



L'énoncé de TP

La partie pratique consiste à utiliser un des outils pour simuler le raisonnement en exploitant

les TBOX et les Abox des exercices précédents.

Ce box est composé d'entités atomiques ayant des concepts et des rôles. Les premiers représentant des classes et les deuxièmes leurs méthodes. Par ailleurs, les entités composées sont des sous-ensembles des concepts.

Code Source :

L'installation de library Python Owl Ready Library

```
+ Code + Text

✓ 4s #%%
!pip install owlready2
from owlready2 import *

onto = get_ontology ("http://testxyz.org/onto.owl") # create ontology using iri

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: owlready2 in /usr/local/lib/python3.8/dist-packages (0.39)
```

Looking in indexes: <https://pypi.org/project/owldispatch/>, <https://pypi.org/project/owldispatch/>
Requirement already satisfied: owlready2 in /usr/local/lib/python3.8/dist-packages (0.39)

```
▶ with onto : # defining our ontology

## Defining concepts ##
class joueur ( Thing ):
    pass
class equipe ( Thing ):
    pass
class contry ( Thing ):
    pass
AllDisjoint ([ joueur , equipe , contry ])

## Defining roles ##
class joue ( joueur >> Thing ):
    pass

class Entraîner ( joueur >> Thing ):
    pass

class Entraîner_par ( ObjectProperty ):
    inverse_property = Entraîner

class joue_par ( ObjectProperty ):
    inverse_property = joue

class PartieDe ( Thing >> Thing ):
    pass

## Defining composed entities ##
class club ( Thing ):
    equivalent_to = [ Thing & PartieDe . only ( contry )]

class Entraîneur ( joueur ):
    equivalent_to = [ joueur & Entraîner . some ( joueur )]

class JoueurSelectionne ( Thing ):
    equivalent_to = [ joueur & Entraîner_par . only ( Entraîneur )]

## defining instances ABOX ##

class Messi ( Thing ):
    equivalent_to = [ joueur & Entraîner_par . only ( equipe )]
class pioG (Thing):
```

✓ 16m 28

```

class JoueurSelectione ( Thing ):
    equivalent_to = [ joueur & Entraîneur_par . only ( Entraîneur )]

## defining instances ABOX ##

class Messi ( Thing ):
    equivalent_to = [ joueur & Entraîneur_par . only ( equipe )]
class pipG (Thing):
    equivalent_to = [ Entraîneur & Entraîneur . only ( equipe )]
class FCB ( equipe ):
    equivalent_to = [ club & Entraîneur . some ( joueur ) & Entraîneur_par . only ( pipG )]

AllDisjoint ([ club , Entraîneur ])
AllDisjoint ([ Messi , pipG ])
#AllDisjoint ([ FCB , pipG , Messi ])

sync_reasoner_pellet ( infer_property_values = True )
onto.save ( file = " tp_rcr.owl ", format = "rdfxml")

###
with onto :

    Espagnol = onto.contry ()

    ronaldo = onto.JoueurSelectione ()

    real = onto.equipe ()
    amine = onto.joueur ()

    MCA = Thing () # club

    ronaldo.joue = [ real ]
    amine.Entraîneur = [ ronaldo ]

    sync_reasoner_pellet ( infer_property_values = True )
    onto.save ( file = " tp_rcr.owl ", format = "rdfxml")

# %%

```

* Owlready2 * Running Pellet...

Voici les résultats déduits par le raisonneur :

```

* Owlready2 * Running Pellet...
  java -Xmx2000M -cp /usr/local/lib/python3.8/dist-packages/owlready2/pellet/log4j-1.2.16.jar:/usr/local/lib/python3.8/dist-packages/owlready2/pellet/jcl-over-slf4j-1.6.4.jar:/usr/local/lib/python3.8/
* Owlready2 * Pellet took 1.8495643138885498 seconds
* Owlready * Equivalent: onto.JoueurSelectione onto.joueur
* Owlready * Equivalent: onto.joueur onto.JoueurSelectione
* Owlready * Equivalent: onto.FCB onto.Messi
* Owlready * Equivalent: onto.FCB onto.pipG
* Owlready * Equivalent: onto.FCB owl.Nothing
* Owlready * Equivalent: onto.Messi onto.FCB
* Owlready * Equivalent: onto.Messi onto.pipG
* Owlready * Equivalent: onto.Messi owl.Nothing
* Owlready * Equivalent: onto.pipG onto.FCB
* Owlready * Equivalent: onto.pipG onto.Messi
* Owlready * Equivalent: onto.pipG owl.Nothing
* Owlready * Equivalent: owl.Nothing onto.FCB
* Owlready * Equivalent: owl.Nothing onto.Messi
* Owlready * Equivalent: owl.Nothing onto.pipG
* Owlready * (NB: only changes on entities loaded in Python are shown, other changes are done but not listed)
* Owlready2 * Running Pellet...
  java -Xmx2000M -cp /usr/local/lib/python3.8/dist-packages/owlready2/pellet/log4j-1.2.16.jar:/usr/local/lib/python3.8/dist-packages/owlready2/pellet/jcl-over-slf4j-1.6.4.jar:/usr/local/lib/python3.8/
* Owlready2 * Pellet took 1.8294269501861572 seconds
* Owlready * Equivalent: onto.JoueurSelectione onto.joueur
* Owlready * Equivalent: onto.joueur onto.JoueurSelectione
* Owlready * Equivalent: onto.FCB onto.Messi
* Owlready * Equivalent: onto.FCB onto.pipG
* Owlready * Equivalent: onto.FCB owl.Nothing
* Owlready * Equivalent: onto.Messi onto.FCB
* Owlready * Equivalent: onto.Messi onto.pipG
* Owlready * Equivalent: onto.Messi owl.Nothing
* Owlready * Equivalent: onto.pipG onto.FCB
* Owlready * Equivalent: onto.pipG onto.Messi
* Owlready * Equivalent: onto.pipG owl.Nothing
* Owlready * Equivalent: owl.Nothing onto.FCB
* Owlready * Equivalent: owl.Nothing onto.Messi
* Owlready * Equivalent: owl.Nothing onto.pipG
* Owlready * Representing onto.JoueurSelectione?: {onto.JoueurSelectione} => {onto.JoueurSelectione, onto.joueur}
* Owlready * Representing onto.joueur?: {onto.joueur} => {onto.Entraîneur}
* Owlready * (NB: only changes on entities loaded in Python are shown, other changes are done but not listed)

```