RDF 1.1 Sémantique des graphes nommés

Michel Gagnon

Séminaire WEST

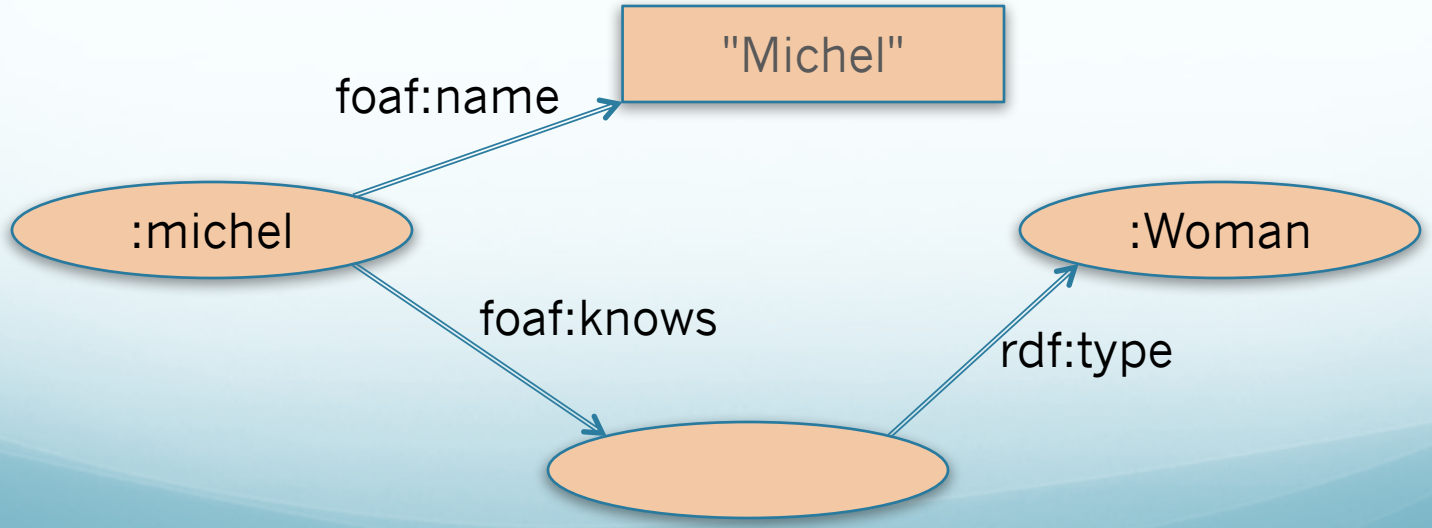
RDF 1.1

- Dans sa version originale, RDF ne prend en considération qu'un seul graphe
- Avec SPARQL est apparu la notion de graphes nommés
- Le W3C a ajouté cette notion dans la spécification de RDF 1.1:

  Un **document RDF** est un document qui encode un *graphe RDF* ou *un ensemble de données RDF* (RDF dataset), en utilisant une syntaxe RDF concrète (RDF/XML, Turtle, RDFa, JSON-LD, TRiG)

Graphe RDF

- Ensemble de triplets *(sujet, prédicat, objet)*
- Chaque nœud du graphe est une IRI, un litéral ou un nœud vide
- Exemple:



Ensemble de données RDF

- Il s'agit d'une collection de graphes RDF qui comprend:
  - Exactement un graphe par défaut, qui n'a pas de nom et qui peut être vide
  - Un ensemble possiblement vide de graphes nommés. Chaque graphe nommé est une paire dont le premier élément est une IRI ou un nœud vide (le nom) et le second est un graphe RDF. Les noms de graphe doivent être uniques dans un ensemble de données RDF
- Des nœuds vides peuvent être partagés par des graphes différents dans l'ensemble de données RDF

Ensemble de données RDF

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

# Graphe par défaut
<http://example.org/michel> dc:publisher :Michel .
<http://example.org/marie> dc:publisher :Marie .

<http://example.org/michel>
{
   [] foaf:name "Michel" ;
      foaf:knows _:b .
}

<http://example.org/marie>
{
    _:b foaf:name "Marie .
}
```

Sémantique des ensembles de données RDF

- Ce que dit le W3C:

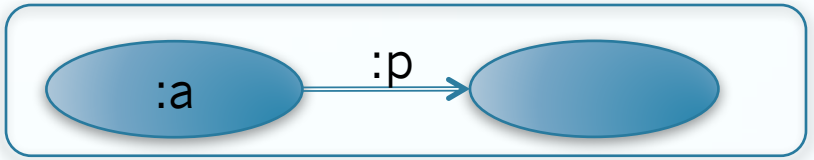
  RDF ne pose aucune restriction formelle sur la nature de la ressource dénotée par le nom d'un graphe nommé, ni sur la relation entre cette ressource et le graphe.

- Il faut donc que chaque application détermine la sémantique qu'elle utilise
- Le RDF Working Group propose plusieurs sémantiques
- Pour comprendre ces sémantiques, il est important de comprendre la différence entre union et fusion de graphes

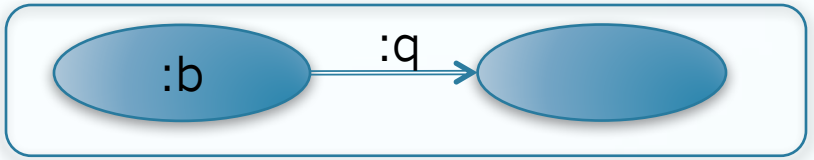
Union et fusion de graphes RDF

Soit les graphes RDF suivants:

G1: `:a :p _:n1 .`



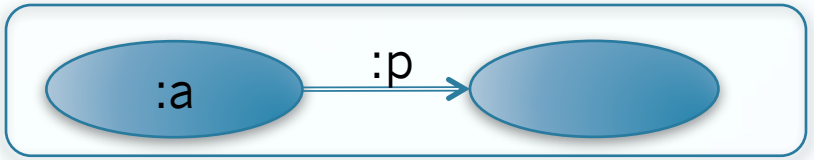
G2: `:b :q _:n1 .`



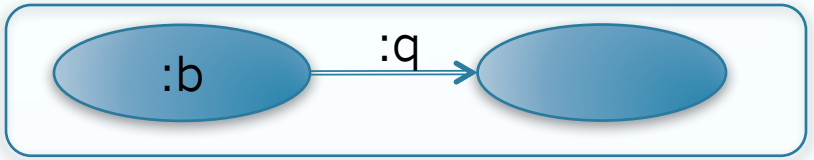
Union et fusion de graphes RDF

Soit les graphes RDF suivants:

G1: `:a :p _:n1 .`

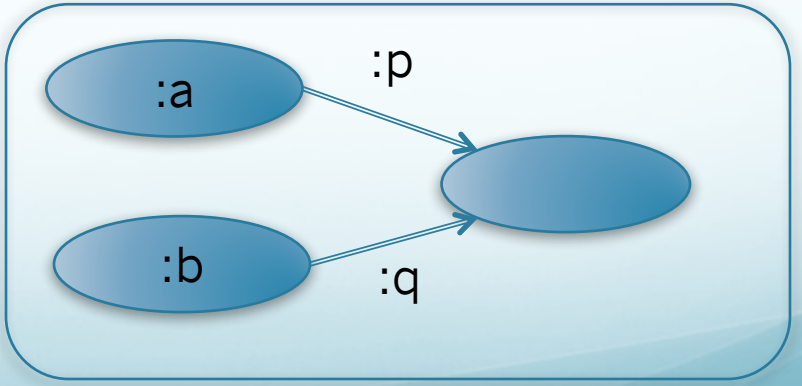


G2: `:b :q _:n1 .`



L'union donnera ceci:

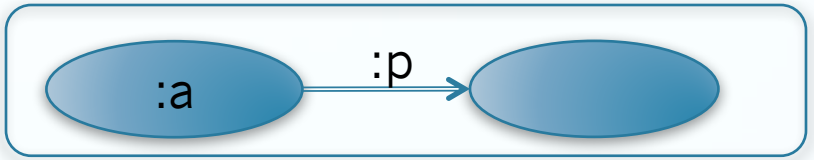
```
:a :p _:n1 .
:b :q _:n1 .
```



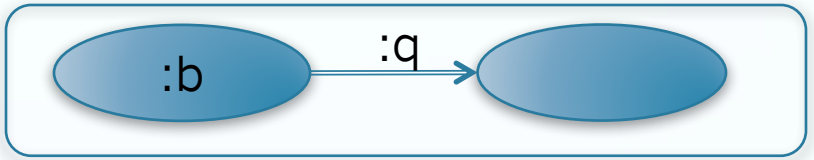
Union et fusion de graphes RDF

Soit les graphes RDF suivants:

```
G1:  :a :p _:n1 .
```

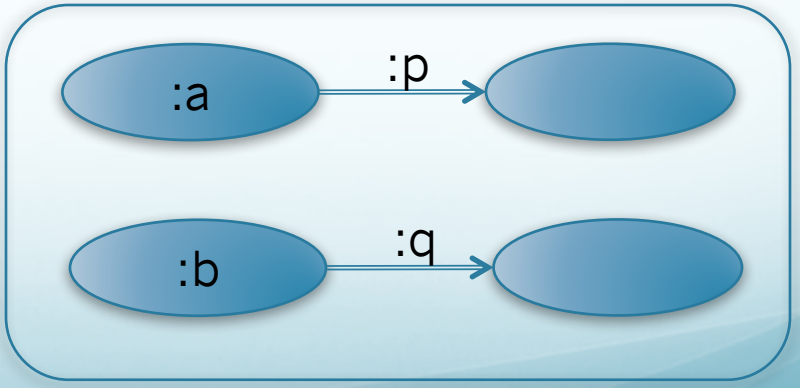



```
G2:  :b :q _:n1 .
```



La fusion donnera ceci:

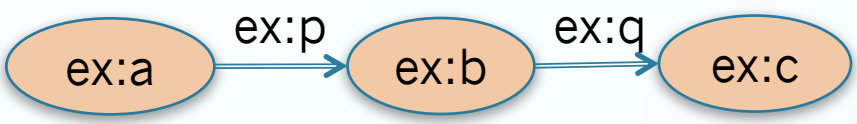
```
:a :p _:x1 .
:b :q _:x2 .
```



Sémantique d'un graphe RDF

Exemple:

```
{ ex:a ex:p ex:b .
  ex:b ex:q ex:c . }
```



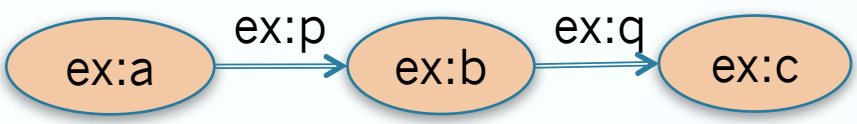
implique

```
{ ex:a ex:p ex:b . }
```

Sémantique d'un graphe RDF

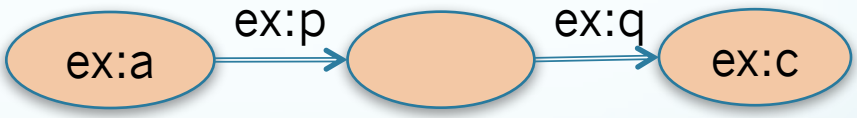
Exemple:

```
{ ex:a ex:p ex:b .
  ex:b ex:q ex:c . }
```



implique

```
{ ex:a ex:p _:n1 .
  _:n1 ex:q ex:c . }
```



Sémantique d'un graphe RDF

Exemple:

```
{ ex:a a ex:C .
  ex:C rdfs:subClassOf ex:D . }
```

implique

```
{ ex:a a ex:D . }
```

Maintenant, passons à la sémantique des ensembles de données RDF...

1 – Pas de sémantique particulière

- Seul le graphe par défaut est pris en compte pour l'interprétation
- Dans ce cas, les graphes nommés ont en quelque sorte un statut de *commentaire*
- Ceci implique que l'inférence ne tiendra pas compte des informations exprimées par les graphes nommés

1 – Pas de sémantique particulière

Exemple:

```
{ ex:a ex:p ex:b . }

ex:g1 { ex:a ex:q  ex:c . }
```

n'implique pas

```
{ ex:a ex:p ex:b ;
       ex:q ex:c . }
```

1 – Pas de sémantique particulière

Exemple:

```
{ ex:a a ex:C . }

ex:g1 { ex:C rdfs:subClassOf ex:D . }
```

n'implique pas

```
{ ex:a a ex:D. }
```

1 – Pas de sémantique particulière

Exemple:

```
{ ex:a ex:p ex:b . }

ex:g1 { ex:a ex:q ex:c . }
```

implique

```
{ }

ex:g2 { ex:x ex:y ex:z . }
```

1 – Pas de sémantique particulière

Exemple:

```
{ ex:g1 ex:auteur ex:michel . }

ex:g1 { ex:a ex:q ex:c . }
```

implique

```
{ ex:g1 ex:auteur ex:michel . }

ex:g1 { ex:x ex:y ex:z . }
```

2 – Sémantique d'union ou de fusion

- Dans ce cas, on considère que les graphes nommés forment tout simplement une partition des informations
- Il faut décider comment on traite les nœuds vides:
  - Si on accepte que des graphes se partagent les mêmes nœuds vides, la sémantique sera la sémantique de RDF appliquée à l'union des graphes
  - Si la portée des nœuds vides se limite à chacun des graphes, on utilisera la sémantique de la fusion des graphes
- Aucune spécification sur la manière d'interpréter les noms

de fusion

Exemple:

```
{ ex:a ex:p ex:b . }

ex:g1 { ex:x ex:q ex:y . }
```

implique

```
{ ex:a ex:p ex:b .
  ex:x ex:q ex:y . }
```

de fusion

Le graphe suivant est inconsistant:

```
{ }

ex:g1 { ex:age rdfs:range xsd:integer . }

ex:g2 { ex:jean ex:age "vingt" . }
```

2 – Sémantique d'union

Exemple:

```
{ ex:a ex:p _:n1 . }

ex:g1 { ex:x ex:q _:n1 . }
```

implique

```
{ ex:a ex:p _:n1 .
  ex:x ex:q _:n1 . }
```

2 – Sémantique de fusion

Exemple:

```
{ ex:a ex:p ex:_n1 . }

ex:g1 { ex:x ex:q _:n1 . }
```

n'implique pas

```
{ ex:a ex:p _:n1 .
  ex:x ex:q _:n1 . }
```

2 – Sémantique de fusion

Exemple:

```
{ ex:a ex:p ex:_n1 . }

ex:g1 { ex:x ex:q _:n1 . }
```

implique

```
{ ex:a ex:p _:n1 .
  ex:x ex:q _:n2 . }
```

3 – Le graphe nommé dénote le graphe lui-même

- L'interprétation d'un nom *n* est la paire *(n,ng)* où *ng* est le graphe nommé
- Le graphe nommé est considéré selon sa forme et non son sens
- Les nœuds vides sont considérés comme des variables existentielles
- Cette sémantique est appropriée si on veut fournir des méta-données sur des graphes
- Peut être utilisé pour représenter une citation

3 – Le graphe nommé dénote le graphe lui-même

Exemple:

```
{ }
ex:g1 { ex:a ex:p ex:b . }
ex:g2 { ex:c ex:q ex:d . }
```

implique

```
{ }
_:n1 { ex:a ex:p ex:b . }
ex:g2 { ex:c ex:q ex:d . }
```

3 – Le graphe nommé dénote le graphe lui-même

Exemple:

```
{ }
ex:g1 { ex:a ex:p ex:b . }
ex:g2 { ex:c ex:q ex:d . }
```

n'implique pas

```
{ }
ex:g1 { [] ex:p ex:b . }
ex:g2 { ex:c ex:q ex:d . }
```

3 – Le graphe nommé dénote le graphe lui-même

Exemple:

```
{ }
ex:g1 { ex:a ex:p ex:b . }
ex:g2 { ex:c ex:q ex:d . }
```

n'implique pas

```
{ }
ex:g1 { }
ex:g2 { ex:c ex:q ex:d . }
```

3 – Le graphe nommé dénote le graphe lui-même

Le graphe suivant est inconsistant:

```
{ ex:age rdfs:range xsd:integer .
  ex:jean ex:age ex:g1}

ex:g1 { ex:jean ex:age "20"^^xsd:integer . }
```

3 – Le graphe nommé dénote le graphe lui-même

Utilisation pour fournir des méta-données:

```
{ex:g1 dc:author ex:michel ;
       dc:date "2010-07-23"^^xsd:date .}

ex:g1 { ex:jean ex:age "20"^^xsd:integer . }
```

3 – Le graphe nommé dénote le graphe lui-même

Imbrication plus complexe:

```
{ex:g1 dc:author ex:michel ;
       dc:date "2010-07-23"^^xsd:date .}

ex:g1 { ex:jean ex:age "20"^^xsd:integer .
        ex:g2 dc:author ex:marie . }

ex:g2 { ex:paul foaf:knows ex:ana . }
```

3 – Le graphe nommé dénote le graphe lui-même

- Dans une variante, le nom ne désigne pas la paire *(n,ng)* mais le graphe *ng* lui-même
- Dans ce cas, on ne pourra pas distinguer deux occurrences identiques d'un même graphe

4 – Chaque graphe nommé définit son propre contexte

- Ici, ce n'est pas le graphe en tant que tel qu'on veut prendre en compte, mais plutôt sa dénotation
- En d'autres mots, chaque graphe nommé est interprété séparément
- Ceci permet que les graphes nommés soient inconsistants entre eux (au raisonneur de se débrouiller avec cela!)
- Soit un graphe nommé *(n,ng)*. L'interprétation de *n* est un graphe RDF qui implique le graphe *ng*

4 – Chaque graphe nommé définit son propre contexte

Exemple:

```
{ }
ex:g1 { ex:Chien rdfs:subClassOf ex:Animal .
        ex:fido a ex:Chien .}
ex:g2 { ex:rex a  ex:Chien . }
```

implique

```
{ }
ex:g1 { ex:fido a  ex:Animal . }
```

4 – Chaque graphe nommé définit son propre contexte

Exemple:

```
{ }
ex:g1 { ex:Chien rdfs:siubClassOf ex:Animal .
        ex:fido a ex:Chien .}
ex:g2 { ex:rex a  ex:Chien . }
```

n'implique pas

```
{ }
ex:g1 { ex:rex a  ex:Animal . }
```

4 – Chaque graphe nommé définit son propre contexte

- Variantes:
  - Le graphe par défaut représente la vérité universelle (ses triplets doivent être vrais pour tous les graphes nommés)
  - Le nom ne représente pas un graphe RDF mais plutôt la ressource associée à un graphe RDF
  - Le type d'interprétation et d'inférence utilisé pourrait être différent pour chaque graphe nommé

Conclusion

- Plusieurs sémantiques possibles pour les graphes nommés
- La sémantique choisie dépend de l'application
- Il faudra définir une manière de déclarer la sémantique utilisée
- Essentiellement, il faut décider:
  - si les graphes nommés doivent être pris individuellement ou non
  - si les nœuds vides sont partagés ou non par les graphes nommés
  - si les graphes nommés doivent être pris littéralement ou si on considère plutôt leur dénotation