Introduction à RDF

Michel Gagnon

Plan

- Présentation de RDF
- Sérialisations N-Triples, RDF/XML et Turtle
- Sémantique et inférence

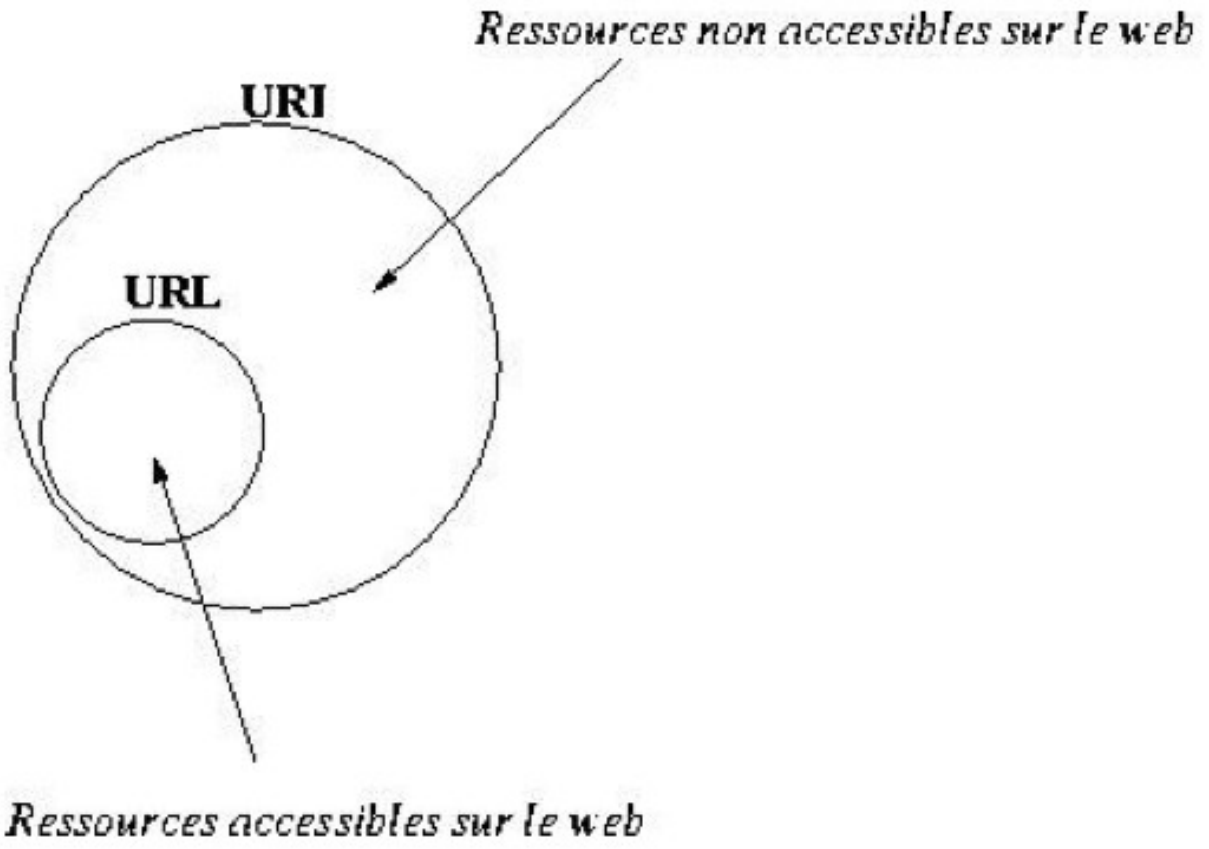
RDF

- Modèle de données pour décrire des ressources du web
- Graphe:
  - les noeuds représentent des ressources
  - les arcs représentent des relations entre ces ressources
- Les ressources sont représentées par leur URI

RDF

- Le graphe est représenté par un ensemble d'énoncés (statements)
- Un énoncé est un triplet <S, P, O>, où
  - S est le sujet
  - P est le prédicat (une propriété)
  - O est l'objet (la valeur de la propriété pour le sujet en question)

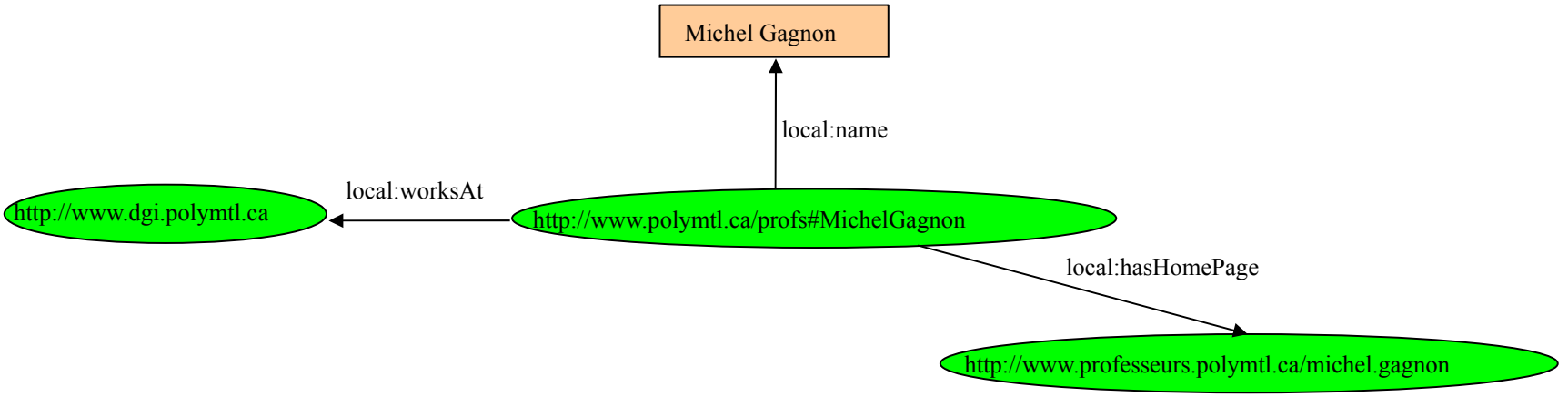
URI


Ressources non accessibles sur le web
URI
URL
Ressources accessibles sur le web

URIref

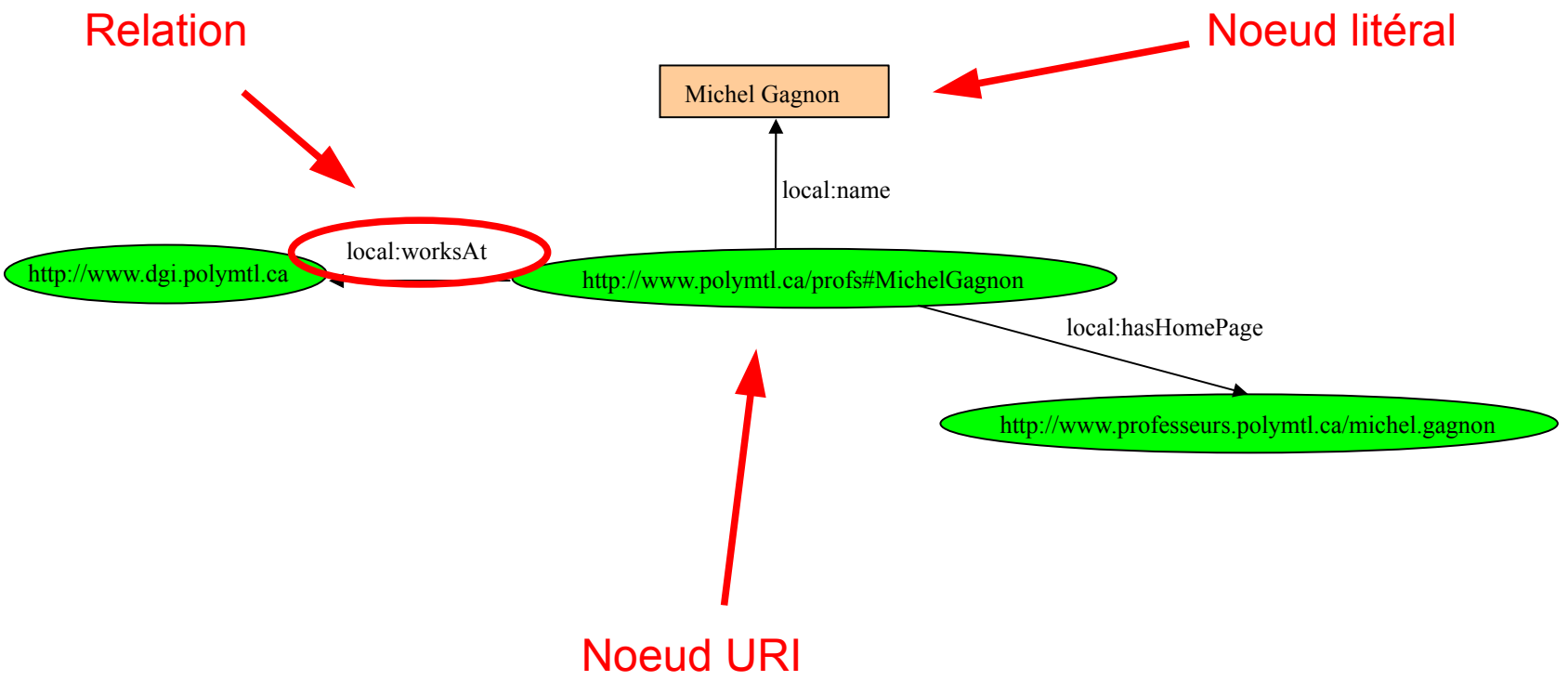
- Plus précisément, les ressources sont identifiées par des URIrefs, c'est-à-dire URI + identificateur de fragment:
  - http://www.polymtl.ca/Profs (URI)
  - #MichelGagnon (Fragment)
  - http//www.polymtl.ca/Profs#MichelGagnon
- En HTML, ceci permet de désigner une section dans un document, alors que pour RDF il ne s'agit que d'un nom donné à une ressource

RDF – Exemple

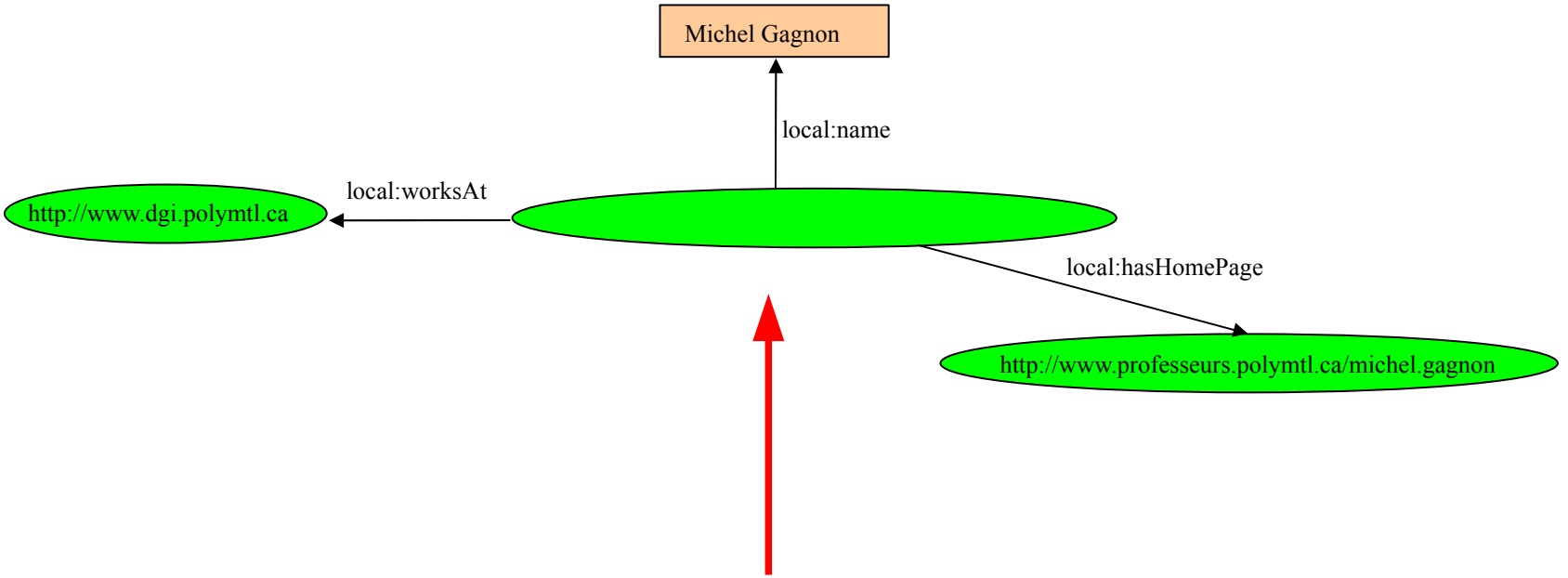


local: http://www.polymtl.ca/vocab#

RDF – Exemple


Relation
Noeud litéral
Michel Gagnon
local:name
local:worksAt
http://www.dgi.polymtl.ca
http://www.polymtl.ca/profs#MichelGagnon
local:hasHomePage
http://www.professeurs.polymtl.ca/michel.gagnon
Noeud URI

RDF – Exemple



Un nœud peut être vide

Syntaxe abstraite

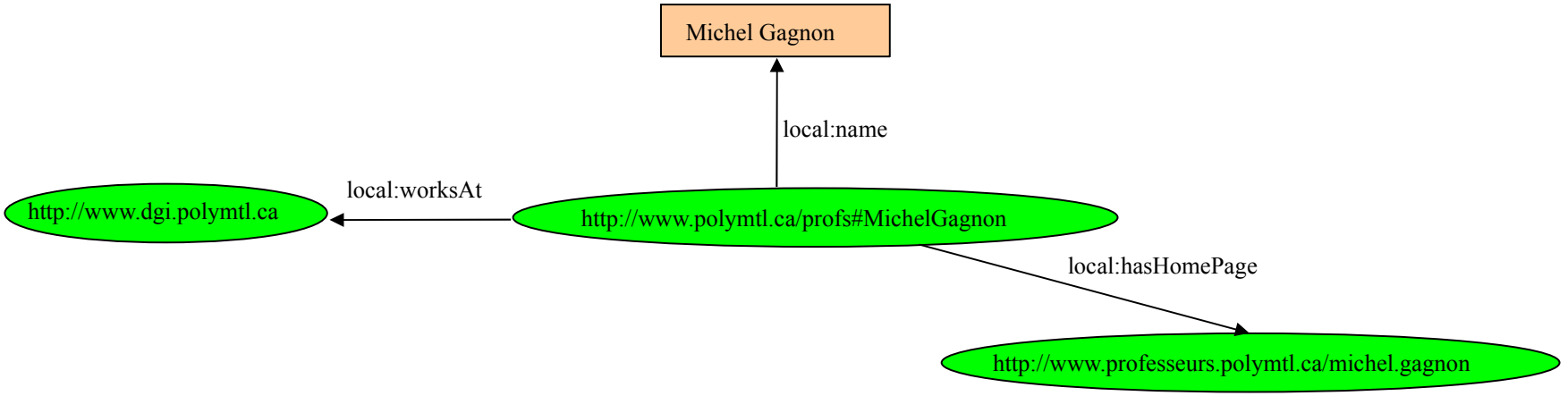
- Collection de triplets
- Une telle collection forme un *graphe RDF*
- Puisque les propriétés sont désignées par des URI, on peut donc les décrire comme n'importe quelle ressource
- Un noeud peut être :
  - Une URI
  - Un littéral
  - Un nœud vide (il désigne en quelque sorte une ressource dont on ne connait pas le nom)
- Deux types de littéraux :
  - Simple: "Michel Gagnon"^^xsd:String
  - Typé: "10"^^xsd:integer

Sérialisation RDF/XML

- Utilise les espaces de nommage
- Balise `rdf:Description` pour regrouper les descriptions d'une ressource
- Pour un nœud vide, on retire l'attribut *about*
- Pour étiqueter un noeud vide, on utilise la balise

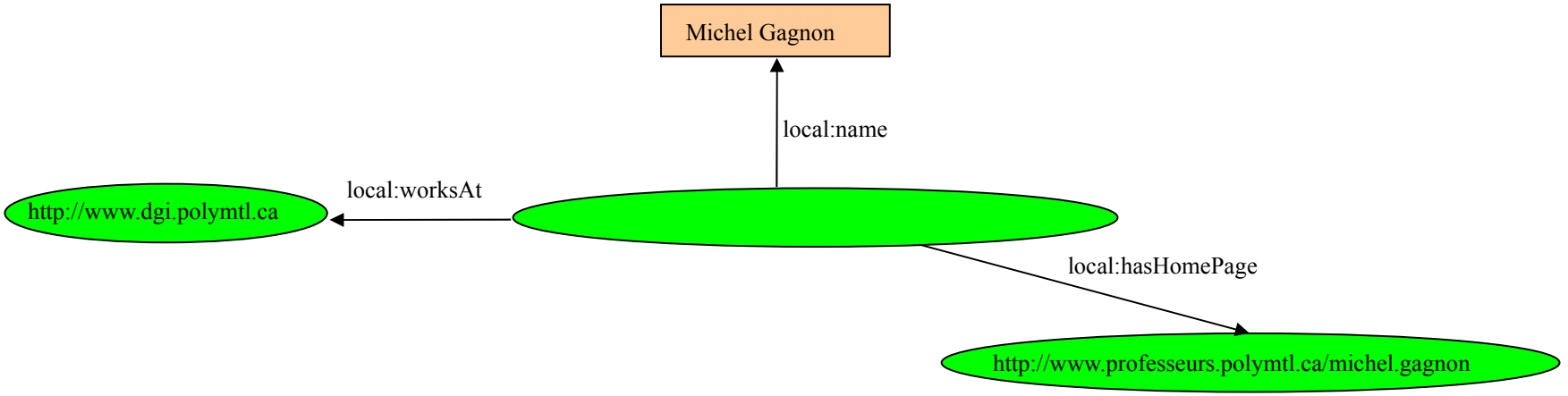
  `rdf:nodeID`
- Pour représenter un littéral typé, on utilise l'attribut `rdf:datatype` dans le prédicat qui relie la ressource à ce littéral
- Il y a souvent plusieurs manières de représenter le même graphe RDF

Sérialisation RDF/XML



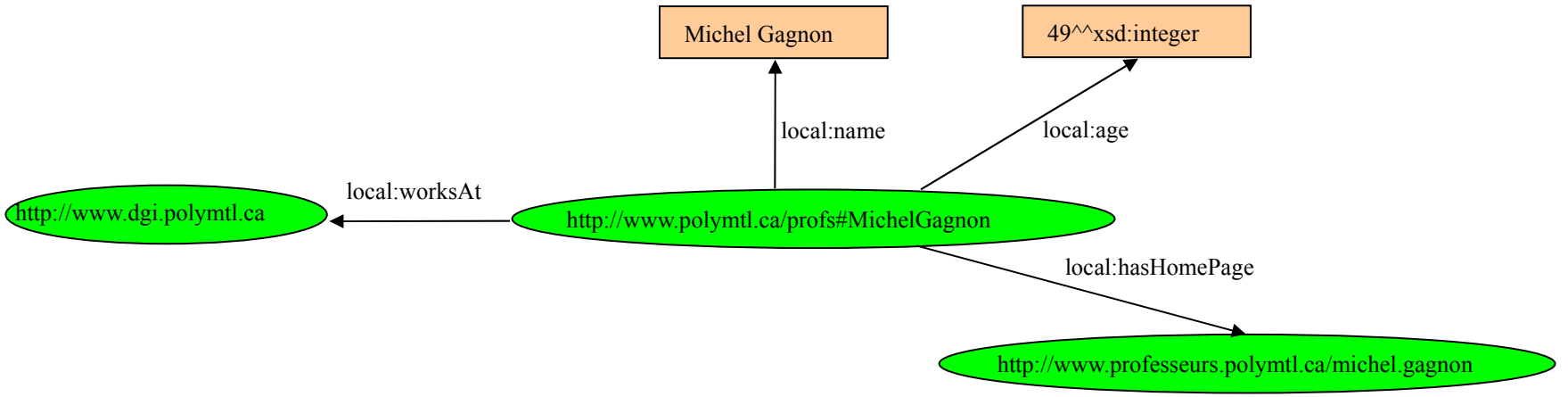

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:local="http://www.polymtl.ca/vocab#">
   <rdf:Description rdf:about="http://www.polymtl.ca/profs#MichelGagnon">
      <local:hasHomePage
            resource="http://www.professeurs.polymtl.ca/michel.gagnon"/>
      <local:worksAt resource="http://www.dgi.polymtl.ca"/>
      <local:name> Michel Gagnon  </local:name>
   </rdf:Description>
</rdf:RDF>
```

Sérialisation RDF/XML




```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:local="http://www.polymtl.ca/vocab#">
   <rdf:Description>
      <local:hasHomePage
            resource="http://www.professeurs.polymtl.ca/michel.gagnon"/>
      <local:worksAt resource="http://www.dgi.polymtl.ca"/>
      <local:name> Michel Gagnon  </local:name>
   </rdf:Description>
</rdf:RDF>
```

Sérialisation RDF/XML

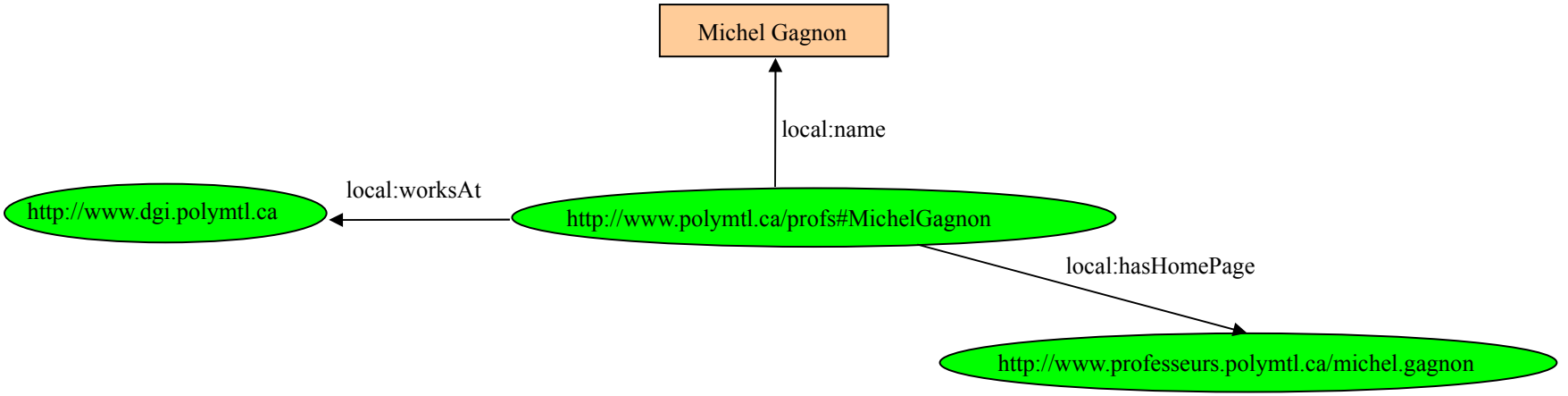



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:local="http://www.polymtl.ca/vocab#">
   <rdf:Description rdf:about="http://www.polymtl.ca/profs#MichelGagnon">
      <local:hasHomePage
            resource="http://www.professeurs.polymtl.ca/michel.gagnon"/>
      <local:worksAt resource="http://www.dgi.polymtl.ca"/>
      <local:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">49</local:age>
      <local:name> Michel Gagnon  </local:name>
   </rdf:Description>
</rdf:RDF>
```

Sérialisation Turtle

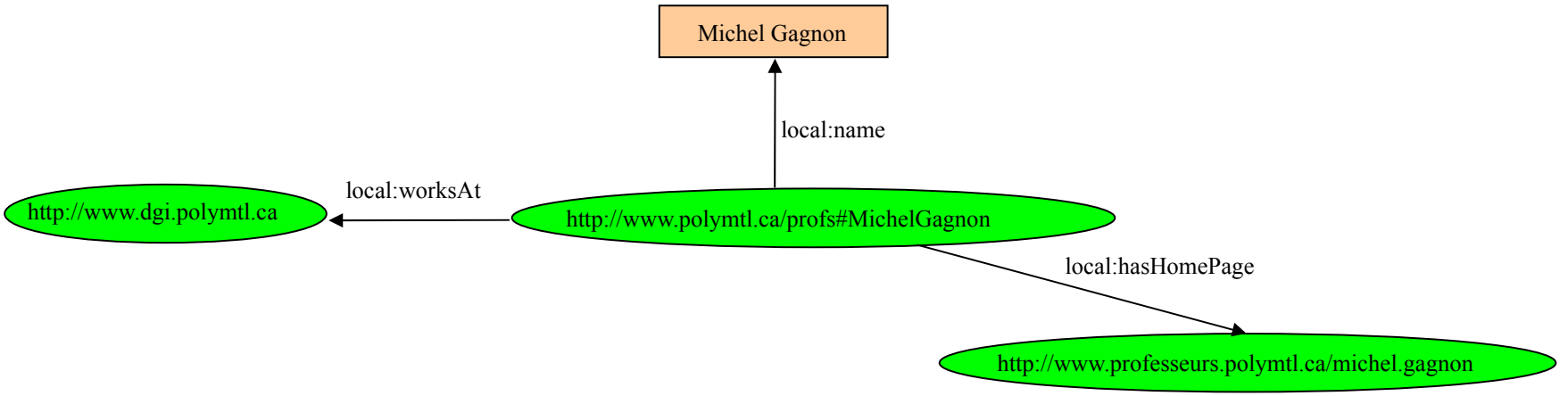
- Permet de spécifier des préfixes
- Permet de combiner des descriptions d'une même ressource :
  - On utilise ; pour grouper des triplets concernant un même sujet
  - On utilise , pour grouper plusieurs instances d'une propriété concernant un même sujet
- Noeud vide représenté par les crochets [ ]
- Toutes les descriptions relatives à un noeud vide peuvent être placées à l'intérieur des crochets

Sérialisation Turtle



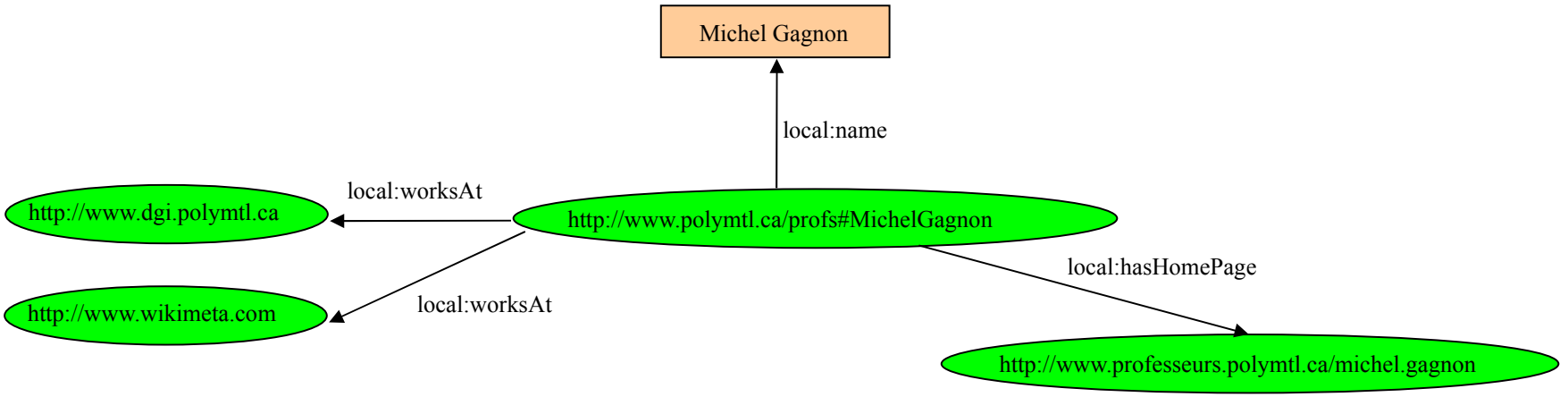

```
@prefix  local: <http://www.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.polymtl.ca/profs#> .

prof:MichelGagnon
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> .

prof:MichelGagnon local:worksAt  <http://www.dgi.polymtl.ca> .

prof:MichelGagnon local:name "Michel Gagnon" .
```

Sérialisation Turtle



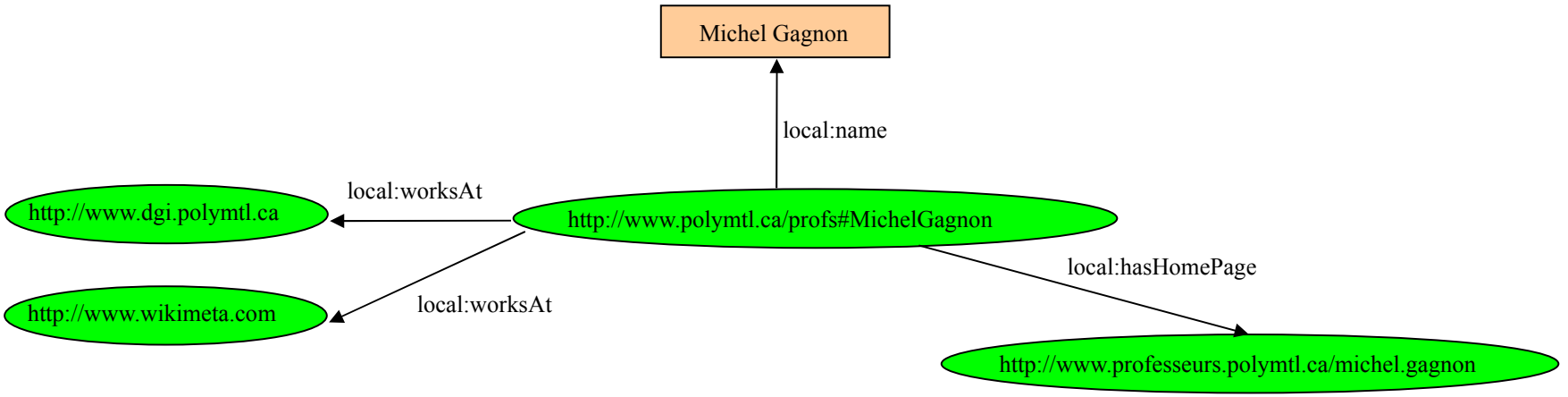

```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

prof:MichelGagnon
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ;
    local:name "Michel Gagnon" .
```

Sérialisation Turtle



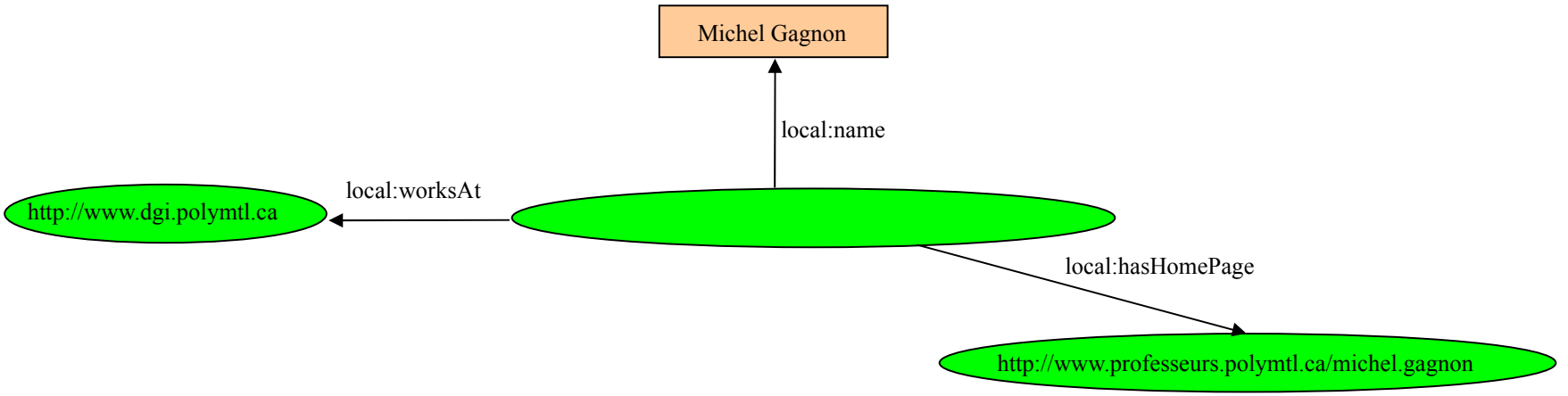

```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

prof:MichelGagnon
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ;
    local:worksAt  <http://www.wikimeta.com> ;
    local:name "Michel Gagnon" .
```

Sérialisation Turtle



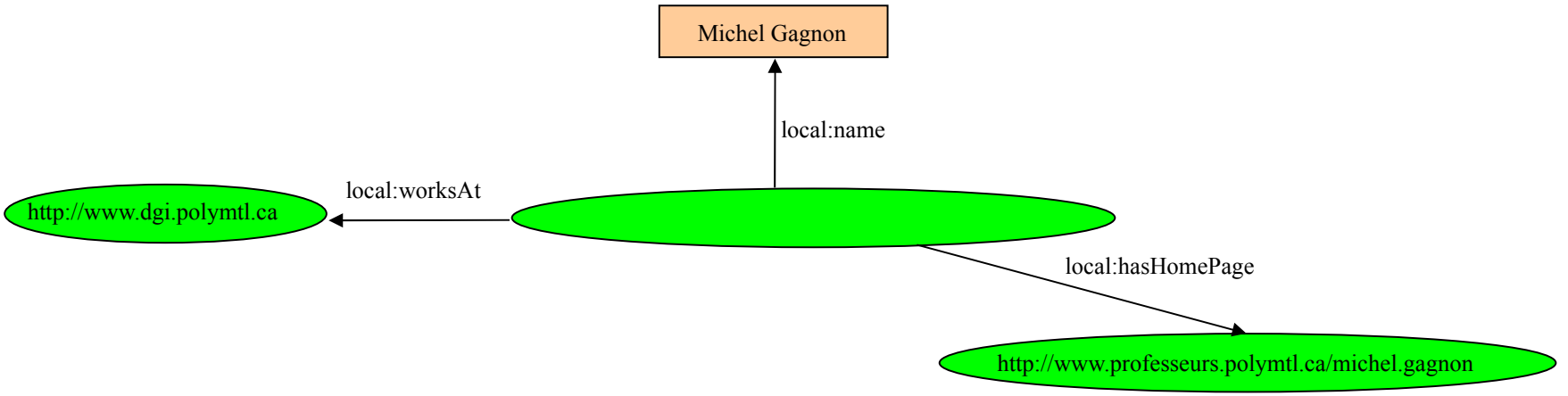

```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

prof:MichelGagnon
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ,
                   <http://www.wikimeta.com> ;
    local:name "Michel Gagnon" .
```

Sérialisation Turtle



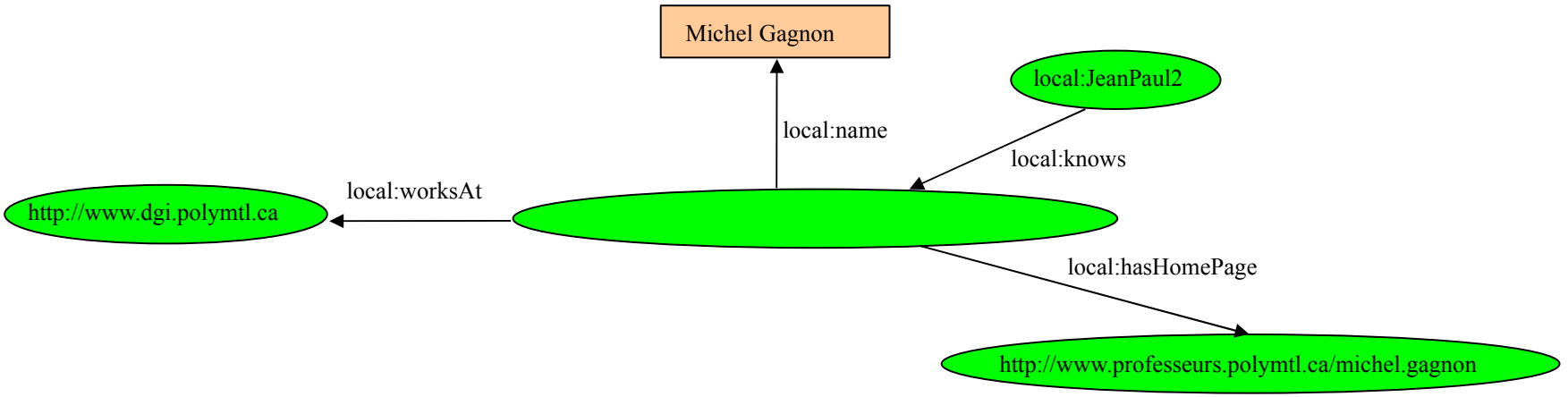

```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

[ ]
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ;
    local:name "Michel Gagnon" .
```

Sérialisation Turtle



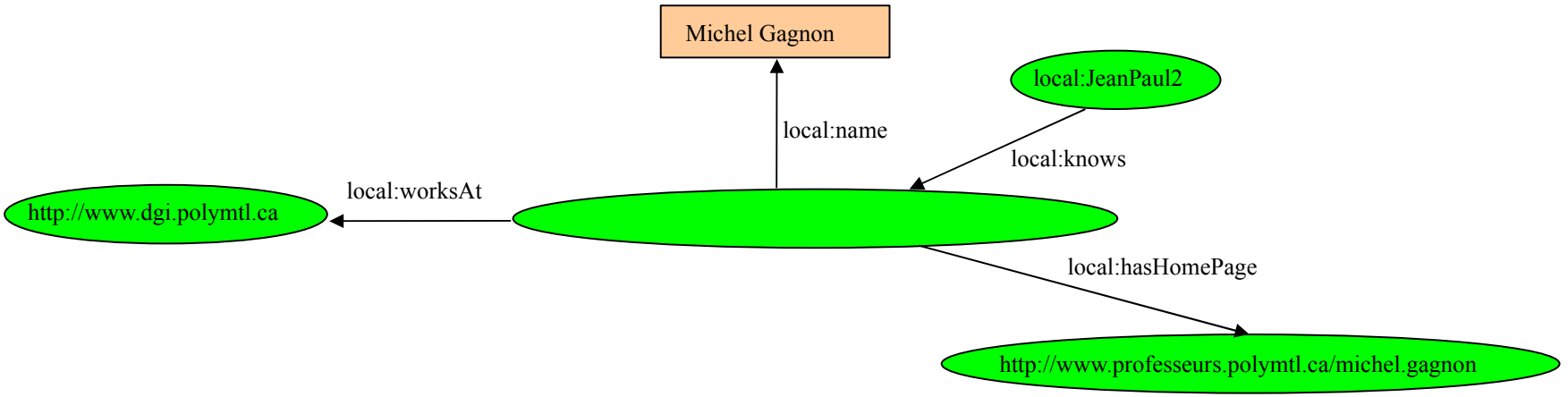

```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

[
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ;
    local:name "Michel Gagnon" .
] .
```

Sérialisation Turtle




```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

_:n1
    local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
    local:worksAt  <http://www.dgi.polymtl.ca> ;
    local:name "Michel Gagnon" .

local:JeanPaul2 local:knows   _:n1 .
```

Sérialisation Turtle

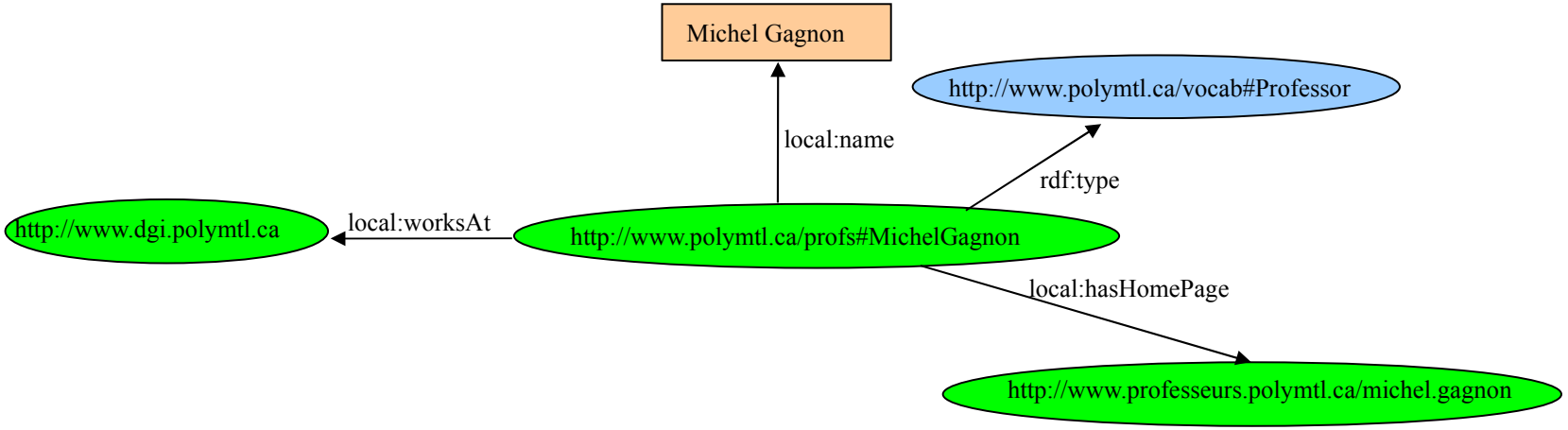



```
@prefix  local: <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:  <http://www.dgi.polymtl.ca/profs#> .

local:JeanPaul2 local:knows
    [ local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
      local:worksAt  <http://www.dgi.polymtl.ca> ;
      local:name "Michel Gagnon" ] .
```

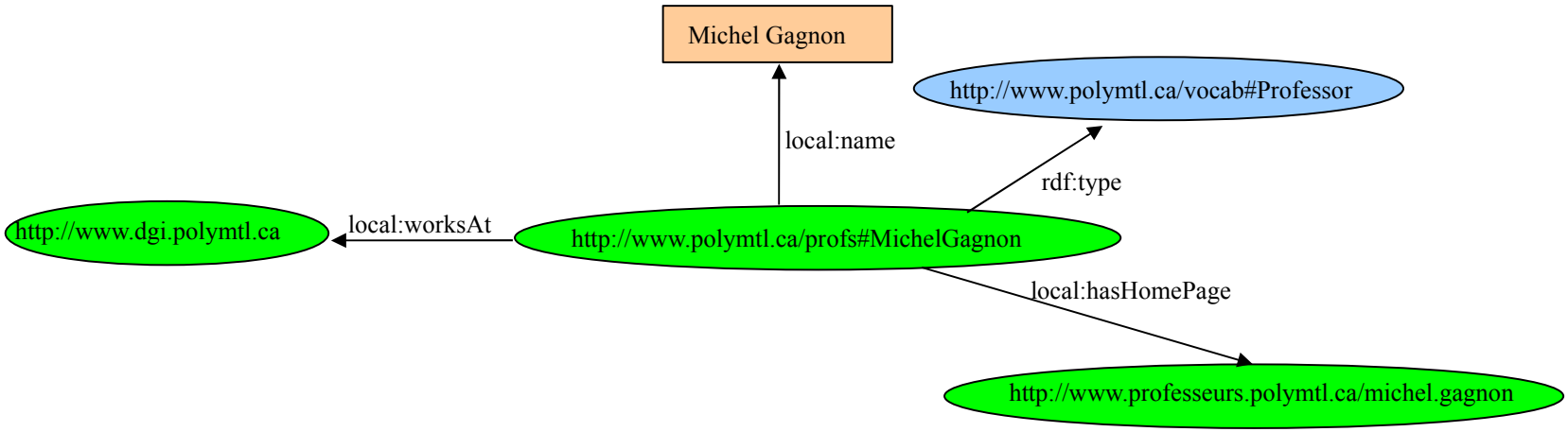
RDF – Déclaration de type de ressource

- Pour identifier le type d'une ressource:
  - Utiliser le prédicat rdf:type pré-défini par RDF
  - Remplacer la balise rdf:Description par le type de la ressource
- À noter qu'une ressource peut avoir plusieurs types
- En Turtle, on peut utiliser le prédicat `a`

Type – Exemple

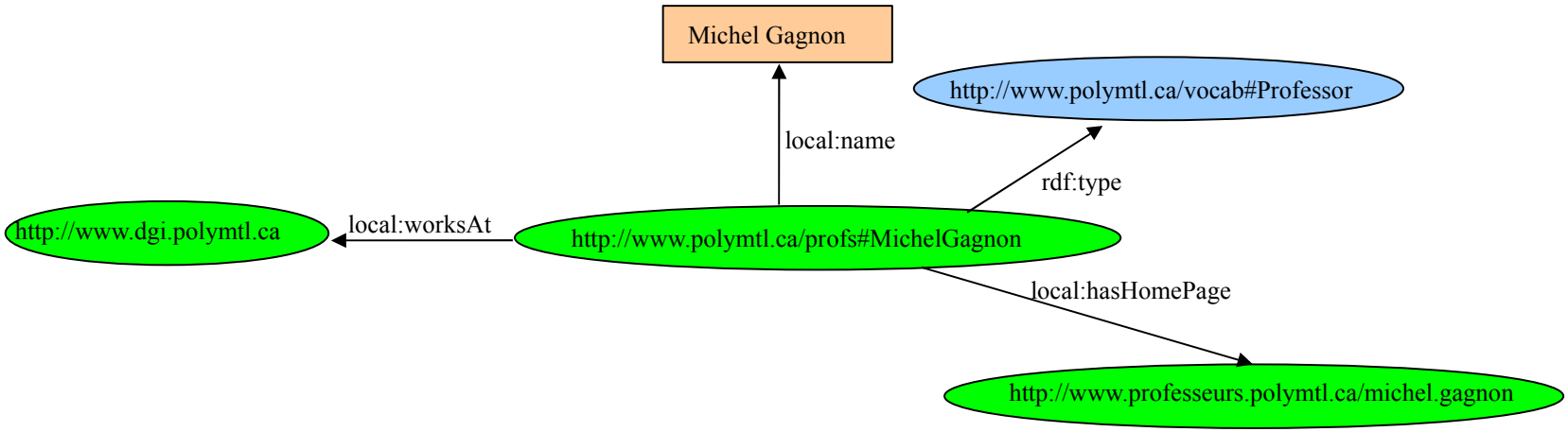


Type – Exemple - Turtle




```
@prefix  local:  <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:   <http://www.dgi.polymtl.ca/profs#> .

prof:MichelGagnon
        rdf:type   local:Professor ;
        local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
        local:worksAt resource  <http://www.dgi.polymtl.ca> ;
        local:name  “Michel Gagnon” .
```

Type – Exemple - Turtle

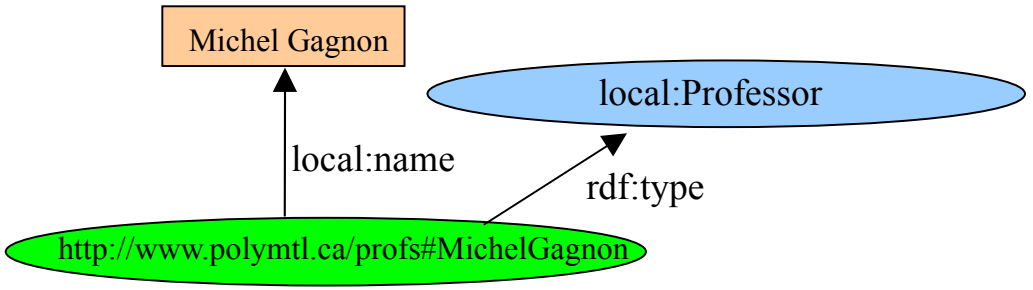



```
@prefix  local:  <http://www.dgi.polymtl.ca/vocab#> .
@prefix  prof:   <http://www.dgi.polymtl.ca/profs#> .

prof:MichelGagnon
        a   local:Professor ;
        local:hasHomePage  <http://www.professeurs.polymtl.ca/michel.gagnon> ;
        local:worksAt resource  <http://www.dgi.polymtl.ca> ;
        local:name  “Michel Gagnon” .
```

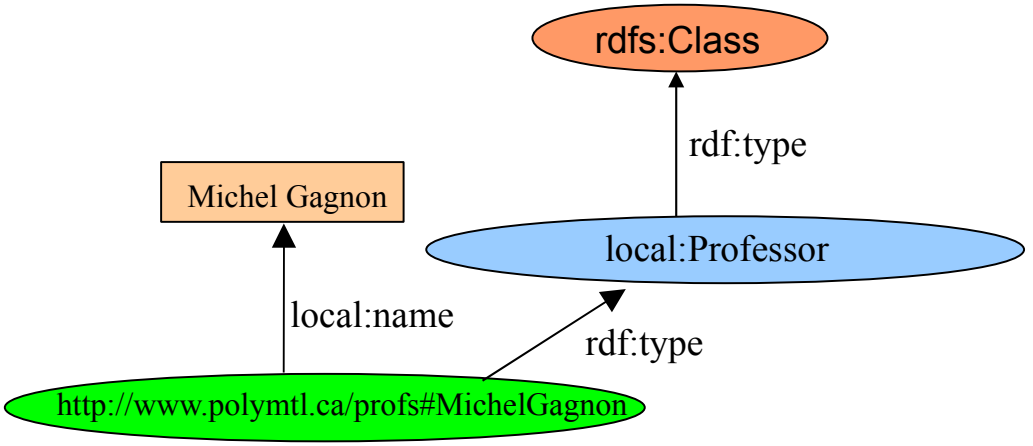
RDF Schema - Classes

- Une ressource peut appartenir à plus d'une classe
- Un type appartient à la classe `rdfs:Class`
- RDFS permet de définir une hiérarchie de classes, grâce au prédicat `rdfs:subClassOf`

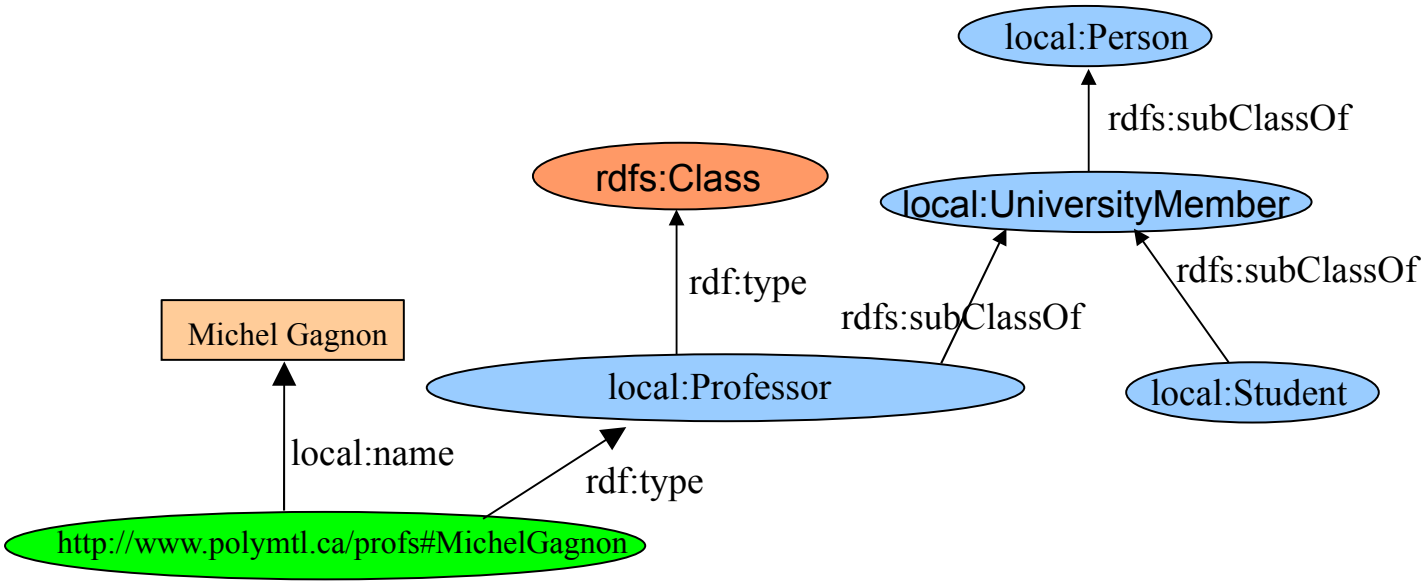
RDF Schema - Classes



RDF Schema - Classes



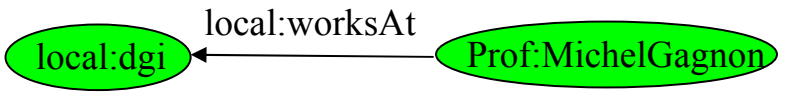
RDF Schema - Classes

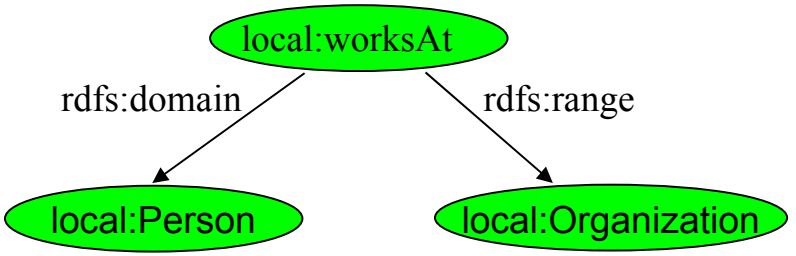


RDF Schema – Propriétés

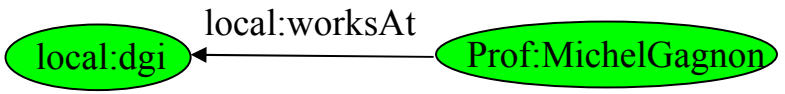
- Toutes les propriétés ont pour type la classe `rdf:Property`
- On peut établir des hiérarchies de propriétés, grâce au prédicat `rdfs:subPropertyOf`
- On peut définir le domaine et l'image d'une propriété, en utilisant les prédicats `rdfs:domain` et `rdfs:range`, respectivement
- Les propriétés sont globales (on peut donc y ajouter des informations n'importe où)

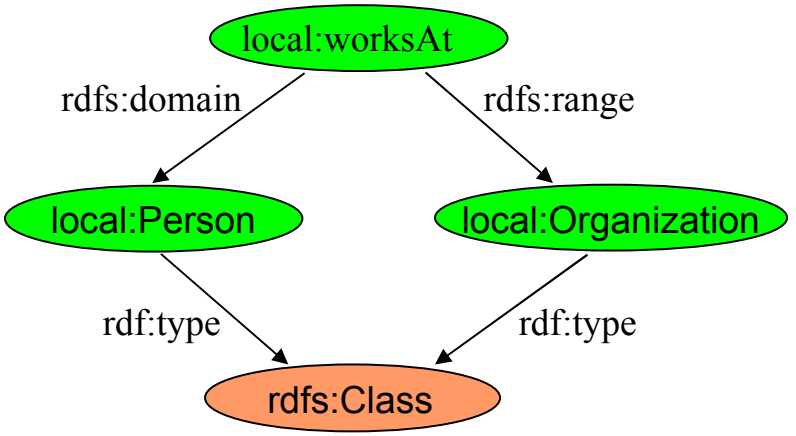
RDF Schema – Propriétés



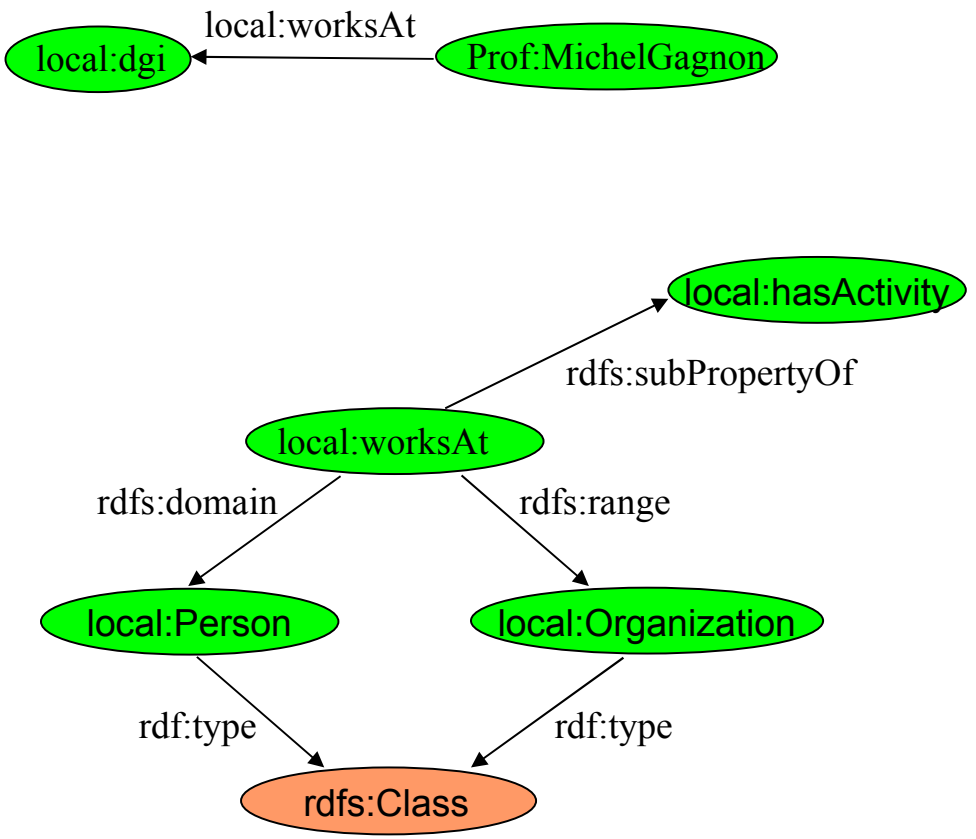


RDF Schema – Propriétés



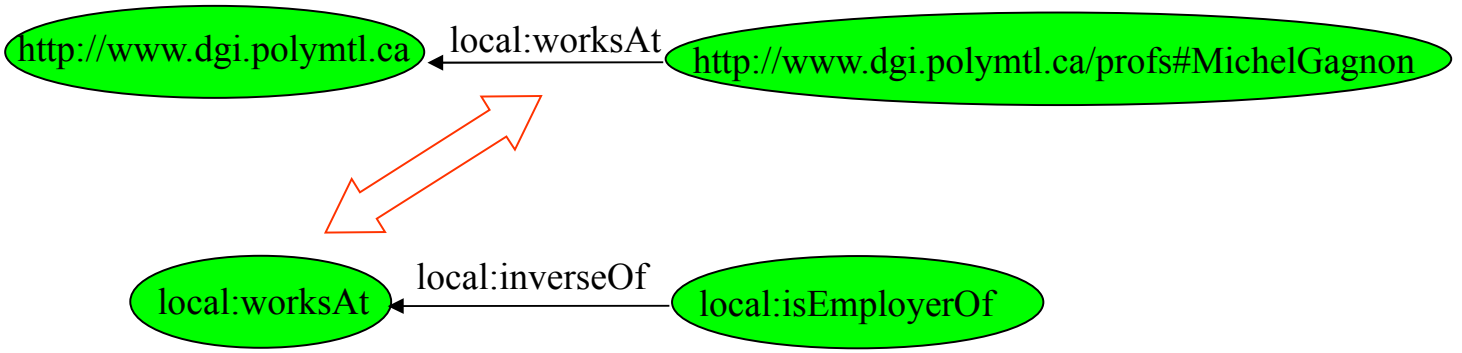


RDF Schema – Propriétés



RDF Schema – Propriétés

- En RDF tout est une ressource, même les propriétés
- Ceci signifie qu'on peut ajouter des descriptions aux propriétés:



Exercice – Dessinez le graphe :

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix poly: <http:/www.polymtl.ca/vocab#> .

poly:x1 poly:p <http://www.polymtl.ca> ;
        a poly:A.
```

Exercice – Dessinez le graphe :

```
@prefix poly: <http://www.polymtl.ca#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

poly:x1 poly:p [];
        a poly:A .
[] poly:q <http://www.polymtl.ca> .
```

Exercice – Dessinez le graphe :

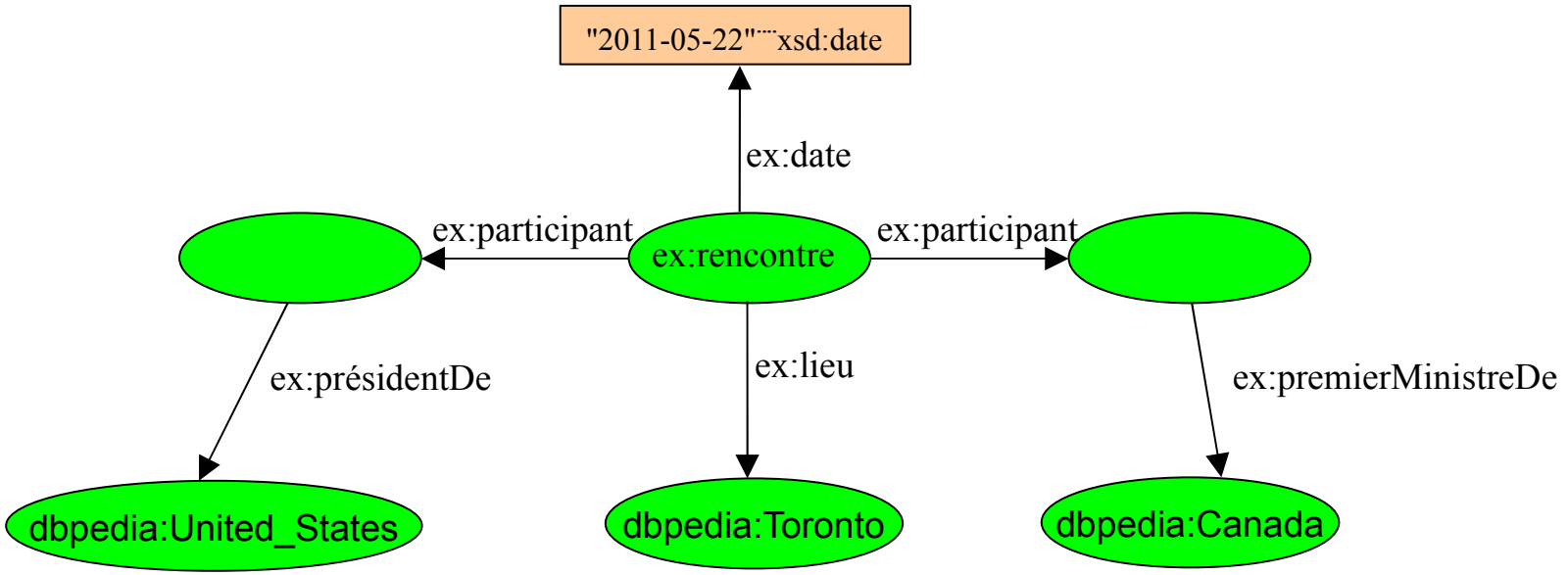
```
@prefix poly: <http://www.polymtl.ca#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

poly:x1 poly:p [ poly:w poly:x2 ];
        a poly:A .
poly:x2 poly:q <http://www.polymtl.ca> .
```

Exercice – Représenter en RDF :

Le président des États-Unis a rencontré le premier ministre du Canada à Toronto le 22 mai 2011.

Exercice – Représenter en RDF :

Le président des États-Unis a rencontré le premier ministre du Canada à Toronto le 22 mai 2011.



Inférence

Si on a

```
aaa p bbb .
```

On peut inférer

```
_:n1 p bbb .
```

`Si aaa` a déjà été remplacé par `_:n1` auparavant.
Sinon, il faut créer un nouveau nœud vide.

Inférence

Si on a

```
aaa p bbb .
```

On peut inférer

```
aaa p _:n1 .
```

`Si bbb` a déjà été remplacé par `_:n1` auparavant.
Sinon, il faut créer un nouveau nœud vide.

Inférence - Exemple

Soit

```
local:MichelGagnon local:worksAt local:dgi .
local:JeanStJean local:worksAt local:dgi .
```

On peut inférer

```
local:MichelGagnon local:worksAt _:n1 .
```

Inférence - Exemple

Soit

```
local:MichelGagnon local:worksAt local:dgi .
local:JeanStJean local:worksAt local:dgi .
```

On peut inférer

```
local:MichelGagnon local:worksAt _:n1 .
local:JeanStJean local:worksAt _:n1 .
```

Inférence - Exemple

Soit

```
local:MichelGagnon local:worksAt local:dgi .
local:JeanStJean local:worksAt local:dgi .
```

On peut inférer

```
local:MichelGagnon local:worksAt _:n1 .
local:JeanStJean local:worksAt _:n1 .
_:n2 local:worksAt _:n1 .
```

Inférence - Exemple

Soit

```
local:MichelGagnon local:worksAt local:dgi .
local:JeanStJean local:worksAt local:dgi .
```

On peut inférer

```
local:MichelGagnon local:worksAt _:n1 .
local:JeanStJean local:worksAt _:n1 .
_:n2 local:worksAt _:n1 .
_:n3 local:worksAt _:n1 .
```

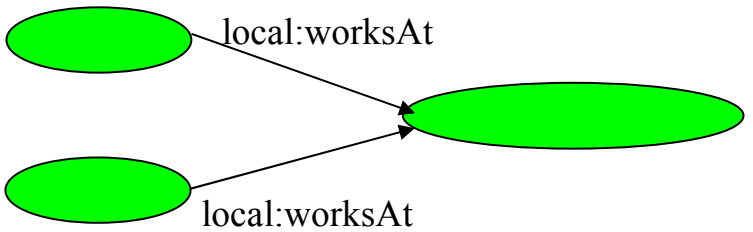
Inférence - Exemple

Ce graphe

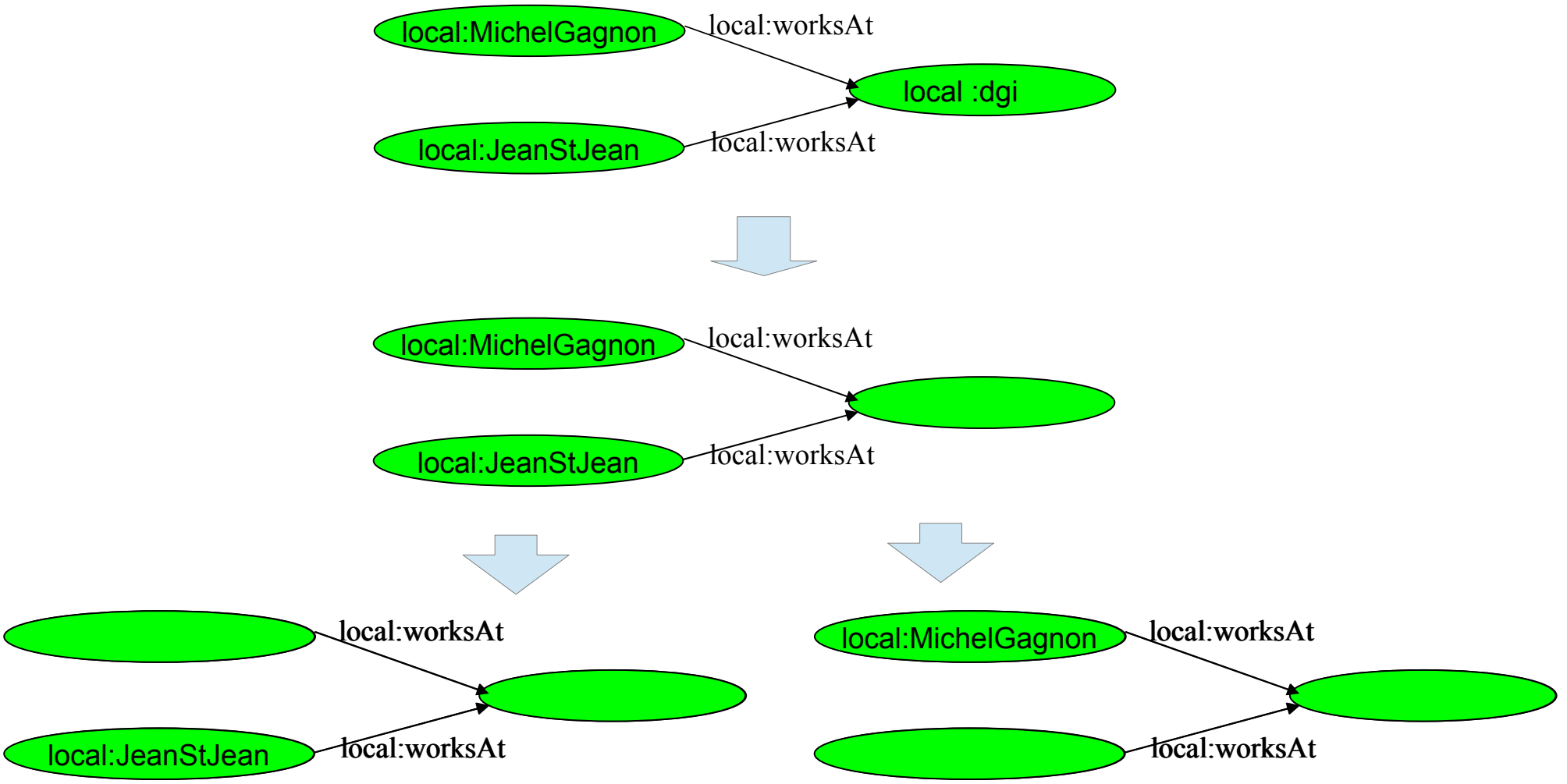
```
_:n2 local:worksAt _:n1 .
_:n3 local:worksAt _:n1 .
```

est équivalent à celui-ci

```
[] local:worksAt _:n1 .
[] local:worksAt _:n1 .
```



Inférence - Exemple



Inférence

Si on a

```
aaa p bbb .
```

On peut inférer

```
p rdf:type rdf:Property .
```

Inférence

Si on a

```
aaa p bbb .
```

On peut inférer

```
aaa rdf:type rdfs:Resource .
```

et

```
bbb rdf:type rdfs:Resource .
```

Inférence

Si on a

```
p rdfs:domain d .
aaa p bbb .
```

On peut inférer

```
aaa rdf:type d .
```

Inférence

Si on a

```
p rdfs:range d .
aaa p bbb .
```

On peut inférer

```
bbb rdf:type d .
```

Inférence - Exemple

Soit

```
local:marieAvec rdfs:domain  local:Homme .
local:marieAvec rdfs:domain  local:Femme .
local:Paul local:marieAvec local:Marie .
```

On peut inférer

```
local:Paul rdf:type local:Homme .
local:Paul rdf:type local:Femme .
```

Donc Paul est à la fois un homme et une femme

Inférence

Si on a

```
c1 rdfs:subClassOf c2 .
aaa rdf:type c1 .
```

On peut inférer

```
aaa rdf:type c2 .
```

Inférence - Exemple

Soit

```
local:Professeur rdfs:subClassOf  local:Person .
local:Michel rdf:type local:Professeur .
```

On peut inférer

```
local:Michel rdf:type local:Person .
```

Inférence

Si on a

```
c1 rdfs:subClassOf c2 .
c2 rdfs:subclassOf c3 .
```

On peut inférer

```
c1 rdf:subClassOf c3 .
```

Inférence

Si on a

```
p1 rdfs:subPropertyOf p2 .
aaa p1 bbb .
```

On peut inférer

```
aaa p2 bbb .
```

Inférence - Exemple

Soit

```
local:aime rdfs:subPropertyOf
      local:eprouveSentimentEnvers .

local:Paul rdf:aime local:Marie .
```

On peut inférer

```
local:Paul rdf:eprouveSentimentEnvers
     local:Marie .
```

Inférence

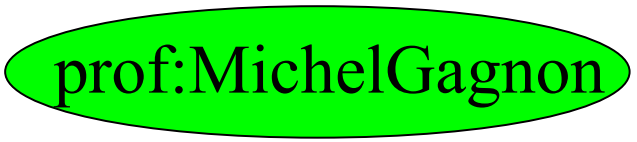
Si on a

```
p1 rdfs:subPropertyOf p2 .
p2 rdfs:subPropertyOf p3 .
```

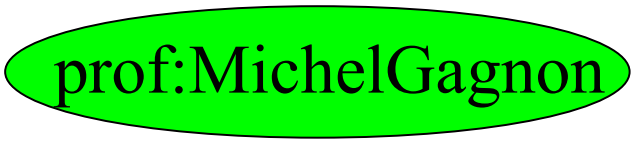
On peut inférer

```
p1 rdf:subPropertyOf p3 .
```

Exercice – Graphe RDF valide?

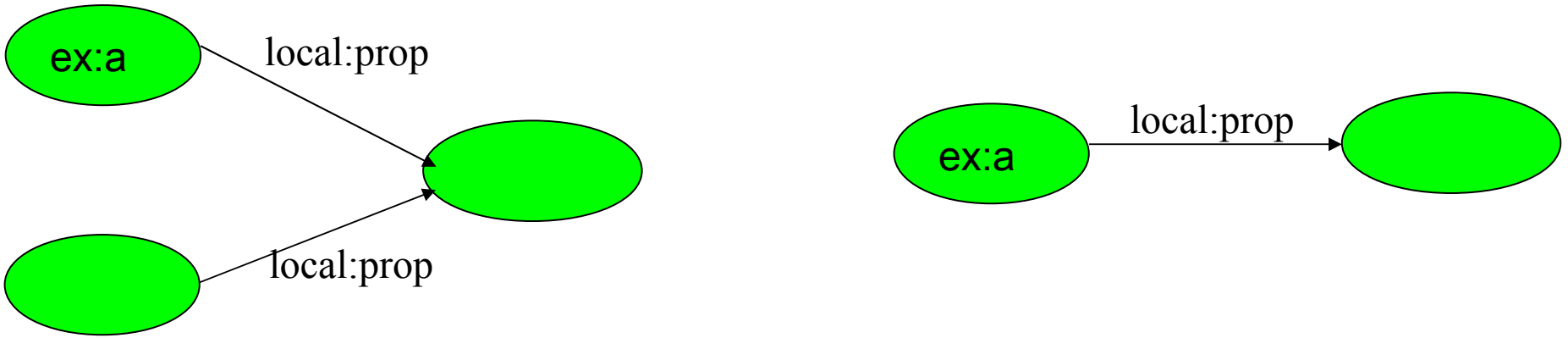


Exercice – Graphe RDF valide?

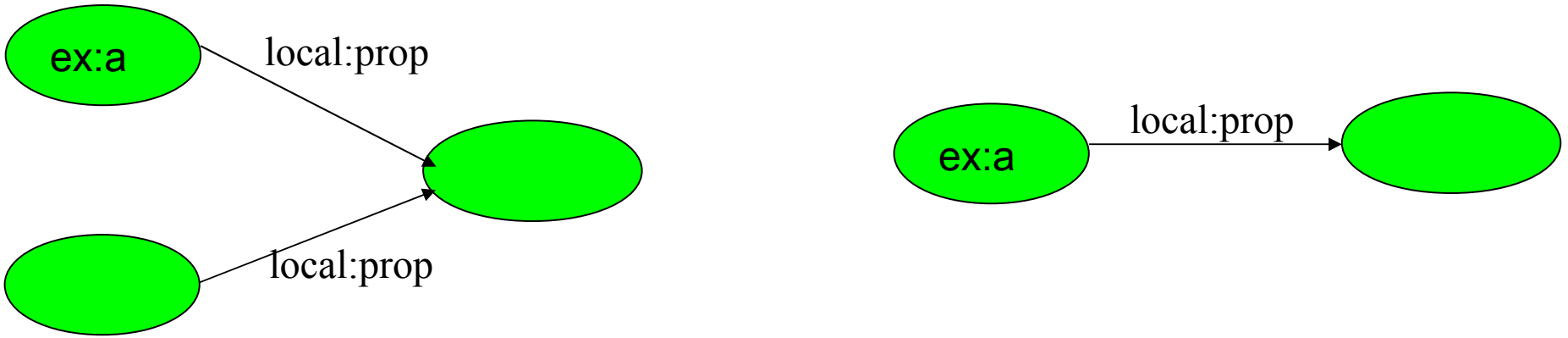


NON!

Exercice – Graphes équivalents?

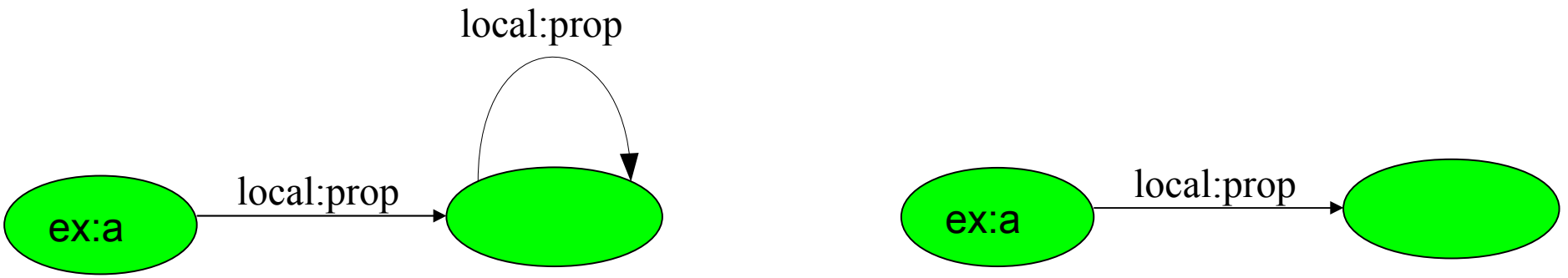


Exercice – Graphes équivalents?

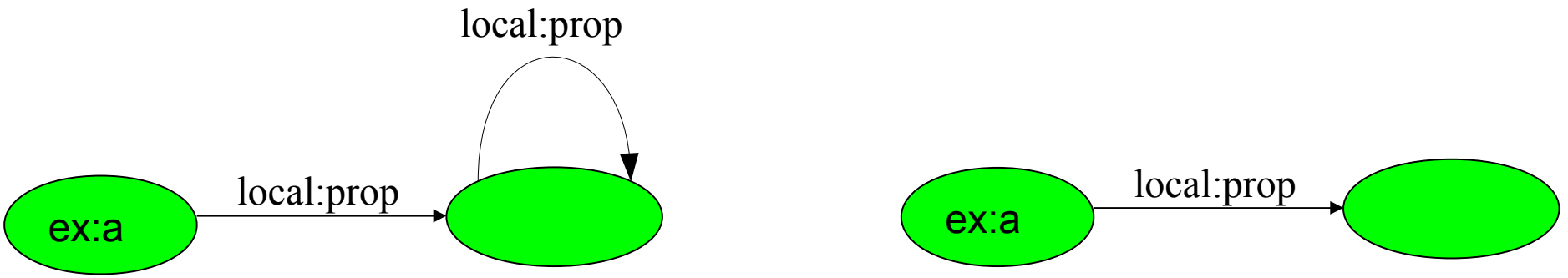


OUI!

Exercice – Graphes équivalents?

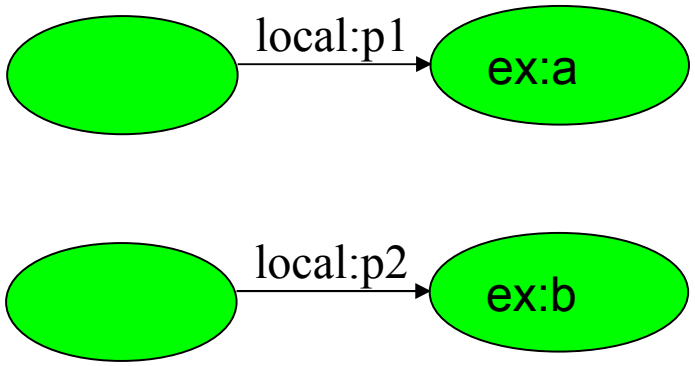


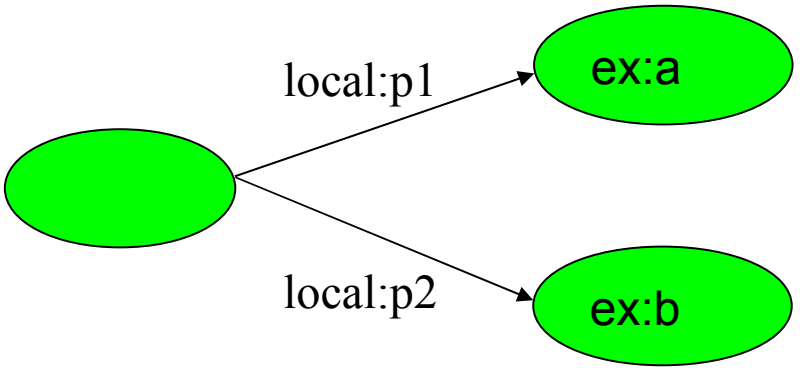
Exercice – Graphes équivalents?



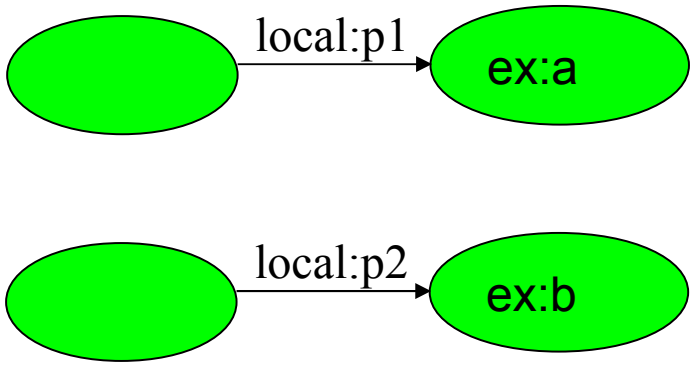
NON!

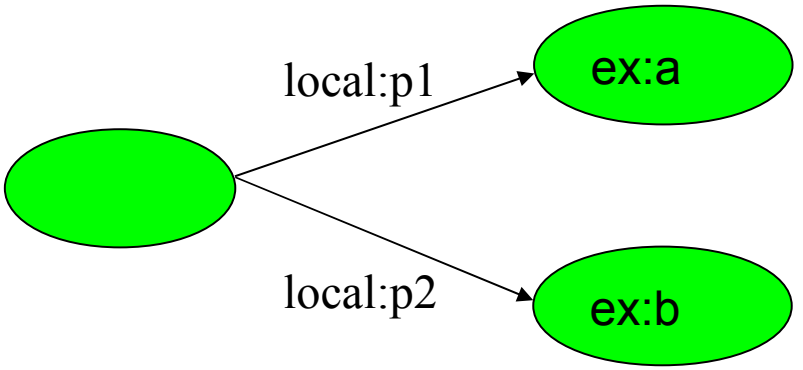
Exercice – Graphes équivalents?





Exercice – Graphes équivalents?





NON!

Montrer qu'un animal a un sentiment envers un autre animal

```
ex:detester rdfs:subPropertyOf ex:avoirSentimentEnvers .
ex:Chat rdfs:subClassOf ex:Animal .
ex:Tom rdf:type ex:Chat .
ex:Jerry rdf:type ex:Animal .
ex:Tom ex:detester ex:Jerry .
```

Monter qu'un animal a un sentiment envers un autre animal

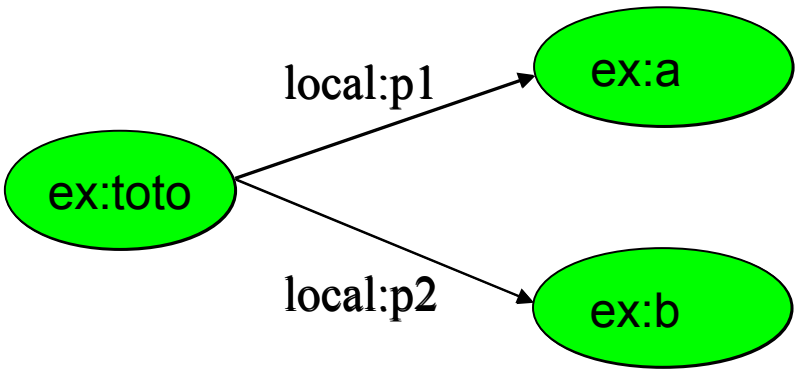
```
(1) ex:detester rdfs:subPropertyOf ex:avoirSentimentEnvers .
(2) ex:Chat rdfs:subClassOf ex:Animal .
(3) ex:Tom rdf:type ex:Chat .
(4) ex:Jerry rdf:type ex:Animal .
(5) ex:Tom ex:detester ex:Jerry .
(6) ex:Tom ex:avoirSentimentEnvers ex:Jerry .      (1,5)
(7) ex:Tom rdf:type ex:Animal .                    (2,3)
(8) _:n1 ex:avoirSentimentEnvers ex:Jerry .        (6)
(9) _:n1 ex:avoirSentimentEnvers _:n2 .            (8)
(10) _:n1 rdf:type ex:Animal .                     (7)
(11) _:n2 rdf:type ex:Animal .                     (4)
```

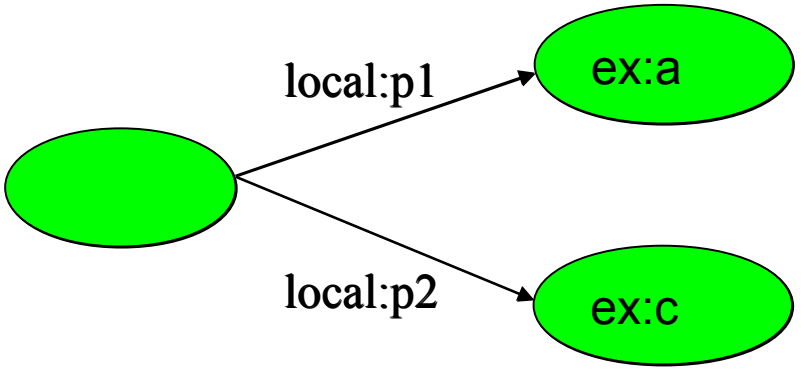
Monter qu'un animal a un sentiment envers un autre animal

```
(1) ex:detester rdfs:subPropertyOf ex:avoirSentimentEnvers .
(2) ex:Chat rdfs:subClassOf ex:Animal .
(3) ex:Tom rdf:type ex:Chat .
(4) ex:Jerry rdf:type ex:Animal .
(5) ex:Tom ex:detester ex:Jerry .
(6) ex:Tom ex:avoirSentimentEnvers ex:Jerry .      (1,5)
(7) ex:Tom rdf:type ex:Animal .                    (2,3)
(8) _:n1 ex:avoirSentimentEnvers ex:Jerry .        (6)
(9) _:n1 ex:avoirSentimentEnvers _:n2 .            (8)
(10) _:n1 rdf:type ex:Animal .                     (7)
(11) _:n2 rdf:type ex:Animal .                     (4)
```

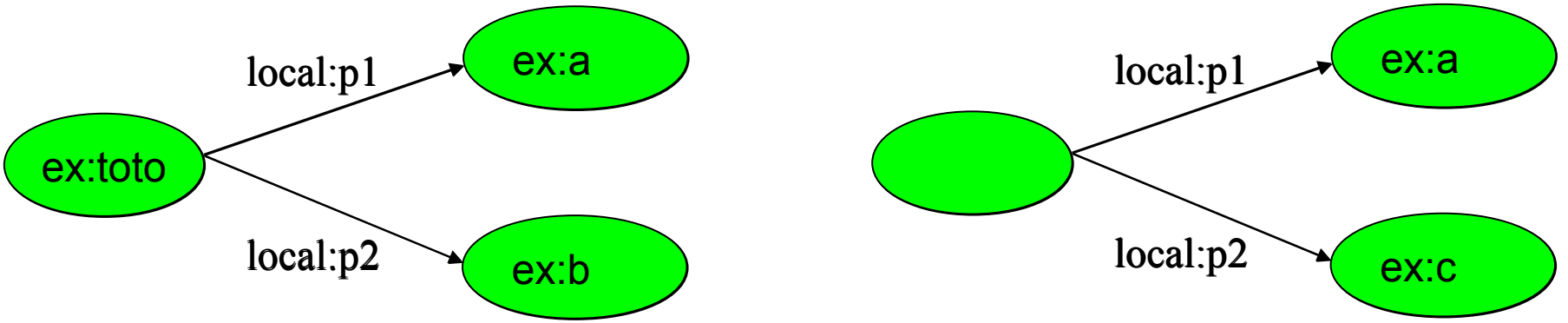
```
[ a ex:Animal ;
  ex:avoirSentimentEnvers  [  a ex:Animal ] ]  .
```

Peut-on avoir une même interprétation pour ces graphes?





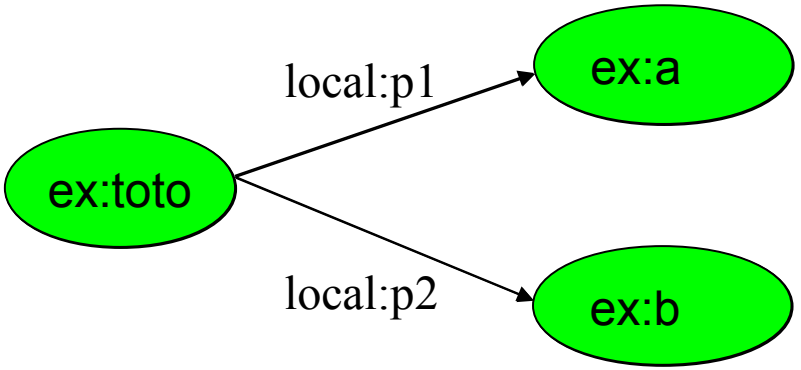
Peut-on avoir une même interprétation pour ces graphes?

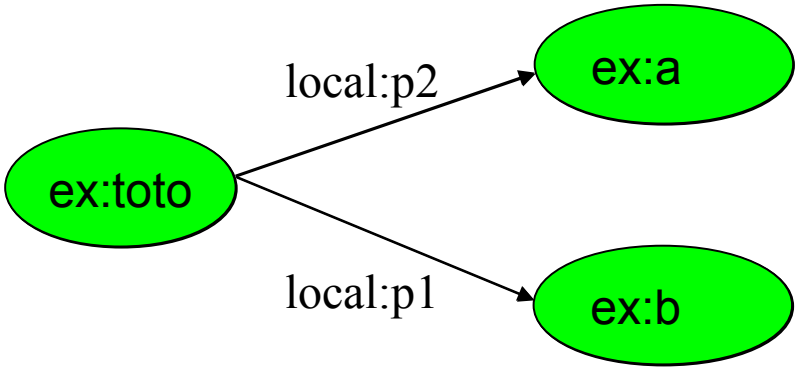


OUI!

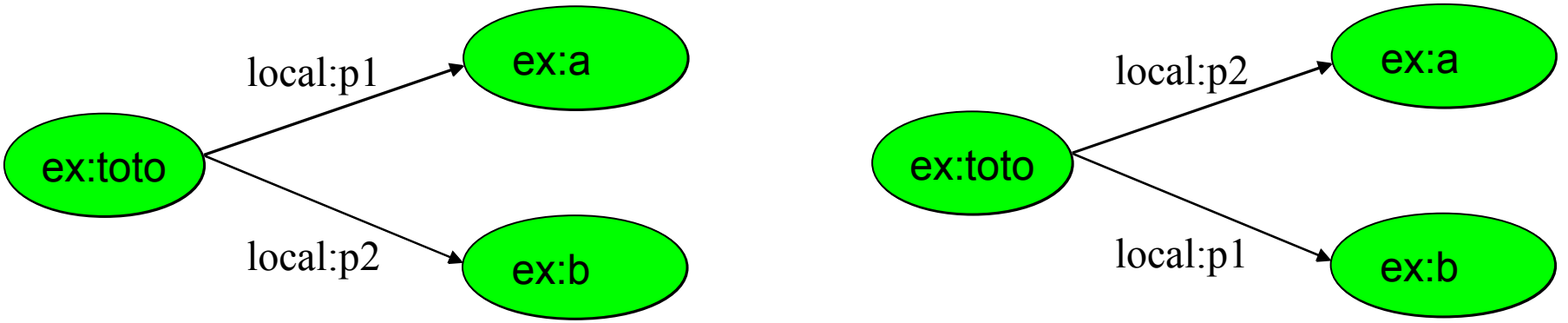
Il suffit que ex:b et ex:c désignent la même entité

Peut-on avoir une même interprétation pour ces graphes?





Peut-on avoir une même interprétation pour ces graphes?



OUI!

Soient T, A et B les entités désignées par ex:toto, ex:a et ex:b, respectivement. Il suffit que les propriétés associées à local:p1 et local:p2 contiennent toutes les deux les paires (T,A) et (T,B).