Semantic Web & Ontologies

Pablo Mollá Chárlez

October 7, 2024

Contents

1 Exercise 3

This exercise is to recall that description logics is not a single logic, but a family of languages, among which is ALC as we discussed in the lecture. Indeed, OWL and OWL 2 standards contain 7 different description logic languages that differ in syntax and expressivity. Consider the university study scenario, in particular an Examination Regulation given in the Appendix, and answer the following questions.

Appendix:

- Students: The admission jury decides which of the students who apply for the program are admissible. One condition for being admissible is having completed at least 4 years of ternary study in Computer Science or a related field (in France: Bac+4). An admissible candidate becomes a student of the program, if he/she has (1) registered in the proper Paris-Saclay establishment (see Web Page of Paris-Saclay University), (2) registered with the study inspector, and (3) paid any associated fees. All of this has to happen for the current year of study, and before the second week of the start of the current year of study. Exceptions can be granted by the study coordinators in agreement with the reference institute and study coordinator. The study inspector maintains the list of students. If a student does not participate regularly in the program for one period, their inscription can be revoked. The rules that the reference institute has put in place for its diploma program apply accordingly.

- Offered Courses: A course is a unit of instruction (typically a sequence of teaching hours, labs, and/or projects) organized by one or several lecturers at Paris-Saclay University. The program encompasses a number of courses called "mandatory" and a number of courses called "optional". The courses and their ECTS credit numbers are proposed by the program coordinators, in agreement with the lecturers. The courses are displayed on the Web page of the program. All obligatory courses will necessarily take place. The optional courses are not guaranteed to take place (they may or may not start, depending on the number of interested students, and other factors). One optional course is the "Module Liberté". It is a placeholder for any data-oriented course at Paris-Saclay University, subject to the approval of the program coordinators. A student can follow at most one Module Liberté. The Module Liberté will necessarily take place. The Module Liberté is worth a predetermined number of ECTS credits, no matter how many credits are awarded by the program that offers the course. Courses typically take place in the first semester of the study year. Students that are not registered for the program can participate in the courses at the discretion of the lecturer.

- Examinations: A course is evaluated by examinations. An examination can be a midterm exam, a final exam, projects, homework, graded practical labs, presentations or similar items. Each examination is graded on a 0-20 scale, with 20 being the maximal grade and 10 being the passing grade. The following applies to all courses except for the Module Liberté. Each 4 course offers two rounds of examinations. Exams of both rounds take place during predetermined periods. For each round, the lecturer defines a grading scheme, i.e., (1) the examinations that play into the grade of the round and (2) the weight of these examinations for the grade of the round. The grading scheme of the second round can include examinations from the first round. The lecturer communicates the grading schemes to the program coordinators before the start of the study year. The grading schemes are displayed on the Web page of the program at most one month after the start of the study year. If a student commits fraud in an examination (including plagiarism), the grade of this examination is zero. If the student copied from another student in the course, the other student also receives a grade of zero for this examination. Upon request by either the students or the lecturer, the case is referred to the jury. The jury can either exonerate the students or ask the reference institute to apply the sanctions for fraud that it has put in place for its diploma program.

Questions:

1. **List the atomic concept names, role names, and individual names for this scenario (for example, Students is an atomic name, hasStudentId is a role, and Pierre can be an individual).**

   Answer

   We categorize the scenario described in the appendix into atomic concept names, role names, and individual names as follows:

   - Atomic Concept Names: These represent the basic classes or categories of entities in the scenario.
     - Student: Represents a person who has applied and been admitted to the program.
     - AdmissibleCandidate: Represents a person who meets the criteria to apply for the program but has not yet registered.
     - RegisteredStudent: Represents a student who has fulfilled all the conditions to be officially part of the program.
     - Course: Represents a unit of instruction in the program.
     - MandatoryCourse: Represents a course that is required for all students.
     - OptionalCourse: Represents a course that students can choose to take.
     - ModuleLiberté: Represents a specific type of optional course.
     - Examination: Represents the evaluation of a course, such as exams, projects, or labs.
     - GradingScheme: Represents the scheme that determines how examinations are graded.
     - Fraud: Represents an instance of academic dishonesty, such as plagiarism.
     - Jury: Represents the committee that oversees examinations and appeals related to fraud cases.
   - Role Names: These describe relationships between concepts.
     - hasStudentId: Represents the relationship between a student and their student ID.
     - hasApplied: Represents the relationship between an individual and their application to the program.
     - hasAdmissibility: Represents the relationship between an individual and the decision of whether they are admissible.
     - hasRegistration: Represents the relationship between a student and their registration in the program.
     - hasPaidFees: Represents the relationship between a student and the payment of fees.
     - hasCourse: Represents the relationship between a student and the course they are enrolled in.
     - hasExamination: Represents the relationship between a course and the associated evaluation or exam.
     - hasGradingScheme: Represents the relationship between a course and its grading scheme.
     - commitsFraud: Represents the relationship between a student and an act of academic dishonesty.

- hasGrade: Represents the relationship between a student and their grade in an exam or course.
- hasSanction: Represents the relationship between a student and any sanctions applied for fraud.
- hasDiploma: Represents the relationship between a given student having obtained a diploma for a given course.

- Individual Names: These represent specific instances of concepts or roles.
  - Pierre: Can represent an individual student.
  - CS101: Can represent a specific course in Computer Science.
  - Exam1: Can represent a specific examination.
  - Semester1: Can represent a specific period in the academic year.
  - Jury1: Can represent the specific jury for a course or examination.

2. **Construct several complex concepts using different Description Logic constructors given below**:

$$\sqcup,\ \sqcap,\ \neg,\ \exists R.C,\ \forall R.C,\ \geq n\ R.C,\ \leq n\ R.C$$

For each of these complex concepts, give its explanation in natural language. For example, $Student \sqcap \exists hasDiploma.Engineering$ **means "all students who have obtained Engineering diploma".**

Answer

Here are several complex concepts constructed using Description Logic (DL) constructors:

(a) $RegisteredStudent \sqcap \exists\ hasCourse.MandatoryCourse$

Explanation: All registered students who are enrolled in at least one mandatory course.

(b) $AdmissibleCandidate \sqcap \neg\ RegisteredStudent$

Explanation: All admissible candidates who are not yet registered as students.

(c) $Student \sqcap \forall\ hasCourse.OptionalCourse$

Explanation: All students who are enrolled only in optional courses.

(d) $RegisteredStudent \sqcap \exists\ hasExamination.Examination \sqcap \forall\ hasGrade. \geq\ 10$

Explanation: All registered students who have taken an examination and passed with a grade of at least 10.

(e) $\exists\ commitsFraud.Fraud \sqcap \forall\ hasSanction.ZeroGrade$

Explanation: All students who have committed academic fraud and have received a sanction of a zero grade.

(f) $Course \sqcap\ \geq 10\ hasStudentId$

Explanation: All courses with at least 10 students enrolled.

(g) $\exists\ hasGradingScheme.\ (\geq 1\ hasExamination \sqcap \forall hasGrade. \geq 15)$

Explanation: Courses that have at least one examination, and in that exam, all students scored at least 15.

(h) $OptionalCourse \sqcap\ \leq 5\ hasStudentId$

Explanation: Optional courses with no more than 5 students enrolled.

3. **Construct a TBox and a ABox for this domain. In Description Logic, an element from TBox or Abox is called an axiom. For example,** $Apprentice = Student \sqcap \exists\, WorkIn.Enterprise$ **can be an element from TBox (so it is called an axiom), and WorksIn(John, Thales) can be an ABox element. You are required to build a TBox having at least 3 TBox axioms and an ABox having at least 3 axioms.**

   Answer

   In Description Logic (DL), the TBox contains axioms about concepts and their relationships, while the ABox contains axioms about individuals and their relationships. Here's the proposed construction of a TBox and an ABox:

   TBox Axioms (concept definitions)

   (a) $RegisteredStudent = Student \sqcap \exists hasRegistration.True$

   Explanation: A registered student is a student who has completed the registration process and holds a registered status which is True.

   (b) $(MandatoryCourse \sqcup OptionalCourse) \sqsubseteq Course$

   Explanation: All mandatory courses are a subset of courses, meaning every mandatory course is also a course.

   (c) $AdmissibleCandidate = Student \sqcap \exists hasAdmissibility.True \sqcap \neg RegisteredStudent$

   Explanation: An admissible candidate is a student who has met the admission criteria but is not yet a registered student.

   ABox Axioms (facts about individuals)

   (a) $Student(Pierre)$

   Explanation: Pierre is an individual who is a student.

   (b) $hasCourse(Pierre, CS101)$

   Explanation: Pierre is enrolled in the course CS101.

   (c) $RegisteredStudent(Pierre)$

   Explanation: Pierre is a registered student.

   (d) $hasExamination(CS101, Exam1)$

   Explanation: The course CS101 includes the examination Exam1.

   (e) $hasGrade(Pierre, Exam1, 18)$

   Explanation: Pierre received a grade of 18 in Exam1.

4. **Can you infer some implicit information from the ontology O =(TBox, ABox) from the TBox and ABox you built above? For example, I can deduce Apprentice(John) based on the TBox**

$$T = \{Apprentice = Student \sqcap \exists\, WorkIn.Enterprise\}$$

and ABox

$$A = \{WorksIn(John, Thales),\ Student(John),\ Enterprise(Thales)\}$$

$\boxed{\textbf{Answer}}$

Implicit information we can infer:

(a) **Pierre is a Student and a RegisteredStudent**

From RegisteredStudent(Pierre), and from the axiom RegisteredStudent = Student$\sqcap\exists$ hasRegistration.True, we can infer that **Pierre is also a student** and has completed the registration process. This is already explicitly stated in the ABox (Student(Pierre)), so it is consistent with the ontology.

(b) **CS101 is a Course**

From the axiom MandatoryCourse $\sqcup$ OptionalCourse $\sqsubseteq$ Course and knowing that Pierre is taking CS101, we can infer that **CS101 must be a course** because all mandatory and optional courses are courses. Even though CS101 is not explicitly declared as a **Course** in the ABox, the axiom structure implies this.

(c) **Pierre must have registered for CS101 before**

From the fact that RegisteredStudent(Pierre) and hasCourse(Pierre, CS101), we can infer that **Pierre must have registered for CS101** as part of the course enrollment process. Even though we don't have a direct ABox axiom stating Pierre's registration for CS101, it's implied by the rules around being a registered student and enrolling in a course.

(d) **Pierre has passed Exam1**

From hasGrade(Pierre, Exam1, 18) and knowing that the passing grade is 10 (based on the examination regulations in the appendix), we can infer that **Pierre has passed Exam1**. This is an implicit result of Pierre's grade being above the threshold for passing.

(e) **AdmissibleCandidate status doesn't apply to Pierre**

From the axiom AdmissibleCandidate = Student $\sqcap\ \exists$ hasAdmissibility.True $\sqcap\ \neg$RegisteredStudent, and knowing that RegisteredStudent(Pierre), we can infer that **Pierre cannot be an admissible candidate** since he is already registered. This is because the definition of admissible candidates excludes those who are already registered students.

(f) **If CS101 is an Optional or Mandatory Course, it is a Course**

While it's not specified in the ABox whether CS101 is an optional or mandatory course, we can infer from the axiom MandatoryCourse $\sqcup$ OptionalCourse $\sqsubseteq$ Course that **CS101 must be one of these** (either optional or mandatory), and thus, implicitly, it must be a course.