INFOIAG: Intelligent Agents 2020
DL Exercises

---

1. Given the following classes "person", "parent", "happyPerson", "marriedPerson", "malePerson" and the relation "hasChild". Represent the following in DL:

   - Father
     father $\equiv$ parent $\sqcap$ malePerson
   - A parent is brave if and only if she/he has exactly two children
     braveParent $\equiv$ parent $\sqcap \geq 2$ hasChild.$\top \sqcap \leq 2$ hasChild.$\top$
   - Strange people are parents who have only married or happy kids
     strangePerson $\sqsubseteq$ parent $\sqcap\forall$ hasChild.(happyPerson $\sqcup$ marriedPerson)
   - Retired people are happy people who have at least one happy married child
     retiredPerson $\sqsubseteq$ happyPerson $\sqcap\exists$ hasChild.(happyPerson $\sqcap$ marriedPerson)
   - Married men do not have children
     marriedPerson $\sqcap$ malePerson $\sqcap\exists$ hasChild.$\top \sqsubseteq \bot$
   - Michael is a father with children named Alice and Bob
     michael:father, (michael, alice):hasChild, (michael,bob):hasChild
   - Charlie is a married man
     charlie:(marriedPerson $\sqcap$ malePerson)

2. The following ontology has been defined in DL:

   - instructor $\equiv$ person $\sqcap$ $\exists$teaches.(course $\sqcup$ lab)
   - projectCourse $\equiv$ course $\sqcap$ lab
   - busyInstructor $\equiv$ instructor $\sqcap$ $\forall$teaches.projectCourse
   - john: instructor
   - simulation: projectCourse
   - (john, simulation): teaches

   Apply the tableaux algorithm to answer the following and show your steps:

   - What is the T-box? What is the A-box? Is this ontology consistent? If not, what has to change to make it consistent?

     - T = {instructor $\equiv$ person $\sqcap$ $\exists$teaches.(course $\sqcup$ lab), projectCourse $\equiv$ course $\sqcap$ lab, busyInstructor $\equiv$ instructor $\sqcap$ $\forall$teaches.projectCourse }
       A = {john: instructor, simulation: projectCourse, (john, simulation):teaches}

     - Using the intersection rule on instructor,
       A = A $\cup$ {john:person, john: $\exists$ teaches.(course $\sqcup$ lab)}

- Using the intersection rule on simulation,
  A = A $\cup$ $\{$simulation:course, simulation:lab$\}$
- No need to add a new instance for the existential restriction because (john, simulation):teaches and simulation:course $\sqcup$ lab
- A = {john: instructor, simulation: projectCourse, (john, simulation):teaches, john:person, john: $\exists$ teaches.(course $\sqcup$ lab), simulation:course, simulation:lab$\}$ and nothing further to add to A. There is no CLASH. Hence, the ontology is consistent.

- Is simulation a lab? (Put the negation of the query $\{$simulation: $\neg$lab$\}$ in A)
  - T = $\{$instructor $\equiv$ person $\sqcap$ $\exists$teaches.(course $\sqcup$ lab), projectCourse $\equiv$ course $\sqcap$ lab, busyInstructor $\equiv$ instructor $\sqcap$ $\forall$teaches.projectCourse $\}$
    A = {john: instructor, simulation: projectCourse, (john, simulation):teaches, simulation: $\neg$lab$\}$

  - The steps in the previous example will be applied, leading to:
    A = {john: instructor, simulation: projectCourse, (john, simulation):teaches, john:person, john: $\exists$ teaches.(course $\sqcup$ lab), simulation:course, simulation:lab, simulation: $\neg$lab$\}$.
  - We need to check if all the branches give a CLASH. In this case, there is a single branch and it gives a CLASH! Yes, we conclude that simulation is a lab.
- Is John a busy instructor? (Put the negation of the query {john: $\neg$busyInstructor$\}$ in A)
  - T = $\{$instructor $\equiv$ person $\sqcap$ $\exists$teaches.(course $\sqcup$ lab), projectCourse $\equiv$ course $\sqcap$ lab, busyInstructor $\equiv$ instructor $\sqcap$ $\forall$teaches.projectCourse $\}$
    A = {john: instructor, simulation: projectCourse, (john, simulation):teaches, john: $\neg$busyInstructor$\}$

  - Apply the steps in 2a, so we have: A = {john: instructor, simulation: projectCourse, (john, simulation):teaches, john:person, john: $\exists$ teaches.(course $\sqcup$ lab), simulation:course,john: $\neg$busyInstructor$\}$

  - Apply intersection on busyInstructor. Note that we have the complement, hence we have to distribute that over the busyInstructor to make it Negation Normal Form. So,
    $\neg$busyInstructor $\equiv$ $\neg$(instructor $\sqcap$ $\forall$teaches.projectCourse)
    $\equiv$ $\neg$instructor $\sqcup$ $\exists$teaches.$\neg$projectCourse.

    A' = A $\cup$ {john:$\neg$instructor$\}$ A''= A $\cup$ {john:$\exists$teaches.$\neg$projectCourse$\}$
  - We have two paths to check for a clash and we can answer yes only if both paths have a clash. A' contains a CLASH because it has both $\neg$instructor and instructor.
  - Make A'' complete by adding the existential restriction rule. A'' = A'' $\cup$ $\{d$:$\neg$projectCourse, (john, d):teaches$\}$. A'' does not have a CLASH.
  - We cannot conclude that John is a busy instructor. At first, this might seem unintuitive because John teaches one course and that is a project course. However, due to

open world assumption, we cannot say that this is the only course that John is teaching and without saying that all his courses are project courses, we cannot conclude that he is a busy instructor.

- If you cannot conclude that John is a busy instructor, what else do you need in your ontology to conclude this?
  For example, we can say that John is not teaching any other course or John only teaches project courses. If we have john:∀teaches.projectCourse as part of the Abox, then the second path above would yield a CLASH because d would become a project course. Having both d as a project course and a not project course would give us the CLASH. Thus, we would be able to conclude that John is a busy instructor.