Department of Computer Science
Technical University of Cluj-Napoca

Knowledge-Based Systems

Laboratory activity

Congenital Heart Defects Ontology

Students: Zaharia Tudorita, Pop Alexandru
Email: *zaharia.tudorita@yahoo.com*

Assoc. Prof.dr. eng. Adrian Groza
Adrian.Groza@cs.utcluj.ro

Contents

Chapter 1

Ontology design

The following is a latex template for your documentation. Three sources of documentation In this project we present the design and implementation of a medical ontology based on Ontobee medical repository. are [3, 2, 1].

1.1 Competency questions

The main questions are formulated to answer patient-specific questions, not necessarily medical related, however having a foundation based on medical knowledge.

Index	Question
QC1	List all congenital heart conditions.
QC2	How can I diagnose at home whether I have heart arrhythmia?
QC3	What is the proper treatment if I have a specific heart defect?
QC4	What does it mean when I get shortness of breath ?
QC5	What does it mean when it feels painful in the chest area ?
QC6	Which heart defects have similar symptoms to the ones present on me?
QC7	Show heart defects that are related physiologically
QC8	Can one follow a more invasive treatment?
QC9	List alternatives to the currently followed treatment
QC10	Show expected side effects in a treatment plan

1.2 Related ontologies

The ontology was searched in the Ontobee repository and some concepts and instances that were reused. The main steps of re-using an ontology were based on

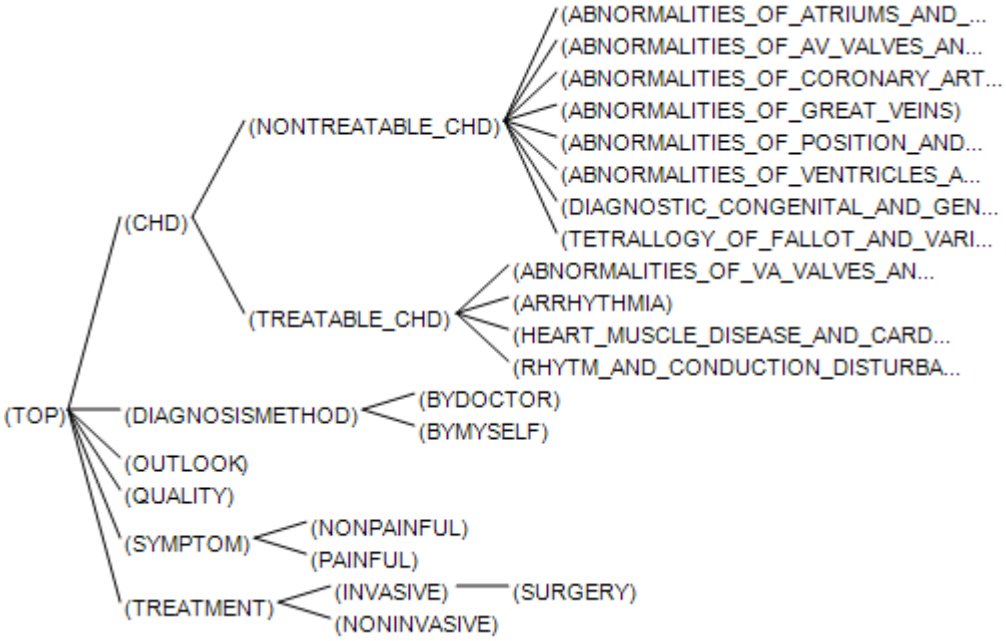
1. The congenital heart defects .owl file was downloaded.
2. Converted from OWL syntax to KRSS syntax using the provided KRSS converter.
3. Surf the ontology with RacerPro and choose instances.
4. Re-use the elements and paste them into the base .racer file.

Given the following congenital heart defects ontology contains congenital diseases.

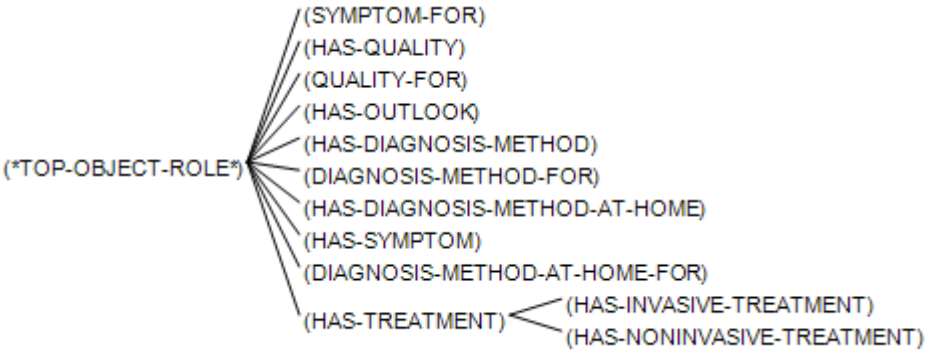
```
(define-primitive-concept Rhytm_and_conduction_disturbances  Treatable_CHD)
(define-primitive-concept Abnormalities_of_great_veins  NonTreatable_CHD)
```

1.3 Tboxes

Conversation form KRSS syntax to latex can be generated with the Manchester convertor http://owl.cs.manchester.ac.uk/tools/webapps/owl-syntax-converter/.
The ontology is presented with the most important concept being CHD (*congenital heart disease*) and the affiliated concepts such as diagnosis method, the outlook of CHD, treatment, quality of treatment, symptoms for various diseases. All these details are presented in the figure below.



As for the roles that are bounded to each of the concept, we can take a short look at the following picture, all the roles take the named convention *has-role*. For example, CHD has-symptom a Symptom, or Symptom is symptom-for CHD.



1.4 Aboxes

The concept assertions are presented in the queries below. The instances asserted will be the following.

```
[QUESTION]: (all-individuals)
[ANSWER]:
            TETRALOGY_OF_FALLOT
            POOR_WEIGHT_GAIN
            CLUBBING_OF_FINGERS
```

```
            FAINTING
            SHORTNESS_OF_BREATH
            CYANOSIS
            PREMATURE_VENTRICULAR_CONTRACTION
            PULSE_OXIMETRY
            ELECTROCARDIOGRAM
            HAS-TREATMENT
            CHEST_PAIN
            PHYSICAL_EXAM
```

To evaluate the role assertion, we will interogate with the following query that checks whether a disease is related to a symptom.

```
[QUERY]:  (individuals-related? Tetralogy_of_Fallot Shortness_of_breath has-symptom)
[ANSWER]:
            T
```

1.5 Rules

creating some rules.. swrl rules etc.

1.6 Queries

The ontology was interrogated with both Racer queries and Fuzzy logic queries, and even SparQL to check and verify the conciseness, accuracy and even the ease of use for each of the following frameworks.

1.6.1 Racer

An eloquent example is displayed in the following table.
For example given a Racer query, we would like to ask the Ontology what symptoms has the cardiac disease Tetralogy of Fallot, by answering with a list of symptoms.

```
[QUERY]:  individual-fillers Tetralogy_of_Fallot has-symptom  ?
[ANSWER]:
            POOR_WEIGHT_GAIN
            CLUBBING_OF_FINGERS
            FAINTING
            SHORTNESS_OF_BREATH
            CYANOSIS
[QUERY]:  concept-instances CHD  ?
[ANSWER]:
            PREMATURE_VENTRICULAR_CONTRACTION
            TETRALOGY_OF_FALLOT
```

1.6.2 Fuzzy

On the other hand, we would like to ask the Fuzzy Learner, if the Tetralogy of Fallot is quite dangerous or not in more general terms. The Learner tells with that the disease is 100% very

dangerous...concerning.

```
Is TETRALOGY_OF_FALLOT instance of Very_Dangerous_Disease ? >= 1.0
Is TETRALOGY_OF_FALLOT instance of Dangerous_Disease ? >= 1.0
```

1.6.3 SparQL

Compared to the previous two languages, the SparQL has the most accessible interface to information, easily being accesed any type of ontology by an URI.
On the other hand, it was a laborous and exhausting process to study and understand the SparQL syntax and create easy queries, not even complex ones. An example of a query created is presented below.

```
[SPARQL]: List all concepts of CHD ?
[RESULT]:
            scimitar syndrome
            tetralogy of Fallot
            dextrocardia
            hypoplastic left heart syndrome
            dextro-looped transposition of the great arteries
            multiple types of congenital heart defects 6
            patent ductus arteriosus
            heart septal defect
            hypoplastic right heart syndrome
```

1.6.4 Competency queries

We will present line by line each of the 10 competency queries, and its result according to Racer.

```
(concept-instances CHD)
[QUESTION]: [C1]: List all congenital heart diseases
[ANSWER]:
            ABNORMALITIES_OF_AV_VALVES_AND_AV_SEPTAL_DEFECT
            ABNORMALITIES_OF_ATRIUMS_AND_ATRIAL_SEPTUM
            ABNORMALITIES_OF_CORONARY_ARTERIES_AND_ARTERIAL_DUCT_AND_PERICARDIUM
            ABNORMALITIES_OF_GREAT_VEINS
            ABNORMALITIES_OF_POSITION_AND_CONNECTION_OF_HEART
            ABNORMALITIES_OF_VENTRICLES_AND_VENTRICULAR_SEPTUM
            DIAGNOSTIC_CONGENITAL_AND_GENERIC_CARDIAC_CODES
            PREMATURE_VENTRICULAR_CONTRACTION
            TETRALOGY_OF_FALLOT
```

```
(concept-instances (some diagnosis-method-at-home-for Arrhythmia))
[QUESTION]: [C2]: How can I diagnose myself if I have arrythmia?
[ANSWER]:
     SELF_REPORTED_SYMPTOMS
     PULSE_OXIMETRY
```

```
(concept-instances (some treatment-for Arrhythmia))
[QUESTION] : [C3]: What are some proper treatments for a specific heart disease?
[ANSWER]:
            SURGICAL_REPAIR
            MEDICAL_MANAGEMENT
            DRUG_TREATMENT
```

```
(individual-fillers shortness_of_breath  symptom-for)
[QUERY]:  [C4]: What does it mean when I have shortness of breath?
[ANSWER]:
            TETRALOGY_OF_FALLOT
            PREMATURE_VENTRICULAR_CONTRACTION
            DIAGNOSTIC_CONGENITAL_AND_GENERIC_CARDIAC_CODES
            ABNORMALITIES_OF_VENTRICLES_AND_VENTRICULAR_SEPTUM
            ABNORMALITIES_OF_CORONARY_ARTERIES_AND_ARTERIAL_DUCT_AND_PERICARDIUM
            ABNORMALITIES_OF_AV_VALVES_AND_AV_SEPTAL_DEFECT
```

```
(individual-fillers chest_area_pain symptom-for)
[QUERY]: [C5]: What does it mean when it feels painful in the chest area?
[ANSWER]:
            PREMATURE_VENTRICULAR_CONTRACTION
            DIAGNOSTIC_CONGENITAL_AND_GENERIC_CARDIAC_CODES
            ABNORMALITIES_OF_VENTRICLES_AND_VENTRICULAR_SEPTUM
            ABNORMALITIES_OF_POSITION_AND_CONNECTION_OF_HEART
            ABNORMALITIES_OF_CORONARY_ARTERIES_AND_ARTERIAL_DUCT_AND_PERICARDIUM
            ABNORMALITIES_OF_ATRIUMS_AND_ATRIAL_SEPTUM
```

```
(retrieve (?disease) (and (?disease fainting has-symptom) (?disease palpitations has-sym
[QUERY]: [C5]: What heart defects have similar symptoms to fainting and palpitations?
[ANSWER]:
DISEASE
ABNORMALITIES_OF_ATRIUMS_AND_ATRIAL_SEPTUM
DISEASE
ABNORMALITIES_OF_POSITION_AND_CONNECTION_OF_HEART
```

```
(retrieve (?disease ?treatment) (and (?disease ?treatment has-treatment) (?treatment Inv
[QUERY]: [C8]: Can patient with heart disease choose an invasive treatment?
[ANSWER]:
DISEASE
TETRALOGY_OF_FALLOT
TREATMENT
SURGICAL_REPAIR
```

1.7 Refined ontology

In this section we discuss about the use of different tools with potential to enhances the ontology design process.

1.7.1 Concept creation

First thing first, the process of creating different concepts can be easily transferred to Natural Language Processing, more exactly tools such as Fred.
The model is asked to create concepts related to heart diseases.

Congenital heart defects (CHDs) are structural abnormalities present at birth that affect the heart's normal function. They are the most common birth defects, affecting approximately 1and severity. They include conditions like atrial septal defect (ASD), ventricular septal defect (VSD), patent ductus arteriosus (PDA), Tetralogy of Fallot (TOF), transposition of the great arteries (TGA), and many others. Causes: The exact causes of CHDs are often unknown, but they can result from genetic factors, environmental influences, or a combination of both. Certain risk factors, such as maternal diabetes, certain medications, smoking during pregnancy, and chromosomal abnormalities, may increase the likelihood of CHDs. Symptoms: Symptoms can vary depending on the type and severity of the defect. Mild defects may cause no symptoms, while more severe cases can lead to symptoms like cyanosis (bluish skin color), rapid breathing, poor feeding, fatigue, and failure to thrive. CHDs are typically diagnosed through prenatal ultrasound screenings or soon after birth through physical examinations. Additional diagnostic tests like echocardiograms, electrocardiograms (ECGs), and cardiac catheterization may be used to evaluate the defect's specifics. Treatment options for CHDs depend on the type and severity of the defect. Some mild defects may not require any intervention and may resolve on their own. Others may need medication to manage symptoms, while more complex cases might require surgical interventions, such as open-heart surgery or catheter-based procedures. Outlook: Advances in medical technology and treatment have significantly improved the long-term outlook for individuals with CHDs. Many children with CHDs go on to lead healthy lives with appropriate medical care and lifestyle modifications. However, ongoing medical follow-up is often necessary throughout adulthood. Living with a CHD can present challenges for individuals and their families. Support groups, counseling, and educational resources are available to help navigate the emotional and practical aspects of managing CHDs.

And afterwards it generates a beautiful graph with muliple concepts in .rdf format. The .rdf format is then uploaded in the RacerPorter IDE to help visualize better the ontology created.

```
(define-primitive-concept Outlook *top*)
(define-primitive-concept Quality *top*)
(define-primitive-concept Surgery Invasive)
```

1.7.2 Ontology population

The congenital heart defect ontology lacks a huge amount of information regarding types of disease and so and so. Given such requirements, it was used as a baseline the popular Chat-GPT model for creating a .csv file, afterwards the content is pre-processed automatically using a python script.
The .csv files has the following structure.

Instance	**Category**
Surgical_repair	Invasive
Related	**has-symptom**
Abnormalities_of_ventricles_and_ventricular_septum	Chest_pain

The following .csv file is then afterwards converted to .racer file using a python script, the script will simply take each entry of the csv and convert it into the Racer syntax. For the table presented above, it will generate the code:

```
(instance Surgical_repair InvasiveTreatment)
(related Tetralogy_of_Fallot Surgical_repair has-treatment)
(related Abnormalities_of_atriums_and_atrial_septum Chest_pain has-symptom)
```

1.7.3 Ontology pattern design

A good practice is to use some ontology pattern designs, especially for more complex queries. The first pattern applied in this project and a backbone of the concept is **Partition pattern**.

```
(equivalent CHD (or Treatable_CHD NonTreatable_CHD))
(disjoint Treatable_CHD NonTreatable_CHD)
```

To remove ambiguity in querying, the **Content pattern** is used such that when a new role is created, the concept will automatically ask for it using some key word such as *is-role-for* or *has-role* as presented in the example below.

Class Names	Singular form and multiple words are divided by _
Roles	for a relation there is *has-role* or *role-of*

An example of a role is described in the following paragraph.

```
(define-primitive-role has-symptom :domain CHD :range Symptom :inverse symptom-for)
```

Last but not least, the **Selector pattern** for selecting specific diseases that are disjoint in terms of treatability, or treatment.

```
(implies Symptom (some symptom-for CHD))
```

1.7.4 Ontology verbalization

More than a good practice, or an inovative tool, the process of *verbalization* translates ontology language into human readable such that it may be easier to look over the heart defects and their properties, and enhances the process of creating new queries based on the existing information in the ontology.
The output of verbalization process is displayed in the following box.

```
Every CHD is something that is a NONTREATABLE_CHD or that is a TREATABLE_CHD.
Everything that is a NONTREATABLE_CHD or that is a TREATABLE_CHD is a CHD.
```

1.8 Ontology evaluation

You can evaluate your ontology against several metrics.

You have to check the ontology consistency and coherence:

(tbox-coherent?)

(tbox-cyclic?)

Number of concepts	29
Number of roles	20
Number of individuals	37
Number of rules	0
Number of inconsistencies	0
Number of axioms	0
DL expressivity	LCUH

Appendix A

Base code

A.0.1 heart_defect_raw.racer

```
(in-knowledge-base disease congenital-heart-diseases)
#|------------PARTITION PATTERN--------------------|#
(equivalent CHD (or Treatable_CHD NonTreatable_CHD))
(equivalent Symptom (or Painful NonPainful))
(equivalent Treatment (or Invasive NonInvasive))
(equivalent DiagnosisMethod (or ByDoctor ByMyself))
(disjoint Treatable_CHD NonTreatable_CHD)
(disjoint Painful NonPainful)
(disjoint Invasive NonInvasive)
(define-primitive-concept Outlook *top*)
(define-primitive-concept Quality *top*)
(define-primitive-concept Surgery Invasive)

#|---------NAMING CONVENTION PATTERN---------------|#
(define-primitive-role has-quality :domain Treatment :inverse quality-for :range Quality
(define-primitive-role has-outlook :domain CHD :range Outlook)

#|---------REUSED ONTOLOGY concepts---------------|#
(define-primitive-concept Abnormalities_of_AV_valves_and_AV_septal_defect  NonTreatable_
(define-primitive-concept Abnormalities_of_atriums_and_atrial_septum  NonTreatable_CHD)
(define-primitive-concept Abnormalities_of_coronary_arteries_and_arterial_duct_and_perio
(define-primitive-concept Abnormalities_of_great_veins  NonTreatable_CHD)
(define-primitive-concept Abnormalities_of_position_and_connection_of_heart  NonTreatabl
(define-primitive-concept Abnormalities_of_ventricles_and_ventricular_septum  NonTreatab
(define-primitive-concept Diagnostic_congenital_and_generic_cardiac_codes  NonTreatable_
(define-primitive-concept Tetrallogy_of_Fallot_and_variants  NonTreatable_CHD)
(define-primitive-concept Heart_muscle_disease_and_cardiomyopathies  Treatable_CHD)
(define-primitive-concept Rhytm_and_conduction_disturbances  Treatable_CHD)
(define-primitive-concept Abnormalities_of_VA_valves_and_great_arteries  Treatable_CHD)
(define-primitive-concept Arrhythmia Treatable_CHD)

#|---------Role definition--------------|#
(define-primitive-role has-symptom :domain CHD :range Symptom :inverse symptom-for)
(define-primitive-role has-treatment :domain CHD :range Treatment)
(define-primitive-role has-invasive-treatment :parent has-treatment :range Invasive)
(define-primitive-role has-noninvasive-treatment :parent has-treatment :range NonInvasiv
```

```
(define-primitive-role has-diagnosis-method :inverse diagnosis-method-for :domain CHD :r
(define-primitive-role has-diagnosis-method-at-home :inverse diagnosis-method-at-home-fo

#|---------ABOX concept assertions---------------|#
(instance Surgical_repair Invasive)
(instance Medical_management NonInvasive)
(instance Echocardiogram ByDoctor)
(instance Electrocardiogram ByDoctor)
(instance Chest_X-ray ByDoctor)
(instance Physical_exam ByDoctor)
(instance Genetic_testing ByDoctor)
(instance Cardiac_CT_scan ByDoctor)
(instance Cardiac_MRI ByDoctor)
(instance Holter_monitor ByDoctor)
(instance Event_monitor ByDoctor)
(instance Stress_test ByDoctor)
(instance Echocardiogram ByMyself)
(instance Pulse_oximetry ByMyself)
(instance Self_reported_symptoms ByMyself)

#|---------Population: role & concept assertions---------------|#
(related Abnormalities_of_AV_valves_and_AV_septal_defect Shortness_of_breath has-symptom
(related Abnormalities_of_AV_valves_and_AV_septal_defect Fatigue has-symptom)
(related Abnormalities_of_AV_valves_and_AV_septal_defect Swelling_in_the_legs has-sympto
(related Abnormalities_of_atriums_and_atrial_septum Palpitations has-symptom)
(related Abnormalities_of_atriums_and_atrial_septum Chest_pain has-symptom)
(related Abnormalities_of_atriums_and_atrial_septum Fainting has-symptom)
(related Abnormalities_of_coronary_arteries_and_arterial_duct_and_pericardium Chest_pain
(related Abnormalities_of_coronary_arteries_and_arterial_duct_and_pericardium Shortness_
(related Abnormalities_of_coronary_arteries_and_arterial_duct_and_pericardium Swelling_i
(related Abnormalities_of_great_veins Cyanosis has-symptom)
(related Abnormalities_of_great_veins Swelling_in_the_legs has-symptom)
(related Abnormalities_of_great_veins Fatigue has-symptom)
(related Abnormalities_of_position_and_connection_of_heart Chest_pain has-symptom)
(related Abnormalities_of_position_and_connection_of_heart Fainting has-symptom)
(related Abnormalities_of_position_and_connection_of_heart Palpitations has-symptom)
(related Abnormalities_of_ventricles_and_ventricular_septum Shortness_of_breath has-symp
(related Abnormalities_of_ventricles_and_ventricular_septum Chest_pain has-symptom)
(related Abnormalities_of_ventricles_and_ventricular_septum Fainting has-symptom)
(related Diagnostic_congenital_and_generic_cardiac_codes Shortness_of_breath has-symptom
(related Diagnostic_congenital_and_generic_cardiac_codes Chest_pain has-symptom)
(related Diagnostic_congenital_and_generic_cardiac_codes Palpitations has-symptom)
(related Tetralogy_of_Fallot Surgical_repair has-treatment)
(related Tetralogy_of_Fallot Medical_management has-treatment)
(related Tetralogy_of_Fallot Cardiac_catheterization has-diagnosis-method)
(related Tetralogy_of_Fallot Magnetic_resonance_imaging has-diagnosis-method)
(related Tetralogy_of_Fallot Electrocardiogram has-diagnosis-method)
(related Tetralogy_of_Fallot Chest_X-ray has-diagnosis-method)
(related Tetralogy_of_Fallot Physical_exam has-diagnosis-method)
(instance Premature_Ventricular_Contraction Arrhythmia)
```

```
(related Premature_Ventricular_Contraction Shortness_of_breath has-symptom)
(related Premature_Ventricular_Contraction Chest_pain has-symptom)
(related Premature_Ventricular_Contraction has-treatment drug_treatment)
(related Premature_Ventricular_Contraction Electrocardiogram has-diagnosis-method)
(related Premature_Ventricular_Contraction Pulse_oximetry has-diagnosis-method-at-home)
(related Tetralogy_of_Fallot Cyanosis has-symptom)
(related Tetralogy_of_Fallot Shortness_of_breath has-symptom)
(related Tetralogy_of_Fallot Fainting has-symptom)
(related Tetralogy_of_Fallot Clubbing_of_fingers has-symptom)
(related Tetralogy_of_Fallot Poor_weight_gain has-symptom)
```

A.0.2 Ontology.py

```
    import os
import time

if os.path.basename(os.getcwd()) ==  "KBS2023_HEART_DEFECT_ONTOLOGY":
    os.chdir("utils")
elif os.path.basename(os.getcwd()) == "FuzzyDL":
    print(os.path.basename(os.getcwd()))
    os.chdir("/../utils")

from pracer.racer_client import RacerClient
import re

class Ontology():

    def __init__(self,RACER_QUERIES):
        super().__init__()
        self._racer = None
        self.RACER_PARAM = ['127.0.0.1', 8088]
        self.JAVA_PARAM = ['java', '-jar']
        self.RACER_QUERIES = RACER_QUERIES
        self.FILE_FUZZY = ''
        self.FILE_RACER = 'C:/Users/Fish/Documents/TUDORITA/AN 4 SEM 2/KBS/KBS2023_HEART
        self.PATH_FUZZY = 'C:/Users/Fish/Documents/TUDORITA/AN 4 SEM 2/KBS/KBS2023_HEART
        self.time = 0
    @property
    def racer(self) -> RacerClient:

        self.time+=1
        if self._racer is None:
            self._racer = RacerClient(self.RACER_PARAM[0],self.RACER_PARAM[1])
            self._racer.connect()
            self._racer.full_reset()
            print("[RACER]: connected...")
            if self.time >= 100:
                raise Exception(" racer timeout ...")

        return self._racer
```

```
    def racer_load_file(self):
        with open(self.FILE_RACER,"r") as file:
            for line in file:
                try:
                    result = self.racer.racer_call(line[1:-2])
                    OK_STATUS = result.get_value()[1:3]
                    ANSWER_STATUS = result.get_value()[1:7]
                    if OK_STATUS != "ok" and ANSWER_STATUS != "answer":
                        raise Exception(f'invalid command in file {line}')
                except Exception as e:
                    print("[ERROR]: ",e)

    def racer_run_queries(self):
        for query in self.RACER_QUERIES:
            print("[QUERY]: ",query[1:-1],' ?')
            result = self.racer.racer_call(query[1:-1])
            self.racer_interpret_result(result.get_value(),query)

    def racer_interpret_result(self,result,query):
        try:
            OK_STATUS = result[1:3]
            ANSWER_STATUS = result[1:7]
            if OK_STATUS != "ok" and ANSWER_STATUS != "answer":
                raise Exception(f'invalid command - {query}')

            matches = re.findall(r"[A-Za-z_]+", result)
            print("[ANSWER]: ")
            for match in matches[1:]:
                print(f'\t\t{match}')

        except Exception as e:
            print("[ERROR]: ",e)

    def run(self):
        try:
            self.racer_load_file()
            self.racer_run_queries()

        except Exception as e:
            print("[ERROR]: ",e)
```

A.0.3 populator.py

```
class Populator():
    def __init__(self):
        self.PATH_INPUT = "C:/Users/Fish/Documents/TUDORITA/AN 4 SEM 2/KBS/KBS2023_HEART
        self.PATH_OUTPUT = "C:/Users/Fish/Documents/TUDORITA/AN 4 SEM 2/KBS/KBS2023_HEAR
```

```
    def populate(self):
        with open(self.PATH_INPUT, "r") as f_in, open(self.PATH_OUTPUT, "a") as f_out:
            # Read the CSV file
            lines = f_in.readlines()

            f_out.write("\n\n;Populated automatically\n\n")

            # Write the instances to the output file
            for line in lines: # assuming that the instances are in lines 1 to 14
                if line.strip() == "":
                    currentFormat = ''
                    f_out.write("\n")
                    continue
                if 'Instance,Category' in line:
                    currentFormat = 'instance'
                    continue
                if 'Related,HasSymptom' in line:
                    currentFormat = 'relatedHasSymptom'
                    continue
                if 'Related,Treatment' in line:
                    currentFormat = 'relatedHasTreatment'
                    continue
                if 'Related,DiagnosisMethod' in line:
                    currentFormat = 'relatedHasDiagnosisMethod'
                    continue

                value1, value2 = line.strip().split(",")

                if currentFormat == 'instance':
                    f_out.write("(instance {} {})\n".format(value1, value2))
                if currentFormat == 'relatedHasSymptom':
                    f_out.write("(related {} {} has-symptom)\n".format(value1, value2))
                if currentFormat == 'relatedHasTreatment':
                    f_out.write("(related {} {} has-treatment)\n".format(value1, value2)
                if currentFormat == 'relatedHasDiagnosisMethod':
                    f_out.write("(related {} {} has-diagnosis-method)\n".format(value1,
```

A.0.4 fuzzy.py

```
    import os

import subprocess

class Fuzzy():
    def __init__(self,FUZZY_QUERIES,PATH_FUZZY="C:/Users/Fish/Documents/TUDORITA/AN 4 SE
        self.commands = ["java","-jar",PATH_FUZZY,PATH_FILE]
        self.FUZZY_QUERIES = FUZZY_QUERIES
        self.PATH_FILE = PATH_FILE
```

```
    def sanity_check(self):
            result = subprocess.run(self.commands, stdout=subprocess.PIPE, stderr=subpro

    def load_query(self):
        for query in self.FUZZY_QUERIES:
            with open(self.PATH_FILE,"a") as f:
                f.write(query)

    def load_file(self):
        data = None
        error = None
        try:
            if os.path.basename(os.getcwd()) == "KBS2023_HEART_DEFECT_ONTOLOGY":
                os.chdir("utils/FuzzyDL")
            elif os.path.basename(os.getcwd()) == "utils":
                os.chdir("FuzzyDL")
            elif os.path.basename(os.getcwd()) != "FuzzyDL":
                 raise Exception("invalid fuzzy path...")

            result = subprocess.run(self.commands, stdout=subprocess.PIPE, stderr=subpro
            data = result.stdout
            error = result.stderr
            if(error != ""):
                raise Exception(" emtpy data set...")
            print("[FUZZY]: ",data)
        except Exception as e:
            print("[ERROR]: ",error)

    def run(self):
        self.load_query()
        self.load_file()
```

A.0.5 sparql.py

```
    from SPARQLWrapper import SPARQLWrapper, JSON

class SPARQL():
    def __init__(self, SPARQL_QUERIES):
        self.sprQL = SPARQLWrapper("http://sparql.hegroup.org/sparql/")
        self.FILE_SPARQL = "C:/Users/Fish/Documents/TUDORITA/AN 4 SEM 2/KBS/KBS2023_HEAR
        self.query = \
        """
        """
        self.questions = [
            "[SPARQL]: Select subclasses of CHD ?",
            "[SPARQL]: List all concepts of CHD ? ",
        ]
        self.queries = SPARQL_QUERIES
```

```
    def load_file(self):
        with open(self.FILE_SPARQL,"r") as file:
            for line in file:
                self.query += line

    def run_query(self,query):
        results = None
        query = self.query +'\n' + query
        try:
            self.sprQL.setQuery(query)
            self.sprQL.setReturnFormat(JSON)
            results = self.sprQL.query().convert()
            print("[RESULT]:\n")
            for result in results["results"]["bindings"]:
                if "label" in result:
                    print("\t\t\t",result["label"]["value"])
                elif "x" in result:
                    print('\t\t\t',result['x']['value'])
                else:
                    print("\t\t\t",result)

        except Exception as e:
            print("[ERROR]: ",e)
        return results

    def run(self):
        self.load_file()
        for index,q in enumerate(self.queries):
            print(self.questions[index])
            self.run_query(q)
```

A.0.6 res_fuzzy.txt

```
(define-fuzzy-concept LowDanger right-shoulder(0,10,5,10))
(define-fuzzy-concept Treatability left-shoulder(0,10,5,10))
(define-fuzzy-concept CompletelyStable crisp(0,10,0,2))
(define-fuzzy-concept MediumStable left-shoulder(0,10,2,5))
(define-fuzzy-concept LowStable left-shoulder(0,10,5,10))
(define-modifier very linear-modifier(3))
(functional danger)
(functional treatable)
(functional progression)
(functional stable)
(functional stability)
(define-concept Dangerous_Disease (and CHD (some danger LowDanger)))
(define-concept Very_Dangerous_Disease (and CHD (some danger (very LowDanger))))
(define-concept Completely_Stable_Disease (and CHD (some stability CompletelyStable)))
(instance TETRALOGY_OF_FALLOT Dangerous_Disease)
```

A.0.7 heart_defect.sparql

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://xmlns.com/foaf/0.1/>
PREFIX obo-term: <http://purl.obolibrary.org/obo/>
```

A.0.8 heart_defect.ipynb

```
from utils.ontology import Ontology
from utils.fuzzy import Fuzzy
from utils.sparql import SPARQL
```

```
from utils.populator import Populator
pop = Populator()
pop.populate()
```

```
RACER_QUERIES = [
'(individual-fillers Tetralogy_of_Fallot has-symptom)',
'(concept-instances CHD)',
'(concept-instances (some has-diagnosis-method-at-home *top*))',
'(retrieve (?y) (and (?x Arrhythmia) (?x ?y has-treatment)))',
'(concept-instances (some has-treatment *top*))',
'(individual-fillers shortness_of_breath symptom-for)',
'(individual-fillers chest_area_pain symptom-for)',
'(individual-fillers fainting symptom-for)',
]
```

```
ontology = Ontology(RACER_QUERIES)
ontology.run()
```

```
FUZZY_QUERIES = [
'(min-instance? TETRALOGY_OF_FALLOT Very_Dangerous_Disease)',
'(min-instance? TETRALOGY_OF_FALLOT Dangerous_Disease)' ]
```

```
fuzzyDL = Fuzzy(FUZZY_QUERIES)
fuzzyDL.run()
```

```
SPARQL_QUERIES = [
'SELECT ?x
?x rdfs:subClassOf* ¡http://www.semanticweb.org/owl/owlapi/turtleCHD¿ .',

'SELECT DISTINCT ?x ?label from ¡http://purl.obolibrary.org/obo/merged/DOID¿
WHERE
?x rdfs:subClassOf obo-term:DOID_1682.
?x rdfs:label ?label.'
]
```

```
sprQL = SPARQL(SPARQL_QUERIES)
sprQL.run()
```

Bibliography

[1] Adrian Groza. *Ontology Engineering with Racer - an activity based approach. 2014.* UT-Press, Cluj-Napoca, 2014.

[2] V Haarslev, R Möller, and M Wessel. RacerPro user's guide and reference manual version 2.0. *Racer Systems GmbH & Co. KG, editor. Germany: Hamburg*, 2012.

[3] Racer Systems GmbH & Co. KG. Racer user guide version 2.0, 2012.