# CS6770 KRR Assignment 9

-ph19b011

-cs19b036

We are tasked with the subtask of instance retrieval within the wider goal of coding a Taxonomy Builder for ALC.

**Problem Statement:**

*Given an ALC KB and a concept description C, prepare the taxonomy using assignment 8 and reuse the ALC Tableau implementation from assignment 7 and write a program to identify all the named individuals that belong to C.*

*The program should scan ConceptName from the console (in the loop until the user wants to exit) at run time and print corresponding named individuals on-screen and also save to file <ConceptName>.individuals.txt.*

*Treat "Thing" as the top concept and "Nothing" as the bottom concept.*

Here, we are assuming that we get a simplified knowledge bases of the form (ABox, TBox, RBox, Objects), and a concept description and we need to check which Individuals belong to the concept.

We do not have information about the functions used in other assignments, but we do know the tasks these functions must perform. Thus, we will make certain assumptions related to the naming and output form of these functions, so that we can call to them in our subtask.

We begin by importing the subsumption hierarchy built in assignment 8, and using it to build out a KB. Here we are assuming the function TaxonomyBuilder.

Note that in the given problem statement, we are required to list the individuals belonging to a particular class each time. Thus, it is sufficient to check all named individuals against the KB using the tableau given from assignment 7. If we were needed to do this for all or the majority of the possible concepts, it might be better to only check the individuals in the parent of the given concept in the subsumption hierarchy. As it currently stands, it is better to check for all individuals once rather than checking for all individuals for all ancestor concepts.

In the instance checking using tableau, we are using the implicit assumption that we are importing NotInstanceOf and Tableu from Assignment 8 and 7 respectively. Also, from util import NotInstanceOf.

In this way, (weeding out bad inputs), we write the code that takes the conceptname as input and prints all the individuals that belong to concept C and saves it to a .txt file as required.