



## CSE 15: Discrete Mathematics Laboratory 6

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 this lab we continue to use Python functions to implement some simple concepts related to functions. As in the previous assignment, we will restrict our focus to *finite* domains and codomains. Assuming  $A$  is the domain and  $B$  is the codomain, we continue to use subsets of  $A \times B$  to represent a function  $f : A \rightarrow B$ , owing to the fact that a function is a special type of relation between  $A$  and  $B$ . For example if  $A = \{dog, cat, fish\}$  and  $B = \{1, 2\}$  the following function:

$$f(dog) = 1$$

$$f(cat) = 2$$

$$f(fish) = 1$$

could be represented in python as follows:

```
>>> A = {'dog', 'cat', 'fish'}
>>> B = {1, 2}
>>> f = {('cat', 2), ('dog', 1), ('fish', 1)}
```

Note that the above code introduces a new syntax to create sets where you list elements between `{` and `}`. Last week we instead created sets from a list of elements. Both options are useful. The following code prints out the function by iterating through the set `f`

```
>>> for i in f:
>>>     print(i)
```

Assuming functions are represented using the structure described above, write Python functions to implement the following requirements.

1. Let  $A$  and  $B$  be finite domains and codomains. Two functions  $f : A \rightarrow B$  and  $g : A \rightarrow B$  are said to be *equal* if  $f(a) = g(a)$  for each element  $a \in A$ . Note that this definition extends also to the case when  $A$  and  $B$  are infinite sets. Write a python function called `equal_functions` that takes as input two functions (call them `f` and `g`) and returned `True` if they are equal and `False` otherwise.
2. Recall the definition of *partial function* given in class. Write a function `is_partial_function` that takes as input the domain `A` and a function `f` and returns `True` if the function is partial and `False` otherwise.
3. Recall the definition of composite function. If  $g : A \rightarrow B$  and  $f : B \rightarrow C$ , then the composition of  $f$  and  $g$  is a function  $f \circ g : A \rightarrow C$  defined as  $(f \circ g)(a) = f(g(a))$ . Write a python function `composite_function` that accepts as input two functions `g` and `f` and returns the composite function  $f \circ g$  as per the above definition. The composite function should be represented as a set, using the same format described above. To keep things simple, you can assume that `g` and `f` are *well formed*, i.e., that the composite function exists. However, if you want to challenge yourself, you can consider the more complex case where you have to check whether the composite function is defined. If it is defined, then return the corresponding set, and if it is not defined return the empty set. How could it be that a composite function is not defined? Consider for example the following case:

```
>>>g = {('cat', 2), ('dog', 1), ('fish', 1)}
>>>f = {(1,True),(0,False)}
```

In this case the composite function  $f \circ g$  is not defined because  $g(cat) = 2$ , but  $f(2)$  is not defined.

While solving the above questions, keep in mind that when you iterate over a set in Python, you should not assume a specific order because it is not defined (i.e., there is no first, second, third element, etc.)