CSE 15: Discrete Mathematics
Laboratory 5

Fall 2020

- **This lab assignment will be graded and its grade will count towards the course grade**.
- **This lab must be solved individually.** You can discuss your ideas with others, but when you prepare your solution you must work individually. Your submission must be yours and yours only. No exceptions, and be reminded that the CSE academic honesty policy discussed in class will be enforced.
- Your solution must be exclusively submitted via CatCourses. Pay attention to the posted deadline because **the system automatically stops accepting submissions when the deadline passes. Late submissions will receive a 0**. You can upload one or more `.py` files.
- Start early.

Introduction

In the past week we have formally introduced the concept of *function* in class, and we have introduced functions in python. In this lab assignment you will practice with both. First, recall that in class we have defined that a function $f : A \to B$ is relation f over $A \times B$ such that for each $a \in A$, there is exactly one $b \in B$ such that $(a, b) \in f$, and in such case we write $b = f(a)$. To keep things simple, in this assignment we will only consider functions where both A and B are finite sets. Recall that A is the domain of f and B is the codomain of f.

When both A and B are finite, one way to represent $f : A \to B$ is to provide an enumeration of all couples $(a, b) \in A \times B$, such that $b = f(a)$ (this is not necessarily efficient, but it does not matter for the time being.) Owing to the fact that f is a special type of relation, i.e., a subset of $A \times B$, a function can therefore be represented as a set (recall the definition of *graph* of a function.) For example if $A = \{dog, cat, mouse\}$ and $B = \{white, black\}$ the following subset ot $A \times B$ represents (the graph of) a function $f : A \to B$.

$$\{(dog, white), (cat, black), (mouse, black)\}$$

In python this structure can be represented by a set of tuples, where each tuple has two elements. The following two lines would build the set given above and the print it.

```
>>> L = [('dog','white'), ('cat','black'),('mouse','black')]
>>> f = set(L)
>>> print(f)
{('cat', 'black'), ('dog', 'white'), ('mouse', 'black')}
```

In the example, first we store the tuples into a list, and then we create a set with those tuples. There are obviously countless other ways to initialize `f` and get the same result. The following example builds a similar structure, but has a problem:

```
>>> L = L = [('dog','white'), ('cat','black'),('dog','black')]
>>> f = set(L)
>>> print(f)
{('dog', 'black'), ('cat', 'black'), ('dog', 'white')}
```

The problem is that in this case `f` is not the graph of a function because there are two entries where `dog` is the first element, and this is contrary to the definition of a function (every element in the domain must be associated with exactly one element in the codomain.)

With the above assumptions we then represents functions by giving their graph. Write two functions in python that satisfy the following specifications:

1. Write a function

   `is_a_graph(A,B,f)`

   that receives three parameters. The first is the domain, the second is the codomain, and the third is a subset of $A \times B$. The function returns `True` if `f` represents the graph of a function $f : A \to B$ and `False` otherwise.

2. Write a function

   `is_surjective(A,B,f)`

   that receives three parameters. The first is the domain, the second is the codomain, and the third is the graph of a function represented as above. The function returns `True` if `f` is the graph of a surjective function and `False` otherwise. You can assume that in this case `f` is the graph of a function, i.e., you can ignore the case when `f` is not the graph of a function.

Suggestions: to solve this exercise it is necessary to analyze all elements in `f`. To this end, you can re-use the code provided in the file `functionsexamples.py` that is linked from the lab assignment. Therein we give the code for a function `is_injective` that can be modified to answer the questions given above.