents RDFL) *Soit un ensemble $\{G_1, \ldots, G_k\}$ de documents RDFL. Nous appelons fusion de ces documents et notons $G = G_1 + \ldots + G_k$ le document obtenu de la façon suivante :*

- *les documents $G'_1, \ldots, G'_k$ sont obtenus respectivement à partir des G_i en renommant leurs variables 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.*

En "version graphe", cette définition peut se reformuler de la façon suivante : le graphe RDFL $G = G_1 + \ldots + G_k$ est obtenu à partir de l'union disjointe de $G_1, \ldots, G_k$ (le graphe dont le dessin est formé des dessins de $G_1, \ldots, G_k$), puis en fusionnant les sommets ayant même étiquette.

Propriété 1 ([10]) *Une interprétation est un modèle pour un ensemble de documents RDFL $\{G_1, \ldots, G_k\}$ si et seulement si cette interprétation est un modèle de $G = G_1 + \ldots + G_k$.*

2.2.3 La subsomption en RDFL

Nous définissons, de manière classique, la subsomption entre deux documents RDFL. G subsume G' veut dire que toute l'information présente dans G est déja dans G', ou que G' contient une réponse à la question G.

Définition 6 (Subsomption RDFL) *Soient G et G' deux documents RDFL. On dit que G subsume G' si tous les modèles de G' sont des modèles de G.*

La propriété 1 nous permet de justifier la généralisation de cette définition.

Définition 7 *Soient $H, G_1, \ldots, G_k$ un ensemble de documents RDFL. On dit que H subsume $\{G_1, \ldots, G_k\}$ si et seulement si H subsume $G_1 + \ldots + G_k$.*

2.3 Subsomption et homomorphismes

Cette partie développe le résultat central de cet article : la reformulation de la subsomption en un homomorphisme d'hypergraphes étiquetés que nous appelons *projection*. Nous montrons tout d'abord comment associer un hypergraphe à une interprétation ou à un document RDFL. Cette reformulation permet d'affirmer que toute preuve que l'interprétation est un modèle du document est une projection de l'hypergraphe associé au document dans celui associé à 'interprétation. Ce premier résultat fonde le théorème principal, qui affirme l'équivalence entre la subsomption et l'existence d'une projection.

2.3.1 Hypergraphe d'une interprétation

Nous codons une interprétation $\langle D, I\rangle$ par un hypergraphe orienté, étiqueté $\mathcal{H}(\langle D, I\rangle)$. Les sommets seront les éléments de I_R, étiquetés par l'ensemble des termes qui peuvent être interprétés par cet élément, et les hyperarcs (une généralisation des arcs à une dimension superieure) encoderont la relation I_{EXT}. Cet hypergraphe est construit de la façon suivante :

- à chaque élément x de I_R nous associons un sommet $h(x)$, initialement étiqueté par l'ensemble vide ($\epsilon(h(x)) = \{\}$) ;
- si u est une uriref de $\mathcal{V}$ et $I_S(u) = x$, alors on rajoute u à $\epsilon(h(x))$;
- si x est un élément de LV (x est la valeur d'un littéral), alors on ajoute cette valeur à $\epsilon(h(x))$ (notons toutefois que le cardinal de l'étiquette d'un tel sommet sera toujours 1) ;
- si l est un littéral syntaxiquement inconsistant et $I_L(l) = x$, alors on rajoute 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.

2.3.2 Hypergraphe d'un document

A tout document RDFL G, nous pouvons associer un hypergraphe orienté, étiqueté $\mathcal{H}(G)$ afin d'homogénéiser ce codage avec celui des interprétations.

- si u est une uriref de G, $h(u)$ est un nouveau sommet étiqueté par $\{u\}$;
- si l est un littéral de G :
  - si l est syntaxiquement inconsistant, alors $h(l)$ est un nouveau sommet étiqueté par $\{l\}$;
  - si VAL(l) étiquette 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 de G, $h(v)$ 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 hypergraphes. C'est une sorte d'homomorphisme (conservation des hyperarcs) préservant les étiquettes.

Définition 8 (Projection) *Soient G et G' deux hypergraphes, obtenus à partir d'une interprétation ou d'un document RDFL via la transformation $\mathcal{H}$. Nous appelons projection de G dans G' une application Π des sommets de G dans les sommets de G' qui respecte les contraintes suivantes :*

- *pour chaque sommet x de G, pour chaque élément e appartenant à l'étiquette $\epsilon(x)$, e doit aussi appartenir à $\epsilon(\Pi(x))$;*
- *pour chaque hyperarc $\langle s, p, o\rangle$ de G, $\langle \Pi(s), \Pi(p), \Pi(o)\rangle$ doit être 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: Immédiate, par reformulation des définitions 2 et 3. Son intérêt réside dans l'identification des preuves à des projections.
($\Rightarrow$) Soit $\langle D, I\rangle$ un modèle du document G (il en existe donc une preuve I'). Prouvons que l'application Π définie ci-dessous 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 $\mathcal{H}(G)$ dont l'étiquette est vide. Ce sommet a donc été obtenu à partir d'une variable v. Alors $\Pi(x) = h(I'(v))$.

Prouvons maintenant que Π 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 Π une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(\langle D, I\rangle)$. Nous exhibons une preuve I' que $\langle D, I\rangle$ est un modèle de G :

- 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 vérifier 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 de $\mathcal{H}(\langle D, I\rangle)$. ■

La preuve de la propriété suivante nous permet d'introduire la notion de *modèle isomorphe d'un document RDFL* (voir preuve Th. 1).

Propriété 3 *Tout document RDFL admet un modèle.*

Preuve: Soit G un document RDFL, et $\mathcal{H}(G)$ son hypergraphe associé. Voir que $\mathcal{H}(G)$ est aussi l'hypergraphe associé à une interprétation du vocabulaire de G. Or il existe une projection (un isomorphisme) de $H(G)$ 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 $H(G)$ (une telle interprétation existe). Alors il existe une projection de $H(G)$ 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. ■

2.3.4 Théorème d'adéquation et de complétude

Avant d'énoncer notre résultat principal, nous commencerons par un lemme préparatoire, dont la démonstration est immédiate.

Lemme 1 *La composition de deux projections est une projection.*

Théorème 1 (Adéquation et complétude) *Un document RDFL G subsume un document RDFL G' ssi 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. Et en particulier, tout modèle de G' contenant le modèle isomorphe de G' est un modèle de G. Choisissons un, $\langle D, I\rangle$, 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 $\langle D, I\rangle$ ssi 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 $\mathcal{H}(\langle D, I\rangle)$. D'après le Lemme 1, $\Pi \circ \Pi'$ 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. ■

La généralisation suivante est une application directe de la propriété 1.

Corollaire 1 *Un document RDFL G subsume un ensemble $\{G_1, \ldots, G_k\}$ de documents RDFL ssi il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}(G_1 + \ldots + G_k)$.*

3 RDF = RDFL + vocabulaire RDF

RDF se construit à partir de RDFL en ajoutant un vocabulaire spécifique doté d'une sémantique particulière.

3.1 Le vocabulaire RDF

RDF ajoute à RDFL le vocabulaire composé des urirefs suivantes :
`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 :value`.

3.2 Interprétation du vocabulaire RDF

Dans ce vocabulaire, seuls sont munis d'une sémantique particulière les urirefs `rdf :type`, `rdf :Property` et `rdf :XMLLiteral`. Nous allons les étudier.

3.2.1 Contraintes supplémentaires pour une interprétation

Propriétés Nous devons indiquer que toutes les propriétés (les éléments de I_P) ont pour type `rdf :Property`. Pour celà, pour chaque ressource x dans I_P, nous rajoutons $\langle x, I($`rdf :Property`$\rangle$ dans $I_{EXT}(I($`rdf :type`$)$.

Littéraux XML `rdf :XMLLiteral` est le seul type de données proposé par RDF. Si l est un littéral typé de type `rdf :XMLLiteral` syntaxiquement consistant, alors nous devons avoir $\langle I_L(l), I($`rdf :XMLLiteral`$)\rangle \in I_{EXT}($`rdf :type`$)$.

Les triplets axiomatiques Toute interprétation devra être un modèle pour les triplets suivants (dits *triplets axiomatiques de RDF*) :

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

3.2.2 Subsomption

Hormis ces modifications des interprétations, les définitions de modèle et de subsomption ne changent pas. Notons toutefois que ceci invalide certaines propriétés immédiates de la subsomption en RDFL. 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.

3.3 Mise à jour du mécanisme d'inférence

Nous généralisons maintenant à RDF le résultat précédent. Les choses ne sont cependant plus si simples. En effet, on peut déduire de n'importe quel document RDF (par exemple le document vide) une infinité de triplets. Il y a l'infinité de triplets axiomatiques $\langle$`rdf :_i rdf :type rdf :Property`$\rangle$ puis l'infinité de triplets $\langle$ l `rdf :type rdf :XMLLiteral`$\rangle$ ou l est un littéral typé par `rdf :XMLLitteral` syntaxiquement consistant. Afin d'obtenir un résultat similaire au précédent (G subsume G' ssi il existe une projection de $\mathcal{H}'(G)$ dans $\mathcal{H}'(G')$), il faudrait donc construire un graphe $\mathcal{H}(G')$ permettant de projeter l'hypergraphe associé à cette infinité de triplets, *i.e.* un graphe $\mathcal{H}(G')$ infini. Cependant, nous n'avons besoin que des triplets suffisants pour projeter $\mathcal{H}(G)$, qui est un graphe fini. Nous utiliserons donc un sous-graphe fini $\mathcal{H}'(G', G)$ de $\mathcal{H}'(G)$ qui n'encodera que l'information nécessaire.

3.3.1 Hypergraphe associé à la subsomption en 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 plus grand entier i tel 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 $\langle l$`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`.

3.3.2 Projection dans l'hypergraphe d'une interprétation

Adaptons la propriété 2 pour les modèles d'un document RDF.

Propriété 4 *Soient G et G' deux documents RDF quelconques, 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)* $\Leftrightarrow$ *(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)* Nous remarquons tout d'abord 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 encore une projection. Donc si $\mathcal{H}'(G', G)$ se projette dans $\mathcal{H}(\langle D, I\rangle)$,nécessairement $\mathcal{H}(G')$ 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 :

- des sommets et des hyperarcs issus des triplets axiomatiques de RDF ;
- des sommets et des hyperarcs issus de l'ensemble X de triplets ;
- 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 de $\mathcal{H}(\langle D, I\rangle)$ contenant son étiquette.

Il ne reste plus qu'à vérifier que pour chacun des hyperarcs $r = \langle p, t, P\rangle$, l'hyperarc $\langle \Pi'(p), \Pi'(t), \Pi'(P)\rangle$ est dans $\mathcal{H}(\langle D, I\rangle)$. 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 $\Pi'(P) = P'$. ■

3.3.3 Subsomption et projection

Étendons maintenant le théorème 1 à la subsomption de documents RDF.

Théorème 2 (Adéquation et complétude) *Soient G et G' deux documents RDF. Alors G subsume G' si et seulement si il existe une projection de $\mathcal{H}(G)$ dans $\mathcal{H}'(G', G)$.*

Preuve: Cette démonstration est très similaire à celle du Th. 1, nous prouverons successivement les deux sens de l'équivalence.

($\Rightarrow$) 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'. 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. 4). Voir que cette projection ne peut se faire que dans le sous-graphe $\mathcal{H}'(G, G')$ de $\mathcal{H}(\langle D, I\rangle)$.
($\Leftarrow$) Identique à la preuuve du Th. 1, par composition de projections.

■

4 Bénéfices de notre reformulation

Nous avons reformulé la subsomption entre documents RDF en une projection d'hypergraphes étiquetés. Examinons maintenant les bénéfices de ce travail. Nous montrons que ce problème est NP-complet, indiquons les rapports entre RDF, Graphes Conceptuels, Logique du premier ordre et CSP, et en retirons cas polynomiaux et algorithmes efficaces.

4.1 Complexité

Nous nous intéressons maintenant au problème de décision suivant :

RDF SUBSOMPTION
Données : Deux documents RDF G et G'.
Question : Est-ce-que G subsume G' ?

Propriété 5 RDF SUBSOMPTION *est un problème NP-complet.*

Preuve: 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 [15] (à partir de K-COLORATION [11]). ■

4.2 RDF et Graphes Conceptuels

Nous montrons que nos hypergraphes sont en fait des graphes conceptuels[2]. Bien que nous n'en rappelions pas les définitions, par manque de place, nous renvoyons le lecteur à [17] pour un exposé synthétique. A chacun de nos hypergraphes H (obtenus par la transformation $\mathcal{H}$ ou $\mathcal{H}'$ à partir de documents

[2]Bien que les Graphes Conceptuels soient habituellement définis comme des graphes bipartis, il s'agit bien des mêmes objets (c'est le biparti d'incidence de l'hypergraphe utilisé, par exemple, dans [1]).

RDF) nous associons un graphe conceptuel très similaire $GC(H)$ (il a la même structure, et des étiquettes similaires).

Le support sur lequel nos graphes conceptuels sont définis est le support trivial suivant :

- l'ensemble des *marqueurs individuels* est constitué des urirefs, des littéraux simples, des valeurs associées aux littéraux typés, ainsi que des littéraux typés eux-mêmes, s'il ne sont pas consistants[3] ;
- 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 sommets étiquetés par $\{\}$ sont maintenant étiquetés par $\top : *$ (ce sont des sommets génériques) ;
- les sommets étiquetés par $\{e\}$ sont maintenant étiquetés par $\top : e$ (ce sont des sommets individuels) ;
- les hyperarcs sont tous étiquetés par I_{EXT}.

La propriété suivante exprime que la projection de nos hypergraphes et la GC-projection de graphes conceptuels sont identiques sur notre support :

Propriété 6 *Soient G et G' deux documents RDF. Alors toute projection Π de $\mathcal{H}(G)$ dans $\mathcal{H}'(G', G)$ est une GC-projection de $\mathcal{G}(G) = GC(\mathcal{H}(G))$ dans $\mathcal{G}(G', G) = GC(\mathcal{H}'(G', G))$ (*i.e. *si $\Pi(x) = y$ dans la projection, x et y ont été transformés respectivement en x' et y' lors de la construction du graphe conceptuel, alors $\Pi(x') = y'$ dans la GC-projection).*

La démonstration est immédiate, mais elle nécessite de nombreuses définitions sur les graphes conceptuels que nous n'avons pas la place de développer ici.

Il s'ensuit tout naturellement (par application du théorème 2) que :

Corollaire 2 *Soient G et G' deux documents RDF. Alors G subsume G' si et seulement si il existe une GC-projection de $\mathcal{G}(G)$ dans $\mathcal{G}(G', G)$.*

Plusieurs auteurs (*e.g.* [4, 7]) ont relevé la similarité entre RDF et les graphes conceptuels. Notre transformation s'appuie sur la sémantique de RDF. Il est ainsi possible via cette transformation de calculer la subsomption en RDF en utilisant un moteur d'inférence de graphes conceptuels.

4.3 Une sémantique logique pour RDF en FOL$(\wedge, \exists)$

Une première conséquence du corollaire 2 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 [20] 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.

[3] Remarquons que, contrairement aux définitions habituelles, notre ensemble de marqueurs individuels est infini. Ceci ne pose aucun problème puisque nous n'utiliserons jamais plus que le sous-ensemble fini de marqueurs utilisés dans les graphes que nous comparons.

Théorème 3 ([6]) *Soient G et G' deux graphes conceptuels définis sur un support $\mathcal{S}$. Alors G se GC-projette dans $nf(G')$ (la forme normale de G') si et seulement si $\Phi(G)$ est conséquence logique de $\Phi(\mathcal{S})$ et de $\Phi(G')$.*

Nous en déduisons immédiatement, en appliquant le corollaire 2 que :

Corollaire 3 *Soient G et G' deux documents RDF. Alors G subsume G' si et seulement si $\Phi(G) = \Phi(\mathcal{G}(G))$ est conséquence logique de $\Phi(G', G) = \Phi(\mathcal{G}(G', G))$.*

Preuve: Par application directe du corollaire 2 et du théorème 3, en remarquant que la traduction par Φ de notre support trivial est un ensemble vide et que le graphe $\mathcal{G}(G', G)$ est déja sous forme normale (par construction du graphe RDF, puis par une remise sous forme normale dissimulée dans la transformation $\mathcal{H}$). ■

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

4.4 RDF et CSP

Un réseau de contraintes (CSP pour Constraint Satisfaction Problem) [16] 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.

[18] proposent une transformation $\mathcal{C}$, associant un CSP à une paire de graphes conceptuels et à un support, qui respecte la propriété suivante :

Propriété 7 *Soient G et G' deux graphes conceptuels définis sur un support $\mathcal{S}$. Alors G se projette dans G' si et seulement si $\mathcal{C}(G, G', \mathcal{S})$ est satisfiable.*

Appliquons de nouveau le corollaire 2, afin de pouvoir calculer la subsomption en RDF en utilisant un solver de contraintes.

Corollaire 4 *Soient G et G' deux documents RDF. Alors G subsume G' si et seulement si $\mathcal{C}(G, G') = \mathcal{C}(\mathcal{G}(G), \mathcal{G}'(G', G), \emptyset)$ est satisfiable.*

4.5 Cas polynomiaux

[9] 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 [18] 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. Nous en déduisons la propriété suivante :

Propriété 8 *Soit $\mathcal{P}$ une propriété structurelle des hypergraphes telle que la retriction du problème CSP aux hypergraphes respectant la propriété $\mathcal{P}$ admet un algorithme polynomial. Alors il existe un algorithme $\mathcal{A}_P$ tel que, pour G, G' deux documents RDF, si $\mathcal{H}(G)$ respecte $\mathcal{P}$, alors $\mathcal{A}_P$ décide en temps polynomial si G subsume G'.*

4.6 Algorithmes

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 [1].

Ce travail peut être adapté de façon immédiate à la projection d'hypergraphes...

4.7 RDF + Règles ?

Un mécanisme de règles a été proposé comme une extension des graphes conceptuels [19]. Le rapport que nous avons établi entre la subsomption RDF et la projection de graphes conceptuels devrait permettre d'adapter ce mécanisme, afin d'en faire une extension de RDF. Cette approche peut être intéressante puisque de nombreux auteurs ont fait part de la nécessité d'avoir un language de règles pour le Web Sémantique.

Des mécanismes encore plus évolués, comme ceux étudiés dans [2, 3], utilisent des règles et des contraintes. Leur adaptation serait également possible.

5 Conclusion et perspectives

Nous avons présenté dans cet article une – longue – démonstration permettant de coder la subsomption en RDF par la projection de graphes conceptuels. Ceci nous permet d'obtenir de façon immédiate des codages permettant de décider de la subsomption RDF en utilisant un moteur de graphes conceptuels, de logique du premier ordre ou un solver de contraintes.

Nous avons également évoqué des pistes importantes permettant de définir des cas polynomiaux pour le problème RDF-SUBSOMPTION, de développer des algorithmes efficaces, et de développer une extension à base de règles du language.

Ce travail devra également être prolongé en adaptant notre théorème principal à la subsomption en RDFS [5].

Références

[1] Jean-François Baget. Simple Conceptual Graphs Revisited : Hypergraphs and Conjuctive Types for Efficient Projection Algorithms. In *Conceptual Structures for Knowledge Creation and Communication, Proc. of ICCS03*, pages 195 – 208. Springer, 2003.

[2] Jean-François Baget and Marie-Laure Mugnier. The $\mathcal{SG}$ Family : Extensions of Simple Conceptual Graphs. pages 205–210, Vol. 1, 2001.

[3] Jean-François Baget and Marie-Laure Mugnier. Extensions of Simple Conceptual Graphs : the Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research*, 16 :425–465, 2002.

[4] T. Berners-Lee. Conceptual Graphs and the Semantic Web, 2001. http ://www.w3.org/DesignIssues/CG.html.

[5] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0 : RDF Schema. W3C working draft, W3C, 2003. `http ://www.w3.org/TR/rdf-schema/`.

[6] Michel Chein and Marie-Laure Mugnier. Conceptual Graphs are also Graphs. In *Quatrièmes journées du Laboratoire d'Informatique de Paris-Nord*, pages 81–97, 1995. also available as a research report, RR-LIRMM 95-003.

[7] O. Corby, R. Dieng, and C. Hébert. A Conceptual Graph Model for W3C Resource Description Framework. In *Proc. of ICCS 2000*, 2000.

[8] D. C. Fallside. XML Schema Part 0 : Primer. W3C candidate recommendation 24 oct. 2000, W3C, 2003. `http ://www.w3.org/TR/2000/CR-xmlschema-0-20001024/`.

[9] G. Gottlob, N. Leone, and F. Scarcello. A Comparison of Structural CSP Decomposition Methods. In *Proc. of IJCAI'99*, pages 394–399, 1999.

[10] P. Hayes. RDF Model Theory. W3C working draft 5 sept 2003, W3C, 2003. `http ://www.w3.org/TR/rdf-mt/`.

[11] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[12] Daniel Kayser. *La représentation des connaissances.* Hermes, 1997.

[13] G. Klyne and J. J. Carroll. Resource Description Framework (RDF) : Concepts and Abstract Syntax. W3C working draft 10 oct. 2003, W3C, 2003. `http ://www.w3.org/TR/rdf-concepts/`.

[14] O. Lassila and R. R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C recommendation, W3C, 1999. `http ://www.w3.org/TR/REC-rdf-syntax`.

[15] L. A. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9 :115–116, 1973. English translation in *Problems of Information Transmission*, 9, pp 265–266.

[16] U. Montanari. Networks of Constraints : Fundamental Properties and Application to Picture Processing. *Information Sciences*, 7(2) :95–132, 1974.

[17] Marie-Laure Mugnier. Knowledge Representation and Reasonings Based on Graph Homomorphism. Number 1867 in Lecture Notes in Artificial Intelligence, pages 172–192. Springer, 2000.

[18] Marie-Laure Mugnier and Michel Chein. Représenter des connaissances et raisonner avec des graphes. 10(1) :7–56, 1996.

[19] Éric Salvat and Marie-Laure Mugnier. Sound and Complete Forward and Backward Chainings of Graphs Rules. Number 1115 in Lecture Notes in Artificial Intelligence, pages 248–262. Springer, 1996.

[20] John F. Sowa. *Conceptual Structures : Information Processing in Mind and Machine.* Addison-Wesley, Reading, MA, 1984.