[0] The Function (and other mathematical and computational preliminaries)

Set terminology and notation

Set: an unordered collection of objects. **Example:** $\{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$

$\in$**:** indicates that an object belongs to a set (equivalently, that the set *contains* the object). For example, $\heartsuit \in \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$.

$A \subseteq B$**:** Read this as "A is a subset of B". This means A and B are sets, and every element of A is also an element of B.

$A = B$**:** Two sets are equal if they contain exactly the same elements. (There is no order among elements of a set.)

A convenient way to prove that A and B are equal is to prove that each is a subset of the other. The proof often consists of two parts:

1. a proof that $A \subseteq B$, and
2. a proof that $B \subseteq A$.

Set expressions

In Mathese, we would write "the set of nonnegative numbers" like this:

$$\{x \in \mathbb{R} \ : \ x \geq 0\}$$

Read this as "The set of consisting of all elements x of the set of real numbers such that x is greater than or equal to 0"

The colon stands for "such that".

There are two parts to this set expression:

- *the part before the colon:* This part specifies where the elements of the set come from, and introduces a variable or variables that can be used in the second part.
- *the part after the colon:* This gives a rule that restricts which elements specified in the first part actually get to make it into the set.

The analogous Python expression is a *set comprehension*:

```
>>> S = {-4, 4, -3, 3, -2, 2, -1, 1, 0}
>>> {x for x in S if x >= 0}
{0, 1, 2, 3, 4}
```

Set expressions

Instead of

$$\{x \in \mathbb{R} \ : \ x \geq 0\}$$

you might see just

$$\{x \ : \ x \geq 0\}$$

if it is considered clear what kind of values x is supposed to take on.

Another example:

$$\{x \ : \ x^2 - \frac{5}{6}x + \frac{1}{6} = 0\}$$

This time, the set consists of just two numbers, $\frac{1}{2}$ and $\frac{1}{3}$.

Set terminology and notation

Cardinality: If a set S is not infinite, we use $|S|$ to denote the number of elements or *cardinality* of the set.

For example, the set $\{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$ has cardinality 4.

Set terminology and notation: Cartesian product

$A \times B$ is the set of all pairs (a, b) where $a \in A$ and $b \in B$.

Example: for $A = \{1, 2\}$ and $B = \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$, $A \times B$ is

$$\{(1, \heartsuit), (2, \heartsuit), (1, \spadesuit), (2, \spadesuit), (1, \clubsuit), (2, \clubsuit), (1, \diamondsuit), (2, \diamondsuit)\}$$

Named for René Descartes. We will meet him later.

Set terminology and notation: Cartesian product

Question: What is the cardinality of $A \times B$ where $A = \{1, 2\}$ and $B = \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$?

Answer: 8

Set terminology and notation: Cartesian product

If A and B are finite sets then $|A \times B| = |A| \times |B|$.

Question: What is the cardinality of $\{1, 2, 3, \ldots, 10, J, Q, K\} \times \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$?

Answer: 52

Tuples in set expressions

The set expression

$$\{(x, y) \in \mathbb{R} \times \mathbb{R} \ : \ y = x^2\}$$

denotes the set of all pairs of real numbers in which the second element of the pair is the square of the first.

This set expression might be abbreviated as

$$\{(x, y) : \ y = x^2\}$$

where you are supposed to guess from context that x and y range over real numbers.

Another example:

$$\{(x, y, z) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R} \ : \ x \geq 0, y \geq 0, z \geq 0\}$$

This is the set of triples consisting of nonnegative real numbers.

For a triple (x, y, z) to be included in the result, *all* the conditions to the right of the colon must be satisfied. (You can read the comma between the conditions as "and".)

We might abbreviate that set expression as

$$\{(x, y, z) \ : \ x \geq 0, y \geq 0, z \geq 0\}$$

The function

Informally, for each *input* element in a set A, a *function* assigns a single *output* element from another set B.

- A is called the *domain* of the function
- B is called the *co-domain*

Formally, a function is a set of pairs (a, b) no two of which have the same first element.

Example: The function with domain $\{1, 2, 3, \ldots\}$ that doubles its input is the set

$$\{(1, 2), (2, 4), (3, 6), (4, 8), \ldots\}$$

Example: The function with domain $\{1, 2, 3, \ldots\} \times \{1, 2, 3, \ldots\}$ that multiplies the numbers forming its input is

$\{((1, 1), 1), ((1, 2), 2), ((1, 3), 3), \ldots, ((2, 1), 2), ((2, 2), 4), ((2, 3), 6), \ldots$
$((3, 1), 3), ((3, 2), 6), ((3, 3), 9), \ldots$

The function

Definition: The output of a given input is called the *image* of that input. The image of q under a function f is denoted $f(q)$.

If $f(q) = r$, we say *q maps to r under f*. In Mathese, we write this as $q \mapsto r$.

The set from which all the outputs are chosen is called the *co-domain*.

We write

$$f : D \longrightarrow F$$

when we want to say that f is a function with *domain* D and *co-domain* F.

Note: When we define a function, we have some flexibility in the choice of co-domain. There might be elements of the co-domain that are *not* images of any elements of the domain.

The function

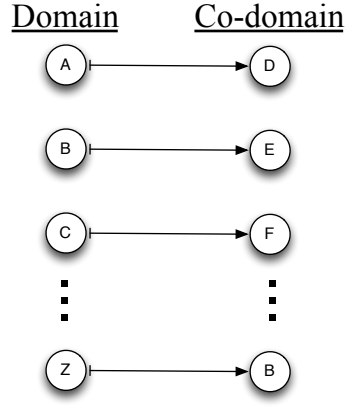
Example: Caesar's Cryptosystem

Each letter is mapped to one three places ahead, wrapping around, so MATRIX would map to PDWULA.

The function mapping letter to letter can be written as:

$$\{('A','D'),('B','E'),('C','F'),('D','G'),\ldots,('W','Z'),$$
$$('X','A'),('Y','B'),('Z','C')\}$$

Both the domain and co-domain are $\{A, B, \ldots, Z\}$.



The function

Definition: The *image* of a function is the set of all images of inputs. Mathese: Im f

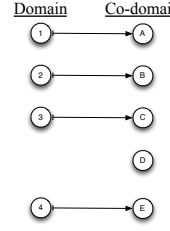
Example: Cosine Function $\cos(x)$

$\cos : \mathbb{R} \longrightarrow \mathbb{R}$, which means the domain is $\mathbb{R}$ and the co-domain is $\mathbb{R}$

The image of $\cos(x)$, Im cos, is $\{x \in \mathbb{R} \ : \ -1 \leq x \leq 1\}$,
which is not the same as the co-domain

Example: The image of the Caesar encryption function is $\{A, B, C, \ldots, Z\}$,
which is the same as the co-domain.

Example: a function $f : \{1, 2, 3, 4\} \longrightarrow \{'A', 'B', 'C', 'D', 'E'\}$



The image of f is Im $f = \{'A', 'B', 'C', 'E'\}$

'D' is in the co-domain but not in the image.

- Some people use "range" to mean co-domain.
- Some people use "range" to mean image.

Because it is used in both ways, I prefer to avoid the word.

The function: Set of functions with given domain and co-domain

Definition: For sets F and D, F^D denotes all functions from D to F.

Example: The set of functions from the set W of words to the set $\mathbb{R}$ of real numbers is $\mathbb{R}^W$.

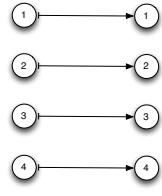
Proposition: For finite sets, $|F^D| = |F|^{|D|}$.

Identity function and composition

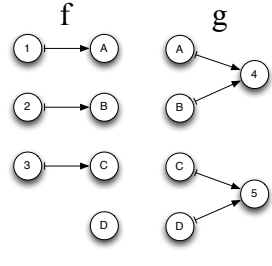
Identity function: for any domain D,

$$id_D : D \longrightarrow D$$

maps each domain element d to itself.



Definition: For functions $f : A \longrightarrow B$ and $g : B \longrightarrow C$, the *functional composition* of f and g is the function $(g \circ f) : A \longrightarrow C$ defined by $(g \circ f)(x) = g(f(x))$



Example: $f : \{1, 2, 3\} \longrightarrow \{A, B, C, D\}$ and $g : \{A, B, C, D\} \longrightarrow \{4, 5\}$

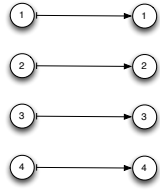
Example: Composition of $g(y) = y^2$ and $f(x) = x + 1$ is $(g \circ f)(x) = (x + 1)^2$.

Identity function and composition

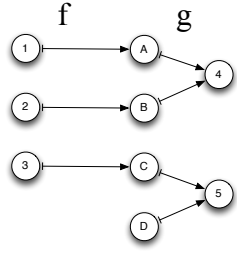
Identity function: for any domain D,

$$id_D : D \longrightarrow D$$

maps each domain element d to itself.



Definition: For functions $f : A \longrightarrow B$ and $g : B \longrightarrow C$, the *functional composition* of f and g is the function $(g \circ f) : A \longrightarrow C$ defined by $(g \circ f)(x) = g(f(x))$



Example: $f : \{1, 2, 3\} \longrightarrow \{A, B, C, D\}$ and $g : \{A, B, C, D\} \longrightarrow \{4, 5\}$

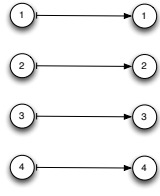
Example: Composition of $g(y) = y^2$ and $f(x) = x + 1$ is $(g \circ f)(x) = (x + 1)^2$.

Identity function and composition

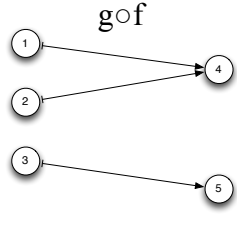
Identity function: for any domain D,

$$id_D : D \longrightarrow D$$

maps each domain element d to itself.



Definition: For functions $f : A \longrightarrow B$ and $g : B \longrightarrow C$, the *functional composition* of f and g is the function $(g \circ f) : A \longrightarrow C$ defined by $(g \circ f)(x) = g(f(x))$



Example: $f : \{1, 2, 3\} \longrightarrow \{A, B, C, D\}$ and $g : \{A, B, C, D\} \longrightarrow \{4, 5\}$

Example: Composition of $g(y) = y^2$ and $f(x) = x + 1$ is $(g \circ f)(x) = (x + 1)^2$.

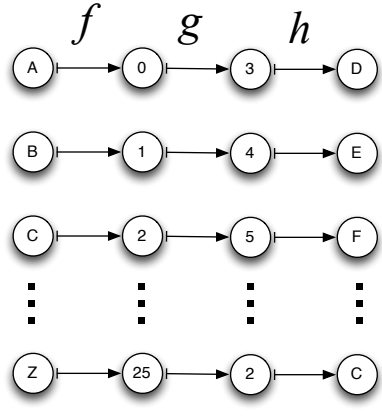
The function: Composition

Example: Define the function $f : \{A, B, C, \ldots, Z\} \longrightarrow \{0, 1, 2, \ldots, 25\}$ by $A \mapsto 0, B \mapsto 1, C \mapsto 2, \cdots, Z \mapsto 25$

Define g on the domain/co-domain $\{0, 1, 2, \ldots, 25\}$ by $g(x) = (x + 3) \bmod 26$

Define h with domain $\{0, 1, 2, \ldots, 25\}$ and co-domain $\{A, \ldots, Z\}$ such that $0 \mapsto A$, $1 \mapsto B$, etc.

Then $h \circ (g \circ f)$ is the Caesar cypher.



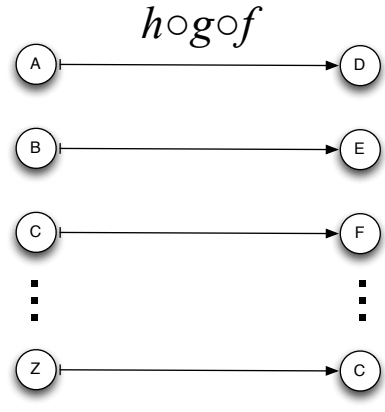
The function: Composition

Example: Define the function $f : \{A, B, C, \ldots, Z\} \longrightarrow \{0, 1, 2, \ldots, 25\}$ by $A \mapsto 0, B \mapsto 1, C \mapsto 2, \cdots, Z \mapsto 25$

Define g on the domain/co-domain $\{0, 1, 2, \ldots, 25\}$ by $g(x) = (x + 3) \bmod 26$

Define h with domain $\{0, 1, 2, \ldots, 25\}$ and co-domain $\{A, \ldots, Z\}$ such that $0 \mapsto A$, $1 \mapsto B$, etc.

Then $h \circ (g \circ f)$ is the Caesar cypher.



The function: Associativity of function composition

Proposition: $h \circ (g \circ f) = (h \circ g) \circ f$

Proof: for any element x of domain of f:

$$\begin{aligned}
(h \circ (g \circ f))(x) &= h((g \circ f)(x)) \text{ by definition of } h \circ (g \circ f)) \\
&= h(g(f(x)) \text{ by definition of } g \circ f \\
&= (h \circ g)(f(x)) \text{ by definition of } h \circ g \\
&= ((h \circ g) \circ f)(x) \text{ by definition of } (h \circ g) \circ f
\end{aligned}$$

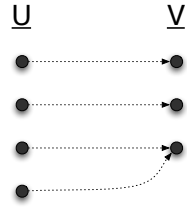
The function: Functional inverse

Definition: Functions f and g are *functional inverses* if $f \circ g$ and $g \circ f$ are defined and are identity functions.

A function that has an inverse is *invertible*.

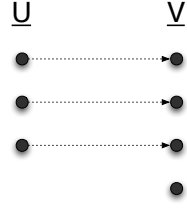
The function

Definition: $f : D \longrightarrow F$ is *one-to-one* if $f(x) = f(y)$ implies $x = y$.



NOT ONE-TO-ONE

Definition: $f : D \longrightarrow F$ is *onto* if for every $z \in F$ there exists an a such that $f(a) = z$.



NOT ONTO

The function

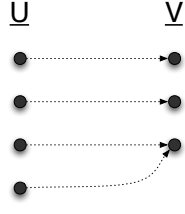
Definition: $f : D \longrightarrow F$ is *one-to-one* if $f(x) = f(y)$ implies $x = y$.

Proposition: Invertible functions are one-to-one.

Proof: Assume an invertible function f is not one-to-one. So there exists $x_1 \neq x_2$ where $f(x_1) = f(x_2) = y$.

Then $f^{-1}(y) = x_1$ but $f^{-1}(y) = x_2$, and both cannot be true, by the definition of function.

QED



NOT ONE-TO-ONE

The function

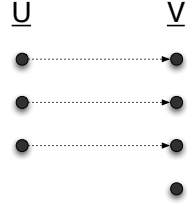
Definition: $f : D \longrightarrow F$ is *onto* if for every $z \in F$ there exists an element $a \in D$ such that $f(a) = z$.

Proposition: Invertible functions are onto

Proof Assume an invertible function f is not onto. So there exists element $\hat{y}$ in co-domain such that for no x does $f(x) = \hat{y}$.

But $f^{-1}(\hat{y}) = \hat{x}$ for some $\hat{x}$, and by the definition of the inverse, $f(\hat{x}) = \hat{y}$, a contradiction.

QED



NOT ONTO

The function

Function Invertibility Theorem:

A function f is invertible if and only if it is one-to-one and onto.

Previous two propositions show that every invertible function is one-to-one and onto.
It is not hard to prove that a function that is one-to-one and onto is invertible.

Procedure

Procedure: a description of a computation that, given an input, produces an output.

Example: `def mul(p,q): return p*q`

Computational Problem: an input-output specification that a procedure might be required to satisfy.

Example: Integer factoring

- *input:* an integer m greater than 1.
- *output:* a pair of integers (p, q) such that $p \times q = m$.

Differences:

- Functions/computational problems don't tell us *how* to compute output from input.
- Many procedures might exist for the same spec.
- A computational problem may allow several possible outputs for each input.
  For integer factoring, for input 12 could have output $(2, 6)$ or $(3, 4)$ or...

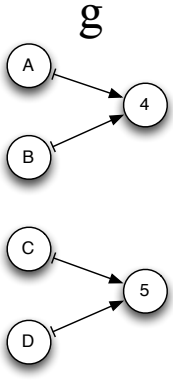
Procedures and functions in Python

We will write procedures in Python, e.g. `def mul(p,q): return p*q`

Often these are called *functions* but we will reserve that term for the mathematical objects.

In Python, we will often use a *dictionary* to represent a function with a finite domain.

The function



can be represented in Python by the dictionary

`{'A':4, 'B':4, 'C':5, 'D':5}`

Probability distribution function

A special kind of function is a *probability distribution function*.
It is used to specify relative likelihood of different outcomes of a single experiment.
It assigns a *probability* (a nonnegative number) to each possible outcome.
The probabilities of all the possible outcomes must sum to 1.

Example:

Probability distribution function for drawing a letter at beginning of the board game Scrabble:

A	9	B	2	C	2	D	4	E	12	F	2	G	3	H	2
I	9	J	1	K	1	L	4	M	2	N	6	O	8	P	2
Q	1	R	6	S	4	T	6	U	4	V	2	W	2	X	1
Y	2	Z	1												

Since the total number of tiles is 98, the probability of drawing an E is 12/98, the probability of drawing an A is 9/98, etc. In Python:

```
{'A':9/98, 'B':2/98, 'C':2/98, 'D':4/98, 'E':12/98, 'F':2/98,
 'G':3/98, 'H':2/98, 'I':9/98, 'J':1/98, 'K':1/98,  'L':1/98,
 'M':2/98, 'N':6/98, 'O':8/98, 'P':2/98, 'Q':1/98,  'R':6/98,
 'S':4/98, 'T':6/98, 'U':4/98, 'V':2/98, 'W':2/98,  'X':1/98,
 'Y':2/98, 'Z':1/98}
```

Probability distribution function: Uniform distributions

Often the probability distribution is a *uniform distribution*. That means it assigns the same probability to each outcome.

To
model rolling a die, the possible outcomes are 1, 2, 3, 4, 5, and 6, and the probabilities are

$$\Pr(1) = \Pr(2) = \Pr(3) = \Pr(4) = \Pr(5) = \Pr(6) = 1/6 \quad .$$

In Python,

```
>>> Pr = {1:1/6, 2:1/6, 3:1/6, 4:1/6, 5:1/6, 6:1/6}
```

To model the flipping of two coins, the possible outcomes are HH, HT, TH, TT and the probability of all outcomes is $1/4$.
In Python,

```
>>> Pr = {('H', 'H'):1/4, ('H', 'T'):1/4,
          ('T', 'H'):1/4, ('T', 'T'):1/4}
```