**Lab 8 Instructions**

In today’s lab you will build a simple knowledge base as a multigraph. In case you are not familiar with the notion of a multigraph then imagine a normal graph. A graph is defined by its nodes and the edges between nodes. The edges may have weights, in which case we call it a weighted graph. All edges are of the same type (i.e. they carry the same type of information). For example, if the nodes represent cities and the weights on the edges represent some distance between a pair of cities, then all edges represent the same quantity i.e. some distance. Now consider a situation where there can be different types of edges and each type carries a certain type of information. For example, consider the following sentences:

1. Jerry is a cat.
2. Cats are mammals.
3. Mammals are animals.
4. All animals are mortal.
5. Cats have four legs.
6. Cats like to drink milk.

We can identify several entities in the above sentences. A partial list of entities is: {Jerry, cat, mammal, animal, leg, milk}.

A careful observation shows that these entities are related by different relations. A partial list of relations in the above sentences is: {“is a”, “are”, “have”, “to drink”}.

Additionally, there can be attributes like: {“mortal”, “four”, “like”}.

These can be represented as a multigraph. In the simplest case we have each entity as a node. Whenever two entities are related then we have a labelled edge between them. Obviously, since we can have different types of relationships, so we have different types of edges. Also, maintain the direction properly. For example, given the sentence “Jerry is a cat” we get two entities namely “Jerry” and “cat” along with a relation “is a”. So, if we want to draw a multigraph, we will get nodes with labels “Jerry” and “cat” and an edge with the label “is a”. However, we need to maintain the direction which, in this case, will be from “Jerry” to “cat”. Once your multigraph is ready implement a traversal procedure that accepts a pair of nodes and returns true if there is a path from the first node to the second and returns false otherwise. For example, there is a path from “Jerry” to “leg” but not from “leg” to “Jerry”.

Your task for today is as follows:

Download an arbitrary paragraph from the net that has around six sentences. Identify all entities (nouns). You may get some pronouns. Resolve them to their respective nouns. Create a list of all entities. Now identify all the relationships. Create a multigraph with the entities as nodes and the relationships as edges.

**Note:** As usual, you should show your work, even if it is partial, during the lab since each lab is being graded.