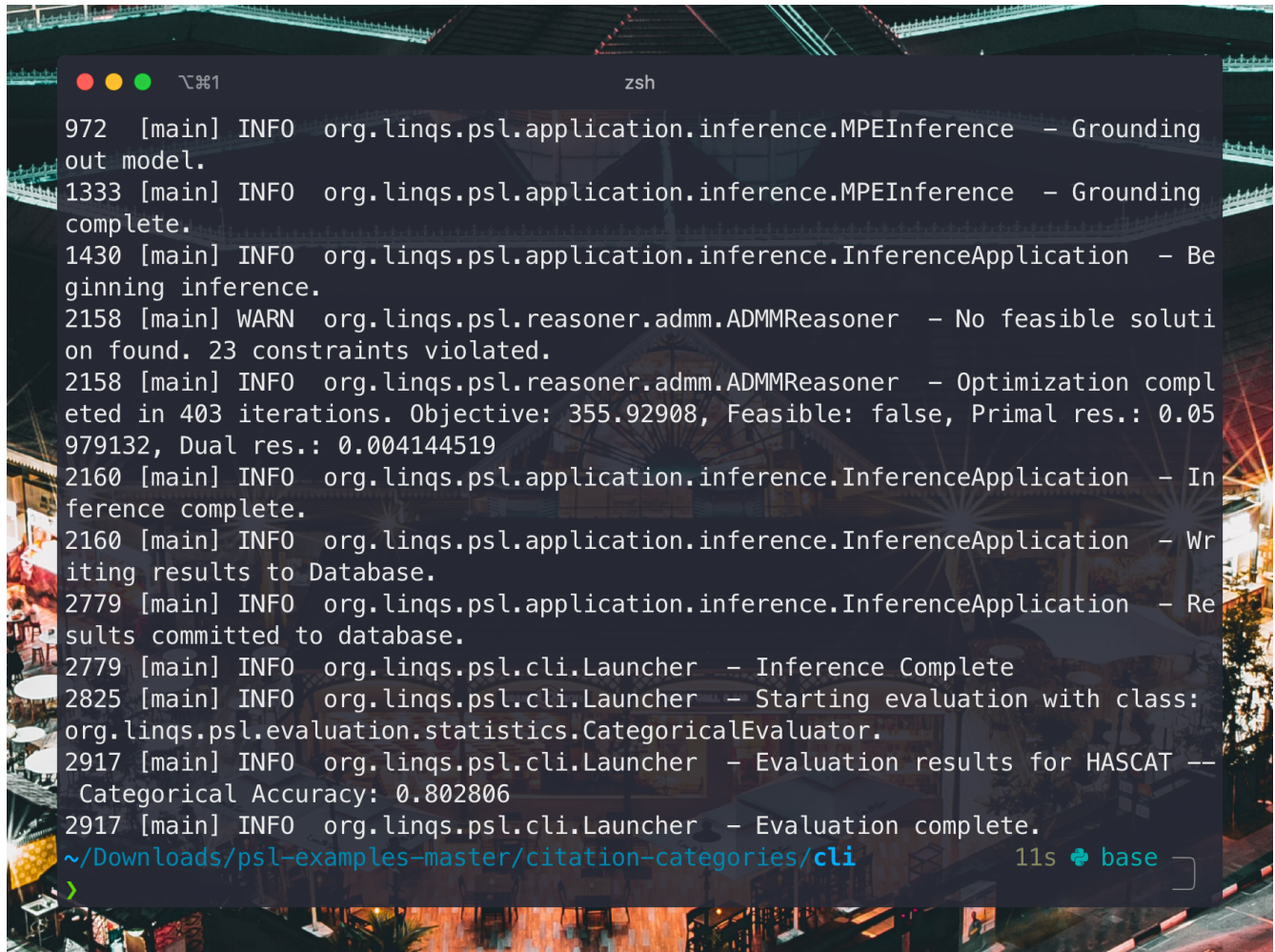


HW6 PSL & OWL

Task 1 PSL

Task 1.1



```
972 [main] INFO org.linqs.psl.application.inference.MPEInference - Grounding
out model.
1333 [main] INFO org.linqs.psl.application.inference.MPEInference - Grounding
complete.
1430 [main] INFO org.linqs.psl.application.inference.InferenceApplication - Be
ginning inference.
2158 [main] WARN org.linqs.psl.reasoner.admm.ADMMReasoner - No feasible soluti
on found. 23 constraints violated.
2158 [main] INFO org.linqs.psl.reasoner.admm.ADMMReasoner - Optimization compl
eted in 403 iterations. Objective: 355.92908, Feasible: false, Primal res.: 0.05
979132, Dual res.: 0.004144519
2160 [main] INFO org.linqs.psl.application.inference.InferenceApplication - In
ference complete.
2160 [main] INFO org.linqs.psl.application.inference.InferenceApplication - Wr
iting results to Database.
2779 [main] INFO org.linqs.psl.application.inference.InferenceApplication - Re
sults committed to database.
2779 [main] INFO org.linqs.psl.cli.Launcher - Inference Complete
2825 [main] INFO org.linqs.psl.cli.Launcher - Starting evaluation with class:
org.linqs.psl.evaluation.statistics.CategoricalEvaluator.
2917 [main] INFO org.linqs.psl.cli.Launcher - Evaluation results for HASCAT --
Categorical Accuracy: 0.802806
2917 [main] INFO org.linqs.psl.cli.Launcher - Evaluation complete.
~/Downloads/psl-examples-master/citation-categories/cli 11s + base
```

A live and full version of screenshot is available here: <https://asciinema.org/a/0UzGSWWM0fAEIwAtrFDqKBjtZ>

Task 1.2

1.2.1 Similarity

- I use Jaro-Winkler as the similarity metric for title. I first lowercase it, remove the punctuation, and then calculate the similarities
- I use the normalized Levenshtein as the similarity metric for year. If one year field is empty, then it will be scored 0.5.

1.2.2 model-manual.psl

A snapshot of terminal output:

```
[main] INFO org.lings.psl.cli.Launcher - Evaluation results for SAMEPAPER -- Accuracy:
0.997671, F1: 0.794979, Positive Class Precision: 0.860507, Positive Class Recall:
0.738725, Negative Class Precision: 0.998394, Negative Class Recall: 0.999263
```

1.2.3 model-learned.psl

A snapshot of terminal output:

```
[main] INFO org.lings.psl.cli.Launcher - Evaluation results for SAMEPAPER -- Accuracy:
0.997281, F1: 0.719608, Positive Class Precision: 0.973475, Positive Class Recall:
0.570762, Negative Class Precision: 0.997366, Negative Class Recall: 0.999904
```

Task 2 OWL

Task 2.1 Pizza Ontology

The tutorial is quite outdated, the interface and lots of functions look different from the current version. Several times, I had trouble figuring out what it was referring to.

Problems encountered on Task 2.1

0:

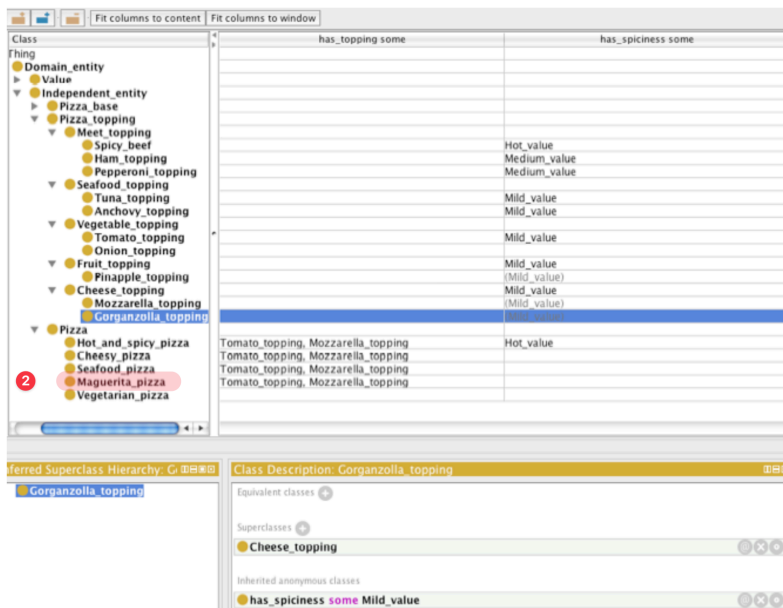
Setup the renderer and how new entities will be created

- Open preferences (**File | Preferences** on windows, **Protege | Preferences** on mac)
- Select the **new entities** tab **1**
- In the **Entity URI** pane, select **auto ID**. When you create a new class, property or individual, Protege will give it a meaningless URI and a readable label. That way if you exchange ontologies, correcting spelling mistakes (by merely changing labels) won't cause the links between the ontologies to break
- Select the **renderer** tab **2**
- Select **Render entities using annotation values**

1: No matching option exist, I choose the closest one: `End with` -> `Auto-generated ID`

2: No matching option exist, I choose the closest one: `Render by annotation property (e.g., rdfs:label, skos:prefLabel)`

1:



1,2:
Different spell

3:
In the previous image,
Quatro_formaggi_pizza
cannot be found.

Finish off the pizza definitions

- Return to the class tab
 - Close the pizzas toppings so that no others can be added
 - Close Margherita_pizza
 - 1 ■ Select Margherita_pizza
 - right click on one of the has_topping restrictions in the **Description** view (on the mac you might need to first select the restriction, then right click).
 - In the menu that appears select **Create closure axiom**. This creates an AllValuesFrom restriction along the property in the restriction you selected. The filler of this new restriction will be a union of all the fillers along this property.
 - Repeat 3 for Quatro_formaggi_pizza and Seafood_pizza.
- 1 and 2 in different spell
 - 3: Qutr...pizza cannot be found on previous image, thus I skipped this one

2.

Add the definitions for other defined classes

If the definitions are more complicated, e.g. for Vegetarian Pizza, then you just have to do it the long way. It is still probably easier to add them as individual restrictions and then convert to a defined class

- Add a definition for a Cheeseey_pizza = Pizza and has_topping min 2 Cheese_topping e.g. that it has at least two kinds of cheese
- Add a definition for Vegetarian Pizza = Pizza and not(has_topping some (Meat_topping or Fish_topping)) e.g. that it has no meat or fish toppings

Cannot found Fish_topping , never mentioned on previous part of tutorial, thus I skipped. Accordingly , Vegetarian_pizza 's definition becomes Pizza and not(has_topping some Meat_topping)

3.

Making corrections

It is the disjoint axioms that are troublesome. If you are changing the class hierarchy:

- If you are moving a class, first remove its disjoint axioms. They will almost certainly be wrong for its new location
- When you are finished, go to a sibling class with disjoint axioms, remove the long disjoint axiom and press **CTRL-J/CMD-J** to add back the disjoint axioms on all primitive siblings.

Anything else, just fix in the usual Class Description view.

- 1: No matching menu item exist, I choose the closest one: remove local disjoint classes axioms...
- 2: Shortcut not working at all

4. Other Bugs I met

1. Using a reasoner may cause the hierarchy of class to be out of order.
2. Save the owl file and re-read it, the hierarchy of class may be incorrect.

Task 2.2 Protégé - `people.owl`

2.2.1 Mad Cow

2.2.1.1. What's the definition of `mad_cow`?

Give both the formal definition and an explanation in your own words.

- Formal:
 - A mad cow is a cow that has been eating the brains of sheep.

```
cow and (eats some (brain and (part_of some sheep)))
```

- My explanation: cows are vegetarian, once eat sheep's brains, thus it gone mad

2.2.1.2. What constraints does `mad_cow` inherit from its superclasses? Do you see any problem with that definition? Why?

- constraints:
 - `eats only (not (animal))`
 - `eats only (not (part_of some animal))`
 - `eats some owl:Thing`
- A vegetarian is defined as an animal that eats no other animals, or parts of animals.
- Mad cows equivalent to `cow and (eats some (brain and (part_of some sheep)))`
 - Mad cows eat sheep's brain, this is a **violation**

2.2.2 Reasoner

2.2.2.1. What happened to the definition of `mad_cow`? Compare with result when not using reasoner. (Hint: check also the class hierarchy (inferred) tab)

After use reasoner, on "class hierarchy (inferred)" tab, `mad cow` \sqsubseteq `owl:Nothing`

2.2.2.2. What happened to the giraffe class? Compare with result when not using reasoner.

- After using reasoner, `Giraffe` becomes the subclass of `vegetarian` and gains the constraints from `vegetarian` as well.

2.2.2.3. Which classes do Tom and Minnie belong to? Compare with result when not using reasoner. (Hint: Minnie has_pet Tom).

	Tom	Minnie
Before	<code>owl:Thing</code>	<code>elderly</code> <code>female</code>
After	<code>owl:Thing</code> <code>cat</code> <code>pet</code>	<code>elderly</code> <code>female</code> <code>old lady</code>

2.2.2.4. List all the person instances. Compare with result when not using reasoner. Which ones were added after you executed the reasoner? Describe why each of the additional instances were added.

Before: Fred, Joe, Kevin, Walt

After: Fred, Joe, Kevin, Walt and

- Mick: dog owner and white van man
- Minnie: old lady
- Pete: pet owner

2.2.2.5. Give a complete description of the instance Mick. Which classes do Daily Mirror belong to? Compare with result when not using reasoner.

	Mick	Daily Mirror
Before	male Mick is male and drives a white van	owl:Thing The paper read by Mick.
After	male dog owner white van man Mick is male and drives a white van	owl:Thing tabloid The paper read by Mick.

2.2.2.6. List all the descendant classes of pet owner.

- animal lower (after reasoning)
- cat owner (after reasoning)
- dog owner (after reasoning)
- old lady (subClass of (has_pet some animal) and (has_pet only cat))

2.2.2.7. How many “pets” must a person have to be considered an animal-lover?

3

2.2.2.8. Do all the “pets” of an animal-lover need to be animals? Does an “old lady” need to be a person?

- Yes, the ranges of has_pet is animal
- Yes, old lady equivlant to elderly and female and person