

Graphe de connaissances : raisonnement et apprentissage

Adrien Coulet, équipe HeKA

Inria Paris &

Centre de Recherche des Cordeliers (Inserm, Univ. Paris Cité, Sorbonne Univ.)





2 façons de concevoir
le monde

2 façons d'acquérir des
connaissances

Plan

1. Définitions

- a) Distinction entre données, information et connaissances
- b) Définitions d'un graphes de connaissances
- c) Les types de graphes de connaissances

2. Les graphes RDF

- a) Syntaxe
- b) Raisonnement
- c) Requête (SPARQL)

3. L'apprentissage sur les graphes de connaissances

- a) Approches classiques
- b) GCN

Des données aux connaissances

- **Données** : ensemble de signes et symboles
- **Informations** : données + sens (ou sémantique)
- **Connaissances** : information assimilée et interprétée + possibilité de mise en action de l'information interprétée

- "Dobrý den, pane. Kolik je hodin?"
- "Je přesně 08:32."

- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."

- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."



- "J'ai un rendez vous prévu à 8h30. Je suis en retard. Il faut que je cours."



- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."

```
1: situation= "à l'heure",  
2: vitesse_transport = "marche",  
3: heure_rdv = 8h30,  
4: heure_actuelle = 8h32,  
5: si heure_actuelle > heure_rdv alors situation = "en retard",  
6: si situation = "retard"  
7: alors vitesse_transport = "cours"  
8: et parole="J'ai un rendez vous prévu à "+heure_rdv+". Je suis "+situation+". Il  
faut que je "+vitesse_transport+"."
```

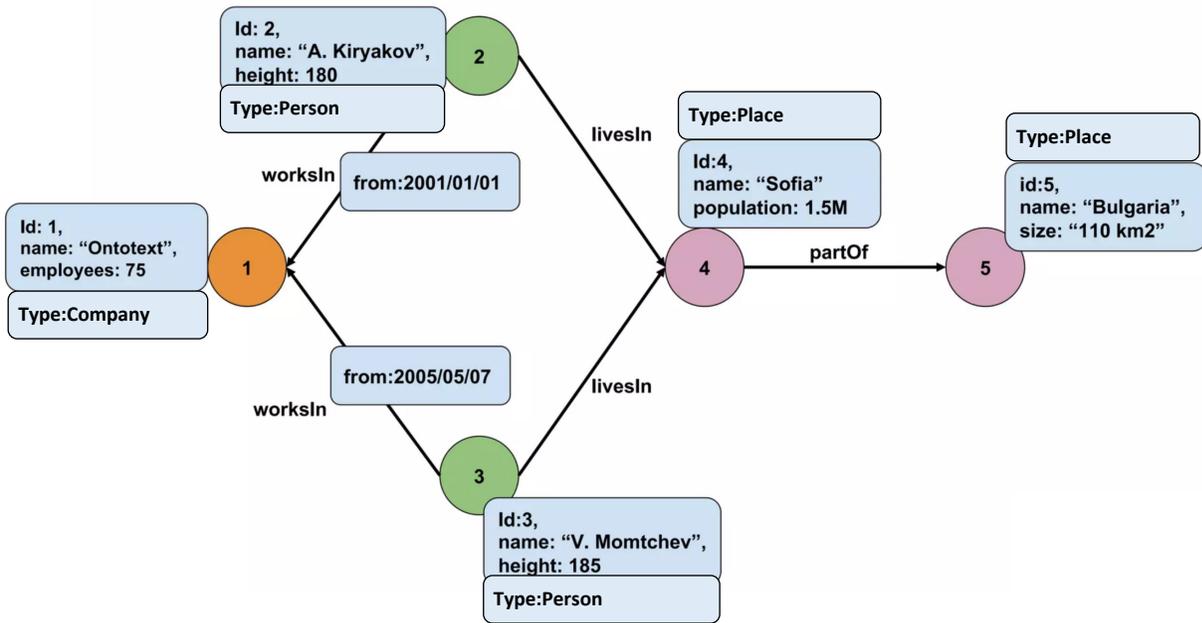
- "J'ai un rendez vous prévu à 8h30. Je suis en retard. Il faut que je cours."



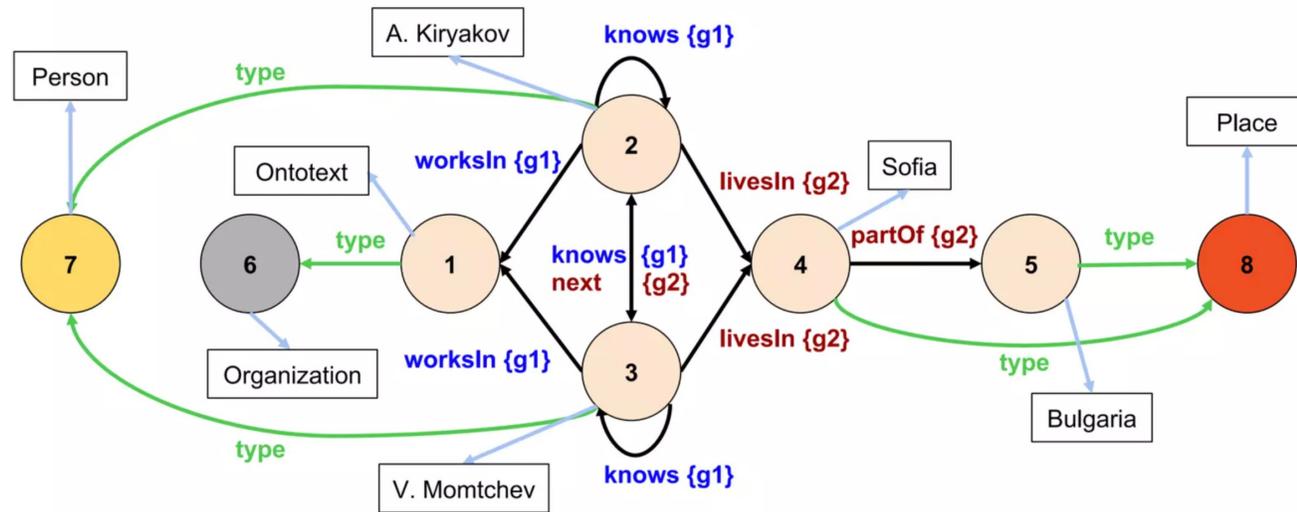
Graphe de connaissances : définitions

- Définition générale :
 - « Un graphe avec des connaissances » (Paulheim, 2017)
- Définition consensuelle :
 - « Un graphe qui cherche à représenter des connaissances et dont les nœuds représentent des entités du monde et les arcs des relations entre ces entités » (Hogan *et al.*, 2021)
- Définition Web sémantique :
 - « Un graphe qui intègre des connaissances diverses dans l'objectif de pouvoir en dériver de nouvelles » (Ehrlinger and Wöß, 2016)
- Différents types de graphes de connaissances
 - Les graphes à propriétés ou graphes attribués (*Property graphs*)
 - Les graphes RDF
 - D'autres : les hypergraphes, ...

Graphe à propriétés vs. Graphe RDF

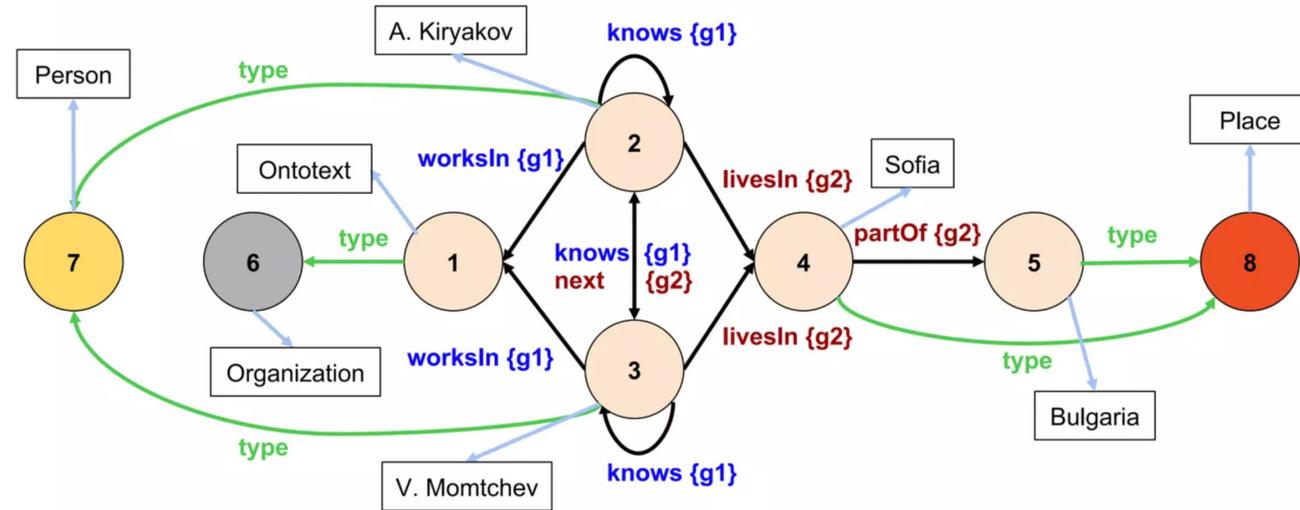
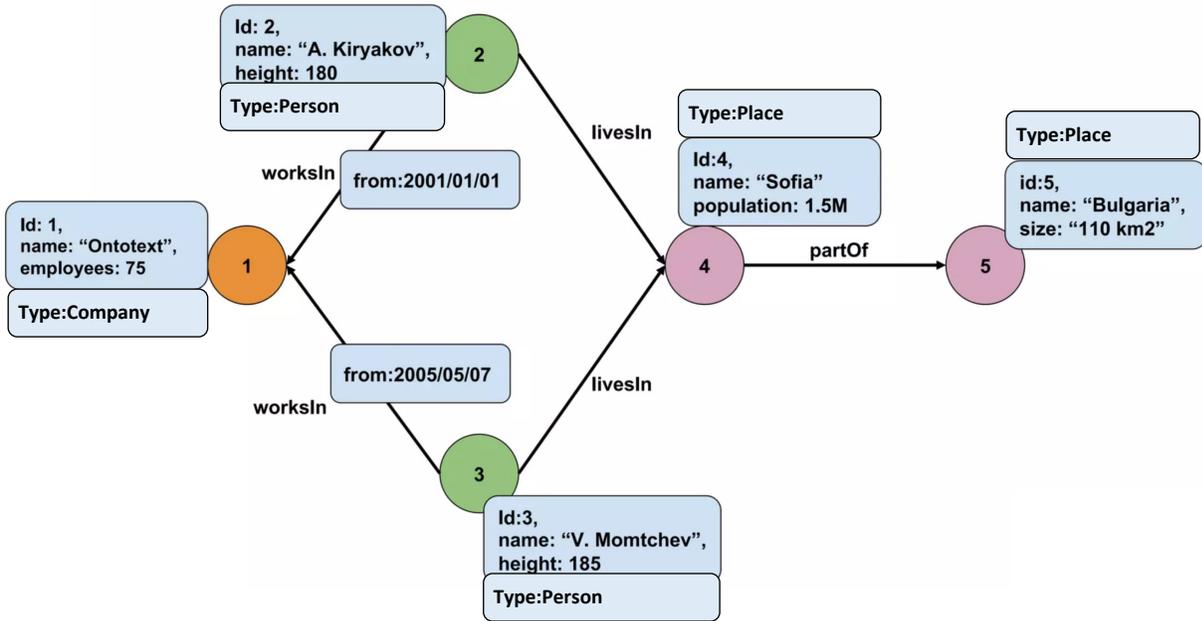


Graphe à propriétés :
Multi-graphe dirigé, étiqueté, cyclique, typé



Graphe RDF :
Multi-graphe dirigé, étiqueté, cyclique, typé

Graphe à propriétés vs. Graphe RDF



Graphe à propriétés :

Multi-graphe dirigé, étiqueté, cyclique, typé

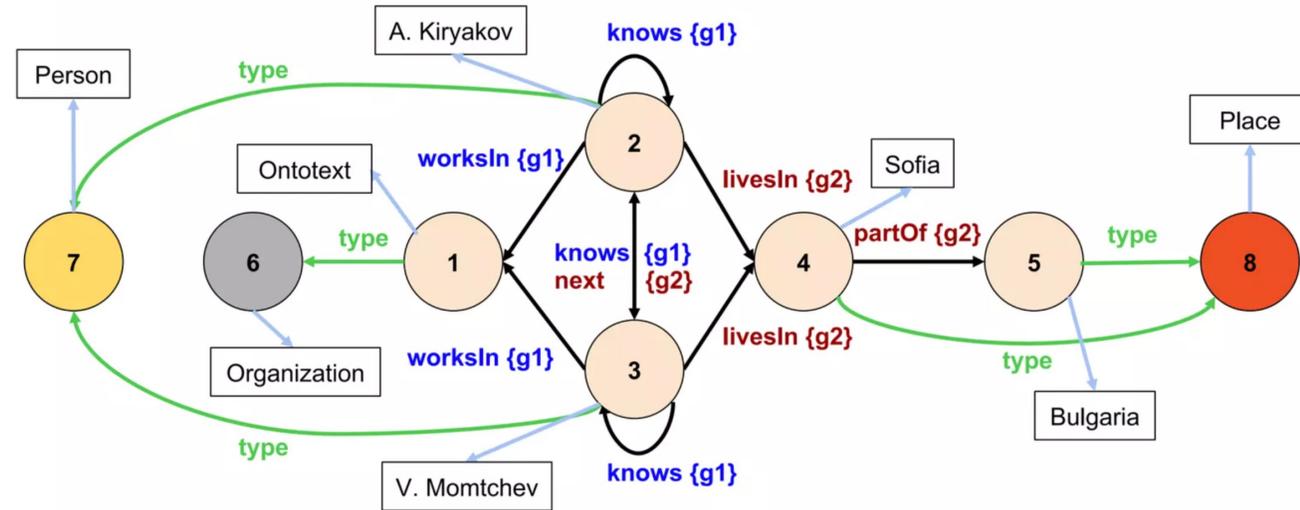
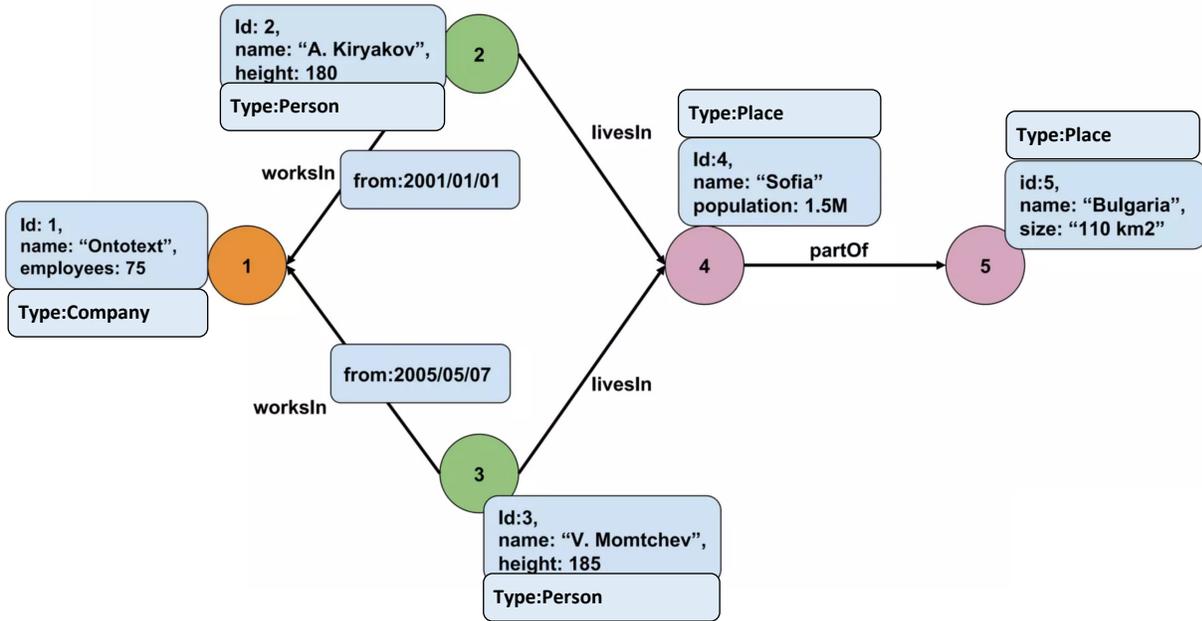
- Nœuds et arcs sont associés à des ensemble de propriétés *i.e.*, des paires (clé,valeur)
- Pas de standard particulier pour les noms de nœuds, arcs et leurs propriétés
- Pas de langage standard d'interrogation(Cypher, GQL, etc.)
- Pas de langages de schéma ou de sémantique formelle associée

Graphe RDF :

Multi-graphe dirigé, étiqueté, cyclique, typé

- Tout est représenté sous forme de nœuds et d'arcs
- Standards de nommage par le W3C
- Langage standard d'interrogation: SPARQL
- Langages de schéma (RDFS, OWL) et sémantique formelle associée, autorisant le raisonnement

Graphe à propriétés vs. Graphe RDF



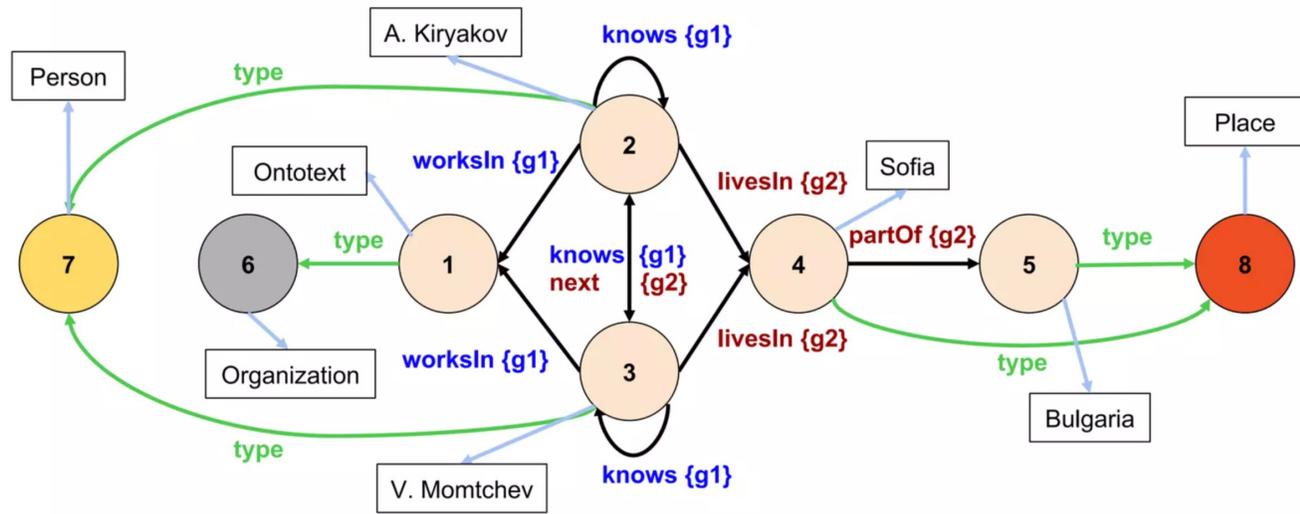
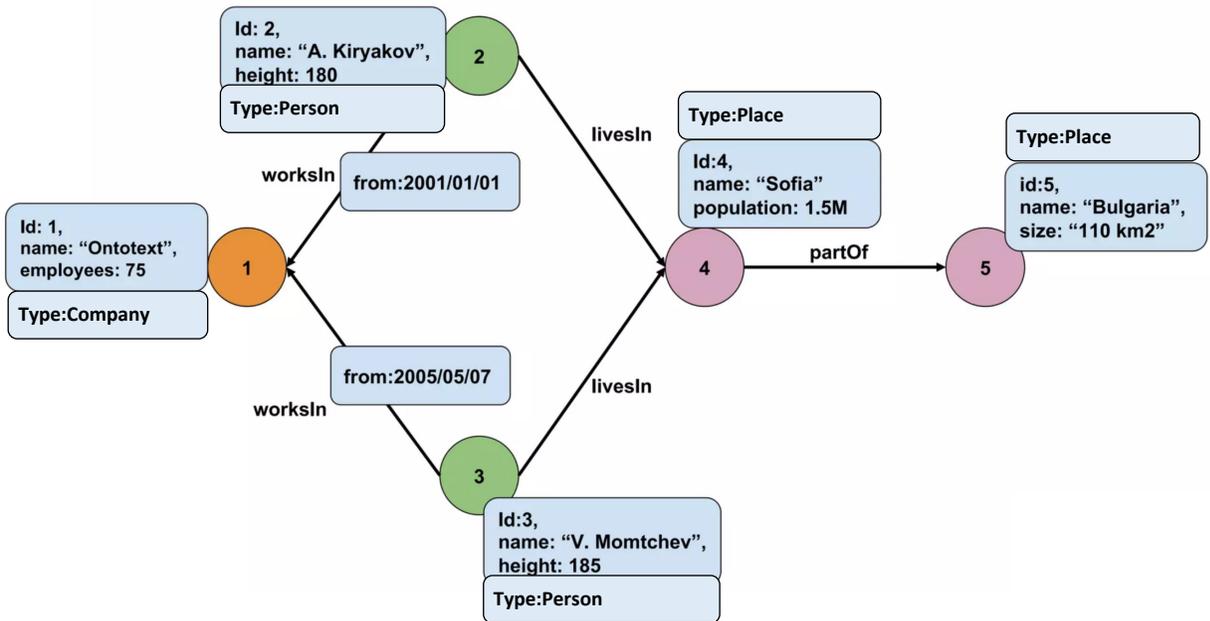
Graphe à propriétés :

- Modèle plus simple, plus compact
- Adapté à l'analyse de données
- Passe plus facilement à l'échelle de très grands graphes

Graphe RDF :

- Modèle plus expressif et associé à des standards
- Adapté à la représentation de connaissances, à l'échange et l'interconnection de données, à l'intégration de données

Graphe à propriétés vs. Graphe RDF



Outils

- Neo4j
- ArangoDB
- TigerGraph
- Oracle Property Graph
- ...

- Virtuoso, OpenLink
- AllegroGraph
- GraphDB, OntoText
- Oracle RDF
- ...

- Amazone Neptune
- ...

Graphe à propriétés vs. Graphe RDF

Exemples d'applications

Complétion de réseau d'interaction protein-protein

Zitnik M, et al, 2019

Repositionnement de médicaments

Gysi DM, et al, 2021

Classification automatique de types cellulaire (*single cell*)

Wang S, et al, 2021

Explication des mécanismes
des traitements

Ruiz C, et al. 2021

Explication des mécanismes
des effets secondaires

Bresso E, et al, 2021

...

Identification de pharmacogènes

Dalleau K, et al. 2017

RDF / RDFS

Un peu d'histoire...

Un peu d'histoire...

- RDF = Ressource Description Framework
- RDF (1990) < XML (1996)
- RDF (1990) pour représenter des méta données des pages Web (auteurs, copyright)
- RDF (2004) pour le Web sémantique

RDF / RDFS

Représentation des graphes RDF

Les briques de base du RDF

- 3 types de nœuds
 - Individu
 - Classe
 - Valeur

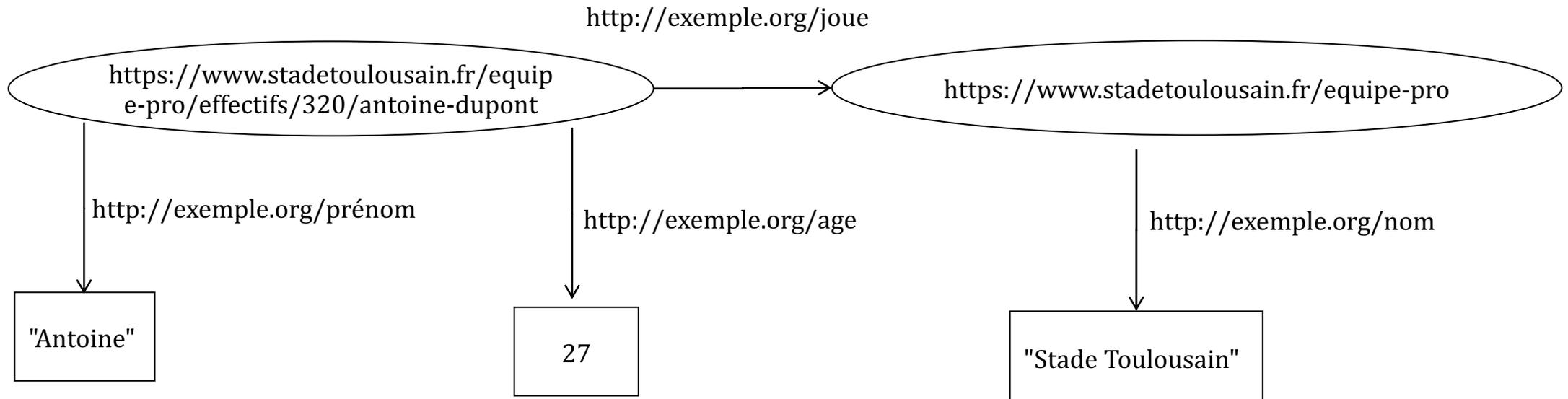
- Relations

individu

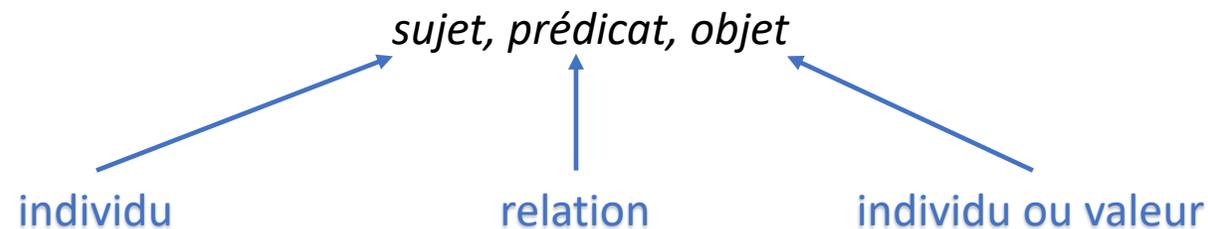
Classe

42





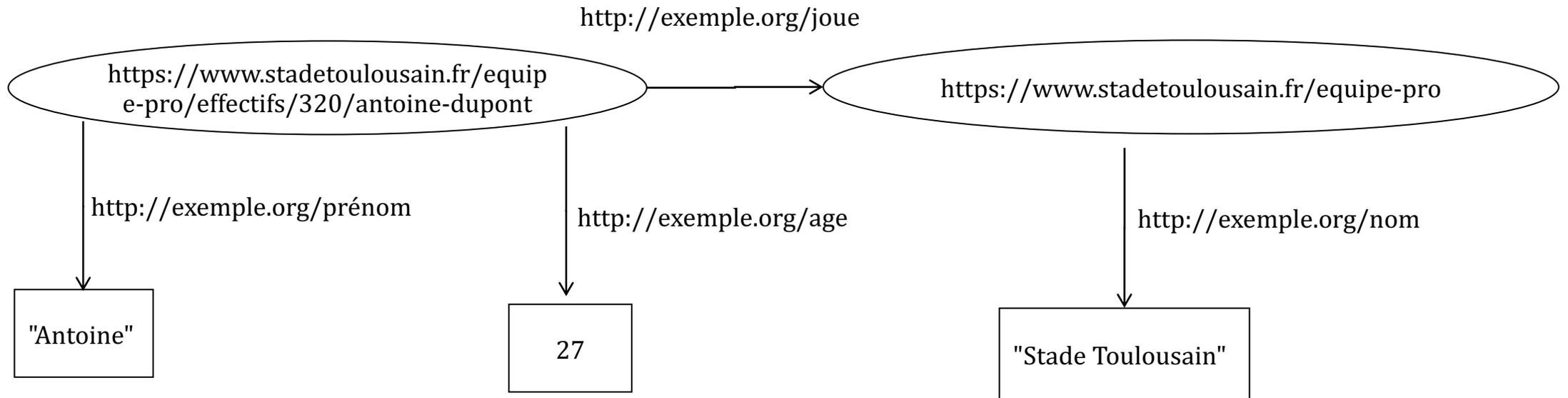
- Unité de base du RDF, le triplet :



- Représente des relations binaires
- Les individus et relations sont identifiés par des URI (de la forme des URL)
- Une valeur ne peut jamais être sujet d'un triplet

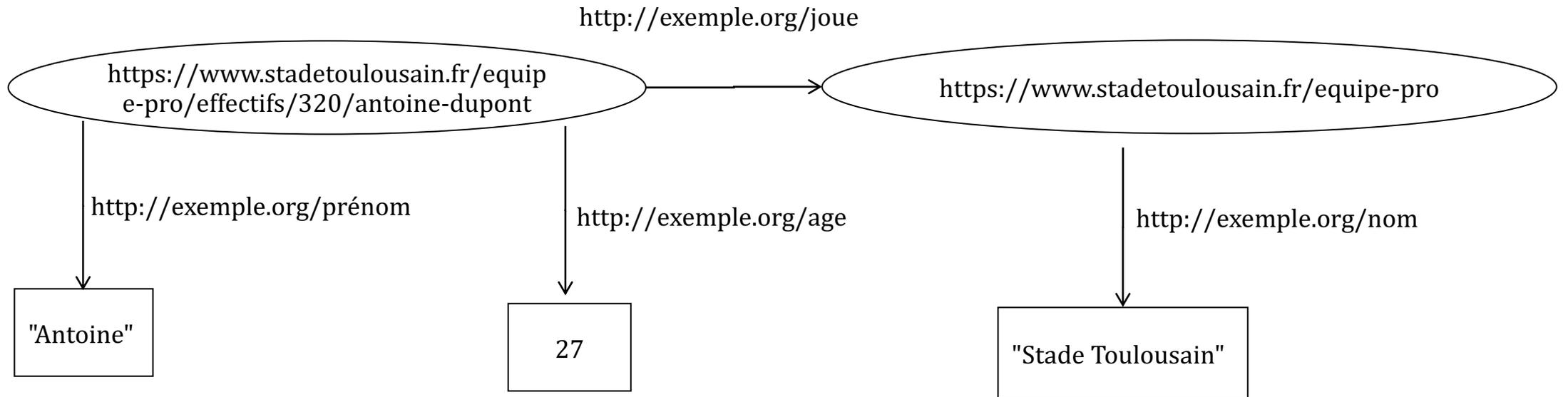
Syntaxe RDF / RDFS

Sérialisation des graphes RDF : le format *Turtle*



Sérialisation du RDF : le format *Turtle*

```
<https://www.stadetoulousain.fr/equipe-pro/effectifs/320/antoine-dupont> <http://exemple.org/joue> <https://www.stadetoulousain.fr/equipe-pro> .
<https://www.stadetoulousain.fr/> <http://exemple.org/age> "27" .
<https://www.stadetoulousain.fr/> <http://exemple.org/nom> "Stade Toulousain" .
```



Sérialisation du RDF : le format *Turtle*

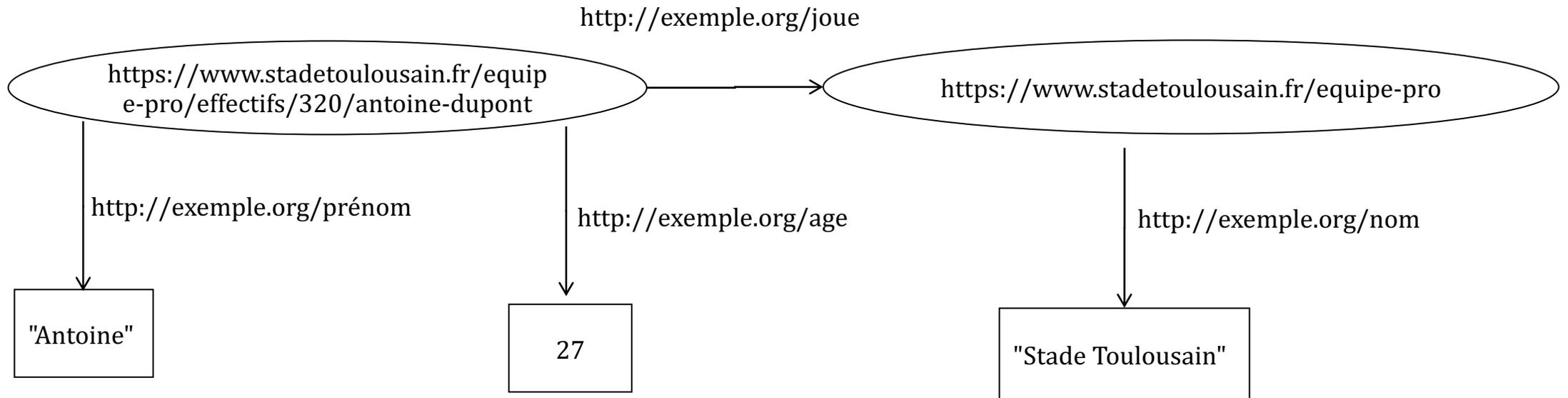
Les préfixes (racines d'URI)

```

@prefix st: <https://www.stadetoulouse.fr/> .
@prefix ex: <http://exemple.org/> .
@prefix eff: <https://www.stadetoulouse.fr/equipe-pro/effectifs/320/> .

st:equipe-pro          ex:nom          "Stade Toulousain".
eff:antoine-dupont    ex:joue          st:equipe-pro .
eff:antoine-dupont    ex:age          "27" .

```



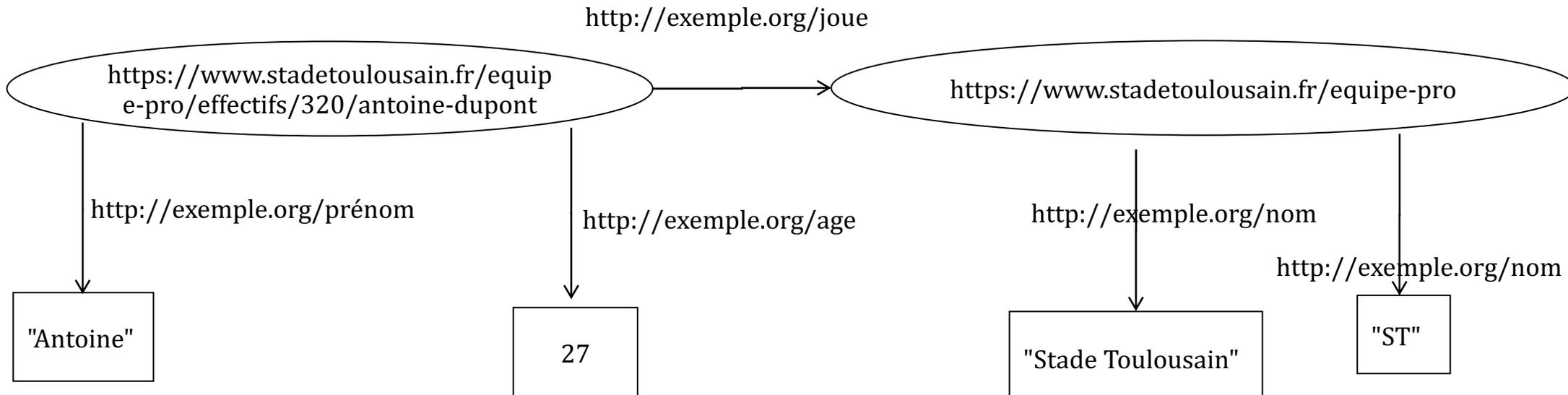
Sérialisation du RDF : le format *Turtle*

Les préfixes (racines d'URI)

```

@prefix st: <https://www.stadetoulousain.fr/> .
@prefix ex: <http://exemple.org/> .
@prefix eff: <https://www.stadetoulousain.fr/equipe-pro/effectifs/320/> .

st:equipe-pro          ex:nom          "Stade Toulousain".
eff:antoine-dupont    ex:joue          st:equipe-pro ; ← ; permet d'éviter
                                                             de répéter le sujet
ex:age                "27" .
  
```



Sérialisation du RDF : le format *Turtle*

Les préfixes (racines d'URI)

@prefix st: <https://www.stadetoulousain.fr/> .

@prefix ex: <http://exemple.org/> .

@prefix eff: <https://www.stadetoulousain.fr/equipe-pro/effectifs/320/> .

st:equipe-pro

ex:nom

"Stade Toulousain" ,

"ST"

eff:antoine-dupont

ex:joue

st:equipe-pro ;

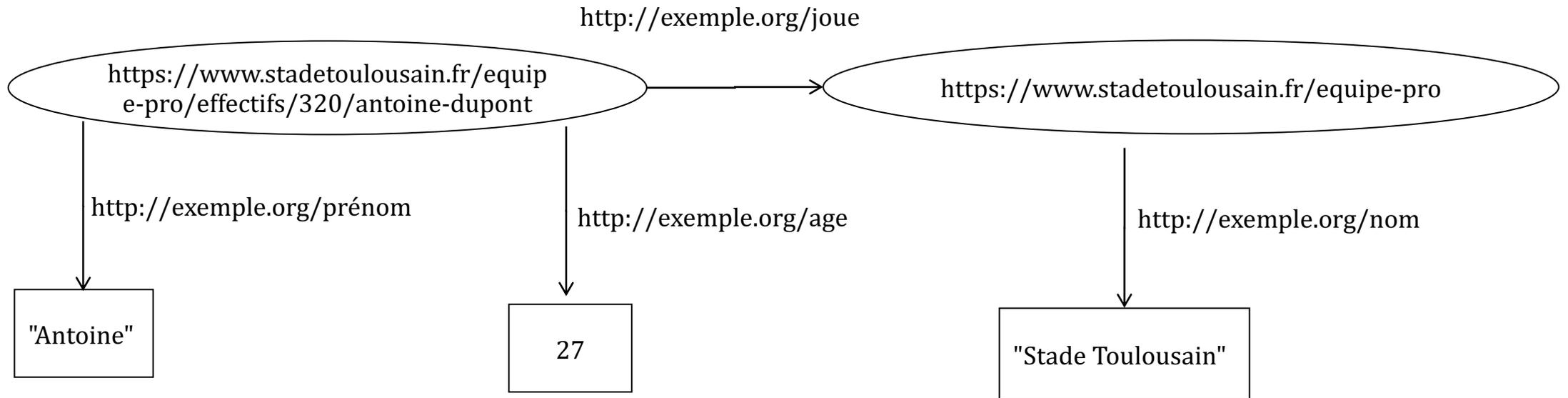
ex:age

"27" .

, permet d'éviter de répéter le sujet et le prédicat

Syntaxe RDF / RDFS

Sérialisation des graphes RDF : le format *RDF/XML*

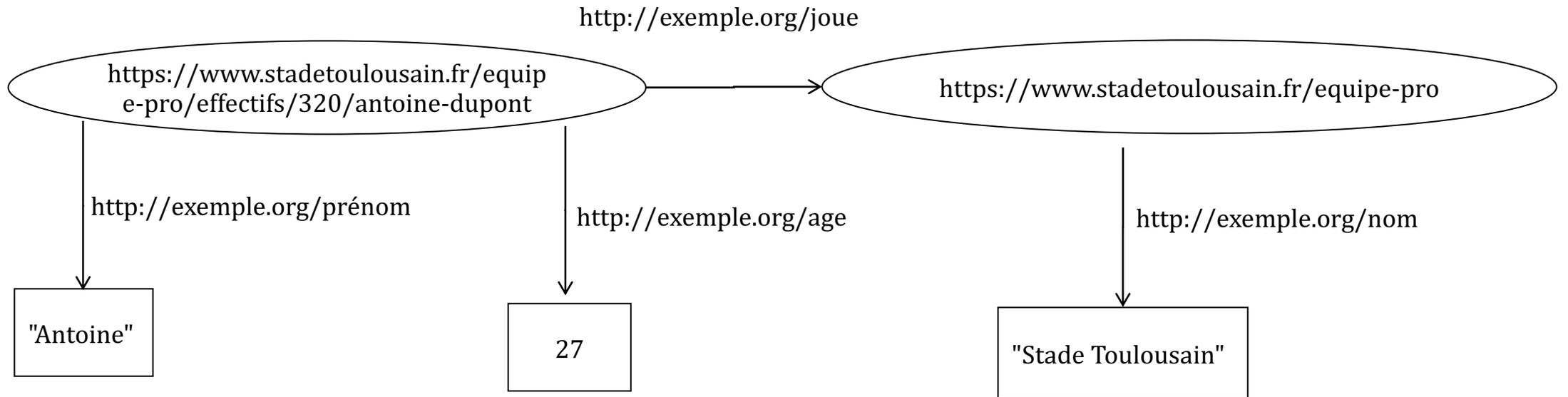


Sérialisation du RDF : le format *RDF/XML*

```

< ?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://exemple.org/ ">

  <rdf:Description rdf:about="https://www.stadetoulousain.fr/equipe-pro/effectifs/320/antoine-dupont">
    <ex:joue>
      <rdf:Description rdf:about="https://www.stadetoulousain.fr/equipe-pro"/>
    </ex:joue>
    <ex:prenom>Antoine</ex:prenom>
  </rdf:Description>
</rdf:RDF>
  
```



Sérialisation du RDF : le format *RDF/XML*

```

< ?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://exemple.org/ ">

  <rdf:Description rdf:about="https://www.stadetoulousain.fr/equipe-pro/effectifs/320/antoine-dupont"
    ex:prenom= "Antoine">
    <ex:joue>
      <rdf:Description rdf:about="https://www.stadetoulousain.fr/equipe-pro"/>
    </ex joue>
  </rdf:Description>
</rdf:RDF>
  
```

Syntaxe RDF / RDFS

Quelques exemples supplémentaires

Exemple 1 : les relations n -aires

@prefix ex: <http://exemple.org/> .

ex:Chutney ex:contientIngrédient "300g de mangue verte";

ex:contientIngrédient "1càc de piment de Cayenne".

Exemple 1 : les relations *n*-aires

```
@prefix ex:      <http://exemple.org/> .

ex:Chutney      ex:contientIngrédient  ex:mangueVerte;
                ex:quantité            "300g";
                ex:contientIngrédient  ex:pimentDeCayenne;
                ex:quantité            "1càc".
```

Exemple 1 : les relations *n*-aires

```
@prefix ex:      <http://exemple.org/> .  
  
ex:Chutney      ex:contientIngrédient  ex:mangueVerte;  
                ex:quantité           "300g";  
                ex:contientIngrédient  ex:pimentDeCayenne;  
                ex:quantité           "1càc".
```

Quelle quantité pour quel ingrédient ?

Exemple 1 : les relations *n*-aires

```
@prefix ex:                <http://exemple.org/> .

ex:Chutney                 ex:contientIngrédient  ex:mangueVerte;
                           ex:contientIngrédient  ex:pimentDeCayenne.
ex:mangueVerte             ex:quantité           "300g" .
ex:pimentDeCayenne         ex:quantité           "1càc" .
```

Exemple 1 : les relations *n*-aires

@prefix ex:	<http://exemple.org/> .	
ex:Chutney	ex:contientIngrédient	ex:mangueVerte;
	ex:contientIngrédient	ex:pimentDeCayenne.
ex:mangueVerte	ex:quantité	"300g".
ex:pimentDeCayenne	ex:quantité	"1càc".
ex:SaladeDeMangue	ex:contientIngrédient	ex:mangueVerte;
	ex:contientIngrédient	ex:banane.
ex:mangueVerte	ex:quantité	"100g".
ex:banane	ex:quantité	"100g".

Exemple 1 : les relations *n*-aires

@prefix ex:	<http://exemple.org/> .	
ex:Chutney	ex:contientIngrédient	ex:mangueVerte;
	ex:contientIngrédient	ex:pimentDeCayenne.
ex:mangueVerte	ex:quantité	"300g".
ex:pimentDeCayenne	ex:quantité	"1càc".
ex:SaladeDeMangue	ex:contientIngrédient	ex:mangueVerte;
	ex:contientIngrédient	ex:banane.
ex:mangueVerte	ex:quantité	"100g".
ex:banane	ex:quantité	"100g".

Quelle quantité pour quelle plat ?

Exemple 1 : les relations *n*-aires

@prefix ex:	<http://exemple.org/> .	
ex:Chutney	ex:contientIngrédient	ex:ingrédient1;
	ex:contientIngrédient	ex:ingrédient2.
ex:ingrédient1	ex:ingrédient	ex:mangueVerte;
	ex:quantité	"300g".
ex:ingrédient2	ex:ingrédient	ex:pimentDeCayenne;
	ex:quantité	"1càc".

Ce processus de modélisation s'appelle la **réification**

Exemple 1 : les relations *n*-aires

- Exemple en *RDF/XML*

```
<rdf:Description rdf:about="http://exemple.org/Chetnay">
  <ex:contientIngrédient rdf:nodeID="ID1"/>
</rdf:Description>
<rdf:Description rdf:nodeID="ID1">
  <ex:ingrédient rdf:resource="http://exemple.org/mangueVerte"/>
  <ex:quantité>300g<ex:quantité/>
</rdf:Description>
```

- Exemple en *Turtle*

@prefix ex:	<http://exemple.org/> .	
ex:Chutney	ex:contientIngrédient	_:id1.
_:id1	ex:ingrédient	ex:mangueVerte;
	ex:quantité	"300g".

Syntaxe RDF / RDFS

RDFS

RDF/RDFS

- Instance (*RDF*)

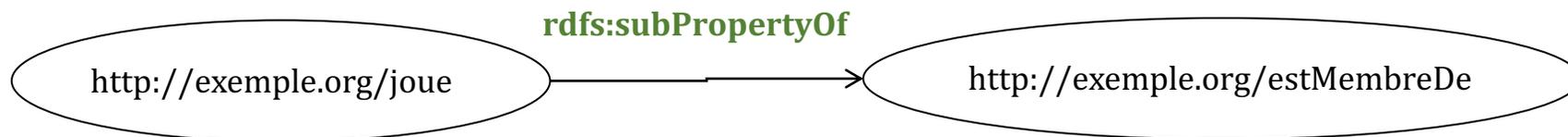
ex:antoine	rdf:type	ex:demi_de_melee.
<i>ou</i>		
ex:antoine	a	ex:demi_de_melee.

- Hiérarchie de classes (*RDFS*)

ex:demi_de_melee	rdf:type	rdfs:Class .
ex:demi_de_melee	rdfs:subClassOf	ex:joueur .

- Hiérarchies de propriétés (*RDFS*)

- rdfs:subPropertyOf



rdfs:Domain & rdfs:Range

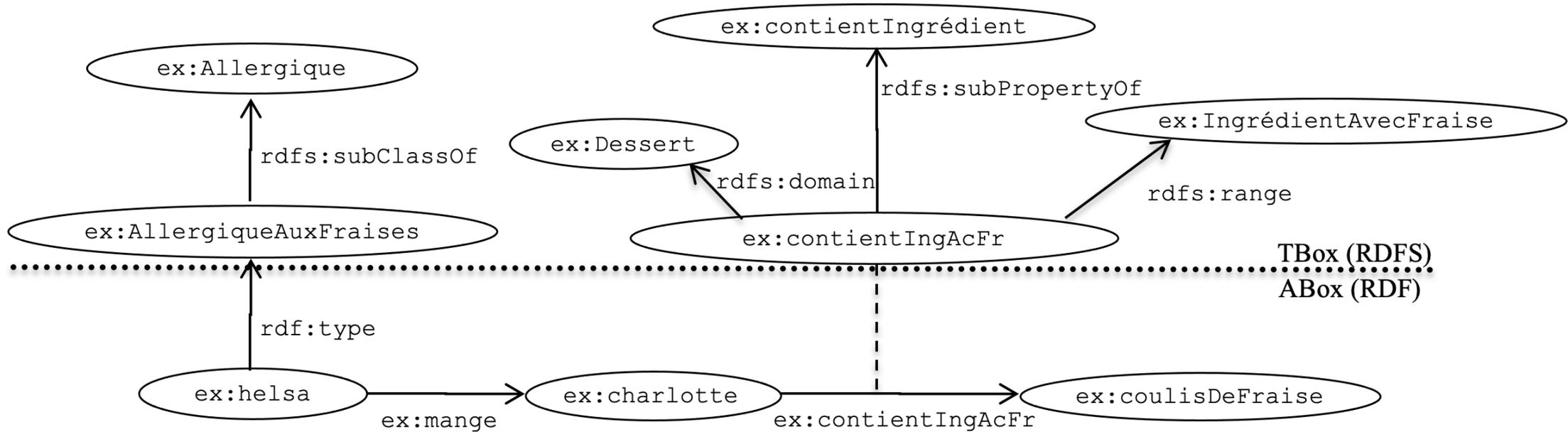
- Domaine et co-domaine d'une propriété

ex:estMariéA	rdfs:domain	ex:Personne.
ex:estMariéA	rdfs:range	ex:Personne.

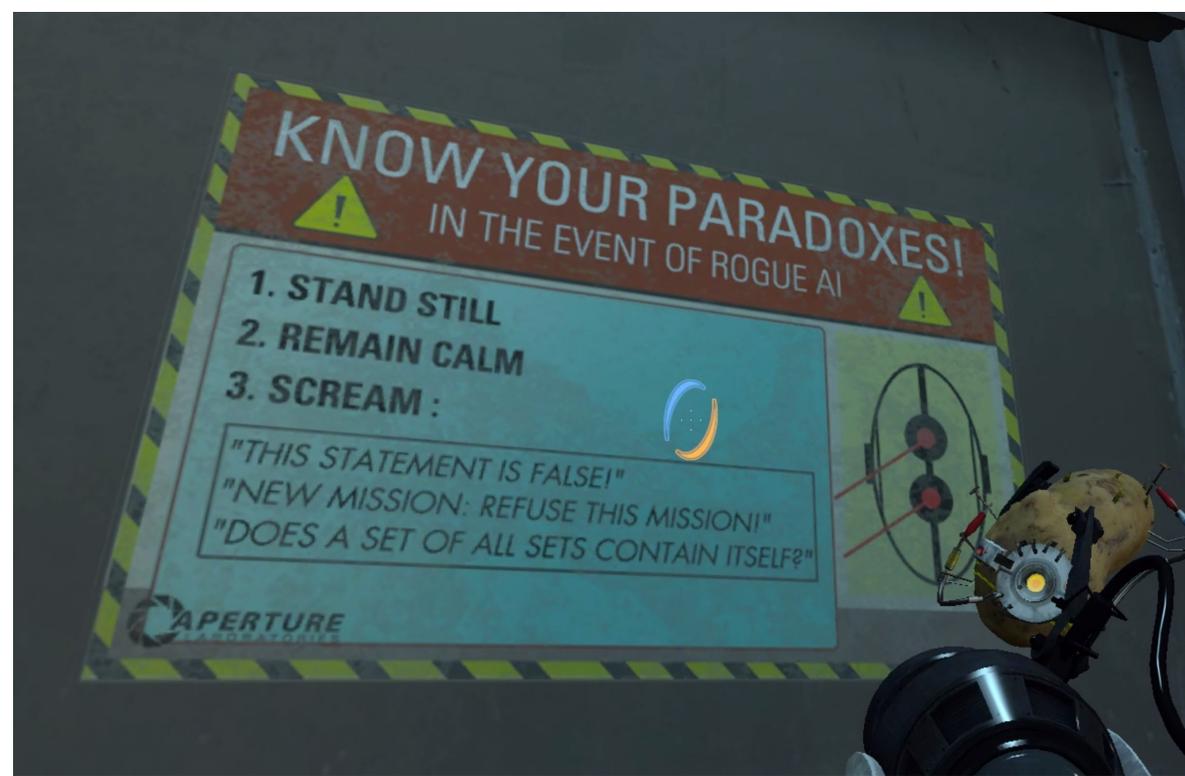
- Peut-on ajouter ... ?

ex:ferme	rdf:type	ex:Bâtiment.
ex:josé	ex:estMariéA	ex:ferme.

Exemple



G_1



Raisonnement avec RDF / RDFS

Raisonnement avec RDFS

Les règles de déduction

$$\frac{r \text{ rdfs:domain } x . \quad u \text{ } r \text{ } y .}{u \text{ rdf:type } x .}$$

La même règle de déduction existe pour l'objet de *rdfs:range*.

$$\frac{x1 \text{ rdfs:subClassOf } x2 . \quad u \text{ rdf:type } x1 .}{u \text{ rdf:type } x2 .}$$

$$\frac{u \text{ rdfs:subClassOf } v . \quad v \text{ rdfs:subClassOf } x .}{u \text{ rdfs:subClassOf } x .}$$

Le même type de règle est défini pour *rdfs:subPropertyOf*

Raisonnement avec RDFS

Les règles de déduction (suite)

$$\frac{r \text{ rdfs:range } x . \quad u \text{ } r \text{ } y .}{y \text{ rdf:type } x .}$$

$$\frac{u \text{ rdfs:subPropertyOf } v . \quad v \text{ rdfs:subPropertyOf } x .}{u \text{ rdfs:subPropertyOf } x .}$$

$$\frac{r1 \text{ rdfs:subPropertyOf } r2 . \quad u \text{ } r1 \text{ } y .}{u \text{ } r2 \text{ } y .}$$

Raisonnement avec RDF

- Règles de déduction supplémentaires

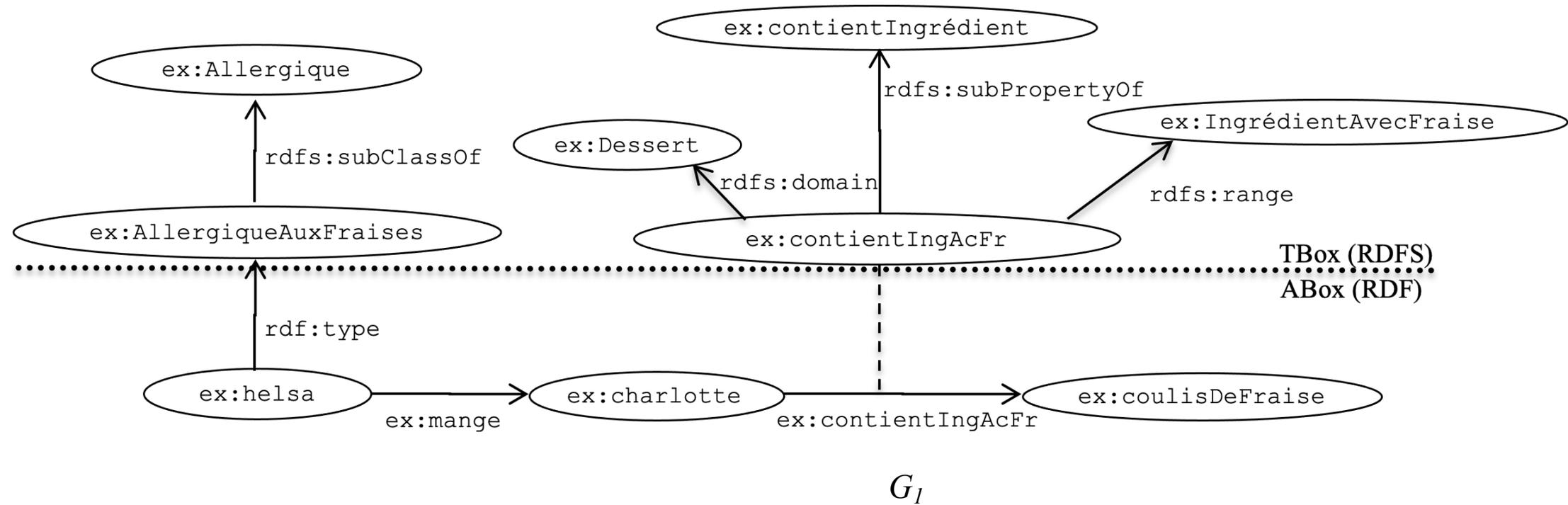
Exemples :

$$\frac{u \ r \ x \ .}{u \ r \ _ :n \ .}$$
$$\frac{u \ r \ x \ .}{_ :n \ r \ x \ .}$$
$$\frac{\text{ex:helsa ex:mange ex:charlotte} \ .}{\text{ex:helsa ex:mange} \ _ :id1 \ .}$$
$$\frac{\text{ex:charlotte ex:contientIngAcFr ex:coulisDeFraise} \ .}{_ :id1 \ \text{ex:contientIngAcFr ex:coulisDeFraise} \ .}$$

"SI Helsa mange la charlotte
ALORS Helsa mange **quelque chose**"

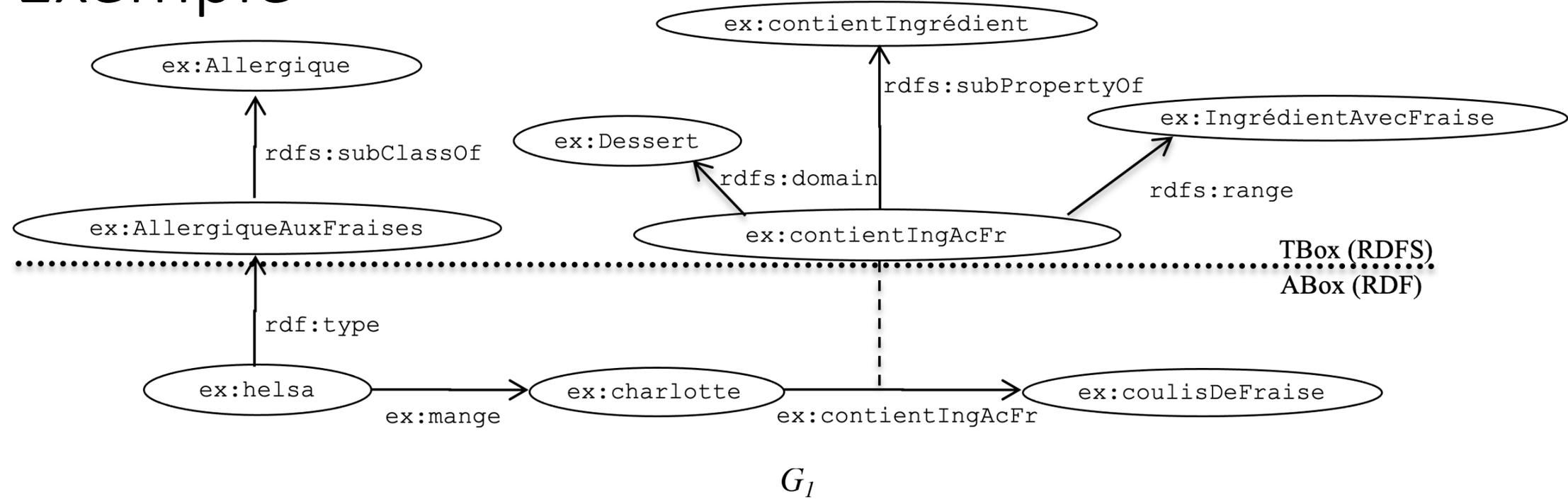
"SI une charlotte contient un ingrédient
avec de la fraise qui est un coulis de fraise
ALORS **quelque chose** contient un
ingrédient avec de la fraise qui est un
coulis de fraise"

Exemple

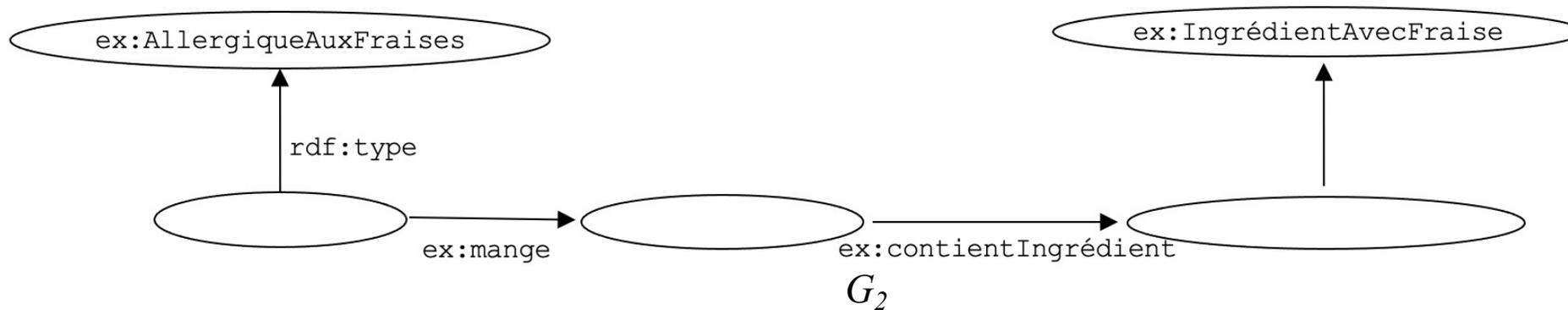


Quelqu'un qui est allergique à la fraise va t-il manger un plat contenant des ingrédients à base de fraise ?

Exemple

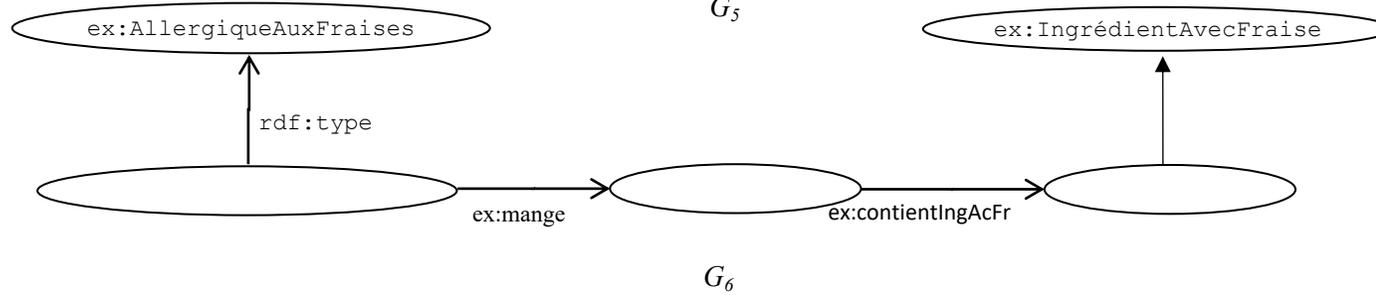
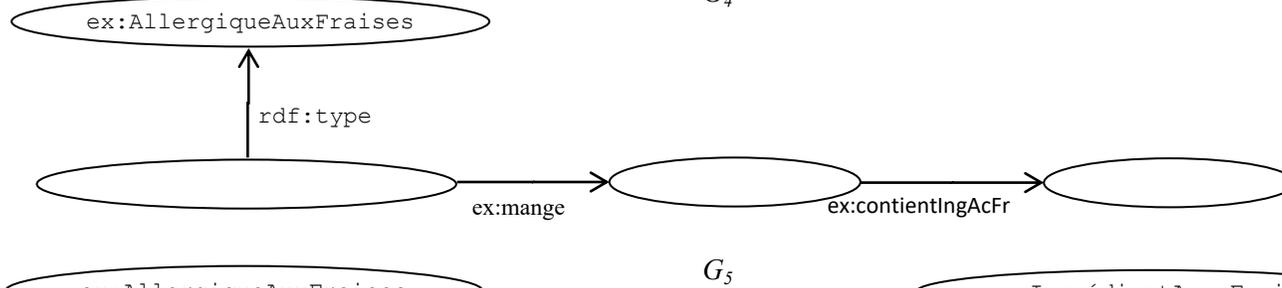
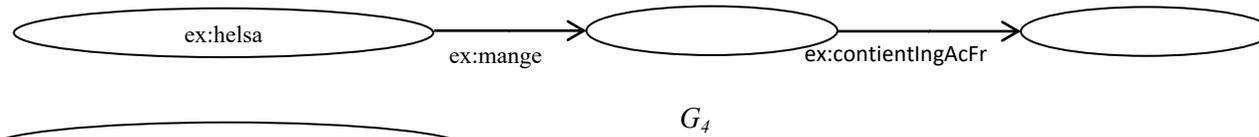
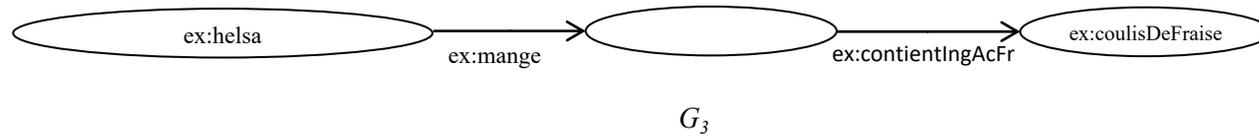
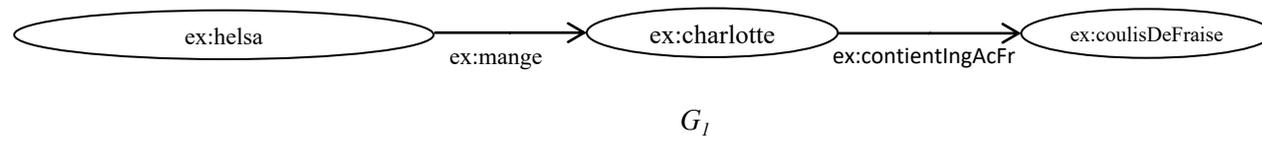


Est ce que quelqu'un qui est allergique aux fraises va manger un plat contenant des ingrédients à base de fraise ?



$G_1 \models G_2$?

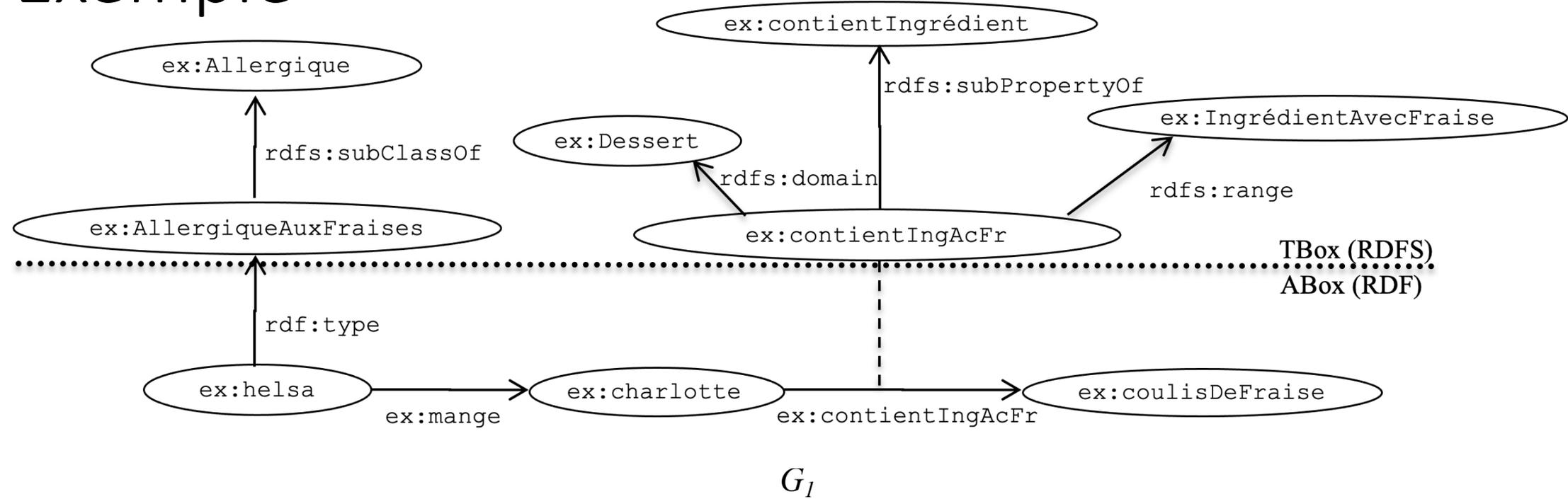
Exemple



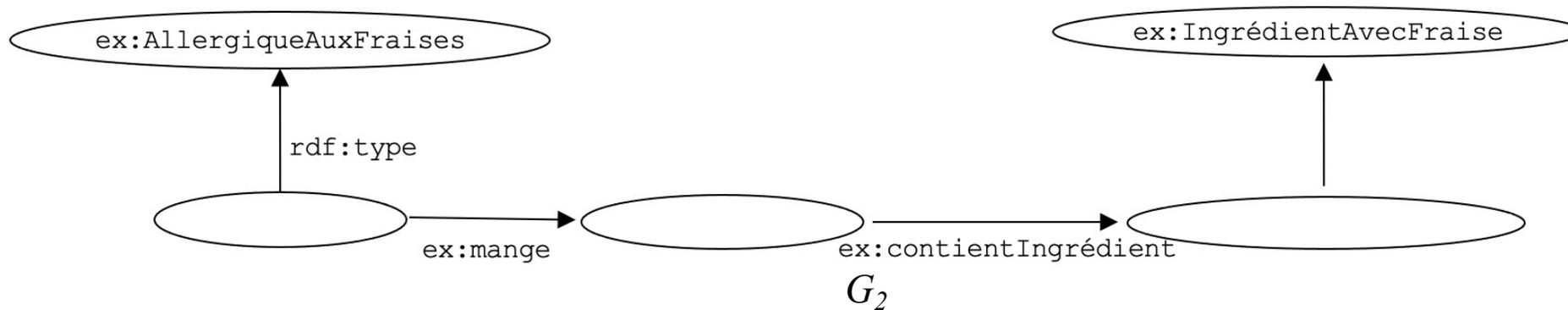
...

$$G_1 \models G_3, G_3 \models G_4, G_4 \models G_5, \dots$$

Exemple



Est ce que quelqu'un qui est allergique aux fraises va manger un plat contenant des ingrédients à base de fraise ?



$G_1 \models G_2$?

Interroger un graphe de
connaissances RDF: SPARQL

SPARQL = "SPARQL Protocol and RDF Query Language"

Structure d'une requête SPARQL

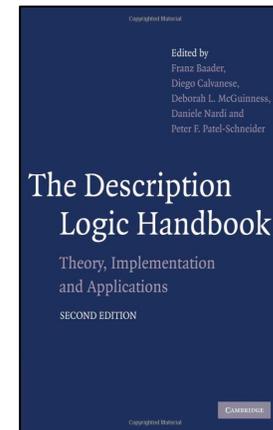
```
SELECT ?sujet ?objet  
WHERE  
{  
  ?sujet ?predicat ?objet.  
}
```

Utilisation de préfixes

```
PREFIX ex: <http://exemple.org/>
SELECT ?predicat ?objet
WHERE
{
    ex:ma_constante ?predicat ?objet.
}
```

Exemple

@prefix ex:	<http://exemple.org/> .	
ex:dlhb	ex:publiéPar	<http://www.cambridge.org/> ;
	ex:titre	"The Description Logic Handbook" ;
	ex:auteur	ex:fBaader ,
		ex:dCalvanese ;
	ex:auteur	ex:dMcGuinness.
ex:om	ex:publiéPar	<http://www.springer.com/>
	ex:titre	"Ontology Matching" ;
	ex:auteur	ex:jEuzenat ;
	ex:auteur	ex:pShvaiko.



Exemple

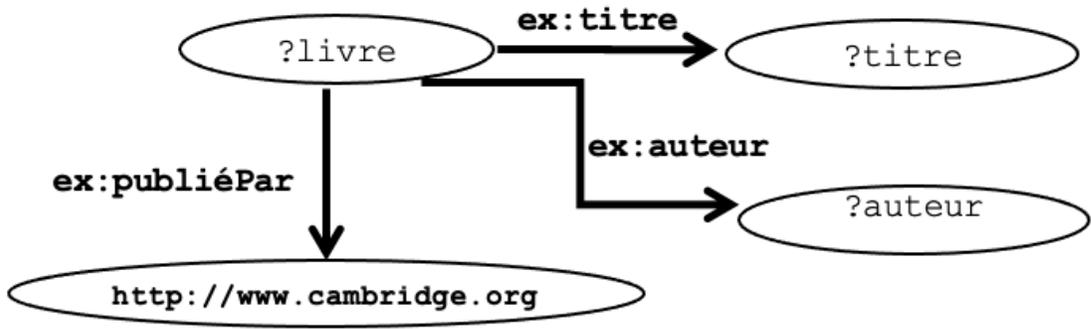
@prefix ex:	<http://exemple.org/> .	
ex:dlhb	ex:publiéPar	<http://www.cambridge.org/> ;
	ex:titre	"The Description Logic Handbook" ;
	ex:auteur	ex:fBaader ,
		ex:dCalvanese ;
	ex:auteur	ex:dMcGuinness.
ex:om	ex:publiéPar	<http://www.springer.com/>
	ex:titre	"Ontology Matching" ;
	ex:auteur	ex:jEuzenat ;
	ex:auteur	ex:pShvaiko.

"Quels sont les titres et noms d'auteurs des livres publiés par Cambridge Press ?"

Exemple

```
@prefix ex: <http://exemple.org/> .  
ex:dlhb                ex:publiéPar      <http://www.cambridge.org/> ;  
                       ex:titre        "The Description Logic Handbook" ;  
                       ex:auteur       ex:fBaader ,  
                                       ex:dCalvanese ;  
ex:om                  ex:auteur       ex:dMcGuinness.  
                       ex:publiéPar    <http://www.springer.com/>  
                       ex:titre        "Ontology Matching" ;  
                       ex:auteur       ex:jEuzenat ;  
                                       ex:pShvaiko.
```

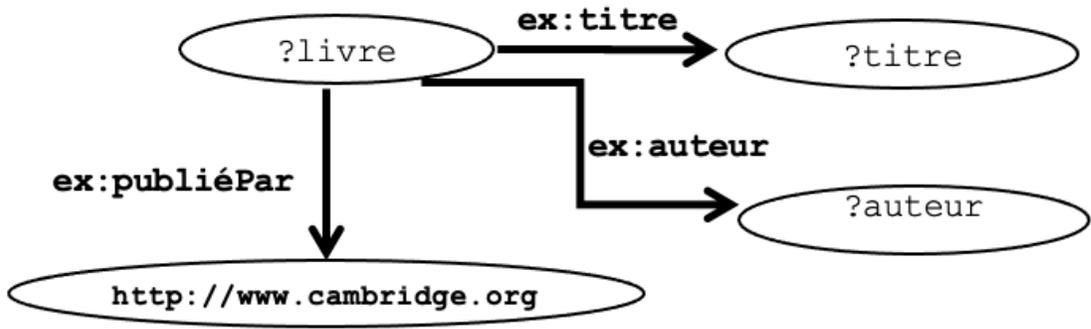
"Quels sont les titres et noms d'auteurs des livres publiés par Cambridge Press ?"



Exemple

@prefix ex: <http://exemple.org/> .		
ex:dlhb	ex:publiéPar	<http://www.cambridge.org/> ;
	ex:titre	"The Description Logic Handbook" ;
	ex:auteur	ex:fBaader ,
		ex:dCalvanese ;
	ex:auteur	ex:dMcGuinness.
ex:om	ex:publiéPar	<http://www.springer.com/>
	ex:titre	"Ontology Matching" ;
	ex:auteur	ex:jEuzenat ;
	ex:auteur	ex:pShvaiko.

"Quels sont les titres et noms d'auteurs des livres publiés par Cambridge Press ?"



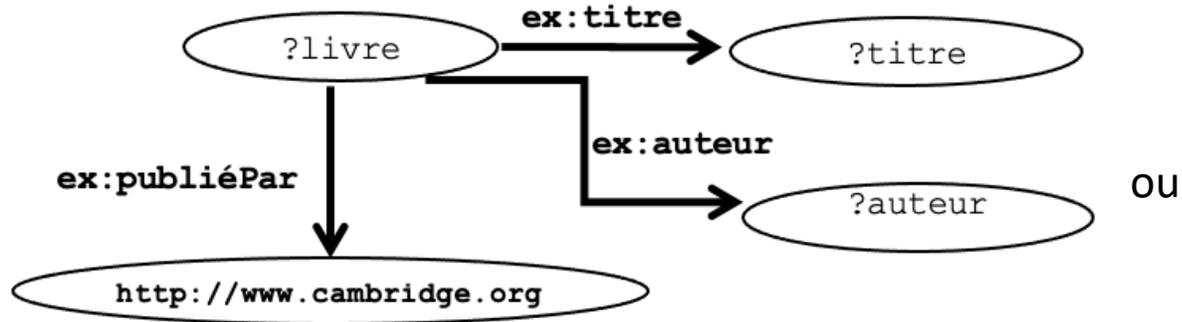
OU

```
PREFIX ex:<http://exemple.org/>
SELECT ?titre ?auteur
WHERE {
  ?livre      ex:publiéPar
              <http://www.cambridge.org/>.
  ?livre      ex:titre      ?titre .
  ?livre      ex:auteur     ?auteur.
}
```

Exemple

```
@prefix ex: <http://exemple.org/> .
ex:dlhb          ex:publiéPar      <http://www.cambridge.org/> ;
                 ex:titre          "The Description Logic Handbook" ;
                 ex:auteur         ex:fBaader ,
                                   ex:dCalvanese ;
ex:om            ex:auteur         ex:dMcGuinness .
                 ex:publiéPar      <http://www.springer.com/>
                 ex:titre          "Ontology Matching" ;
                 ex:auteur         ex:jEuzenat ;
                 ex:auteur         ex:pShvaiko .
```

"Quels sont les titres et noms d'auteurs des livres publiés par Cambridge Press ?"



OU

```
PREFIX ex:<http://exemple.org/>
SELECT ?titre ?auteur
WHERE {
  ?livre ex:publiéPar <http://www.cambridge.org/>.
  ?livre ex:titre ?titre .
  ?livre ex:auteur ?auteur.
}
```

Résultat :

Titre	Auteur
"The Description Logic Handbook"	http://exemple.org/fBaader
"The Description Logic Handbook"	http://exemple.org/dCalvanese
"The Description Logic Handbook"	http://exemple.org/dMcGuinness

Tests et *Filter*

`FILTER` permet de faire des tests et des comparaisons

Tests en SPARQL : FILTER

`FILTER` permet de faire des tests et des comparaisons

```
PREFIX ex:<http://exemple.org/>
SELECT ?livre
WHERE
{
    ?livre      ex:publiéPar    <http://www.springer.com/> .
    ?livre      ex:prix         ?prix
    FILTER ( ?prix < 100)
}
```

Avec `FILTER`, on peut tester l'égalité (=) et utiliser les opérateurs arithmétiques +, -, *, /.

Tests en SPARQL : FILTER

On peut aussi utiliser des expressions régulières

```
PREFIX ex:<http://exemple.org/>
SELECT ?livre
WHERE
{
    ?livre    ex:titre    ?titre
    FILTER ( REGEX(?title, "^Handbook of") )
}
```

et d'autres fonctions prédéfinies : isURI(a), isBLANK(a), isLITERAL(a), LANG(a), REGEX(a,b), etc.

SPARQL

- Beaucoup de fonctionnalités similaire à SQL
 - Agrégation, négation, formatage ...
- Mais d'autres, impossibles en SQL :
 - L'utilisation de **patrons de chemins**
 - Les **requêtes fédérées**

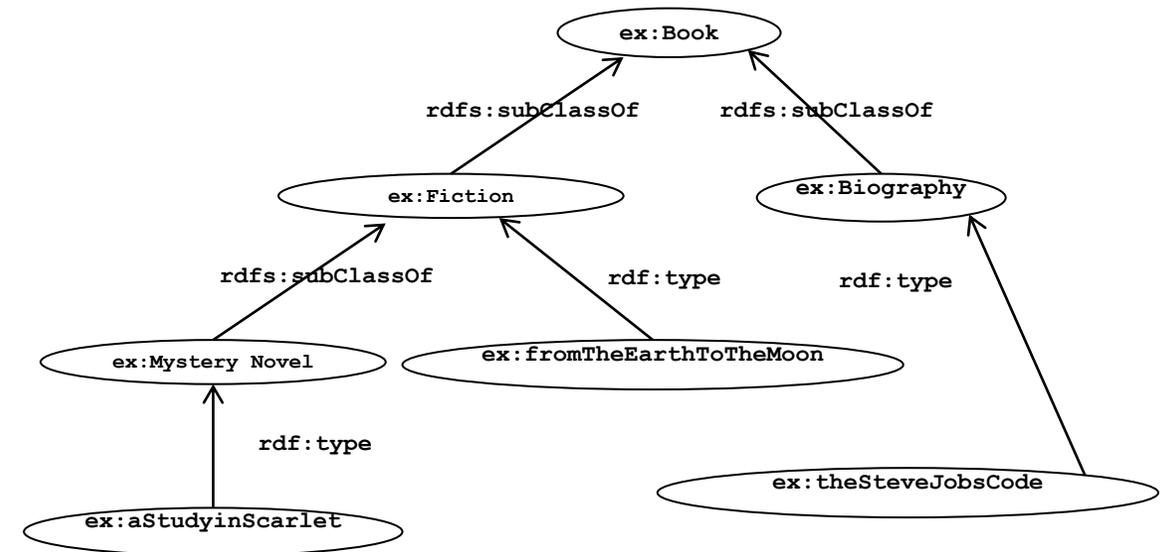
SPARQL1.1 et les chemins

- Les patrons de chemins permettent de retrouver des entités reliées indirectement

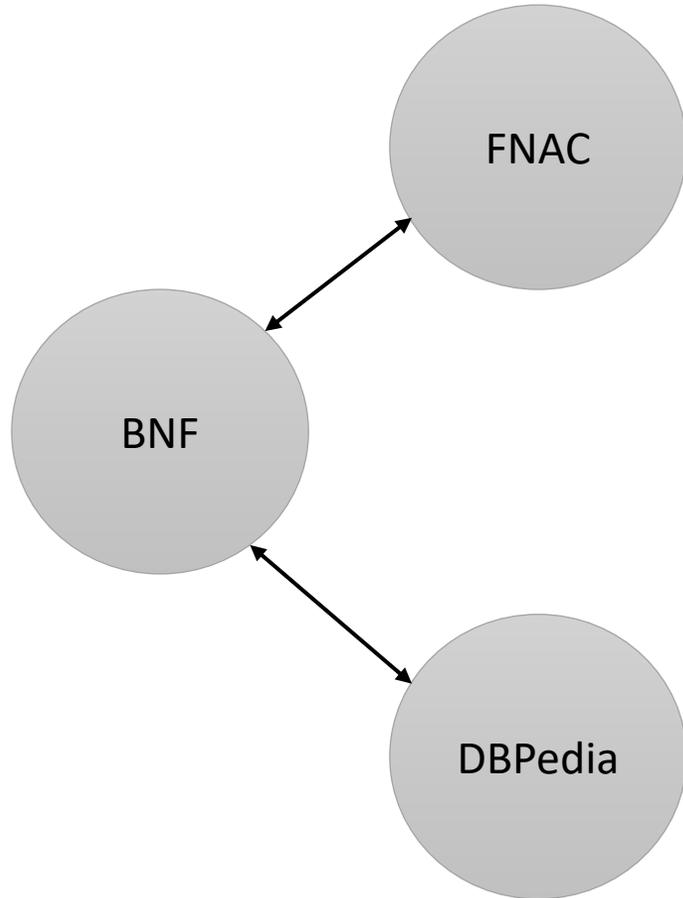
```
PREFIX ex:<http://example.org>
SELECT * WHERE {
  ?x a ex:Book
}
```

```
PREFIX ex:<http://example.org>
SELECT * WHERE {
  ?x rdf:type/rdfs:subClassOf* ex:Book
}
```

```
PREFIX ex:<http://example.org>
SELECT * WHERE {
  ?x ex:coauthorWith+/ex:name "Einstein"
}
```



Requêtes fédérées : à cheval sur plusieurs graphes RDF liés

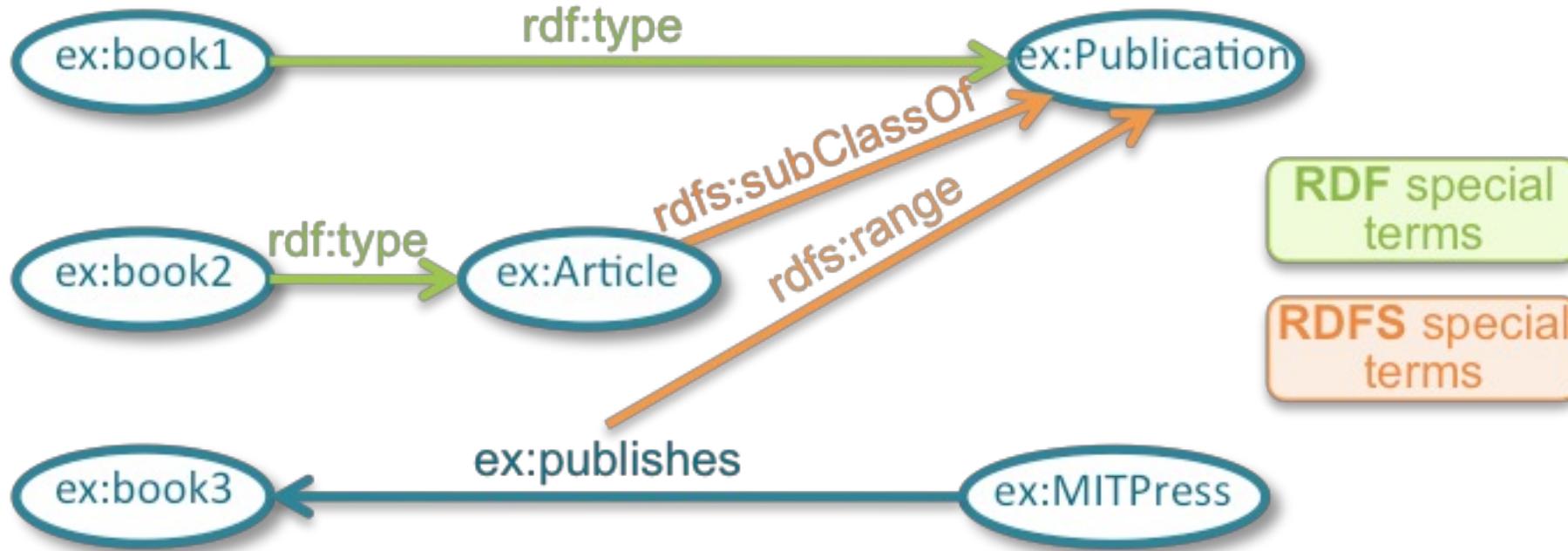


3 graphes de connaissances,
sur des serveurs différents,
Avec des URI en communs

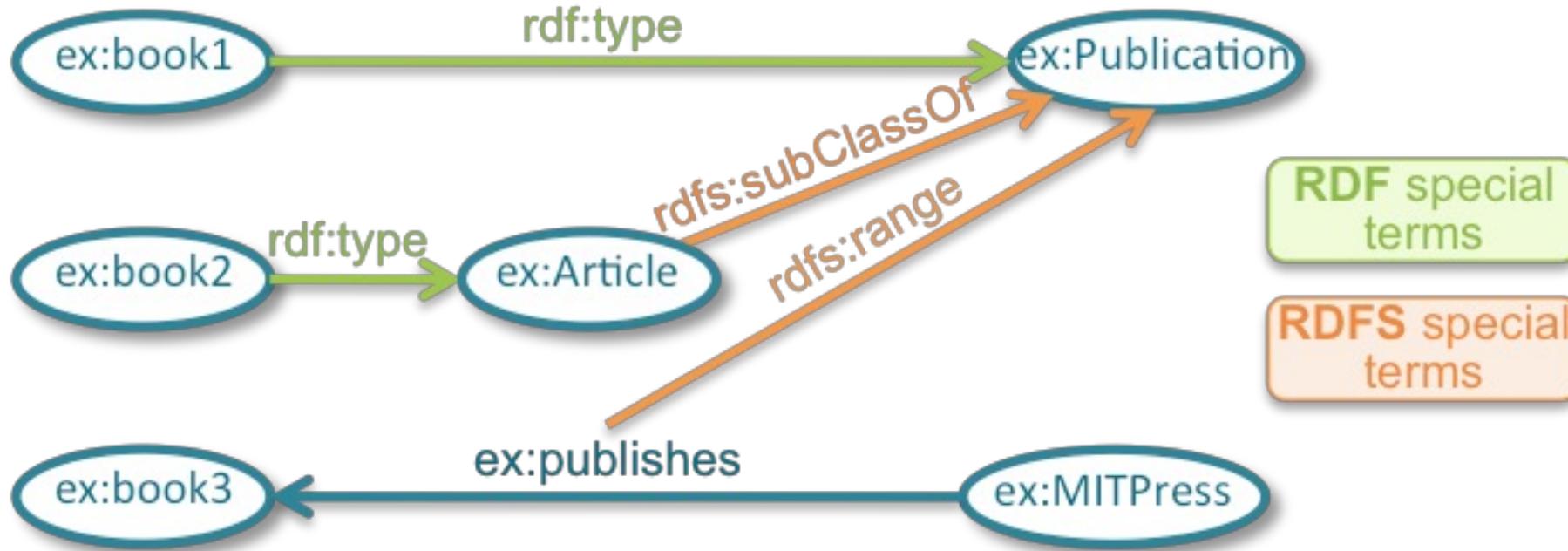
Les auteurs dont au moins un livre dont le titre commence par « the lord » est à moins de 10€ à la Fnac, et dont au moins un enfant est éditeur ?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: http://www.w3.org/2000/01/rdf-schema#
PREFIX bnf: [...] PREFIX fnac: [...] PREFIX dbpedia: [...]
SELECT distinct ?author ?kid
WHERE {
    # Get authors of books with title « the lords of ...»
    ?author bnf:auteur ?book .
    ?book bnf:titre ?titre
    FILTER ( REGEX(?titre, "^the lord") ).
    # The rest of the query will be applied on two different servers
    SERVICE <http://www.fnac.fr/sparql> {
        ?book fnac:price ?p ;
        FILTER ( ?p < 10 ).}
    SERVICE <http://dbpedia.org/sparql> {
        ?author a dbpedia:Author.
        ?author dbpedia:hasChild ?kid.
        ?kid a dbpedia:Editor .}
}
```

SPARQL et le raisonnement : distinction entre triplées assérés et inférés



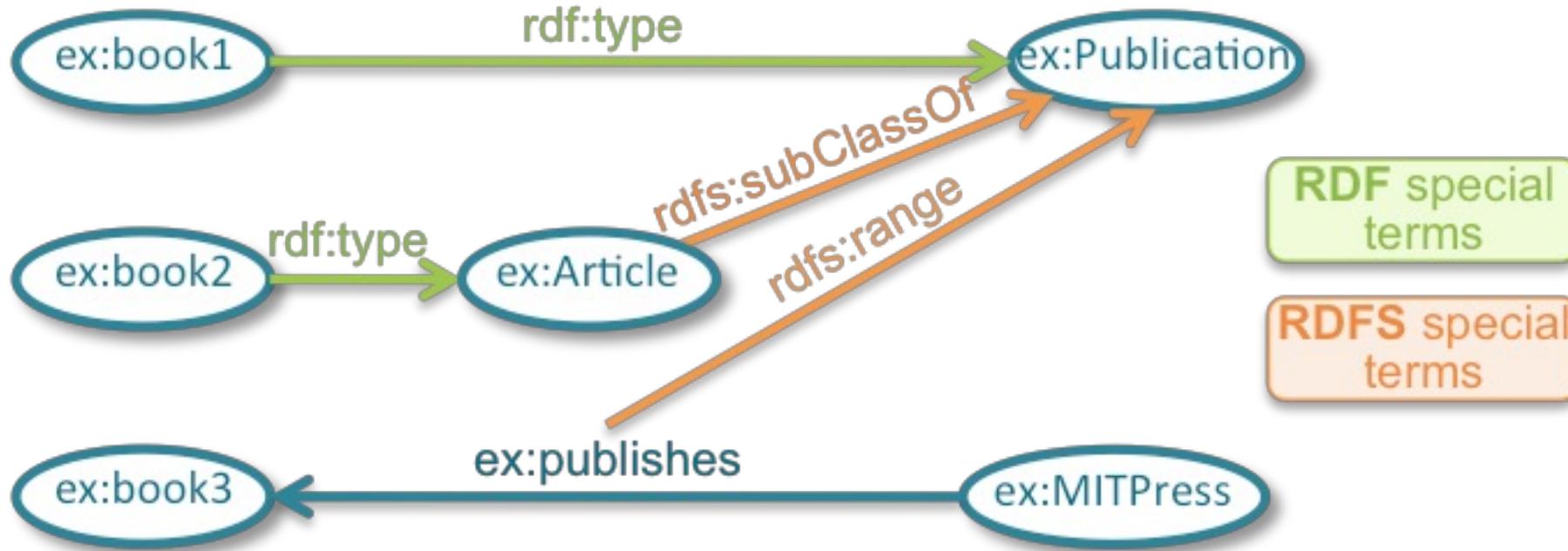
SPARQL et le raisonnement : distinction entre triplées assérés et inférés



Une requête pour trouver toutes les publications :

```
PREFIX ex:<http://example.org>
SELECT * WHERE {
  ?x a ex:Publication
}
```

SPARQL et le raisonnement : distinction entre triplées assérés et inférés



Une requête pour trouver toutes les publications :

```
PREFIX ex:<http://example.org>
SELECT * WHERE {
  ?x a ex:Publication
}
```

```
ex:book1 ou
ex:book1, ex:book2 ou
ex:book1, ex:book2, exbook3
```

Ca dépend du SPARQL endpoint !

Le projet Bio2RDF pour des exemple de requêtes

- Bio2RDF
 - Projet communautaire
 - Open Source
 - Quelques règles simples de nommage
 - Partage des scripts de RDFization : <https://github.com/bio2rdf/bio2rdf-scripts>
 - ...
 - Pas de maintenance active des serveurs
 - Mais données téléchargeables : <https://download.bio2rdf.org/#/>
- Exemple de requête : <https://bio2rdf.org/sparql>

Exemple 2

- <https://sparql.uniprot.org/>

Les proteines humaines qui sont impliquées dans le métabolisme du glucose (GO:0006006)

```
SELECT (COUNT(DISTINCT(?protein)) AS ?pc)
```

```
WHERE {
```

```
  ?protein rdf:type up:Protein ;
```

```
  up:reviewed true ;
```

```
  up:organism taxon:9606 ;
```

```
  up:classifiedWith GO:0006006 .}
```

```
SELECT (COUNT(DISTINCT(?protein)) AS ?pc)
```

```
WHERE {
```

```
  ?protein rdf:type up:Protein ;
```

```
  up:reviewed true ;
```

```
  up:organism taxon:9606 ;
```

```
  up:classifiedWith|(up:classifiedWith/rdfs:subClassOf) GO:0006006 .}
```

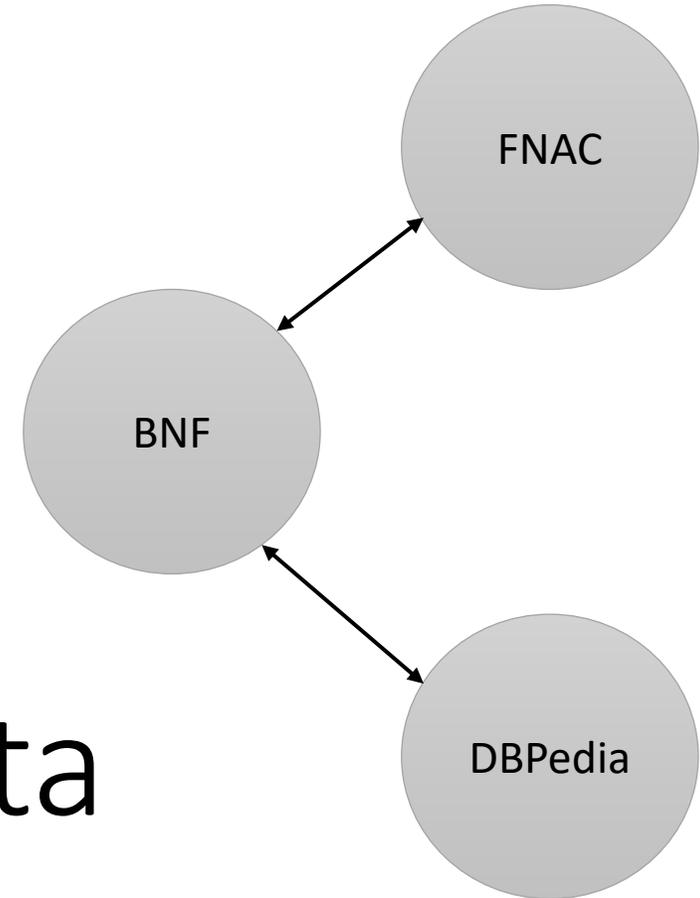
117>61 : pourquoi ?

```
SELECT distinct ?c ?l
```

```
WHERE{ ?c rdfs:subClassOf+ GO:0006006 .
```

```
  ?c rdfs:label ?l.}
```

LOD: Linked Open Data



LOD: Initiative communautaire de partage et de liaison des données

Linked Open Data 5★

★ On the Web (under open license)

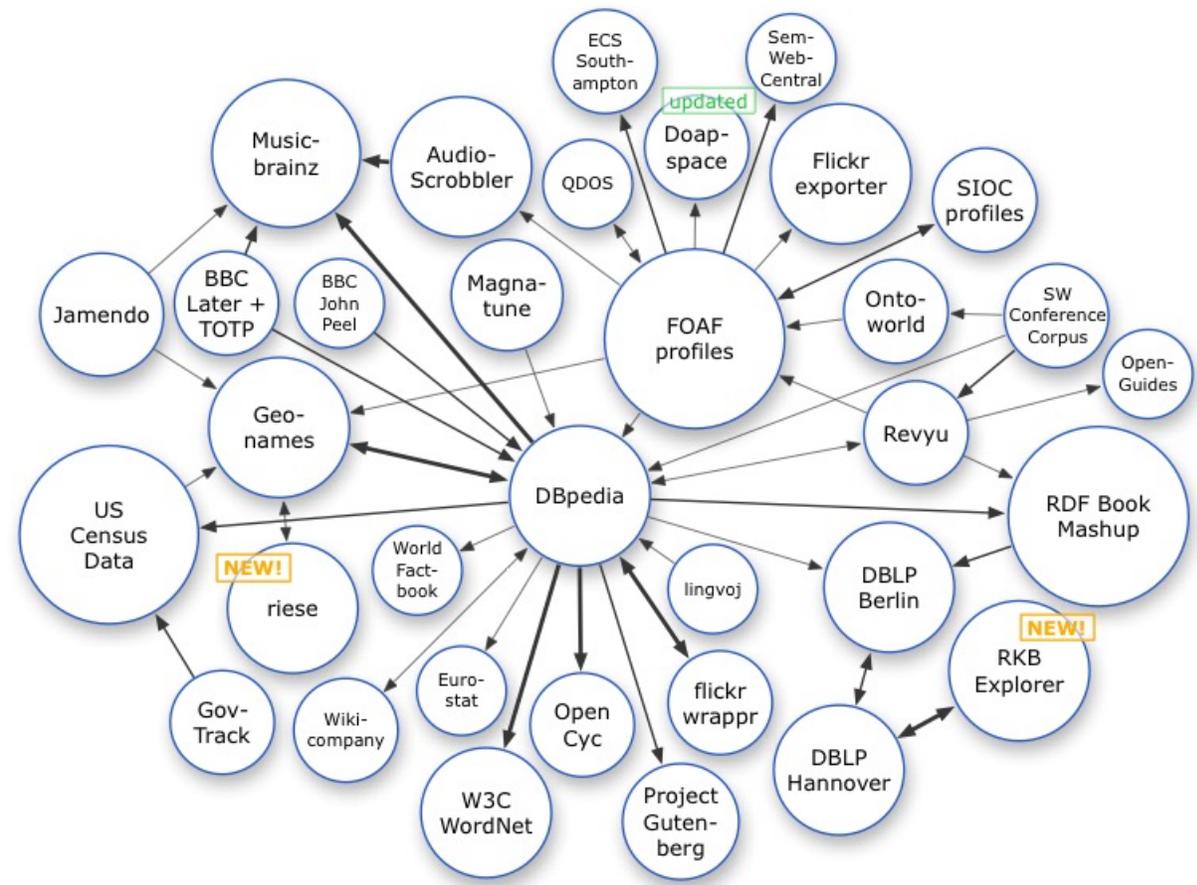
★★ Machine-readable format

★★★ Non-proprietary format

★★★★ RDF standards

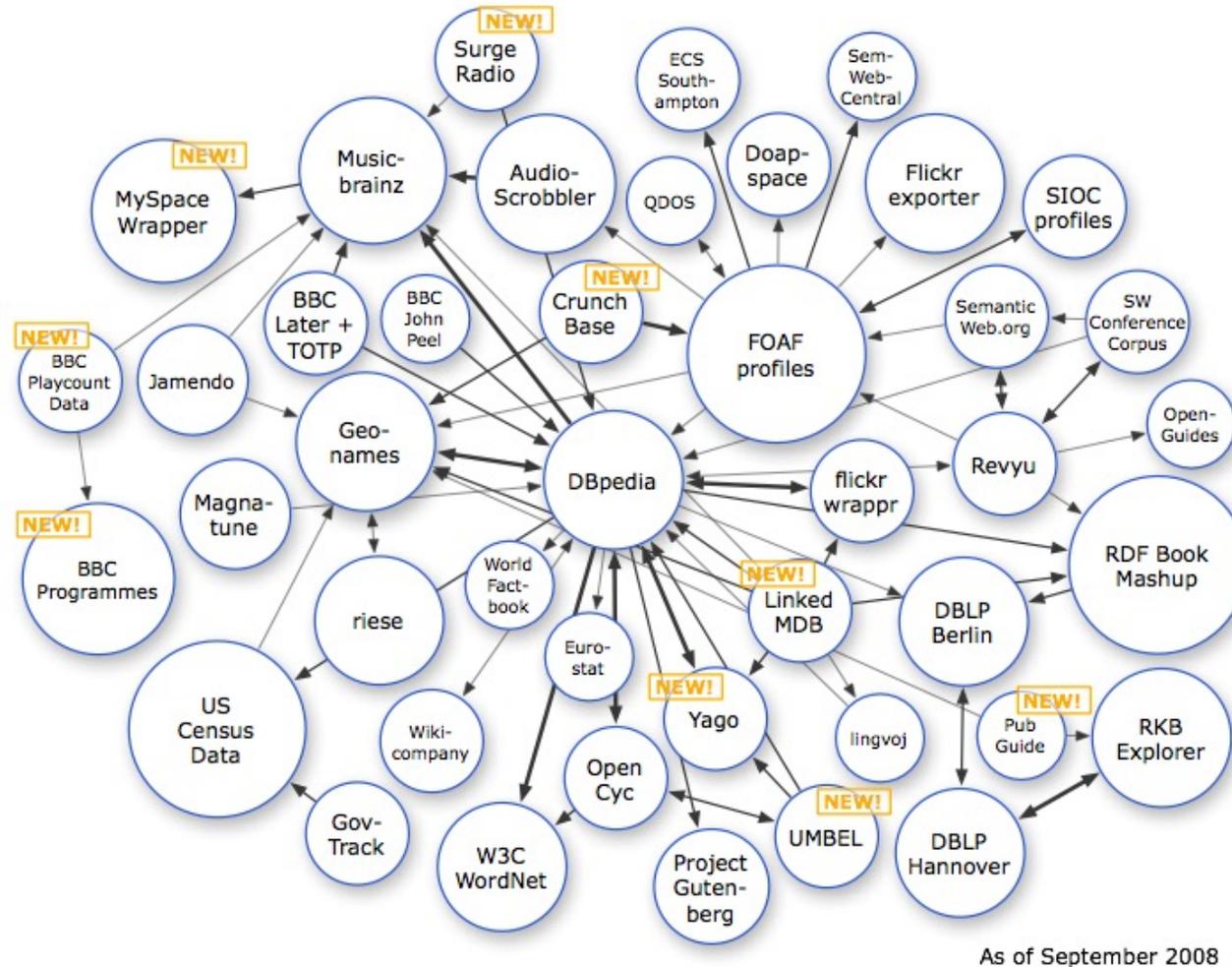
★★★★★ Linked RDF

The LOD Cloud, March 2008

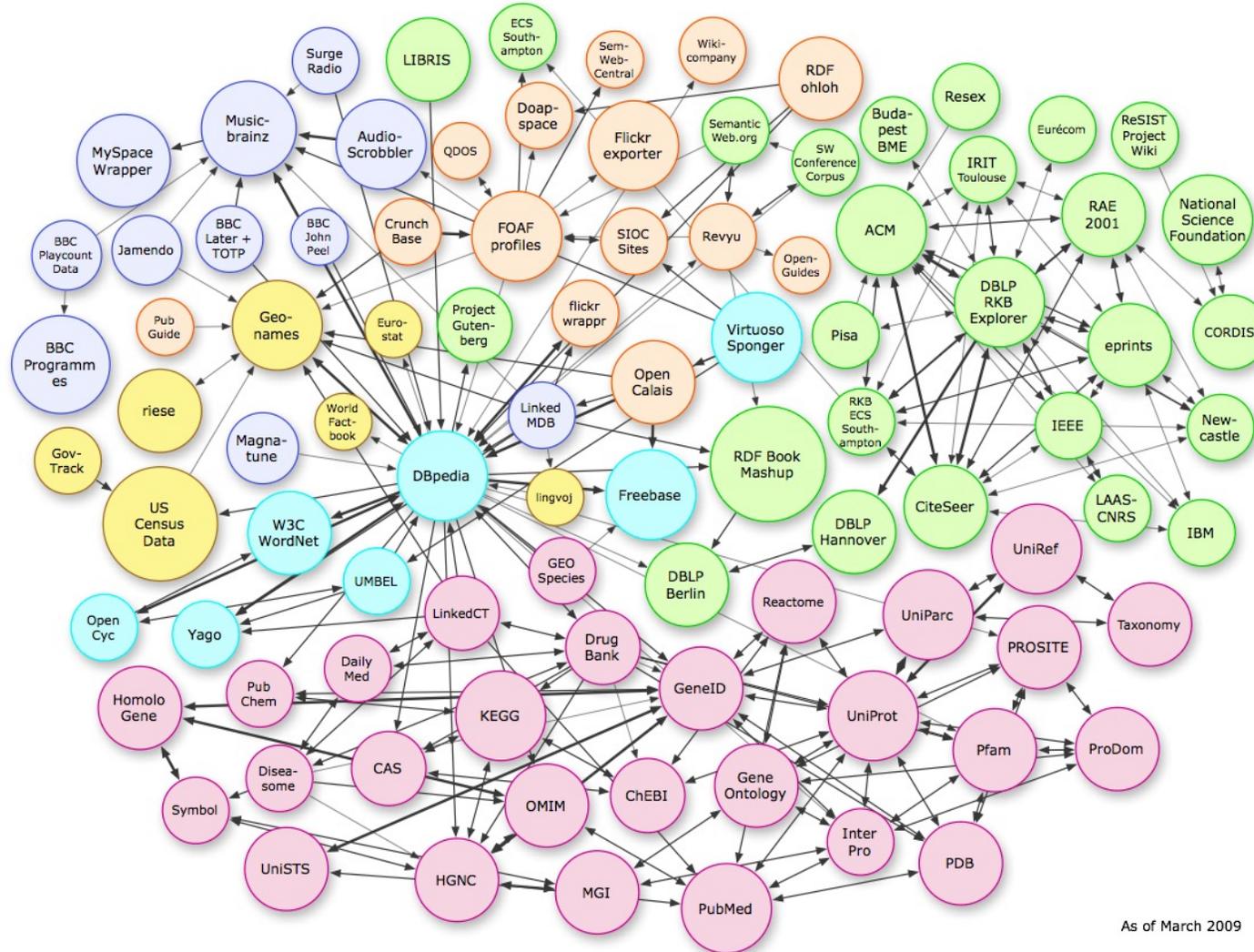


"Linking Open Data cloud diagram 2008, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

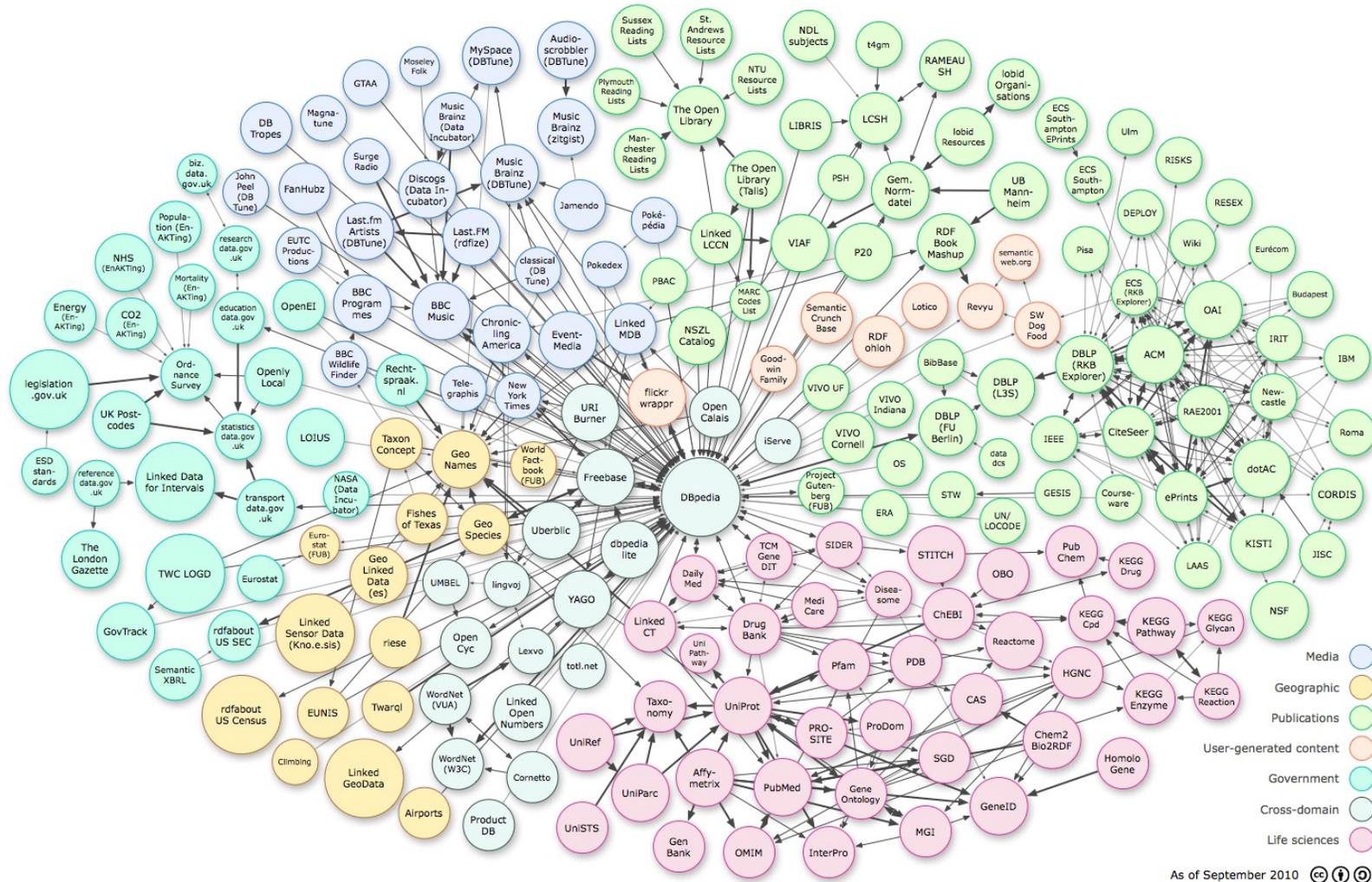
The LOD Cloud, September 2008



The LOD Cloud, March 2009

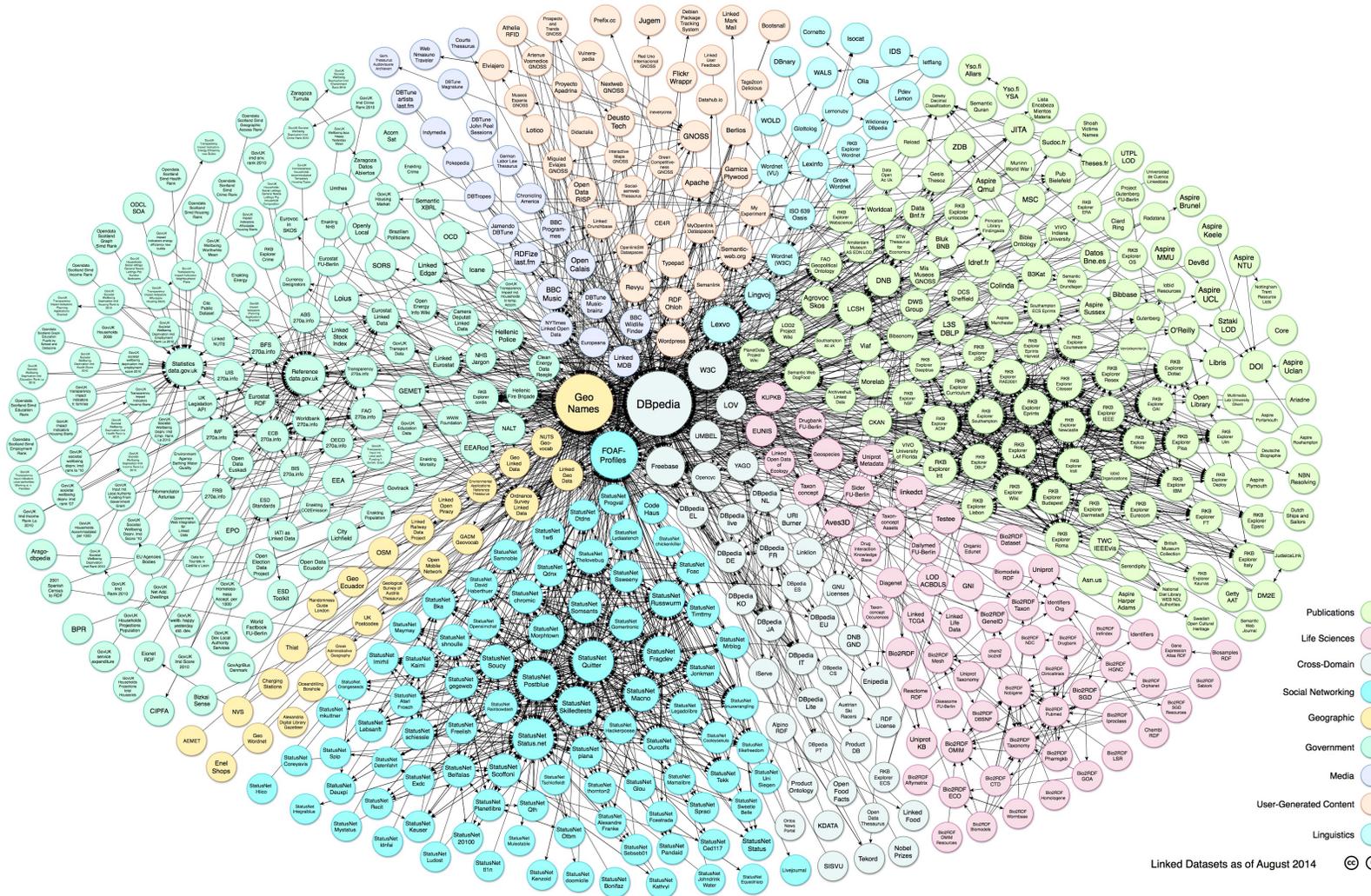


The LOD Cloud, September 2010



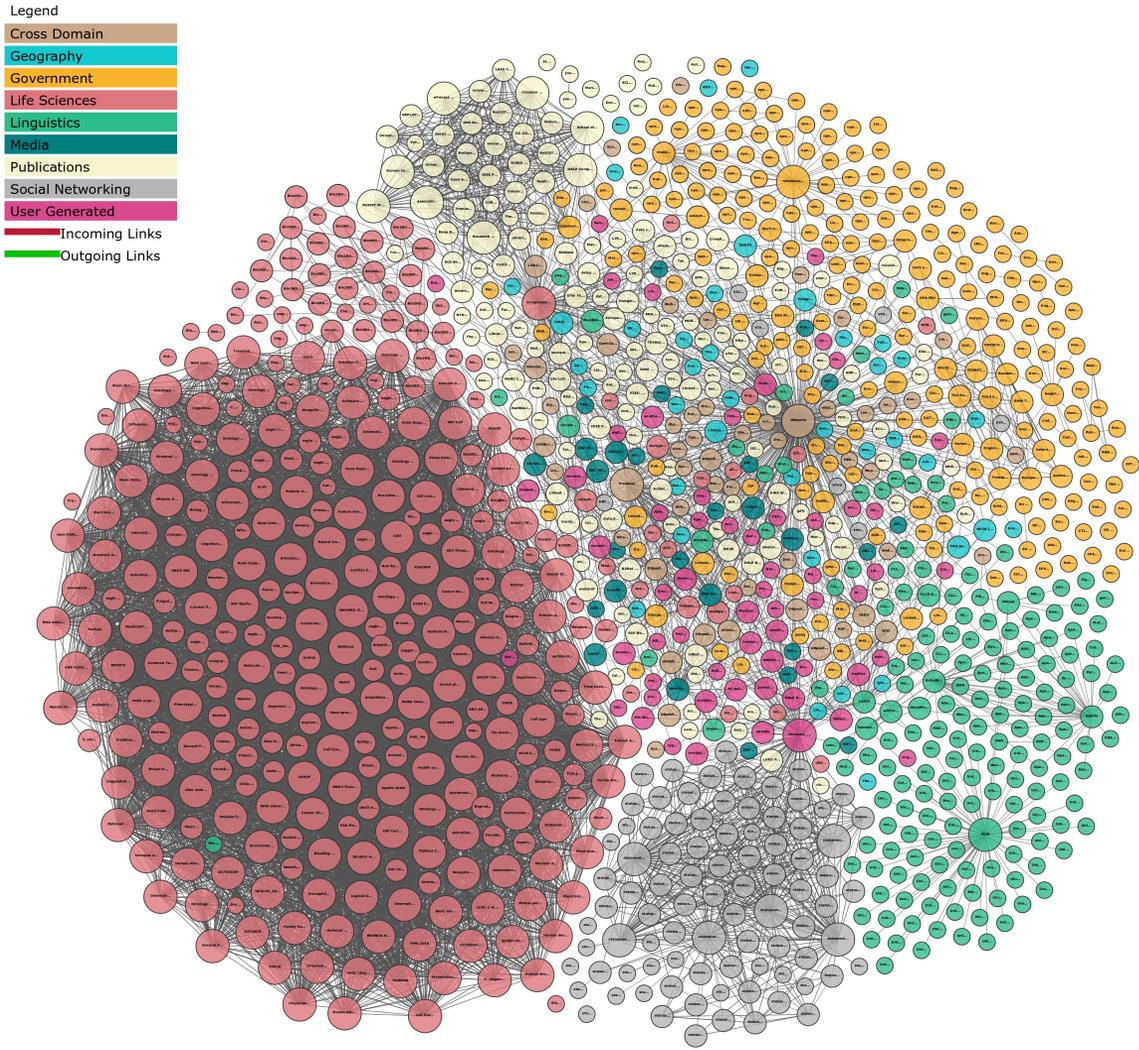
"Linking Open Data cloud diagram 2010, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

The LOD Cloud, August 2014



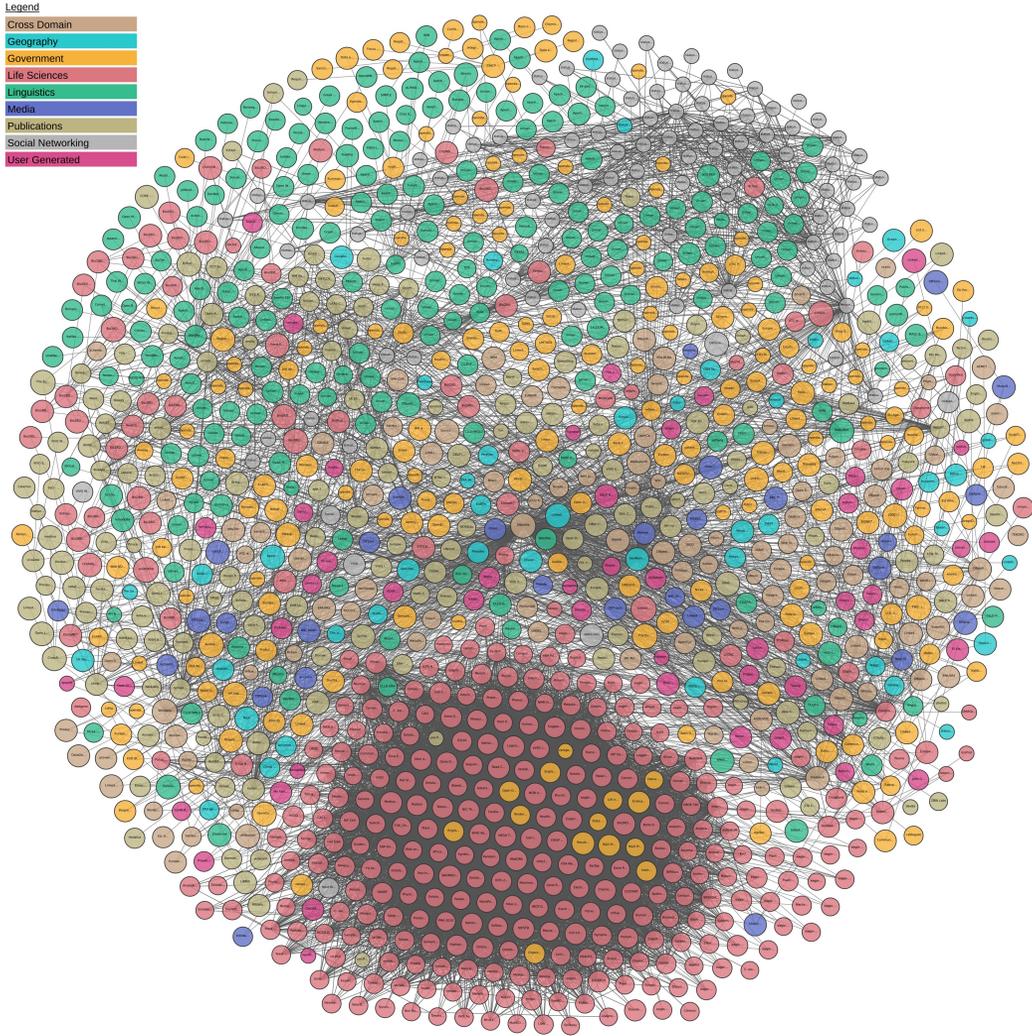
"Linking Open Data cloud diagram 2014, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

The LOD Cloud, August 2017



"Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

The LOD Cloud, August 2018



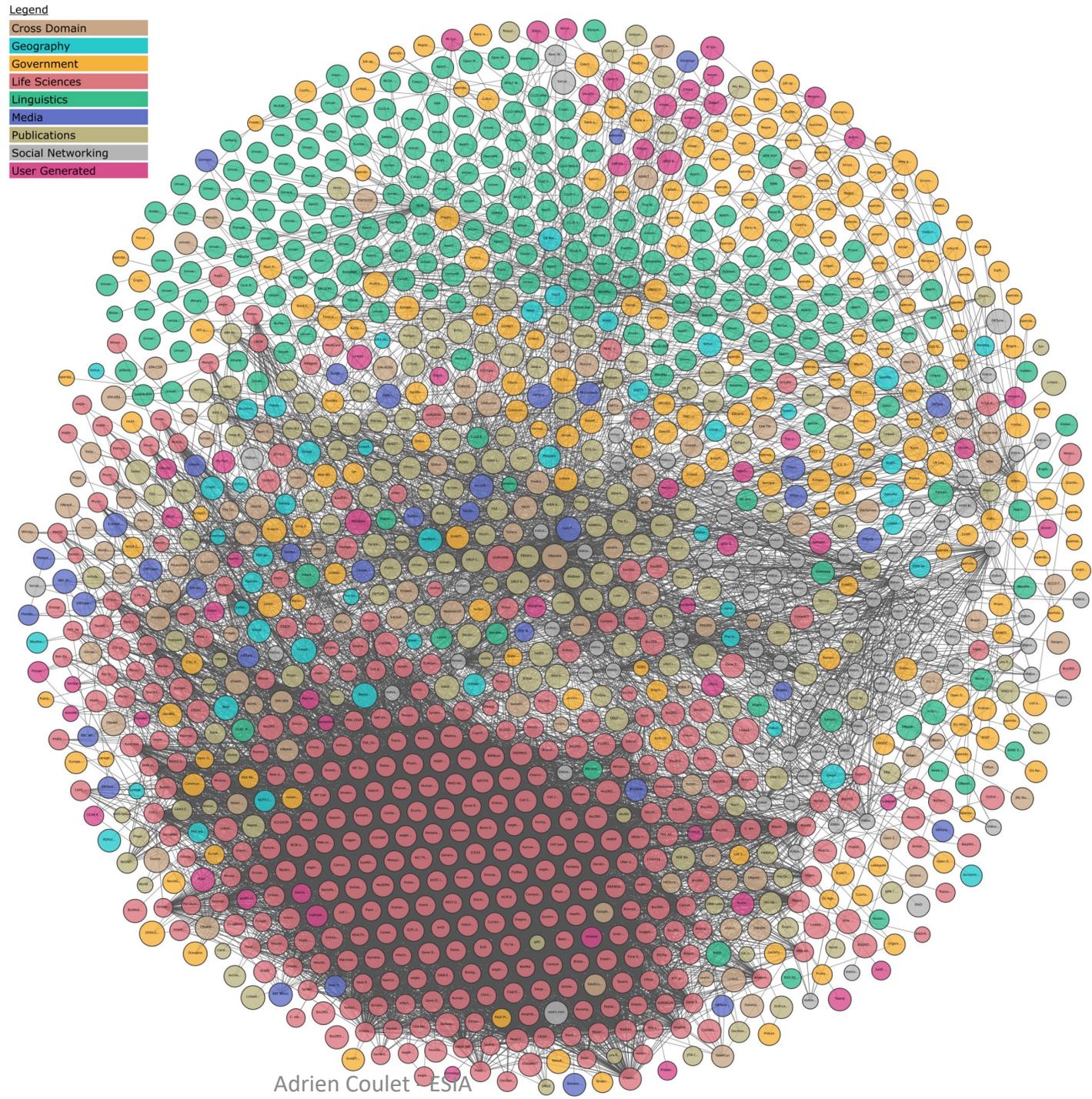
The Linked Open Data Cloud from lod-cloud.net



"Linking Open Data cloud diagram 2018, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

The LOD Cloud, May 2020

The dataset currently contains
1,255 datasets with **16,174** links



"Linking Open Data cloud diagrams,
by Andrejs Abele, John P. McCrae, Paul Buitelaar,
Anja Jentsch and Richard Cyganiak.
<http://lod-cloud.net/>"

5stars inspired FAIR principles

Linked Open Data 5★

★ On the Web (under open license)

★★ Machine-readable format

★★★ Non-proprietary format

★★★★ RDF standards

★★★★★ Linked RDF

The FAIR Guiding Principles

Findable:

- F1 Data and metadata are assigned a globally unique and persistent identifier
- F2 Data are described with rich metadata (defined by R1 below)
- F3 Metadata clearly and explicitly include the identifier of the data it describes
- F4 Data and metadata are registered or indexed in a searchable resource

Accessible:

- A1 Data and metadata are retrievable by their identifier using a standardized communications protocol
 - A1.1 The protocol is open, free, and universally implementable
 - A1.2 The protocol allows for an authentication and authorization procedure, where necessary
- A2 Metadata are accessible, even when the data are no longer available

Interoperable:

- I1 Data and metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2 Data and metadata use vocabularies that follow FAIR principles
- I3 Data and metadata include qualified references to other (meta)data

Reusable:

- R1 Data and metadata are richly described with a plurality of accurate and relevant attributes
 - R1.1 Data and metadata are released with a clear and accessible data usage license
 - R1.2 Data and metadata are associated with detailed provenance
 - R1.3 Data and metadata meet domain-relevant community standards

LOD...

- Pour les données biomédicale de références (DrugBank, UniProt, ClinicalTrial.gov, Sider, etc.) => vu
- Pour le web :
 - DBPedia
 - Yago
- Pour le contenu de la littérature :
 - méthodes d'extraction de relation + normalization => RDF
- Pour les données patients : ?

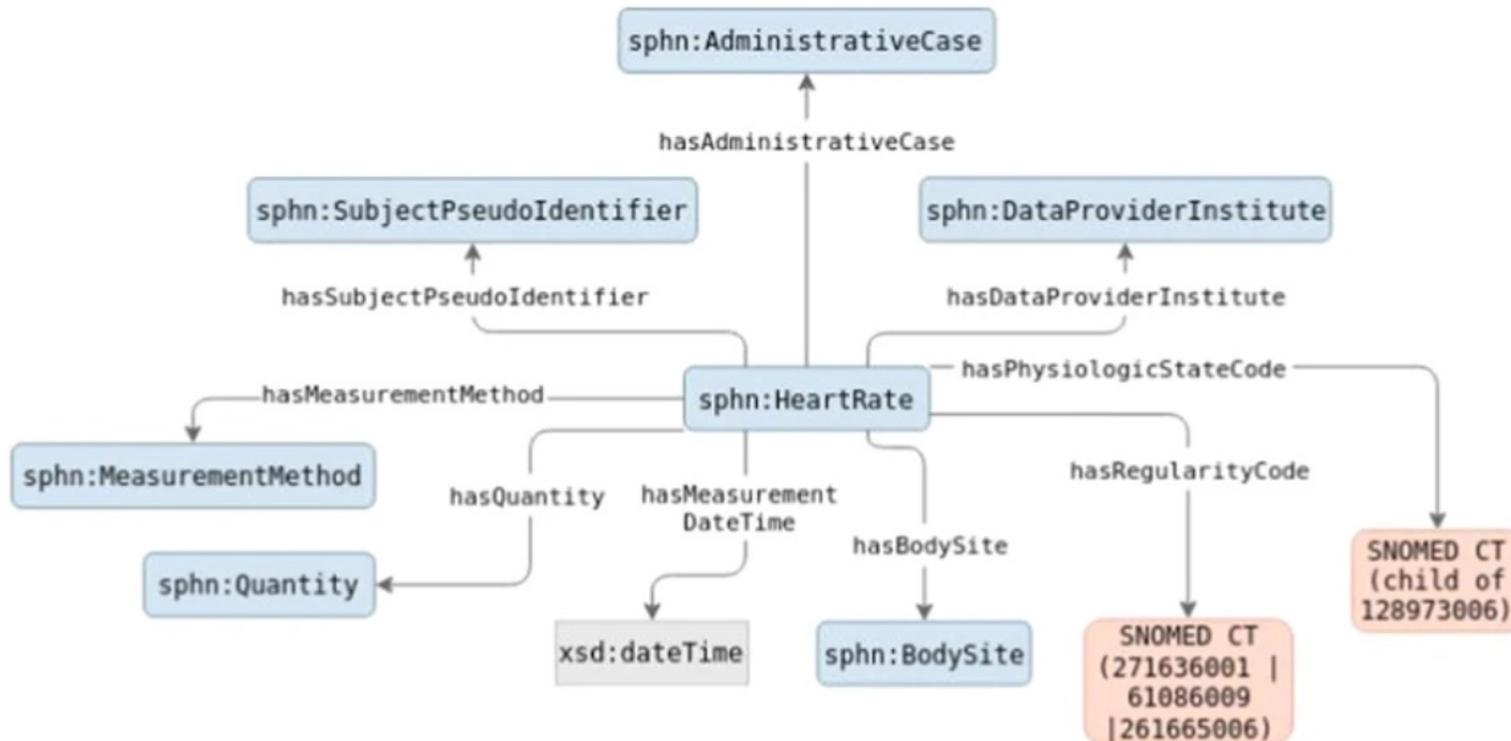
LOD...

- Pour les données biomédicale de références (DrugBank, UniProt, ClinicalTrial.gov, Sider, etc.) => vu
- Pour le web :
 - DBPedia
 - Yago
- Pour le contenu de la littérature :
 - méthodes d'extraction de relation + normalization => RDF
- Pour les données patients : ?
 - Des efforts en court

Des graphes de connaissances pour représenter (et lier) les données patients (1/2)

Principalement deux initiatives :

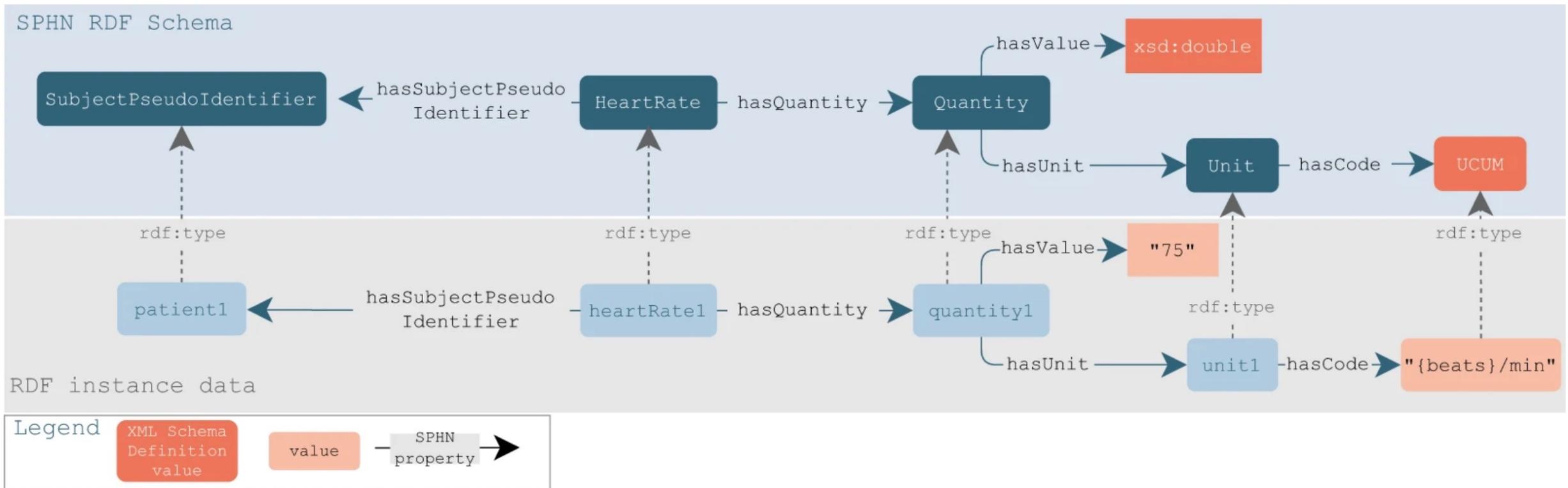
- Version RDF de données suivant le schéma commun OHDSI (l'OMOP CDM)
- Le Swiss Personalized Health Network (SPHN), par le SIB



Des graphes de connaissances pour représenter (et lier) les données patients (2/2)

Principalement deux initiatives :

- Version RDF de données suivant le schéma commun OHDSI (l'OMOP CDM)
- Le Swiss Personalized Health Network (SPHN), par le SIB



Plan

1. Définitions

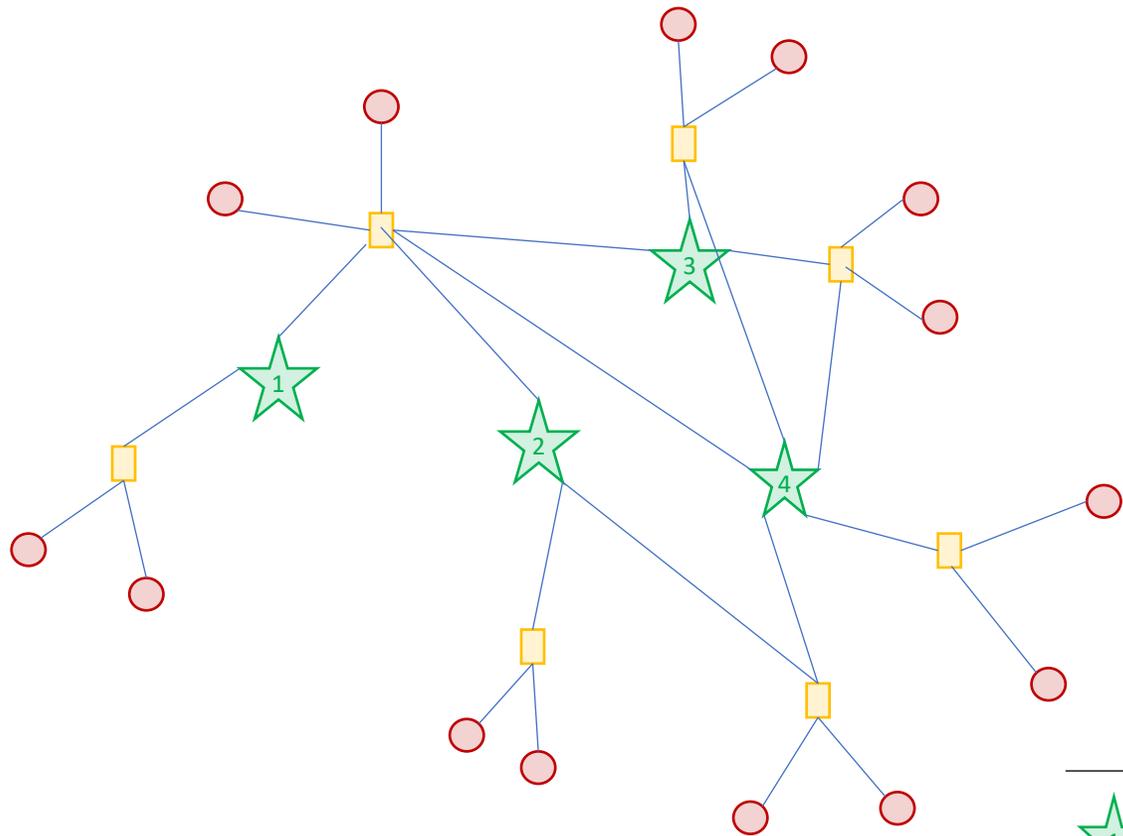
- a) Distinction entre données, information et connaissances
- b) Définitions d'un graphes de connaissances
- c) Les types de graphes de connaissances

2. Les graphes RDF

- a) Syntaxe
- b) Raisonnement
- c) Requête (SPARQL)

3. L'apprentissage sur les graphes de connaissances

- a) Approches classiques
- b) GCN



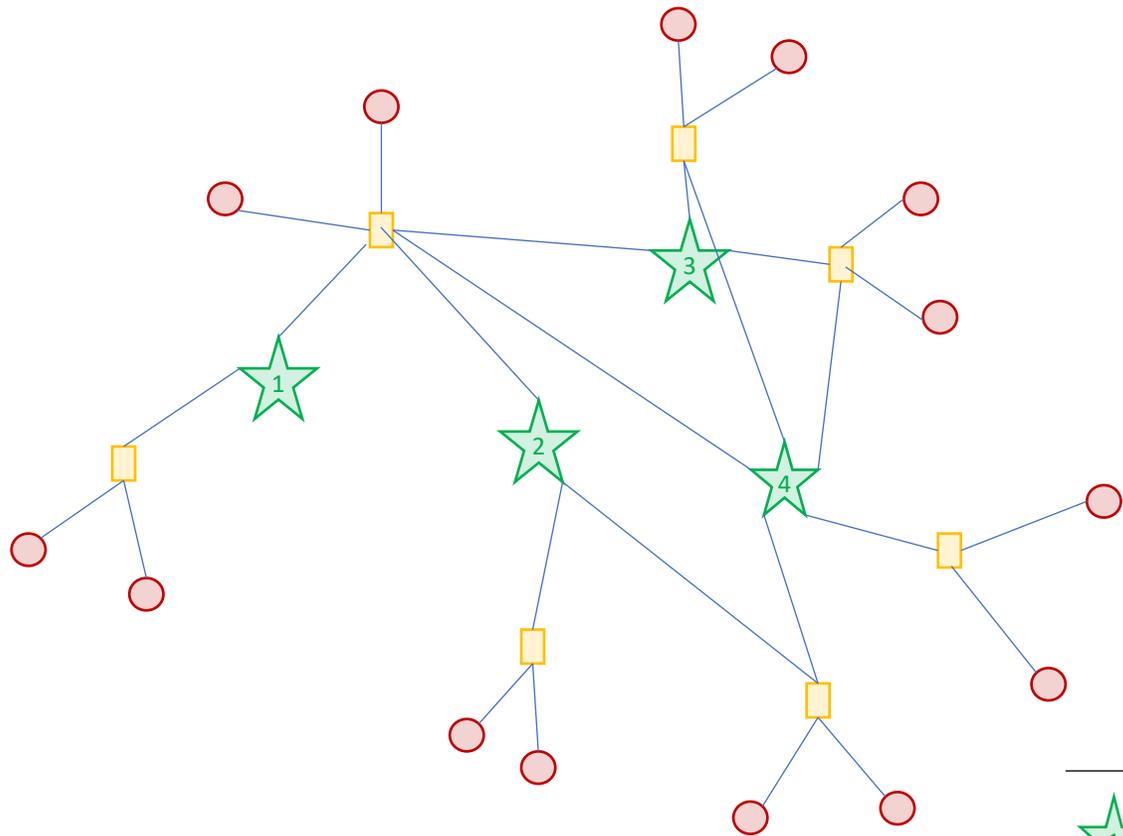
★ Target node

□ Neighbourhood = 1

○ Neighbourhood = 2

Node features

★ 1	
★ 2	
★ 3	
★ 4	

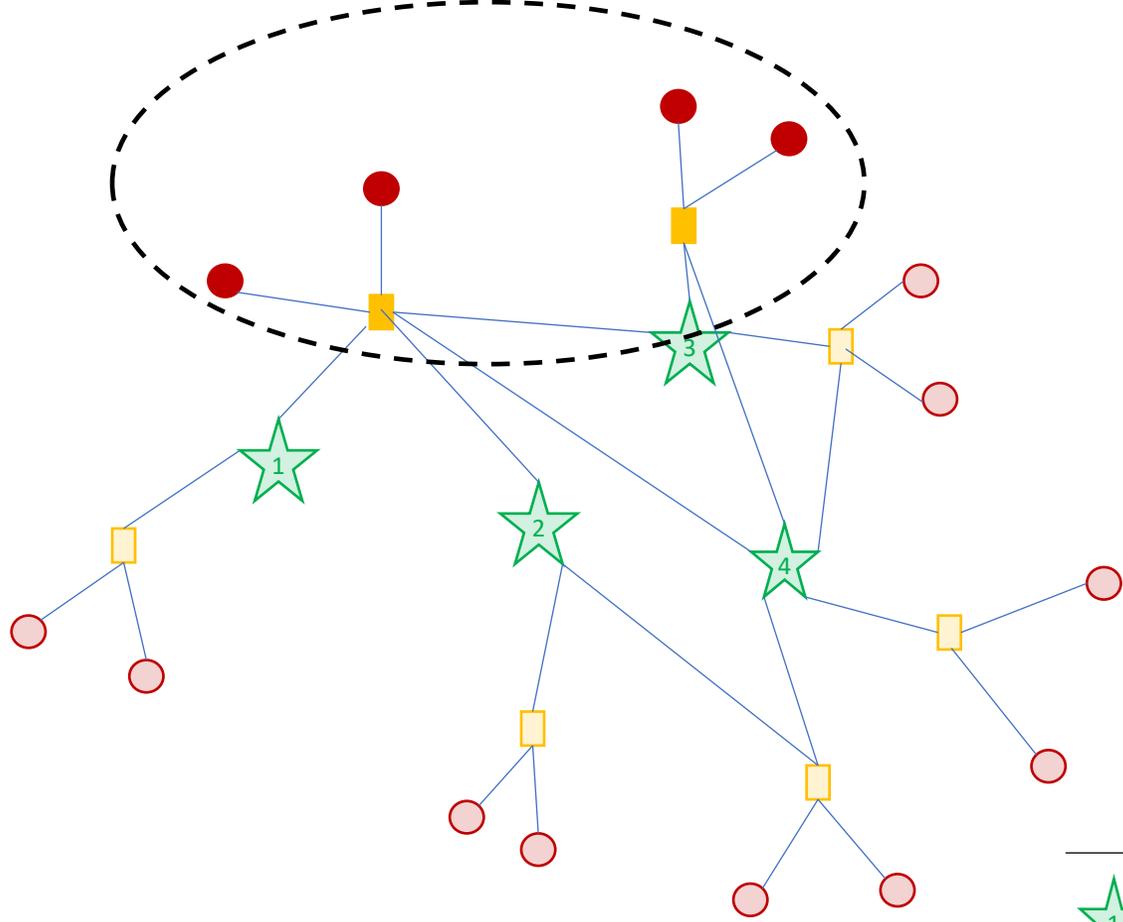


Node features



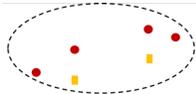
Bag of nodes

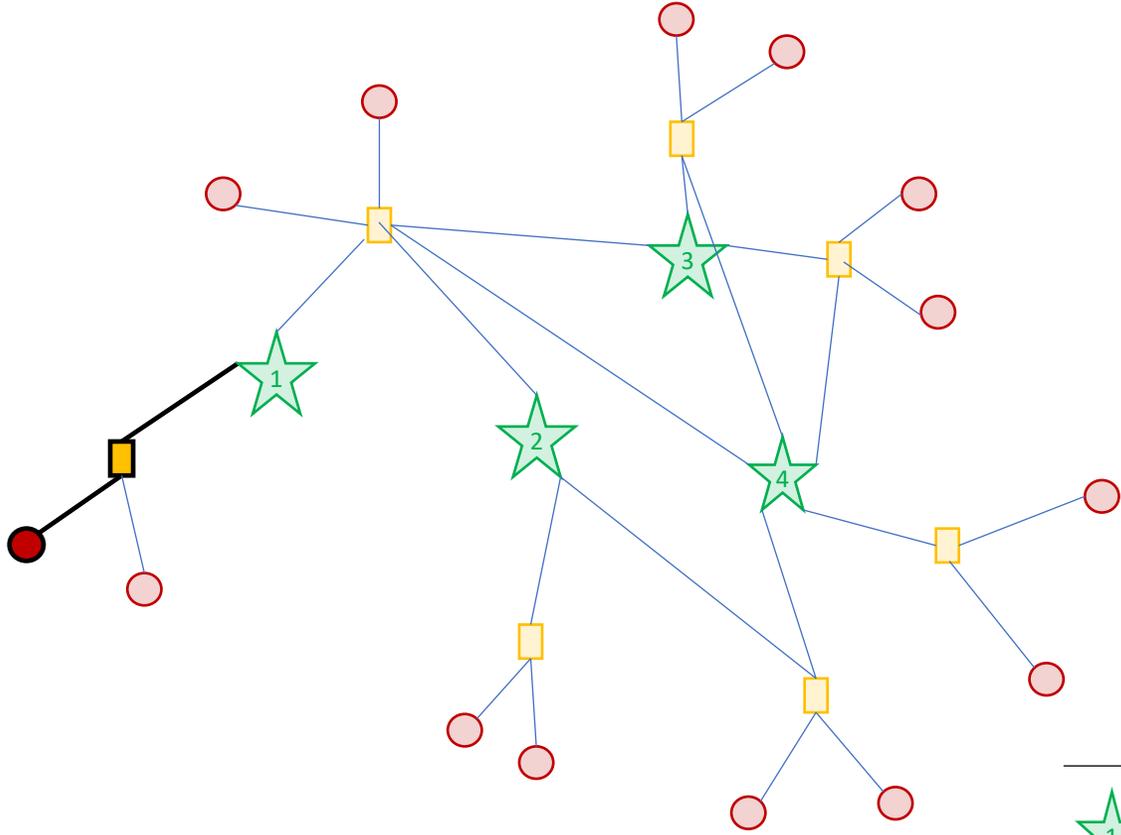
	
	
	
	



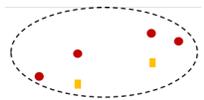
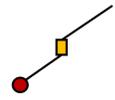
Node features

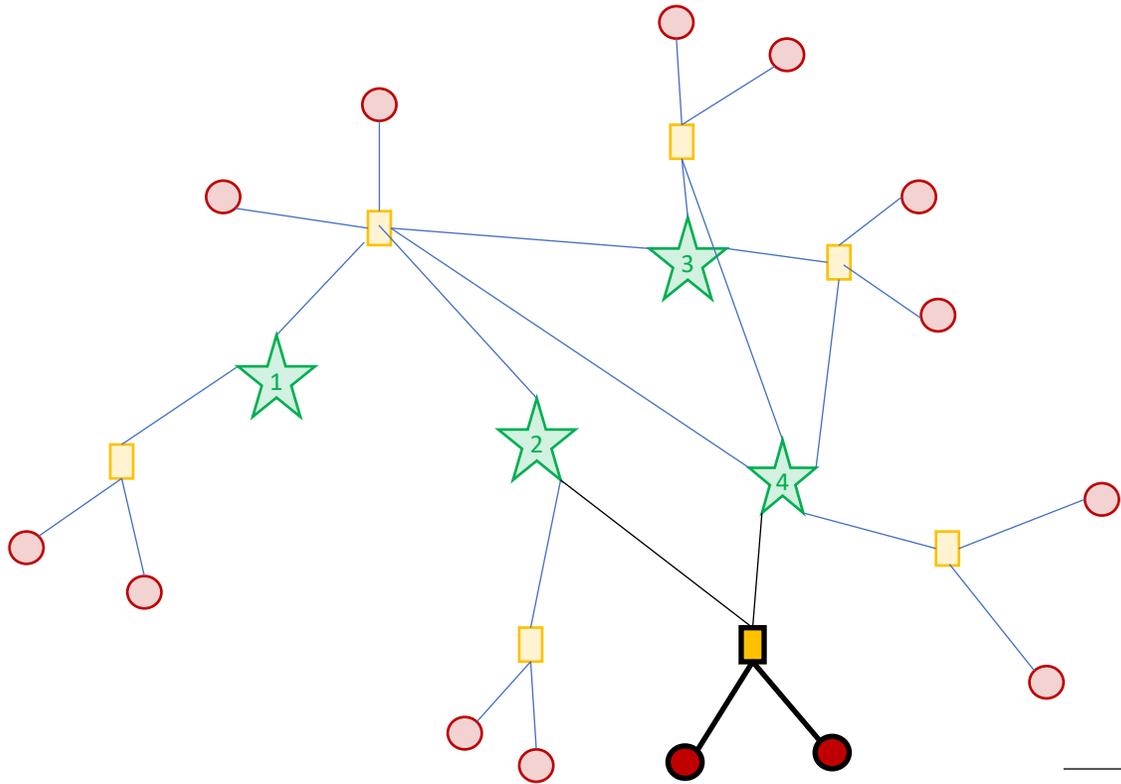
Bags of nodes

	 ...		
			
			
	X		
	X		



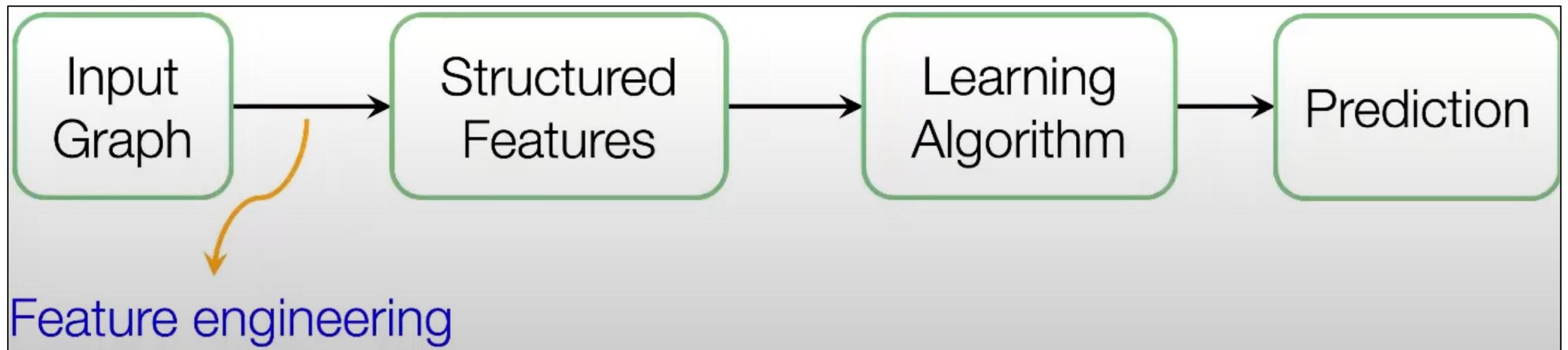
Node features

	Bags of nodes		Paths	
		...		...
			X	
				
	X			
	X			

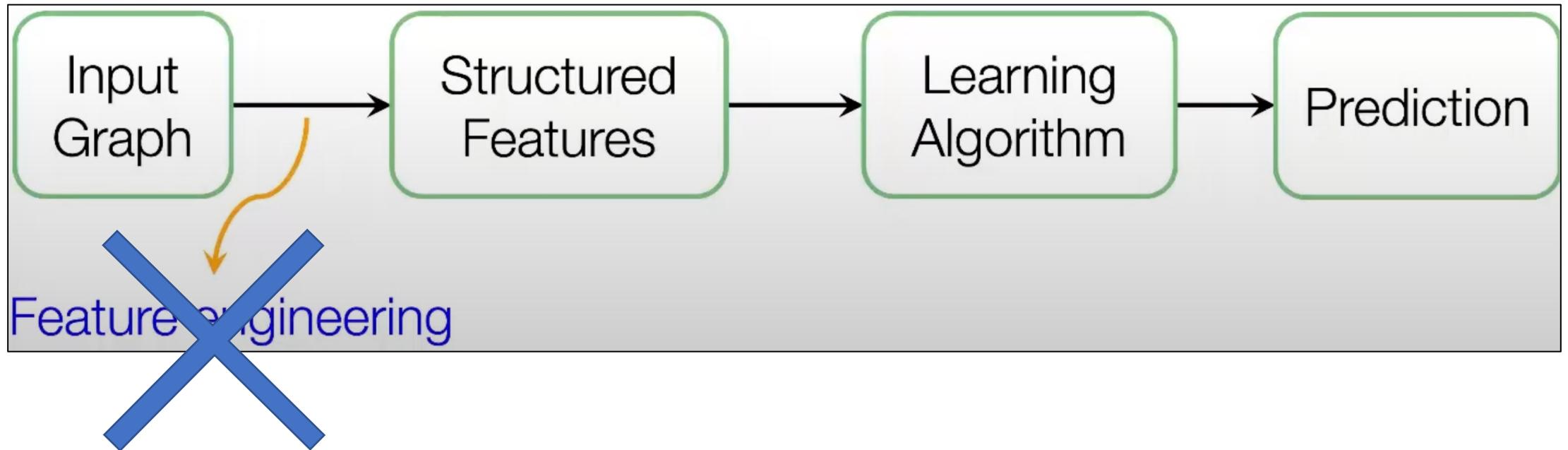


Node features

	Bags of nodes		Paths		Trees	
	
			X			
					X	
	X					
	X				X	



From feature engineering to representation learning



Random walks & node2vec

Random walks & node2vec

- Idea: if random walk starting from a node u visits v with a high probability, u and v are similar
- Similar nodes are nearby in the embedding space
- Finding node u embedding \mathbf{z}_u that minimize \mathcal{L} :

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} - \log \left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^T \mathbf{z}_n)} \right)$$

sum over all nodes u sum over nodes v seen on random walks starting from u predicted probability of u and v co-occurring on random walk

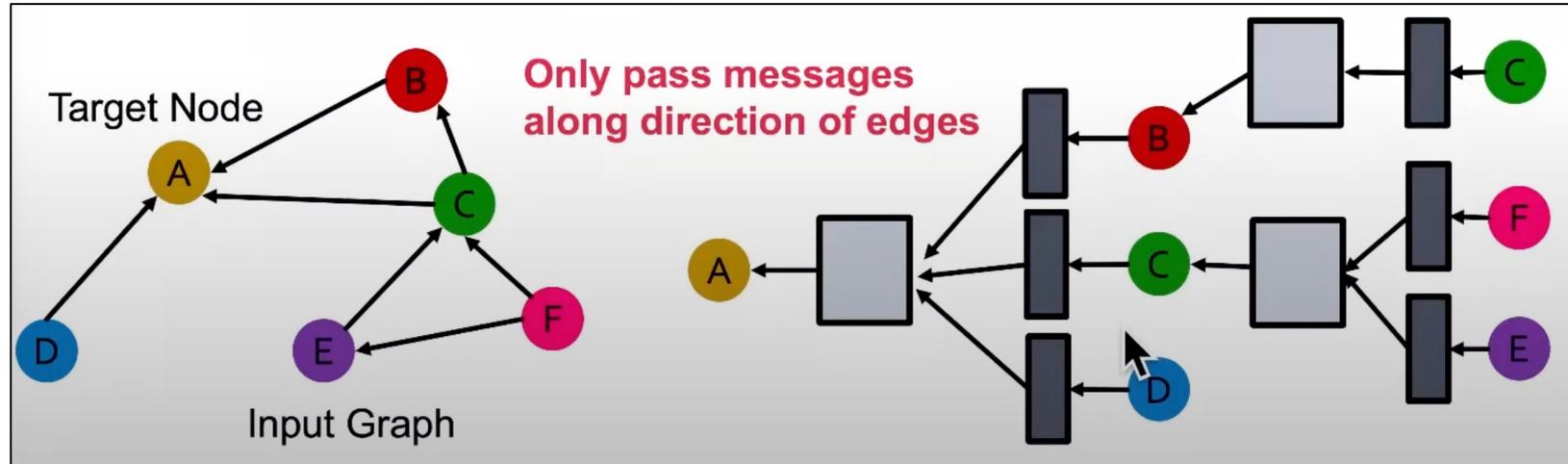
↑ or $P(v|\mathbf{z}_u)$

i.e., a softmax function

Graph Convolutionnal Networks

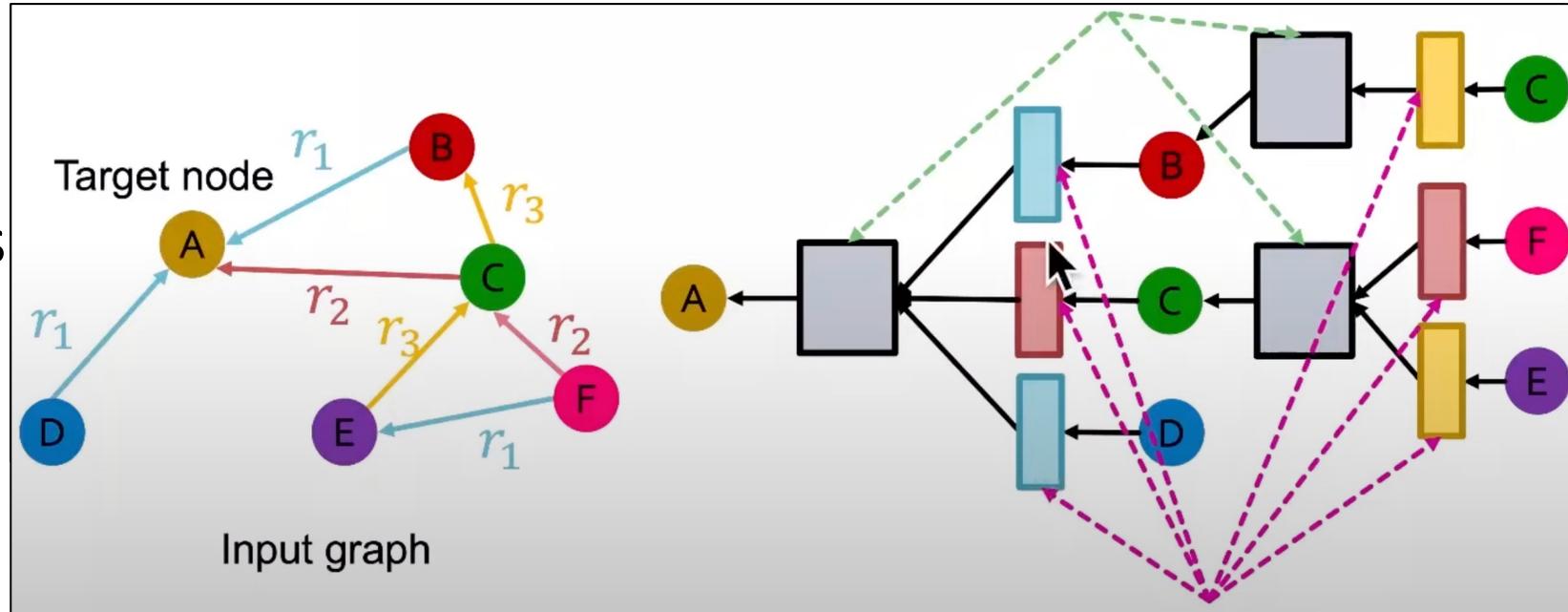
Message passing framework

- intuition



Message passing framework

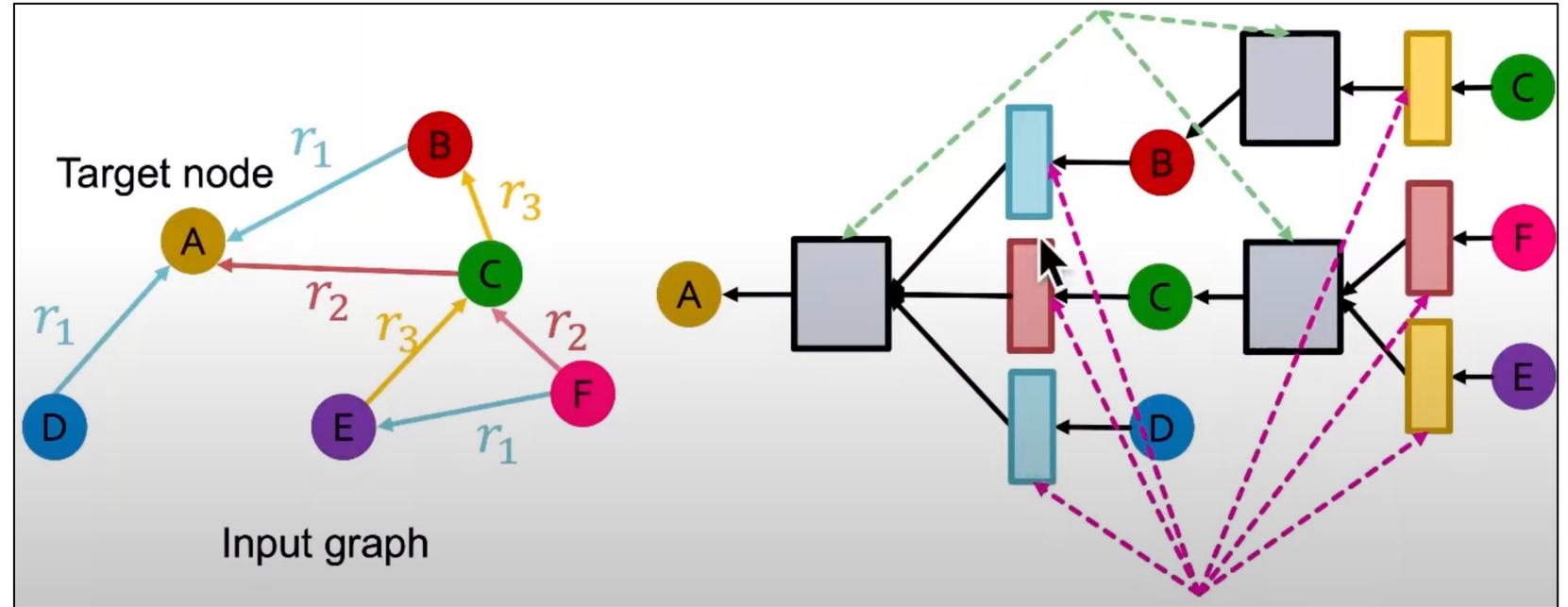
- Intuition, with various types of relations



Message passing framework

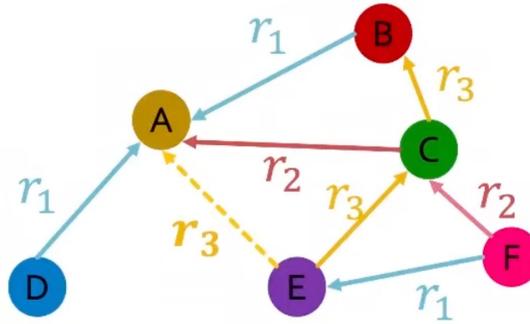
- Intuition, with various types of relations

- Kipf and Welling, 2017



$$h_i^{(l+1)} = \sigma \left(\underbrace{\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{C_{i,r}} W_r^{(l)} h_j^{(l)}}_{\text{Neighborhood}} + \underbrace{W_0^{(l)} h_i^{(l)}}_{\text{Self-connection}} \right)$$

Link prediction with GCN



- (E, r_3, A) ?
- Use GCN to score (E, r_3, A) :
 - Final layers for A and E provides their embeddings h_A^L, h_E^L
 - With a relation specific function:

$$f_r(h_E, h_A) = h_E^T W_r h_A^L$$

- Define a set of positive and of negative (always a tradeoff!)
 - Define a loss, e.g., cross entropy
-
- Other tasks from graph embeddings: node **classification, clustering, ...**

Exemples d'applications

Complétion de réseau d'interaction protein-protein

Zitnik M, et al, 2019

Repositionnement de médicaments

Gysi DM, et al, 2021

Classification automatique de types cellulaire (*single cell*)

Wang S, et al, 2021

Explication des mécanismes
des traitements

Ruiz C, et al. 2021

Explication des mécanismes
des effets secondaires

Bresso E, et al, 2021

...

Identification de pharmacogènes

Dalleau K, et al. 2017

- 2 façons d'acquérir des connaissances
 - Raisonnement symbolique
 - Apprentissage supervisé

- Question de recherche :
Comment combiner les deux ?



Des pistes à partir d'embeddings de graphes de connaissances (TransE, MulE, etc.)
où *embedding et fonction de perte sont adaptées en fonction du type de nœuds du graphe (e.g., objet vs. Classe d'une hiérarchie ou objet vs. valeur)*

Merci pour votre attention

Contact: `adrien.coulet@inria.fr`

References

- Lisa Ehrlinger and Wolfram Wöß. “Towards a Definition of Knowledge Graphs”. In: Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - Vol. 1695. CEUR Workshop Proceedings. CEUR-WS.org, 2016. url: <http://ceur-ws.org/Vol-1695/paper4.pdf>.
- Heiko Paulheim. “Knowledge graph refinement: A survey of approaches and evaluation methods”. In: Semantic Web 8.3 (2017), pp. 489–508. doi: 10.3233/SW-160218. <url:https://doi.org/10.3233/SW-160218>.
- Aidan Hogan *et al.* 2021. Knowledge Graphs. ACM Comput. Surv. 54, 4, Article 71 (May 2022), 37 pages. <https://doi.org/10.1145/3447772>.
- Gysi DM, et al. Network Medicine Framework for Identifying Drug Repurposing Opportunities for COVID-19. ArXiv [Preprint]. 2020 Apr 15:arXiv:2004.07229v2. Update in: Proc Natl Acad Sci U S A. 2021 <https://pubmed.ncbi.nlm.nih.gov/32550253/>
- Wang S, et al. Leveraging the Cell Ontology to classify unseen cell types. Nat Commun. 2021 Sep 21;12(1):5556. doi: 10.1038/s41467-021-25725-x. PMID: 34548483 <https://pubmed.ncbi.nlm.nih.gov/34548483/>
- Ruiz C, et al. Identification of disease treatment mechanisms through the multiscale interactome. Nat Commun. 2021 Mar 19;12(1):1796. doi: 10.1038/s41467-021-21770-8. PMID: 33741907
- Zitnik M, et al. Evolution of resilience in protein interactomes across the tree of life. Proc Natl Acad Sci U S A. 2019 Mar 5;116(10):4426-4433. doi: 10.1073/pnas.1818013116. Epub 2019 Feb 14. PMID: 30765515
- Bresso E, et al.. Investigating ADR mechanisms with Explainable AI: a feasibility study with knowledge graph mining. BMC Med Inform Decis Mak. 2021 May 26;21(1):171. doi: 10.1186/s12911-021-01518-6. PMID: 34039343
- Dalleau K, et al. Learning from biomedical linked data to suggest valid pharmacogenes. J Biomed Semantics. 2017 Apr 20;8(1):16. doi: 10.1186/s13326-017-0125-1. PMID: 28427468;
- Thomas N. Kipf and Max Welling (2017). “Semi-Supervised Classification with Graph Convolutional Networks”. In: ICLR 2017, <https://openreview.net/forum?id=SJU4ayYgl>.