# Project 1 Analysis

1) If you used different arrays/structures/classes to represent the different types of sets, would it be possible to have overloaded methods or operations that would provide the correct functionality regardless of whether or not you were using sets, multi sets, or fuzzy sets? Why or why not? (You do not need to code this, just answer the question).

When it comes to some set operations on plain sets vs. fuzzy sets, we would have a problem since the nature of a set operation on a fuzzy set is entirely different when operated on a normal set. For instance, computing the complement of a normal set means you take the opposite truth value of the bit string representing what is included in the set. So, everything not included in the set is included in the complement. When it comes to taking the complement of a fuzzy set, since we are dealing with members and degrees of membership, we look at the degree of membership, which is a number between 0 and 1. The complement will be the difference of 1 and the degree of membership. An overloaded method for the complement set operation would have to have two entirely different implementations to handle the different types of sets.

Possible but not feasible.

2) Would it be possible to use the same data set/structure/class to store sets, multi sets, and fuzzy sets? Why or why not?

It would be possible to have one class that holds fields for each of the different types of sets. For instance, you could have a class called Sets and then fields: setA and setB (normal sets), multiSetA and multiSetB (multisets), and fuzzySetA and fuzzySetB (fuzzysets). Set specific set operation methods could be implemented where applicable. May as well have specialized data structures. Organizationally this would not be good. Conciseness over spaghetti.

3) How easy or difficult is it to determine the type of set that you need to use based on the users query? Why?

Difficult because user input is very subjective vs. the objective nature of computers. Natural language processing is difficult in general.

4) Is it possible to store the data from one type of set (plain sets, multi sets, fuzzy sets) in another type? Would you need to lose data in order to do so? Why?

Multisets could be stored in fuzzy sets and you wouldn’t lose any data. Plain sets can be stored in multisets. Multi sets have weights that would be hard to represent in a multi set or plain set. Plain sets are the simplest, multisets are intermediate, and fuzzy sets are the most complex. You can store simpler types of sets in more complicated sets, but not the other way around without losing data.

5) Discuss what implications your answers to questions 1 – 4 have for someone trying to code an interface which would allow users to type in natural language queries.

First implication is that we cannot store all sets in the same way, which means we have to delineate which one we should be using based on the user’s query.

‘Which of our customers attended SLCC?’ – comparing set of customers to set of those who attended SLCC and performing intersection on them.

‘Which would our customers like more, plain apple pie, apple pie with vanilla ice cream, or apple pie with butter pecan ice cream?’ – data for the weights of customers liking each item of food stored in fuzzy set and then we can find the highest weight to return.

Someone designing an interface to handle these types of queries and must determine associations to figure out what methods to call to handle the query. Challenging to program an interface to choose which platform to use based on how a user inputs a query.

6) Discuss what implications your answers to questions 1 – 5 have for someone trying to code an interface which would allow users to access arbitrary types of data using natural language.

See above answer

In a very fundamental sense, the answers to questions 5 and 6 have not only shaped how the internet has evolved but has equally shaped how people interact with the internet. One of the definitions of a ‘digital native’ is comfortable and adept at finding and interpreting answers to questions in an online world.

7) Considering your answers to 5 and 6 above, do you, and programmers in general, have a good feel as to how people use data in today’s world? Why do you say that? There is no wrong answer to this, I just want you to think deeply about how you view data vs how a typically internet user mighty view data.

People have a good understanding of how people are using data in our modern age due to necessity. We believe that programmers in general have a good understanding of how people use data because programmers are naturally digital natives delving into the use of software that collects data to make the applications and services better for the user. It is important to have a good understanding of how people use data to create applications that are user friendly and something a person wants to use.