Department of Computer Science
Technical University of Cluj-Napoca

Knowledge-Based Systems

Laboratory activity
Ontology title: Covid Vaccination in Romania
Students: Marcus Pop, Rares Popa

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

Contents

Chapter 1

Rules and policies

1. This assignment is 30% from the final grade.

2. Before each deadline, you have to send your work (latex documentation/code) in your own directory on moodle.cs.utcluj.ro

3. *Laptop policy*: you can use your own laptops as long you have Linux. One goal of the laboratory is to increase your competency in Linux.

4. When submitting your project do not include these guidelines.

Grading.

- 10 means that you did very impressive work or more efficient that I expected or handled a lot of special cases for realistic scenarios.

- 7 means that you: i) constantly worked during classes, ii) you proved competent to use the tool and its expressivity for a realistic ontology, iii) you understood theoretical concepts on which the tool rely on.

- 8,9 mean that your code quantity and the significance of your results are taken into consideration for grades 8 and 9.

- 5 means that you managed to develop something of your own, functional, with your own piece of code substantially different from the examples available.

- You obtain less than 5 in one of the following situations: i) few code written by yourself; ii) too much similarity with the provided examples; iii) non-seriosity (i.e. re-current late at classes, playing games, worked for other disciplines, poor/unprofessional documentation of your work,

- You got 2 if you present the project but fail to submit the documentation or code. You got 1 if you do not present your project before the deadline. You got 0 for any line of code taken from other parts that appear in section *My own code.* For information on TUCN's regulations on plagiarism do consult the active regulations. If your grade is 0, 1, or 2, you do not satisfy the preconditions for participating to the written exam. The only possibility to increase your laboratory grade is to take another project in the next year, at the same class, at to make all the steps again.

Table 1.1: Deadlines.

Activity	**Deadline**
Competency questions. You have to submit using this template a list of competency questions in natural language to which your ontology could provide answers.	W_2
Reused ontologies and knowledge sources. You have to submit the ontologies that you have been identified for possible reuse. These ontologies should be translated into racer format.	W_3
Defining main concepts and roles. You have to submit the racer file (TBox) which contains the main concepts and roles of your ontology.	W_5
Populating ontology. You have to send the ABoxes. The individuals can be asserted automatically (e.g., from databases, natural language processing, etc.) or manually.	W_6
Defining rules. You have to include into your racer file the rules built on top of your ontology.	W_7
Defining queries. You have to include into your racer file the queries used to test your ontology.	W_8
Refining ontology. You have to send the racer and owl file of your ontology. The refined ontology should rely on various ontology design patterns.	W_{11}
Evaluating and documenting ontology. You have to evaluate your ontology against various ontology evaluation metrics and to write your final report (this template).	W_{12}
Presenting ontology. You have to present and send your beamer presentation.	W_{13}

Plagiarism. Most of you consider plagiarism only a minor form of cheating. This is far from accurate. Plagiarism is passing off the work of others as your own to gain unfair advantage.

During your project presentation and documentation, I must not be left with doubts on which parts of your project are your work or not. Always identify both: 1) who you worked with and 2) where you got your part of the code or solution. You should sign the declaration of originality.

Describe clearly the starting point of your solution. List explicitly any code re-used in your project. List explicitly any help (including debugging help, design discussions) provided by others (including colleagues or teaching assistant). Keep in mind that it is your own project and not the teaching assistant's project. Learning by collaborating does remain an effective method. You can use it, but don't forget to mention any kind of support.

Class attendance. You are free to manage your classes aiming to submit the project earlier, as long as you meet all the constraints and deadlines. However, it is mandatory to participate at the final public presentation of your ontology.

Chapter 2

Ontology documentation

2.1 Competency questions and Use Cases

Pandemia actuală de Covid-19 a afectat majoritatea sectoarelor socio-economice existente, iar acum la peste un an de la izbucnirea epidemiei, procesul de vaccinare a început pe întreg mapamondul. Scopul acestei ontologii este de a modela și analiza procesul de vaccinare pe teritoriul României.

Principalele aspecte pe care le vom considera sunt:

- Ce reacții adverse pot exista în cazul vaccinării cu un anumit ser, ținând cont de starea de sănătate a pacientului.
- Aproximări ale datelor de vaccinare.
- Corelarea numărului de vaccinări cu numărul de cazuri pozitive pe zi.

2.2 Related ontologies

Ontologia pe care am utilizat-o în proiectul nostru este :

- DBpedia (https://wiki.dbpedia.org/develop/datasets)

Informații actualizate zilnic privind vaccinarea în România: https://datelazi.ro/

2.3 Tboxes

Classes

ADULT

ADULT $\equiv$ PERSON $\sqcap$ $\exists$ HAS-AGE Error107 $\sqcap$ $\exists$ HAS-AGE Error108
ADULT $\sqsubseteq$ $\neg$CHILD $\sqcap$ $\neg$TEENAGER
ADULT $\sqsubseteq$ $\neg$ SENIOR
ADULT $\sqsubseteq$ $\neg$ CHILD
ADULT $\sqsubseteq$ $\neg$ TEENAGER

CHILD

CHILD ≡ PERSON ⊓ ∃ HAS-AGE Error103
CHILD ⊑ ¬ SENIOR
CHILD ⊑ ¬ TEENAGER
CHILD ⊑ ¬ ADULT

COUNTY

DISEASE

ESSENTIAL-JOB

FEMALE

FEMALE ⊑ ¬MALE
FEMALE ⊑ ¬ MALE

JOB

JOB ≡ ESSENTIAL-JOB ⊔ MEDICAL-JOB ⊔ OTHER-JOB

MALE

MALE ⊑ ¬ FEMALE

MEDICAL-JOB

OTHER-JOB

PERSON

PERSON ≡ FEMALE ⊔ MALE
PERSON ≡ ADULT ⊔ CHILD ⊔ SENIOR ⊔ TEENAGER

RESTRICTION-TYPE

SCENARIO

SENIOR

SENIOR ≡ PERSON ⊓ ∃ HAS-AGE Error104
SENIOR ⊑ ¬ADULT ⊓ ¬CHILD ⊓ ¬TEENAGER
SENIOR ⊑ ¬ CHILD
SENIOR ⊑ ¬ TEENAGER
SENIOR ⊑ ¬ ADULT

SIDEEFFECT

TEENAGER

TEENAGER ≡ PERSON ⊓ ∃ HAS-AGE Error105 ⊓ ∃ HAS-AGE Error106
TEENAGER ⊑ ¬CHILD
TEENAGER ⊑ ¬ SENIOR
TEENAGER ⊑ ¬ CHILD
TEENAGER ⊑ ¬ ADULT

Thing

VACCINATION-PHASE

VACCINE

VACCINETYPE

maxInclusive

minInclusive

simpleType

Object properties

HAS-AGE

HAS-DISEASE

$\exists$ HAS-DISEASE Thing $\sqsubseteq$ PERSON
$\top \sqsubseteq \forall$ HAS-DISEASE DISEASE

HAS-JOB

$\exists$ HAS-JOB Thing $\sqsubseteq$ PERSON
$\top \sqsubseteq \forall$ HAS-JOB JOB

HAS-NEXT

$\exists$ HAS-NEXT Thing $\sqsubseteq$ VACCINATION-PHASE
$\top \sqsubseteq \forall$ HAS-NEXT VACCINATION-PHASE

HAS-RESTRICTION

$\exists$ HAS-RESTRICTION Thing $\sqsubseteq$ COUNTY
$\top \sqsubseteq \forall$ HAS-RESTRICTION RESTRICTION-TYPE

HAS-SIDE-EFFECT

$\exists$ HAS-SIDE-EFFECT Thing $\sqsubseteq$ VACCINE
$\top \sqsubseteq \forall$ HAS-SIDE-EFFECT SIDEEFFECT

HAS-TYPE

$\exists$ HAS-TYPE Thing $\sqsubseteq$ VACCINE
$\top \sqsubseteq \forall$ HAS-TYPE VACCINETYPE

IN-SCENARIO

$\exists$ IN-SCENARIO Thing $\sqsubseteq$ COUNTY
$\top \sqsubseteq \forall$ IN-SCENARIO SCENARIO

LIVES

$\exists$ LIVES Thing $\sqsubseteq$ PERSON
$\top \sqsubseteq \forall$ LIVES COUNTY

VACCINATE-IN-PHASE

$\exists$ VACCINATE-IN-PHASE Thing $\sqsubseteq$ PERSON
$\top \sqsubseteq \forall$ VACCINATE-IN-PHASE VACCINATION-PHASE

VACCINATE-WITH

$\exists$ VACCINATE-WITH Thing $\sqsubseteq$ PERSON
$\top \sqsubseteq \forall$ VACCINATE-WITH VACCINETYPE

2.4 Aboxes

2.5 Atomic Concepts

1. Person
2. Male
3. Female
4. Gender
5. Child
6. Teenager
7. Adult
8. Senior
9. Disease
10. Vaccination Phase
11. Job
12. County
13. City
14. Vaccine
15. Vaccine Type
16. Side Effect

2.6 Roles

1. (lives :domain Person :range County)
2. (has-disease :domain Person :range Disease)
3. (has-job :domain Person :range Job)
4. (vaccinate-in-phase :domain Person :range Vaccination-Phase)

5. (vaccinate-with :domain Person :range VaccineType)
6. (has-side-effect :domain Vaccine :range SideEffect)
7. (has-type :domain Vaccine :range VaccineType)
8. (in-scenario :domain County :range Scenario)
9. (has-restriction :domain County :range Restriction-Type)
10. (has-next :domain Vaccination-Phase :range Vaccination-Phase)

2.7 Rules

2.7.1 Vaccinated in Phase 1 Rule

```
(define-rule (?x Phase-1 vaccinate-in-phase)
  (and (?x Person) (?x ?z has-job) (?z Medical-Job)))
```

Scopul acestei reguli este de a defini mulțimea populației cu drept de vaccinare în Etapa 1. Din această categorie face parte doar personalul medical.

2.7.2 Vaccinated in Phase 2 Rule

```
(define-rule (?x Phase-2 vaccinate-in-phase)
 (and (?x Person)
      (or (and (?x ?z has-job) (?z Essential-Job)) (and (?x ?w has-disease) (?w Disease)))))
```

Regula descrie ce parte a populației se poate vaccina in Etapa 2, fiind vorba de persoanele ce activează în domenii esențiale și cei bolnavi cronic.

2.7.3 Vaccinated in Phase 3 Rule

```
(define-rule (?x Phase-3 vaccinate-in-phase)
  (and (?x Person)))
```

2.7.4 If a county has an incidence bigger than .6 then it is in the Red scenario

```
(define-rule (?x Red in-scenario)
  (?x (and County (an has-incidence) ( > has-incidence 0.6))))
```

2.7.5 Incidence in interval (0.4, 0.6] -¿ Yellow

```
(define-rule (?x Yellow in-scenario)
  (?x (and County (an has-incidence) ( > has-incidence 0.4) (<= has-incidence 0.6))))
```

2.7.6 Incidence smaller than 0.4 -¿ Green Scenario

```
(define-rule (?x Green in-scenario)
  (?x (and County (an has-incidence) ( <= has-incidence 0.4))))
```

2.7.7 Counties in RED scenario are in lockdown

```
(define-rule (?x Lockdown has-restriction)
  (and (?x Red in-scenario)))
```

2.7.8 Counties in Yellow scenario only have hour restricitions

```
(define-rule (?x Hour-Restriction has-restriction)
  (and (?x Yellow in-scenario) ))
```

2.7.9 Counties in Green scenario dont have restrictions

```
(define-rule (?x No-Restriction has-restriction)
  (and (?x Green in-scenario) ))
```

2.7.10 Childs, Teenagers and Senior should be vaccinated with arn-based vaccines

```
(define-rule (?x arn-based vaccinate-with)
  (or (?x Child) (?x Teenager) (?x Senior) ))
```

2.7.11 Adults should be vaccinated with viral-vector vaccines

```
(define-rule (?x viral-vector vaccinate-with)
  (or (?x Adult)) )
```

2.8 Queries NRQL

2.8.1 Q1: Câte tipuri de vaccin sunt disponibile?

```
(evaluate (length (retrieve-concept-instances
    'Vaccine (current-abox) (all-individuals))))
```

2.8.2 Q2: Care sunt localitățile județului Cluj?

```
(retrieve (?x)
  (and (?x City)
  (Cluj ?x part_of)))
```

2.8.3 Q3: Care sunt reacțiile adverse posibile în urma vaccinării cu AstraZeneca?

```
(retrieve (?x )
  (and (?y SideEffect)
  (az ?y has_side_effect)))
```

2.8.4 Q4: Care sunt vaccinurile create pe bază de arn viral?

```
(retrieve (?x )
  (and (?x Vaccine)
  (?x arn_based has_type)))
```

2.9 Reusing other ontologies

Am utilizat DBPedia pentru a popula automat ontologia cu date despre populația unui județ. Datele sunt preluate utilizând urmatoarea interogare SPARQL:

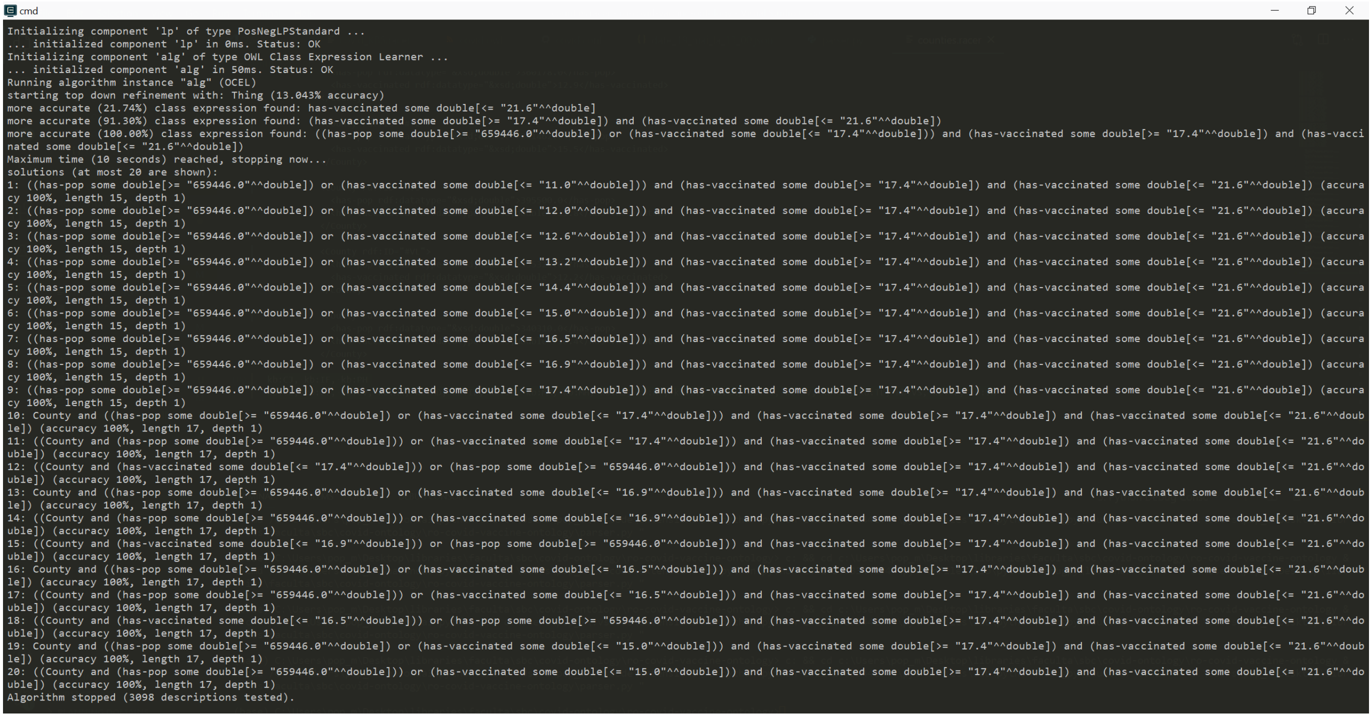
```
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX page: <https://dbpedia.org/page/>

select distinct(?county) ?population where {
?county dbo:type ?type.
?county  dbo:populationTotal ?population.
FILTER(?type = dbr:Counties_of_Romania)
}
```

Rezultatele acestei interogări sunt:

```
(attribute−filler Vaslui_County 395500 has−pop)
(attribute−filler Cluj_County 691106 has−pop)
(attribute−filler Olt_County  415530 has−pop)
(attribute−filler Maramures_County  516562 has−pop)
(attribute−filler Harghita_County 304969 has−pop)
(attribute−filler Hunedoara_County  396253 has−pop)
(attribute−filler Ialomita_County 293352 has−pop)
(attribute−filler Iasi_County 772348 has−pop)
(attribute−filler Ilfov_County  999562 has−pop)
(attribute−filler Satu_Mare_County  329079 has−pop)
(attribute−filler Tulcea_County 201462 has−pop)
(attribute−filler Constanta_County  684082 has−pop)
(attribute−filler Covasna_County  210177 has−pop)
(attribute−filler Calarasi_County 285050 has−pop)
(attribute−filler Salaj_County  343347 has−pop)
(attribute−filler Vrancea_County  340310 has−pop)
(attribute−filler Valcea_County 355320 has−pop)
(attribute−filler Neamt_County  470766 has−pop)
(attribute−filler Prahova_County  762886 has−pop)
(attribute−filler Suceava_County  634810 has−pop)
(attribute−filler Mures_County  550846 has−pop)
(attribute−filler Mehedinti_County  303878 has−pop)
(attribute−filler Teleorman_County  360178 has−pop)
```

2.10 DL-Learner

Pentru a utiliza DL-Learner, am convertit ontologia în format OWL folosind comanda ’save-kb’. Problema pe care dorim să o implementăm este următoarea: ”Predicting whether the number of cases will be high(positive class) or low(negative class) based on population and percentage of vaccinated people”.



```
Initializing component 'lp' of type PosNegLPStandard ...
... initialized component 'lp' in 0ms. Status: OK
Initializing component 'alg' of type OWL Class Expression Learner ...
... initialized component 'alg' in 50ms. Status: OK
Running algorithm instance "alg" (OCEL)
starting top down refinement with: Thing (13.043% accuracy)
more accurate (21.74%) class expression found: has-vaccinated some double[<= "21.6"^^double]
more accurate (91.30%) class expression found: (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double])
more accurate (100.00%) class expression found: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "17.4"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vacci
nated some double[<= "21.6"^^double])
Maximum time (10 seconds) reached, stopping now...
solutions (at most 20 are shown):
1: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "11.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
2: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "12.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
3: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "12.6"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
4: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "13.2"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
5: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "14.4"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
6: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "15.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
7: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "16.5"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
8: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "16.9"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
9: ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "17.4"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^double]) (accura
cy 100%, length 15, depth 1)
10: County and ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "17.4"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^doub
le]) (accuracy 100%, length 17, depth 1)
11: ((County and (has-pop some double[>= "659446.0"^^double])) or (has-vaccinated some double[<= "17.4"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
12: ((County and (has-vaccinated some double[<= "17.4"^^double])) or (has-pop some double[>= "659446.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
13: County and ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "16.9"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^doub
le]) (accuracy 100%, length 17, depth 1)
14: ((County and (has-pop some double[>= "659446.0"^^double])) or (has-vaccinated some double[<= "16.9"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
15: ((County and (has-vaccinated some double[<= "16.9"^^double])) or (has-pop some double[>= "659446.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
16: County and ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "16.5"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^doub
le]) (accuracy 100%, length 17, depth 1)
17: ((County and (has-pop some double[>= "659446.0"^^double])) or (has-vaccinated some double[<= "16.5"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
18: ((County and (has-vaccinated some double[<= "16.5"^^double])) or (has-pop some double[>= "659446.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
19: County and ((has-pop some double[>= "659446.0"^^double]) or (has-vaccinated some double[<= "15.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^doub
le]) (accuracy 100%, length 17, depth 1)
20: ((County and (has-pop some double[>= "659446.0"^^double])) or (has-vaccinated some double[<= "15.0"^^double])) and (has-vaccinated some double[>= "17.4"^^double]) and (has-vaccinated some double[<= "21.6"^^do
uble]) (accuracy 100%, length 17, depth 1)
Algorithm stopped (3098 descriptions tested).
```

2.11 Automatic ontology population

Datele sunt preluate sub form de JSON de pe site-ul datelazi.ro, iar acestea conțin informații referitoare la situația vacinării pe fiecare județ. Pentru parsarea fișierului, am utilizat un script de Python care scrie instrucțiunile necesare populării ontologiei sub formă de text.
Parseul utilizat:

```python
import json
import os

dirname = os.path.dirname(os.path.realpath(__file__))

lut = {
    'VS': 'Vaslui_County',
    'CJ': 'Cluj_County',
    'OT': 'Olt_County',
    'MM': 'Maramures_County',
    'HR': 'Harghita_County',
    'HD': 'Hunedoara_County',
    'IL': 'Ialomita_County',
    'IS': 'Iasi_County',
    'IF': 'Ilfov_County',
    'SM': 'Satu_Mare_County',
    'TL': 'Tulcea_County',
    'CT': 'Constanta_County',
    'CV': 'Covasna_County',
    'CL': 'Calarasi_County',
    'SJ': 'Salaj_County',
    'VN': 'Vrancea_County',
    'VL': 'Valcea_County',
    'NT': 'Neamt_County',
    'PH': 'Prahova_County',
    'SV': 'Suceava_County',
    'MS': 'Mures_County',
    'MH': 'Mehedinti_County',
    'TR': 'Teleorman_County'
}

pop = {
  'Vaslui_County':  395500,
  'Cluj_County':  691106,
  'Olt_County': 415530,
  'Maramures_County': 516562,
  'Harghita_County':  304969,
  'Hunedoara_County': 396253,
  'Ialomita_County':  293352,
  'Iasi_County':  772348,
  'Ilfov_County': 999562,
  'Satu_Mare_County': 329079,
  'Tulcea_County':  201462,
  'Constanta_County': 684082,
  'Covasna_County': 210177,
  'Calarasi_County':  285050,
  'Salaj_County': 343347,
  'Vrancea_County': 340310,
  'Valcea_County':  355320,
  'Neamt_County': 470766,
  'Prahova_County': 762886,
  'Suceava_County': 634810,
  'Mures_County': 550846,
  'Mehedinti_County': 303878,
  'Teleorman_County': 360178
}

if __name__ == '__main__':
  path = os.path.join(dirname, 'date_19_mai_la_13_00.json')

  with open(path, 'r') as fd:
    data = json.loads(fd.read())
    counties_cases = data['currentDayStats']['countyInfectionsNumbers']
    counties_incidence = data['currentDayStats']['incidence']
    counties_vaccinated = data['currentDayStats']['countyVaccinatedPercent']
    counties_cases_yesterday = data['historicalData']['2021-05-18']['countyInfectionsNumbers']

    rules_cases, rules_incidence, rules_vaccinated, owl_kb = [], [], [], []
    i = 1

    # dl-learner
    pos, negs = [], []

    for county in counties_cases.keys():
      if county in lut:
        today_cases = counties_cases[county] - counties_cases_yesterday[county]
        cases = f'(attribute-filler {lut[county]} {today_cases} has-cases)\n'
        incidence = f'(attribute-filler {lut[county]} {counties_incidence[county]} has-incidence)\n'
        vaccinated = f'(attribute-filler {lut[county]}  {counties_vaccinated[county]} has-vaccinated)\n'

        rules_cases.append(cases)
        rules_incidence.append(incidence)
        rules_vaccinated.append(vaccinated)

        # dl-learner
        owl = f"""
```

```
            <County rdf:ID="#{county}">
                <has-pop rdf:datatype="&xsd;double">{float(pop[lut[county]])}</has-pop>
                <has-vaccinated rdf:datatype="&xsd;double">{counties_vaccinated[county]}</has-vaccinated>
            </County>
        """

        if today_cases > 25:
            pos.append(f'kb:{county}')
        else:
            negs.append(f'kb:{county}')

        owl_kb.append(owl)
        i += 1

# print(len(rules), len(lut.keys()))
assert len(rules_cases) == len(lut.keys())

# print(rules)
with open(os.path.join(dirname, 'counties.racer'), 'w') as fd_counties:
    fd_counties.writelines(rules_cases)
    fd_counties.write('\n')
    fd_counties.writelines(rules_incidence)
    fd_counties.write('\n')
    fd_counties.writelines(rules_vaccinated)

    # dl-learner
    fd_counties.write('\n')
    fd_counties.writelines(owl_kb)

    fd_counties.write('\n')
    fd_counties.writelines(', '.join(pos))
    fd_counties.write('\n')
    fd_counties.writelines(', '.join(negs))
```

Rezultatele se scriu în fișierul counties.racer.

2.12 Ontology verbaliser

Pentru a verbalisa ontologia, am utilizat un tool scris în Prolog.
Tool Utilizat: `https://github.com/Kaljurand/owl-verbalizer` Dupa rularea scriptului utilizând SWI-Prolog, rezultatele sunt scrise în fișierul natural_language.out.

Every JOB is something that is an ESSENTIAL-JOB or that is a MEDICAL-JOB or that is an OTHER-JOB.

Everything that is an ESSENTIAL-JOB or that is a MEDICAL-JOB or that is an OTHER-JOB is a JOB.

Every PERSON is something that is an ADULT or that is a CHILD or that is a SENIOR or that is a TEENAGER.

Everything that is an ADULT or that is a CHILD or that is a SENIOR or that is a TEENAGER is a PERSON.

Every PERSON is something that is a FEMALE or that is a MALE.

Everything that is a FEMALE or that is a MALE is a PERSON.

Every SENIOR is a PERSON that HAS-AGE an Error92.

Every PERSON that HAS-AGE an Error92 is a SENIOR.

Every TEENAGER is a PERSON that HAS-AGE an Error93 and that HAS-AGE an Error94.

Every PERSON that HAS-AGE an Error93 and that HAS-AGE an Error94 is a TEENAGER.

Every ADULT is something that is not a CHILD and that is not a TEENAGER.

No FEMALE is a MALE.

Every SENIOR is something that is not an ADULT and that is not a CHILD and that is not a TEENAGER.

No TEENAGER is a CHILD.

No ADULT is a CHILD.

No ADULT is a SENIOR.

No ADULT is a TEENAGER.

2.13 Design Patterns

Am utilizat 3 DP în ontologia noastră:

- Partition Design Pattern

```
(disjoint Male Female)
(disjoint Child Teenager Adult Senior)
```

- Constituency Design Pattern

```
(equivalent Person (or Male Female))
(equivalent Person (or Child Teenager Adult Senior))
```

- Criterion Design Pattern

```
(equivalent Child (and Person (max has-age 12)))
(equivalent Teenager (and Person (min has-age 13) (max has-age 19)))
(equivalent Adult (and Person (min has-age 20) (max has-age 64)))
(equivalent Senior (and Person (min has-age 65)))
```

Table 2.1: Ontology evaluation metrics in Racer

Number of concepts	21
Number of roles	29
Number of individuals	67
Number of rules	11
DL expressivity	LCUH(D)-

2.14 Ontology evaluation

Tbox este coerent, iar ciclurile sunt: ((MALE) (FEMALE) (PERSON) (CHILD) (TEENAGER) (ADULT) (SENIOR))

Appendix A

Examples of ontology engineering with Racer

You can find the following ontologies in the directory SampleOntology on Moodle.

1. Ontological modelling of wind energy systems
2. Developing an ontology for the Goal Structuring Notation
3. An ontology for Romanian tourism
4. An ontology vor vehicular networks
5. An ontology of sentiments (developed by Cornea Iulia, Popa Andrada as lab assignment)
6. An ontology of nutrition (developed by Petrican Teodor, Stan Ciprian Adrian as lab assignment)
7. An ontology for tourism recommendation (developd by A. Suciu as his master thesis)

The ontologies developed by your collegues in the previous years are available at https://ontohub.org/ under two repositories:

- OntologyBuildingCompetition
- BOC2016

Appendix B

Your original code

This inclues both Racer or JAVA if you decided to use OWLAPI.
This section should contain only code developed by you, without any line re-used from other sources. This section helps me to correctly evaluate your amount of work and results obtained. Including in this section any line of code taken from someone else, leads to failure of KBS class this year.
Sample of a Racer ontology (developed by Borsos and Varady):

Intelligent Systems Group