Discrete Structures!

CMPSC 102

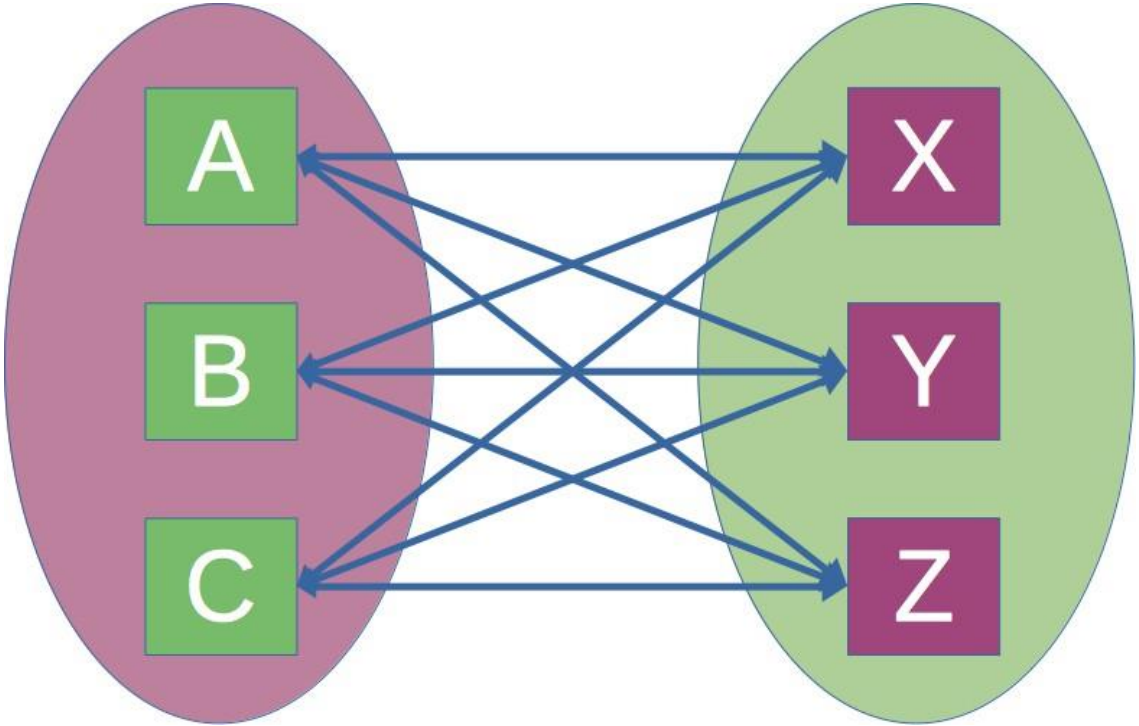
ALLEGHENY COLLEGE

Mathematical Terminology

- Mathematical terminology is a vocabulary for discussing Python programs
  - **Set**: an unordered collection of different entities
  - **Sequence**: an ordered collection of entities
  - **Relation**: a set that relates pairs of things with each other
  - **Mapping**: a set of ordered pairs in every element is unique (sometimes called a "function" in mathematics)
- Can you find these mathematical concepts in the Python programs? For instance: *What is a file*?

Mathematical Terminology

Is this a function?



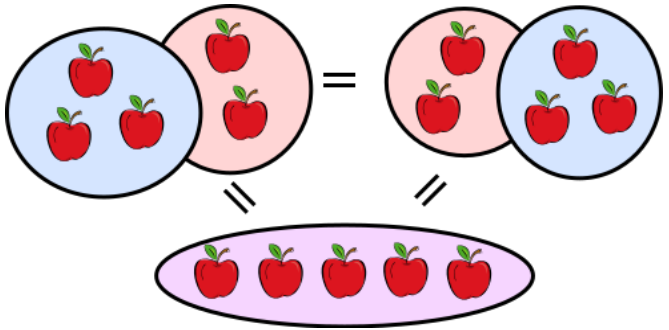
General Properties of Real Numbers

Property	Addition	Multiplication
Commutative	$a + b = b + a$	$a \cdot b = b \cdot a$
Associative	$a + (b + c) = (a + b) + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c$
Distributive	$a \cdot (b + c) = a \cdot b + a \cdot c$	$a \cdot (b + c) = a \cdot b + a \cdot$
Identity	$a + 0 = a$	$a \cdot 1 = a$
Inverse	$a + (-a) = 0$	$a \cdot \frac{1}{a} = 1$

Properties - Commutative

- The term "commutative" is used in several related senses.
- A binary operation $*$ on a set S is called *commutative* if: $x * y = y * x$ for all $x, y \in S$
  - An operation that does not satisfy the above property is called *non-commutative*.

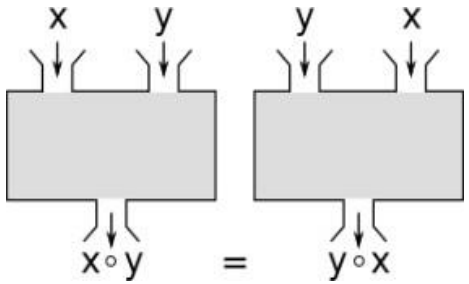
- One says that x *commutes* with y under $*$ if: $x * y = y * x$
- A binary function $f : A \times A \to B$ is called *commutative* if: $f(x, y) = f(y, x)$ for all $x, y \in A$



Properties - Examples

Commutative

- The operator each side of equation do not create inequality
- Think operators like: Addition, multiplication, division



Not Commutative

- The operator each side of equation creates inequality
- Think operators like: subtraction
- $x - y \neq y - x$; $5 - 3 \neq 3 - 5$

Properties - Non-Commutative operations

- Washing and drying clothes resembles a noncommutative operation; washing and then drying produces a markedly different result to drying and then washing.
- Putting on left and then right socks on feet is commutative
- Putting on shirt and then sweater is not-commutative

Strings:

```
a = "face"
b = "book"
a + b == b + a # run the test! "facebook" != "bookface"
print(f"\n The result of the test is: {a + b == b + a}")
```

Connecting Math and Python

- Program **variables** and their associated types exist in both **discrete mathematics** and in **Python programs**.
- Connecting **mathematical variables** to **Python variables**:
  - Python variables have descriptive names like temperature_celcius
  - Python variables can also store character strings like music
  - Python variables have **practical limitations** not faced by mathematical ones! What are they? Why do they exist? Why is it important to know about them?

Practical Variable Limitations in Python

Programming has computational limits

Python Output:

```
>>> 2**2**8 # a really long number
115792089237316195...584007913129639936

>>> 2**2**10 # a very, very long number!!
17976931...6329624224137216

>>> 2**2**100
^CTraceback (most recent call last):
  File "stdin", line 1, in module
KeyboardInterrupt
```

- Mathematical thinking is infinite unlike computational wisdom

Practical Variable Limitations in Python

More computational limits

Python Output:

```
>>> 1.0 == 1.1
False
>>> 1.0 == 1
True
>>> 'h' + 'i' + '!'
'hi!'
>>> .33333 + .33333 + .33333 == 1
False
>>> .33333333333 + .33333333333 + .3333333333 == 1
False
>>> 1/3
0.3333333333333333
>>> 1/3 + 1/3 + 1/3 == 1
True
```

Test Your Understanding

- Understanding the **connections** between **mathematics** and **programming**:
  - **Q1**: What is a **mapping** in the mathematics?
  - **Q2**: What is a **function** in mathematics and Python?
  - **Q3**: What are the **limits** for variables in the Python language?
  - **Q4**: What kinds of computational limits exist in Python? Or for any programming?