

Knowledge Representation Laboratory 9: OWLReady

CITS3005

This laboratory will look at presenting ontologies and deduction using the flask framework with owlready2.

1. Getting Started:

- set up a virtual environment: `python -m venv venv`.
- activate the environment: `source venv/bin/activate`
- install flask and owlready2: `pip install owlready2 flask`
- save the code from workshop 9 (<https://teaching.csse.uwa.edu.au/units/CITS3005/workshops/ws09-flask.pdf>) in `bacteria-flask.py`
- Run with `python bacteria-flask.py`
- Open `http://localhost:5000/` in your browser.
- Inspect the code, and experiment to make sure you understand how it works.

2. Add deduction and search:

- save and run the code on the following page, for searching the ontology.
- add in some additional search features, such as string matching.
- edit the web page to include instructions and better formatting.

3. Apply to your project!

- Using this as a starting point, adapt the code to apply to the UWA handbook knowledge base (or a small part of it).
- What additional search features can you include?

```

# File decision_support.py
from owlready2 import *
onto = get_ontology("owl/bacteria.owl").load()
from flask import Flask, request
app = Flask(__name__)
@app.route('/')
def entry_page():
    html = """<html><body>
<h3>Enter the bacteria characteristics:</h3>
<form action="/result">
Gram:<br/>
<input type="radio" name="gram" value="True"/> Positive<br/>
<input type="radio" name="gram" value="False"/> Negative<br/> <br/>
Shape:<br/>
<input type="radio" name="shape" value="Round"/> Round<br/>
<input type="radio" name="shape" value="Rod"/> Rod<br/> <br/>
Groupings:<br/>
<select name="groupings" multiple="multiple">
    <option value="Isolated">Isolated</option>
    <option value="InPair">InPair</option>
    <option value="InCluster">InCluster</option>
    <option value="InSmallChain">InSmallChain</option>
    <option value="InLongChain">InLongChain</option>
</select><br/>
<br/>
<input type="submit"/>
</form>
</body></html>"""
    return html

ONTO_ID = 0
@app.route('/result')
def page_result():
    global ONTO_ID
    ONTO_ID = ONTO_ID + 1
    onto_tmp = get_ontology("http://tmp.org/onto_%s.owl#" % ONTO_ID)
    gram = request.args.get("gram", "")
    shape = request.args.get("shape", "")
    groupings = request.args.getlist("groupings")
    with onto_tmp:
        bacterium = onto.Bacterium()
        if gram == "True": bacterium.gram_positive = True
        elif gram == "False": bacterium.gram_positive = False
        if shape:
            shape_class = onto[shape]
            bacterium.has_shape = shape_class()
        for grouping in groupings:
            grouping_class = onto[grouping]
            bacterium.has_grouping.append(grouping_class())
        close_world(bacterium)
        sync_reasoner([onto, onto_tmp])
    class_names = []
    for bacterium_class in bacterium.is_a:
        if isinstance(bacterium_class, ThingClass):
            class_names.append(bacterium_class.name)
    class_names = ",".join(class_names)
    html = """<html><body>
<h3>Result: %s</h3>
</body></html>""" % class_names
    onto_tmp.destroy()
    return html

import werkzeug.serving
werkzeug.serving.run_simple("localhost", 5000, app)

```