

# COMP718: Ontologies and Knowledge Bases

## *Exercises Lecture 4—hints for the answers*

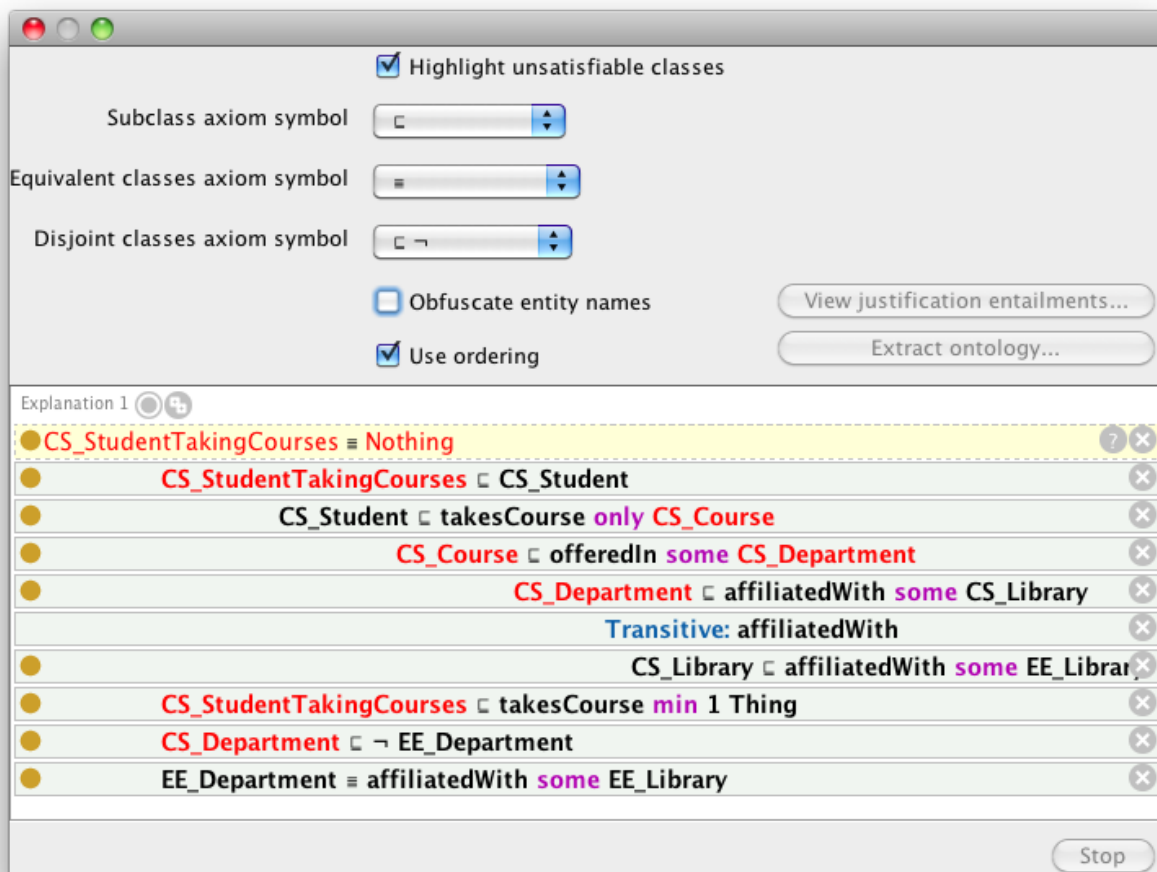
**Lecturer: Dr. Maria Keet**

School of Mathematics, Statistics, and Computer Science,  
University of KwaZulu-Natal, South Africa  
email: keet@ukzn.ac.za, home: <http://www.meteck.org>

March 4, 2012

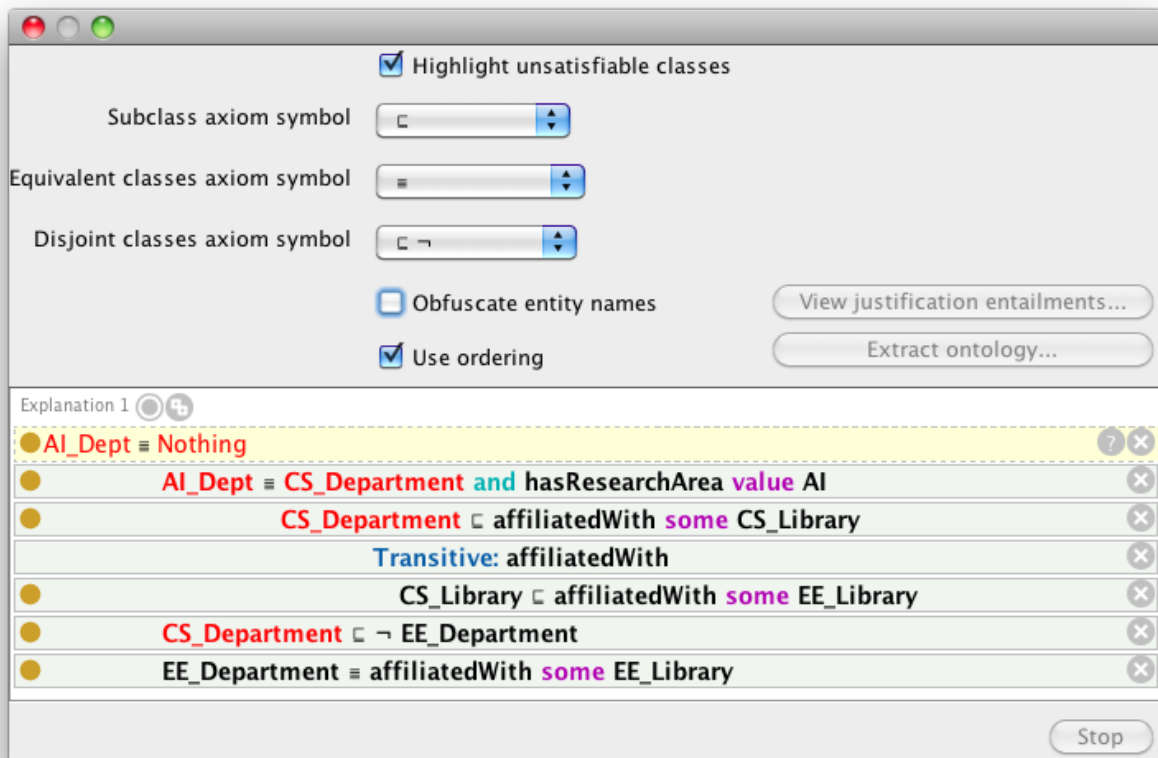
### Reasoning

- *Instance classification and KB consistency.* The answers for the `university1.owl` exercises can be found at <http://owl.man.ac.uk/2005/07/sssw/university.html>.
- *Cleaning up a ‘dirty’ ontology.* Let us randomly have a look at a deduction and its explanation (click on the “?” right from the deduction in Protégé) as a first step toward figuring out why so many classes are unsatisfiable (i.e., equivalent to `Nothing`, or  $\perp$ ). Take the explanation for `CS_StudentTakingCourses`:

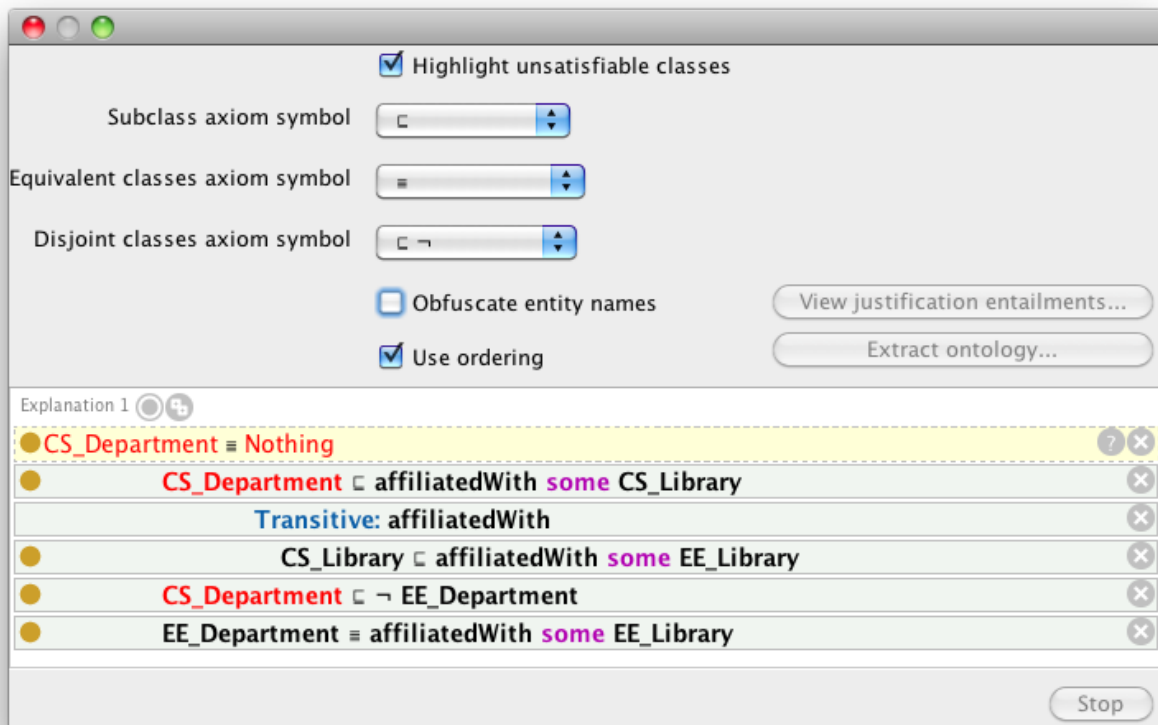


This `CS_StudentTakingCourses` has a long explanation of why it is unsatisfiable, and we see that some of the axioms that it uses to explain the unsatisfiability also have unsatisfiable classes. Hence, it is a good idea to set this aside for a while, as it is a knock-on effect of the others that are unsatisfiable.

Let us have a look at the unsatisfiability regarding departments.



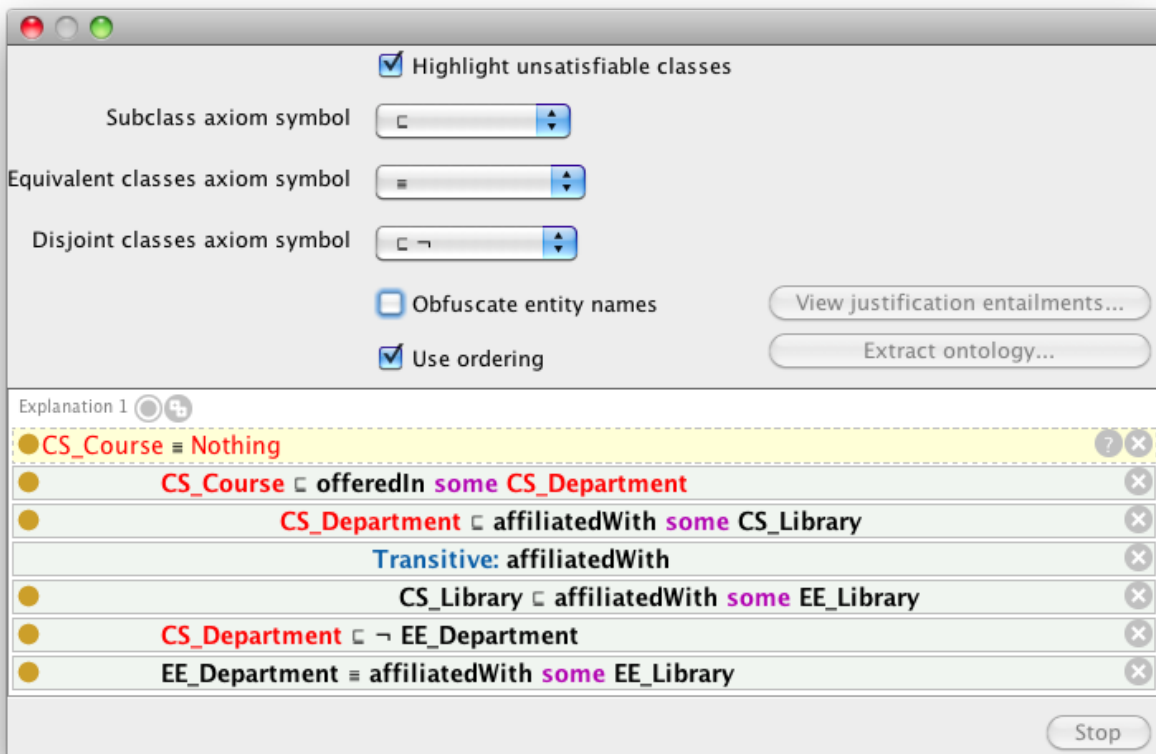
So, the AI\_Dept is unsatisfiable because its superclass CS\_Dept is, i.e., it is a knock-on effect from CS\_Dept. Does this give sufficient information as to say why CS\_Dept is inconsistent? In fact, it does. See the next screenshot, which is the same as lines 3-7, above.



CS\_Dept is unsatisfiable, because it is affiliatedWith some CS\_Library that, in turn (by transitivity), is affiliatedWith some EE\_Library that belongs to the EE\_Dept, which is disjoint from CS\_Dept. Two 'easy' options to get rid of this problem are to remove the transitivity or

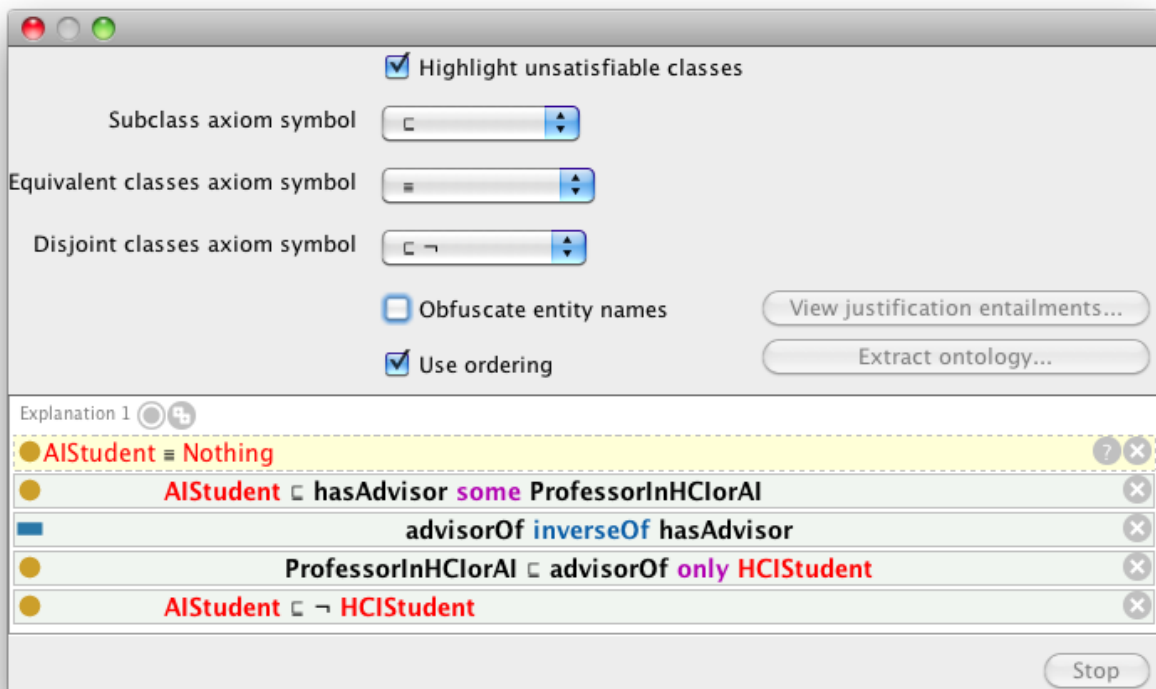
to remove the disjointness. Alternatively, we could revisit the domain knowledge; e.g., CS library may not be `affiliatedWith` EE library, but is, `adjacentTo` or `disjoint` with the EE library.

Let us now consider why `CS_course` is unsatisfiable:

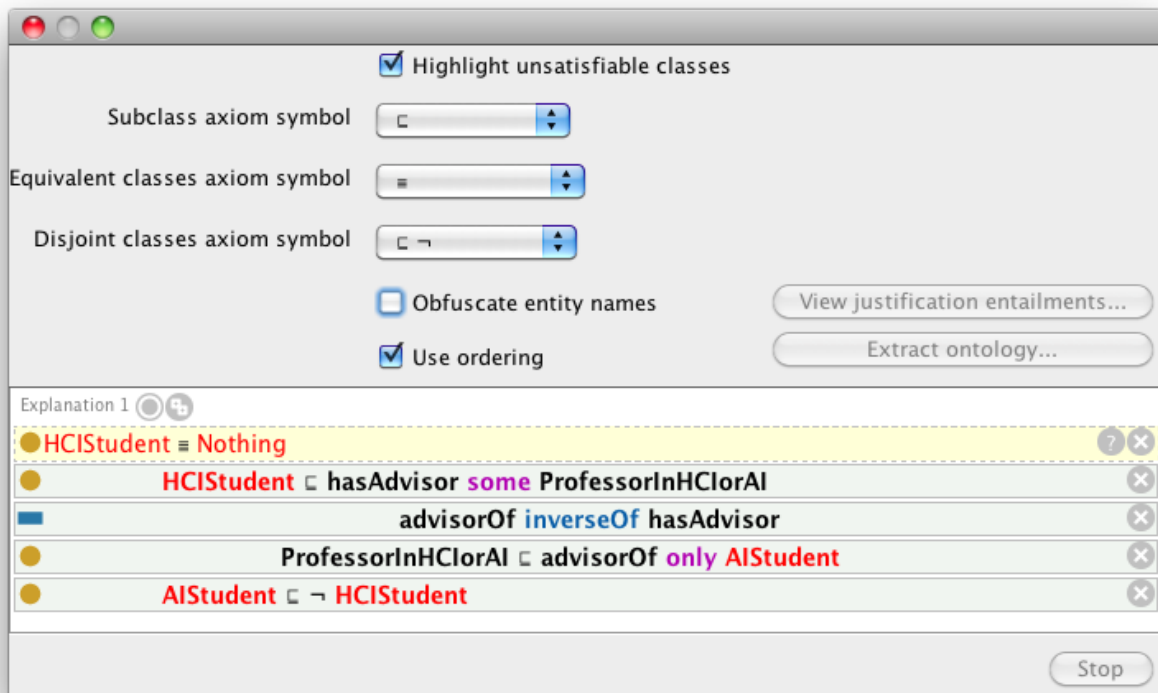


We have again that the real problem is `CS_Department`; fix that one, and `CS_course` is satisfiable, too.

There is a different issue with `AIStudent`. From the explanation in the next screenshot, we can see immediately it has something to do with the inconsistency of `HCISStudent`.



But looking at `HCISStudent` for a clue does not help us further in isolating the problem:



Considering the axioms in the explanation only, one can argue that the root of the problem is the disjointness between `AISTudent` and `HCISStudent`, and remove that axiom to fix it. However, does it really make sense to have the union `ProfessorInHCIorAI`? Not really, and therefore it would be a better fix to change that one into two separate classes, `ProfessorInHCI` and `ProfessorInAI` and have them participating in `ProfessorInHCI`  $\sqsubset \forall$ advisorOf.HCISStudent and `ProfessorInAI`  $\sqsubset \forall$ advisorOf.AISTudent, respectively.

Last, we have a problem of conflicting cardinalities with `LecturerTaking4Courses`: it is a subclass of `TeachingFaculty`, which is restricted to taking at most 3 courses, which is in conflict with the “exactly 4” of `LecturerTaking4Courses`. This can be fixed by changing the cardinality of either one, or perhaps a lecturer taking 4 courses is not a sub- but a sister-class of `TeachingFaculty`.

