

# Task1: OWL using Protege

## Task 1-1: Class & Property Hierarchies

- Created a new owl ontology file with name Vehicle.owl
- Configured protege regarding naming of entities according to the guide.
- Saved the file.
- Installed Pellet Reasoner and restarted.
- Created classes according to the hierarchy.
- Renamed the three classes to the provided URI
- Selected the classes and set them disjoint with any same level classes.
- Created the data properties and set their ranges to xds:integer only.

## Task 1-2: Add object property to clas

- Select Vehicles class and set it as subclass of "hasBrand some Brand" and "hasSpeed some xds:integer"

## Task 1-3: Add datatype properties to class

- Set each Vehicle subclasses to its brand. eg: "hasBrand exactly 1 Honda".
- Created instances on "Individuals by Class" window and set their properties accordingly.

## Task 1-4: Automatically add instances to class using 'equivalent to' property

- Created a subclass under Vehicle called ExpensiveCar.
- Set it as subclass of "hasMoney some xds:integer[>=40000]"
- Right click it and convert to defined class
- Ctrl+R to sync the reasoner. 3 instances are now part of this class.

## Task2: PSL

Other than required data, I also created Similar Title and Year(not only Date). Some titles have very small difference, but they should be counted in. Some Movies have very different publish date, but regarding year they must be the same.

Regarding the PSL rules, I believe there are only a few possibilities:

- they have the exactly same Title AND same Year(or Date)/Director
- they have similar Title AND same Date/Director
- any combination of same Title, Similar Title, Date, Director

Regarding the result evaluation:

```
In [ ]: gt = {}
with open("Movie_PSL/data/same_movie_truth.txt") as file:
    for line in file.readlines():
        a,b,t = line.strip().split("\t")
        gt[(a,b)] = float(t)
```

```
In [ ]: target = {}
with open("Movie_PSL/inferred-predicates/SAMEMOVIE.txt") as file:
    for line in file.readlines():
        a,b,t = line.strip().split("\t")
        target[(a,b)] = round(float(t),0)
```

```
In [ ]: TP = 0
FP = 0
FN = 0
for k,v in gt.items():
    if v:
        try:
            pred_v = target[k]
            if pred_v < v:
                print(k)
                FN += 1
            else:
                TP += 1
        except KeyError:
            print(k)
            FN += 1
    else:
        try:
            pred_v = target[k]
            if pred_v:
                FP += 1
        except KeyError:
            pass
```

```
('a-122', 'b-120')
('a-360', 'b-346')
('a-1326', 'b-1148')
```

```
In [ ]: precision = TP/(TP+FP)
recall = TP/(TP+FN)
F1 = 2*precision*recall/(precision+recall)
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1: {F1}")
```

```
Precision: 1.0
Recall: 0.9822485207100592
F1: 0.9910447761194029
```

There are 3 False Negatives. I reviewed them and found out that they may have the same Title or Director, but their date are completely different. One of them has 2014-10 and 2012-11. But I didnt have time to do a similar Director Name check, which I developed a Waterman Algorithm but takes too long for all the combinations.