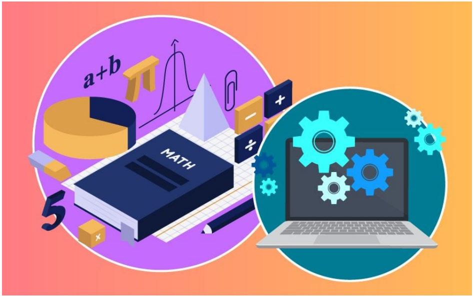
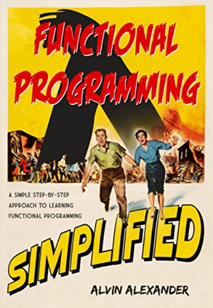




Advanced Math and Functional Programming

Dr. Jim Newton

June 12, 2022

MGS 2022: Creating a Culture of Belonging

applications open November 1 - 30, 2021

Bias, Inclusion, Belonging

Mathematics is unbiased. It is one of the few persuits which is free of politics, nationality, religion, gender, sexuality, age, race, handicap, or of anything else that so often divides us.

Overview

Objectives

Mathematics: Ameliorate your love for Mathematics.

Computer Science: Develop a sense of functional programming.

Communication: Confidently defend your ideas.

Tactic

- Theory — Learn mathematics and computer science concepts.
- Practice — Create code relating to the theory of various units.

Ambitious Syllabus

Week	Day	Unit	Activities
1	Mon	0, 1	Environment Set-up, Hello World
1	Tue	2	Sets and Functions, Theory, First Program
1	Wed	2	Coding Exercises, Presentation
1	Thu	3	Abstract Algebra, Theory
1	Fri	3	Group Project, Coding Exercises
2	Mon	3	Review, Coding Exercises
2	Tue	4,5	Analysis: Convergence, and Sums
2	Wed	4,5	Coding Exercises, Presentation
2	Thu	6,7	Analysis: Derivative, Integral
2	Fri	6,7	Coding Exercises, Presentation

Setup the Environment

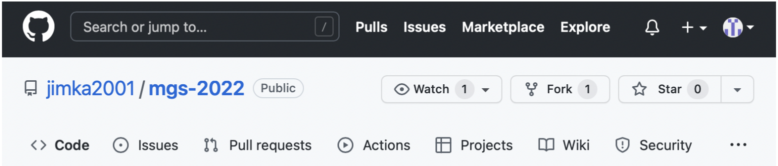
Create GitHub Account

Create a GitHub account using an abstract user name.
Don't use your real name.
Open: `https://github.com/join`

GitHub Project

Open: `https://github.com/jimka2001/mgs-2022`

Fork the Repository



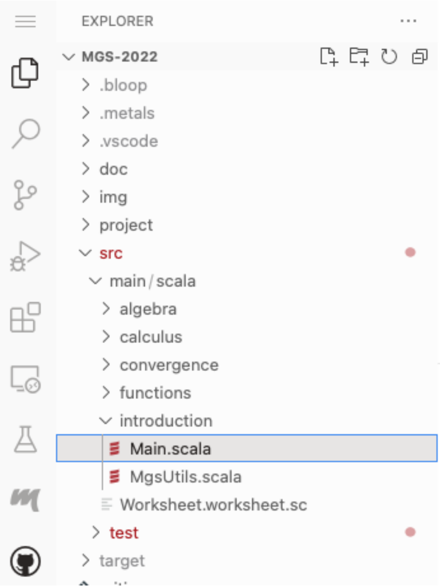
Open the GitPod Workspace

Prepend

```
http://gitpod.io/#
```

to the URL already in the web browser.

Open Main.scala



Ready to edit and run

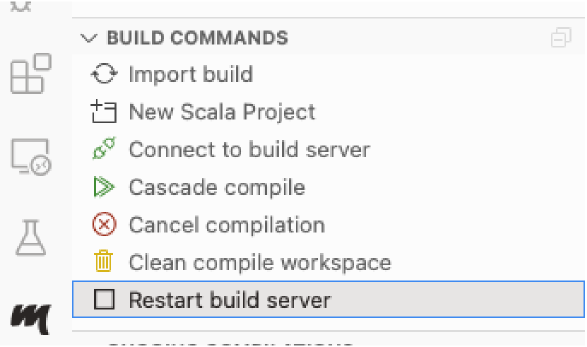
Main.scala ×

src > main > scala > introduction > Main.scala > {} introduction

```scala
package introduction

run | debug
object Main {

  def main(args: Array[String]): Unit = {
    println("hello world")
    println("")
  }
}

```

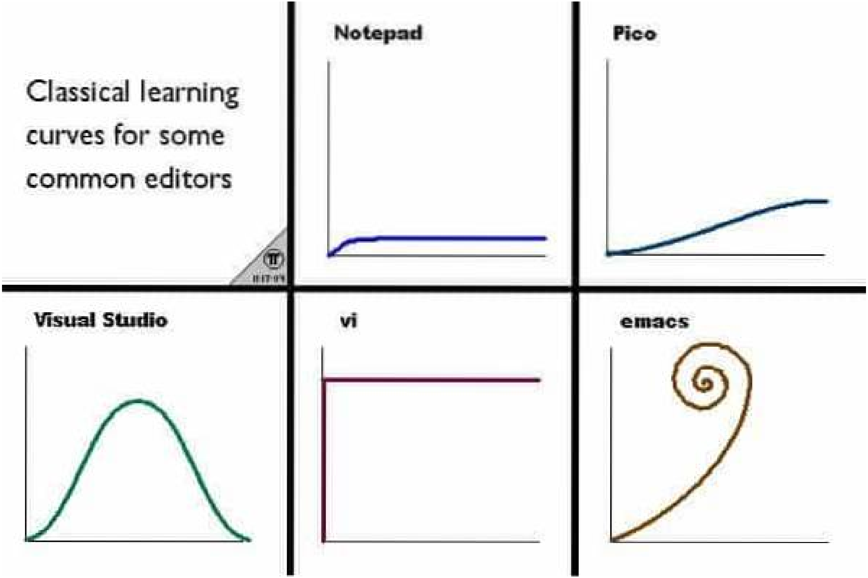
Build Server

You may need to restart the guild-server.



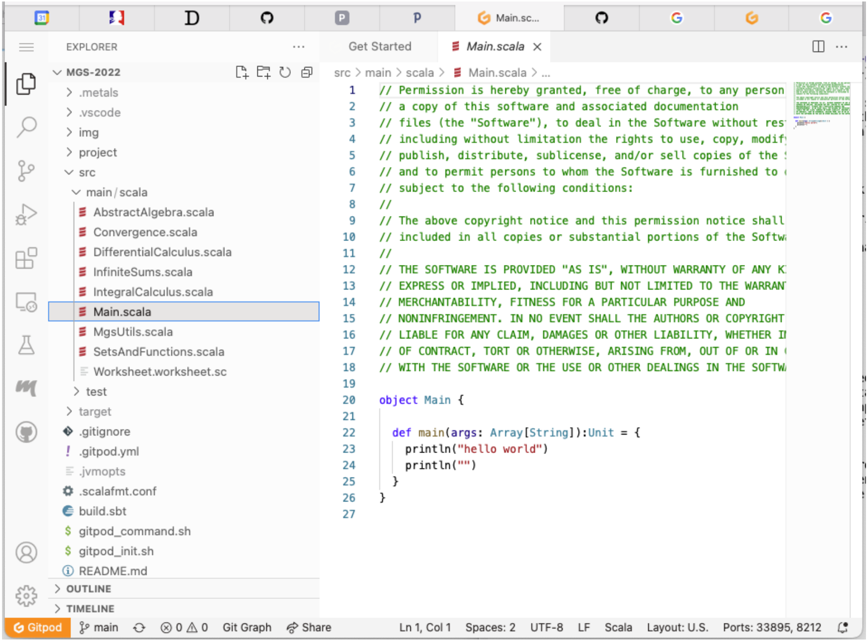
Understanding the Development Flow

Text Files

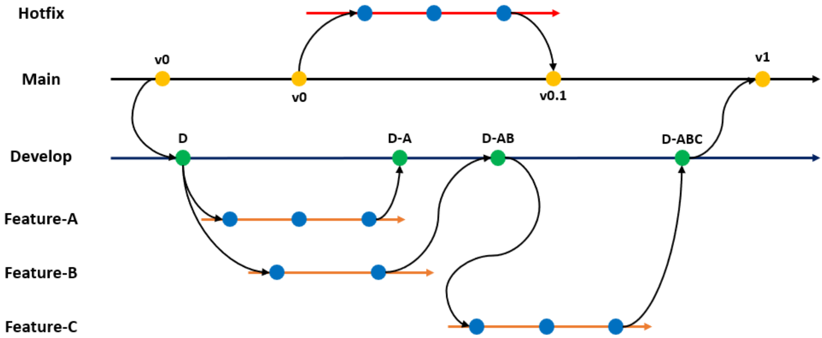


Classical learning curves for some common editors
Notepad
Pico
Visual Studio
vi
emacs

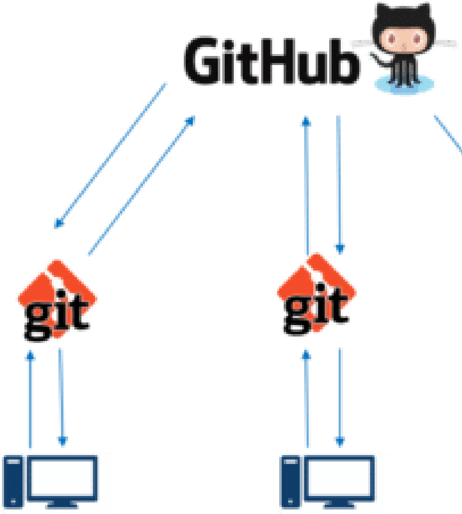
VS Code


Main.sc...
EXPLORER
Get Started
Main.scala
MGS-2022
.metals
.vscode
img
project
src
main / scala
AbstractAlgebra.scala
Convergence.scala
DifferentialCalculus.scala
InfiniteSums.scala
IntegralCalculus.scala
Main.scala
MgsUtils.scala
SetsAndFunctions.scala
Worksheet.worksheet.sc
test
target
.gitignore
.gitpod.yml
.jvmopts
.scalafmt.conf
build.sbt
gitpod_command.sh
gitpod_init.sh
README.md
OUTLINE
TIMELINE
src > main > scala > Main.scala > ...
// Permission is hereby granted, free of charge, to any person
// a copy of this software and associated documentation
// files (the "Software"), to deal in the Software without res
// including without limitation the rights to use, copy, modif
// publish, distribute, sublicense, and/or sell copies of the
// and to permit persons to whom the Software is furnished to
// subject to the following conditions:
//
// The above copyright notice and this permission notice shall
// included in all copies or substantial portions of the Softw
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY K
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRAN
// MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
// NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER I
// OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
// WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTW
object Main {
def main(args: Array[String]):Unit = {
println("hello world")
println("")
}
}
Gitpod
main
0 0
Git Graph
Share
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Scala
Layout: U.S.
Ports: 33895, 8212

Version Control



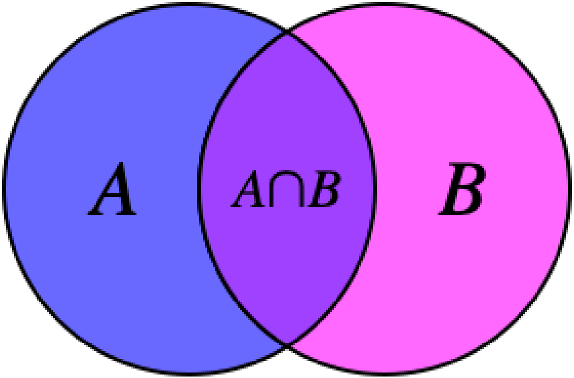
git and GitHub



GitPod

Sets and Functions

What is a set?



What is a set?

We won't really answer this question, because it is *too complicated*.

We will rely on intuition.

For more information see: Zermelo–Fraenkel Set Theory (ZF or ZFC).

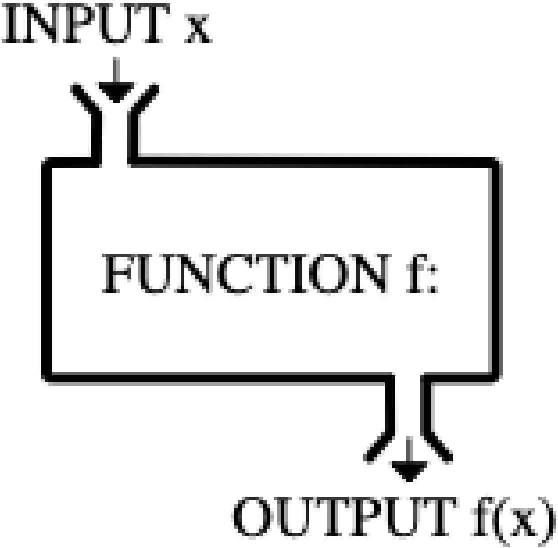
YouTube: Axioms of set Theory – Lecture 2, Frederic Schuller

Some Important Sets

- $\mathbb{N}$ — natural numbers
- $\mathbb{Z}$ — integers
- $\mathbb{Q}$ — rational numbers
- $\mathbb{R}$ — real numbers
- $\mathbb{R}^2$ — ordered pairs of real numbers
- $\mathbb{C}$ — complex numbers

What is a function?

What is a function?



What is a function?

You may already have some intuition about functions.

- A functions may have a name: sin, cos, and log.
- A function may lack a name: $\frac{x^2-1}{x^2+2x+1}$.

What is a function?

Definition (function)

A *function*, f, with domain X and range Y, denoted

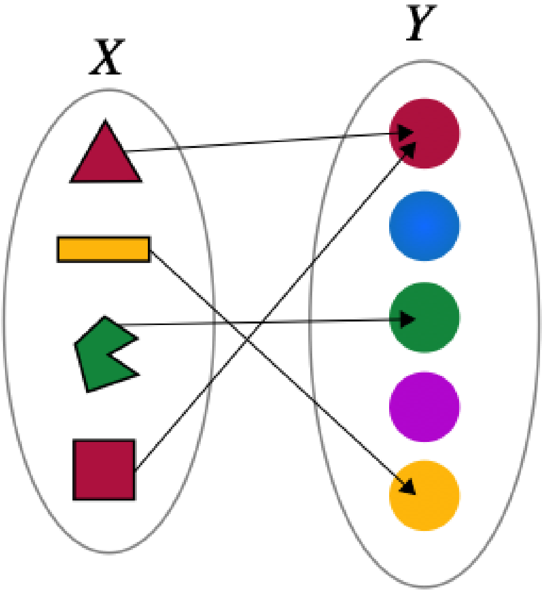
$$f : X \to Y$$

is a correspondence between two sets.

If $x \in X$, then $f(x)$ designates a unique, well-determined, element of Y.

$$x \in X \implies f(x) \in Y\,.$$

Correspondence between sets



Examples of Functions

$$f(x) = 3x + 1$$

$$f : \mathbb{N} \to \mathbb{N} \text{ by } f(x) = 3x + 1$$

Domain and range may be different

$$f : \mathbb{R}^2 \to \mathbb{R} \text{ by } f(x, y) = 3x - 2y + 1$$

Functions defined by cases

$$|x| = \begin{cases} x & ;\text{if } x > 0 \\ 0 & ;\text{if } x = 0 \\ -x & ;\text{if } x < 0 \end{cases}$$

Functions defined by recurrence

$$m^n = \underbrace{m \times m \times \ldots \times m}_{n \text{ times}}$$

$$m^n = \begin{cases} 1 & ; \text{if } n = 0 \\ m \times m^{n-1} & ; \text{if } n > 0 \\ \frac{1}{m^{-n}} & ; \text{if } n < 0 \text{ and } m \neq 0 \end{cases}$$

Functions defined by recurrence

$$n! = 1 \times 2 \times \ldots \times n$$

$$n! = \begin{cases} 1 & ;\text{if } n = 0 \\ n \times (n-1)! & ;\text{if } n > 0 \end{cases}$$

Fibonacci numbers by recurrence

$$F(n) = \begin{cases} 1 & ;\text{if } n = 1 \\ 1 & ;\text{if } n = 2 \\ F(n-1) + F(n-2) & ;\text{if } n > 2 \end{cases}$$

Subsets of size n

$$\mathbb{P}_n(S) = \begin{cases} \{\emptyset\} & ;\text{if } n = 0 \\ \big\{\{x\} \cup y \mid x \in S, y \in \mathbb{P}_{n-1}(S \setminus \{x\})\big\} & ;\text{if } n > 0 \end{cases} \tag{1}$$

Equivalently

$$\mathbb{P}_n(S) = \begin{cases} \{\emptyset\} & ;\text{if } n = 0 \\ \bigcup_{x \in S} \big\{\{x\} \cup y \mid y \in \mathbb{P}_{n-1}(S \setminus \{x\})\big\} & ;\text{if } n > 0 \end{cases} \tag{2}$$

Looping

Why do we need loops?

- to do something (side effect)
- to collect something
- to detect something

Why do we need loops?

- to do something (side effect)
  - `for`
- to collect something
  - `for` + `yield`
- to detect something
  - `exists` and `forall`

Simple loops

Construct simple loops with `for`.

```
for{ x <- List(10,20,30,40)
  } println(x)

for{ x <- Set(10,20,30,40)
  } println( x * x )
```

Concentric loops

We can put loops inside loops, and we can use `List`, `Vector`, `Set` etc.

```
for{ x <- Set(10,20,30,40)
     y <- Vector(5,6,7)
   } println(x + y)
```

Build collections by looping

Add the keyword `yield` to collect results.

```
for{ x <- Set(10,20,30,40)
     y <- Vector(5,6,7)
   } yield x + y
```

Loops with conditionals

Use `if` to filter out certain iterations.

```
for{ x <- Set(10,20,30,40)
     y <- Vector(5,6,7)
     if x + y < 30
   } yield x + y
```

Loops with temporary variables

Use = to capture intermediate values

```
for{ x <- Set(10,20,30,40)
     y <- Vector(5,6,7)
     z = x + y
     if z > 21
   } yield x + y
```

Loops combining multiple things

```
for{ x <- set1
     y <- set2
     if y + x < 100
     z <- set3
     w = fibonacci(x + y)
     if w < factorial(z)
  } println(List(x,y,z,w))
```

Boolean valued loops

Use `exists` to detect whether *at least one* value in a collection meets some condition: matches a *predicate*

```
Set(10,20,30,31,32).exists{
  x => x < 0
}

List(10,20,30,31,32).exists{
  x => x % 2 == 1
}
```

Boolean valued loops

Use `forall` to detect whether *every* value in a collection meets some condition: matches a *predicate*

```
Set(10,20,30,31,32).forall{
  x => x > 0
}

List(10,20,30,31,32).forall{
  x => x % 2 == 0
}
```

Looping between bounds

Use `to` create a collection of integers in a `Range`.

```
(0 to 10).forall{
  x => factorial(x) < 100
}

(1 to 100 by 2).exists{
  x => x * x == 64
}

for{ n <- 1000 to 1 by -3
  if n % 2 == 0
} print(n)

for{ n <- 2 to 1000
  if n % 2 == 0
} yield n
```

Concentric Boolean loops

```
(1 to 100).exists{
  a => List(1,2,3,4).forall{
    b => a*a + b*b > 100
  }
}
```

Challenging exercises

1. Collect all prime numbers between n and m.
2. Collect all Pythagorean triples between 1 and n.
3. Find (print or collect) all solutions to $a^3 + b^3 + c^3 = 1$ for a, b, c in range of $-n$ to n.
4. Taxi cab numbers: For a given n, find numbers between $-n$ and n which are the sum of two cubes in two different ways. E.g., $1729 = 12^3 + 1^3 = 9^3 + 10^3$.
5. Linear Diophantine Equations: Given integers a, b, c, and n, find all integer solutions ($|x| < n$ and $|y| < n$) to the equation $ax + by = c$. *E.g.*, $2x + 4y = 28$ has $x = 12$, $y = 1$ as solution but also $x = 2$, $y = 6$.

Instructions

1. Choose one challenge:
2. Work as individual or team.
3. Create a new file in the `functions` directory.
4. Try to solve the challenge using `for`, `yield`, `exists`, and `forall`.
5. (Optional) If possible, can you make it faster? *E.g.*, decrease the search space.
6. When finished and working, submit a *pull request*
7. Show, explain, and defend your solution to your fellow scholars.

Abstract Algebra

Finding roots of polynomial

If

$$ax^2 + bx + c = 0\,,\, a \neq 0$$

then

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\,.$$

What about higher order polynomials?

Given the roots, we can *easily* find the coefficients.

Simply multiply

$$(x - r_1)(x - r_2)\dots(x - r_n)$$

to arrive at

$$x^n + a_1 x^{n-1} + a_2 x^{n-2} \dots + a_{n-2} x^2 + a_{n-1} x^1 + a_n$$

However, given the coefficients, it is extremely difficult to find the roots.

Example quartic polynomial

$$(x - p)(x - q)(x - r)(x - s) =$$

$$\begin{aligned} x^4 &- (p + q + r + s)x^3 \\ &+ (pq + pr + ps + qr + qs + rs)x^2 \\ &- (pqr + pqs + prs + qrs)x \\ &+ pqrs \\ &= x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \end{aligned}$$

Example quintic polynomial

$$(x-p)(x-q)(x-r)(x-s)(x-t) =$$

$$\begin{aligned}
x^5 &- (p+q+r+s+t)x^4 \\
&+ (pq+pr+ps+pt+qr+qs+qt+rs+rt+st)x^3 \\
&- (rst+qst+qrt+qrs+pst+prt+prs+pqt+pqs+pqr)x^2 \\
&+ (pqrs+pqrt+pqst+prst+qrst)x^2 \\
&- pqrst
\end{aligned}$$

$$= x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

Subsets of size n

A polynomial of degree n has n roots $\{r_1, r_2, r_3, \ldots, r_n\}$. Thus it has n coefficients, $\{a_n, a_{n-1}, \ldots, a_1, a_0\}$.

To compute coefficient, a_k, we must find all the subsets of size k, multiply each subset together, and add up the products.

Recall this recursive function?

$$\mathbb{P}_n(S) = \begin{cases} \{\emptyset\} & ;\text{if } n = 0 \\ \big\{\{x\} \cup y \mid x \in S, y \in \mathbb{P}_{n-1}(S \setminus \{x\})\big\} & ;\text{if } n > 0 \end{cases}$$

Counting subsets

How many subsets of size k are there from a subset of size n?

Counting subsets

How many subsets of size k are there from a subset of size n?

$$\binom{n}{k} = \frac{n!}{k! \times (n-k)!}$$

Summary

Given the roots of a polynomial of degree 5 or greater, it is *easy* to compute the coefficient. However, given the coefficients, it is *difficult* to compute the roots.

Exceptions

Even if sometimes finding the roots is *difficult*, sometimes it is *easy*.
Consider this 5th degree polynomial.

$$x^5 - 1$$

An obvious root is 1, because $1^5 - 1 = 0$. Voilà!
Therefore, we can factor out $x - 1$ to achieve:

$$x^5 - 1 = (x - 1)(x^4 + x^3 + x^2 + x + 1)$$

But we have a formula (albeit unwieldy) to find the roots of a quartic (degree-4).

Story of Evariste Galois

Questions about the video

1. Which nationality was Evariste Galois?
2. What were Galois's important contributions?
3. Why was Galois misunderstood?
4. Why did he *think* he was begin constantly rejected by the best schools?
5. What are some of the disappointments he faced?
6. Which of the events were beyond his control? And which were his own fault?
7. How did he die?
8. Before his death he wrote three letters? What were their contents?

Vocabulary

- Napoleon
- Monarchist
- Republican
- Democratic
- Maths
- Continued fractions
- Apocryphal
- Alexander Dumas
- Victor Hugo

Story of Evariste Galois

Story of Evariste Galois

Monoid

Definition (Monoid)

$(S, \circ)$ is called a *monoid* if

1. Closure: $a, b \in S \implies a \circ b \in S$.
2. Associative: $a, b, c \in S \implies (a \circ b) \circ c = a \circ (b \circ c)$
3. Identity: $\exists e \in S$ such that $a \in S \implies a \circ e = e \circ a = a$

Examples of monoid

- $(\mathbb{N}, +)$, the set of natural numbers under addition is a monoid.
- $(\mathbb{Z}, -)$, the set of integers under subtraction is NOT a monoid. Why?
- The set of even integers under addition is a monoid.
- The set of even integers under multiplication is a NOT monoid. Why?
- $(\mathbb{R}, \times)$, the set of positive real numbers under multiplication is a monoid.

Examples of monoid

- The set of 2×3 matrices under addition is a monoid.
- The set of 2×3 matrices under multiplication is not a monoid. Why?
- However the set of 3×3 matrices under multiplication is a monoid.

Examples of monoid

- The set of subsets of a given set using the operation of union is a monoid. What is the identity element?
- The set of subsets of a given set using the operation of intersection is a monoid. What is the identity element?

Examples of monoid

- The set polynomials with integer coefficients, $\mathbb{Z}[x]$, using the operation of multiplication.
- What about $(\mathbb{R}[x], \times)$
- $\ldots (\mathbb{R}[x], +)$?
- $\ldots (\mathbb{Q}[x], +)$?
- What about polynomials with 3×3 matrices of reals as coefficients?

Examples of monoid

Let S be the set of functions with $\mathbb{R}$ as domain and range.
If $f, g \in S$, let $f \circ g$ be defined as follows:
$x \in \mathbb{R} \implies (f \circ g)(x) = f(g(x))$.

- Is S a monoid under this definition of $\circ$.
- What if we change $\mathbb{R}$ to $\mathbb{Z}$?
- What if we change $\mathbb{R}$ to $\mathbb{Q}[x]$?

The free monoid

Let $\Sigma = \{a, b, c, d\}$.
The set, $\mathcal{L}(\Sigma)$, of all sequences of finite length $(x_1, x_2, \ldots, x_n)$ for which $x_i \in \Sigma$ for $i = 1, 2, \ldots, n$, $(n \geq 0)$.
Let $+$ denote sequence concatenation. *E.g.*,

$$(a, c, a, a) + (d, a, c, a, b) = (a, c, a, a, d, a, c, a, b)\,.$$

$\mathcal{L}(\Sigma)$ is a monoid.
What is its identity element?
Is it a commutative monoid?

Clock addition

The integers, $\{1, 2, 3, ..., 12\}$ form a additive monoid.

$+$	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12	1
2	3	4	5	6	7	8	9	10	11	12	1	2
3	4	5	6	7	8	9	10	11	12	1	2	3
4	5	6	7	8	9	10	11	12	1	2	3	4
5	6	7	8	9	10	11	12	1	2	3	4	5
6	7	8	9	10	11	12	1	2	3	4	5	6
7	8	9	10	11	12	1	2	3	4	5	6	7
8	9	10	11	12	1	2	3	4	5	6	7	8
9	10	11	12	1	2	3	4	5	6	7	8	9
10	11	12	1	2	3	4	5	6	7	8	9	10
11	12	1	2	3	4	5	6	7	8	9	10	11
12	1	2	3	4	5	6	7	8	9	10	11	12

Clock multiplication

The integers, $\{1, 2, 3, ..., 12\}$ form a multiplicative monoid.

*	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	2	4	6	8	10	12
3	3	6	9	12	3	6	9	12	3	6	9	12
4	4	8	12	4	8	12	4	8	12	4	8	12
5	5	10	3	8	1	6	11	4	9	2	7	12
6	6	12	6	12	6	12	6	12	6	12	6	12
7	7	2	9	4	11	6	1	8	3	10	5	12
8	8	4	12	8	4	12	8	4	12	8	4	12
9	9	6	3	12	9	6	3	12	9	6	3	12
10	10	8	6	4	2	12	10	8	6	4	2	12
11	11	10	9	8	7	6	5	4	3	2	1	12
12	12	12	12	12	12	12	12	12	12	12	12	12

Examples of power() function for monoid

```
def power(b: Int, n: Int): Int = {
  if (n == 0)
    1
  else
    b * power(b, n - 1)
}

def power(b: String, n: Int): Int = {
  if (n == 0)
    ???
  else
    ???
}
```

(Slow) Power (exponentiation)

In which algebraic structures can we use these equations?

$$x^n = \begin{cases} e & ;\text{if } n = 0 \\ x \circ x^{n-1} & ;\text{if } n > 0 \end{cases} \tag{3}$$

How many applications of $\circ$ are needed for this algorithm?

The `power` function for monoid

For a monoid, M, with operation $\circ$, the previous algorithm, *slow power*, computes x^n, for $x \in M$, by $n-1$ applications of $\circ$.

Question: Can we do better?

What is the minimum number of applications necessary to compute x^n?

Fast Power (exponentiation)

Which algebraic structures can we use these equations?

$$x^n = \begin{cases} e & ;\text{if } n = 0 \\ x & ;\text{if } n = 1 \\ x \circ x^{n-1} & ;\text{if } n \text{ is odd} \\ (x^{\frac{n}{2}}) \circ (x^{\frac{n}{2}}) & ;\text{if } n \text{ is even} \end{cases} \tag{4}$$

Question: Is the case for $n = 1$ necessary?

What is a group?

Definition (Group)

$(S, \circ)$ is called a *group* if

1. $(S, \circ)$ is a monoid.
2. Inverse: $\forall a \in S \; \exists a^{-1} \in S$ such that $a \circ a^{-1} = a^{-1} \circ a = e$

If $a \circ b = b \circ a$ for all $a, b \in S$, then we call S an *Abelian* group.

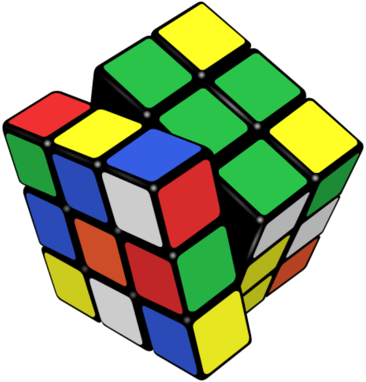
Examples of group

- The set of integers, $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$, is a group under integer addition. Why?
  - $(\mathbb{Z}, +)$ is a monoid with 0 being the identity.
  - If $a \in \mathbb{Z}$ there exists $b \in \mathbb{Z}$ such that $a + b = 0$. *E.g.*, $12 + (-12) = 0$
- The integers under multiplication is not a group. Why?

Examples of group

Is the set of rotations of the Rubik's cube a group?

If so, what is the identity, and what are the inverses?



Examples of group

Is the set of 3×3 matrices of real numbers is a group.

Why? or Why not?

Examples of group

- Is the set of subsets of a given set, G, using the operation of union a group. Why? Why not?
- The set subsets of a given set, G, using

$$A \circ B = (A \cap \overline{B}) \cup (\overline{A} \cap B)$$

  as the operation, is a group.
  - Every element is its own inverse.
  - The identity element is the empty set, $\emptyset$.

Clock multiplication

The 11-clock

*	1	2	3	4	5	6	7	8	9	10	11
1	1	2	3	4	5	6	7	8	9	10	11
2	2	4	6	8	10	1	3	5	7	9	11
3	3	6	9	1	4	7	10	2	5	8	11
4	4	8	1	5	9	2	6	10	3	7	11
5	5	10	4	9	3	8	2	7	1	6	11
6	6	1	7	2	8	3	9	4	10	5	11
7	7	3	10	6	2	9	5	1	8	4	11
8	8	5	2	10	7	4	1	9	6	3	11
9	9	7	5	3	1	10	8	6	4	2	11
10	10	9	8	7	6	5	4	3	2	1	11
11	11	11	11	11	11	11	11	11	11	11	11

Examples of group

The non-11 elements of the 11-clock

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

What is a ring?

Definition (Ring)

$(S, +, \times)$ is called a *ring* if

1. $(S, +)$ is an Abelian group.
2. $(S, \times)$ is a monoid.
3. Distributive: $a, b, c \in S \implies a \times (b + c) = (a \times b) + (a \times c)$.

Examples of ring

- The integers $(\mathbb{N}, +, \times)$ is not a ring. Why?
- The integers $(\mathbb{Z}, +, \times)$ is a ring.
- The integers $(\mathbb{Q}, +, \times)$ is a ring.
- The integers $(\mathbb{R}, +, \times)$ is a ring.
- The integers $(\mathbb{C}, +, \times)$ is a ring.
- The set of $n \times n$ matrices, for a fixed value of $n > 0$ is a ring.
- The set of 2×3 matrices is not a ring. Why?

Examples of ring

- $\mathbb{Z}[x]$, polynomials with integer coefficients such as $3x^4 + 2x^2 - 5x + 1$?
- The *normalized* subset of $\mathbb{Q}[x]$ with leading coefficient equal to 1?
  - What are the additive inverses?
- Let S be a ring, the consider the set of polynomials with coefficients in S.

What is a field?

Definition (Field)

$(F, +, \times)$ is called a *field* if

1. $(F, +, \times)$ is an Abelian Ring.
2. $(F \setminus 0, \times)$ is a (Abelian) group, where 0 is the identity under $+$.

Examples of field

1. The rational numbers, $\mathbb{Q}$?
2. The set of $n \times n$ matrices?
3. The set of 2×2 matrices of the form $\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$ where $a, b \in \mathbb{Q}$?
4. The complex numbers
5. The integers modulo 12?

Examples of field

The integers modulo any prime such as 7?

$+$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$\times$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Review of algebraic structures

1. A *monoid* is a set where we can add.
2. A *group* is a set where we can add and subtract.
3. A *ring* is a set where we can add, subtract, and multiply.
4. A *field* is a set where we can add, subtract, multiply, and divide.

Convergence

Infinite Sums