I217E: Functional Programming

Nao Hirokawa

JAIST

Term 2-1, 2022

http://www.jaist.ac.jp/~hirokawa/lectures/fp/

Contents of This Course

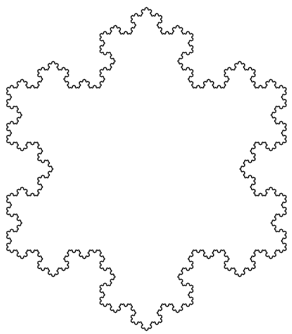
Schedule

10/12	introduction	11/9	interpreters
10/14	algebraic data types I	11/11	compilers
10/19	algebraic data types II	11/16	termination
10/21	program reasoning	11/18	confluence
10/26	applications	11/25	verification
10/28	data structures I	11/30	review
11/2	data structures II	12/5	exam
11/4	computational models		

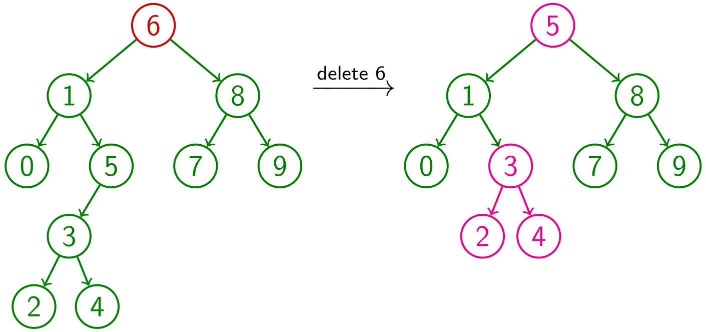
Evaluation

exam (60) + reports (40)

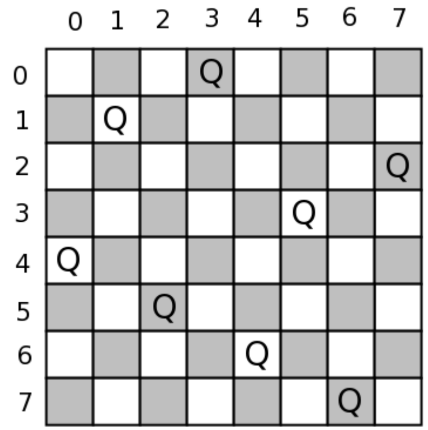
Drawing Fractals



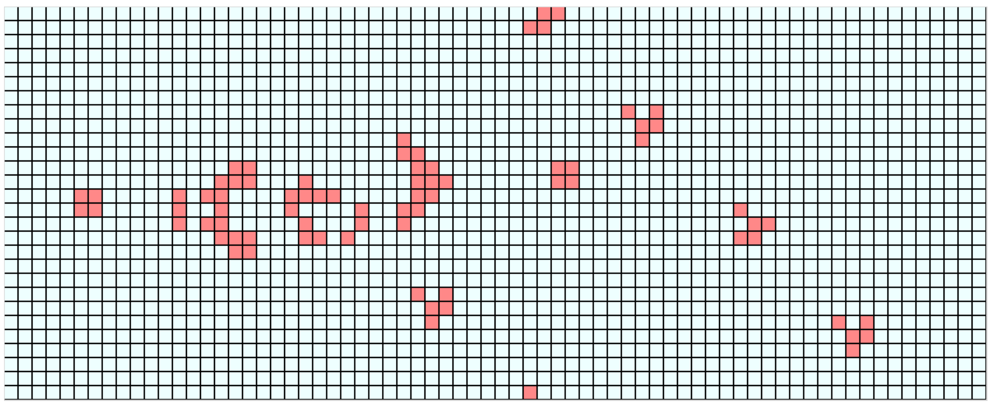
Data Structures and Algorithms



N-Queen Problem Solver



Game of Life (Report Assignment in 2021)



Mini-Haskell Interpreter

sample input:

```
main = qsort [5,1,4,2,3]

qsort []       = []
qsort (x : xs) =
  qsort (filter (<= x) xs) ++ [x] ++
  qsort (filter (>  x) xs)
                :
```

output:

```
[1,2,3,4,5]
```

Final Exam (Closed Book)

I217E: Functional Programming — Exam 2021

Programs should be written in Haskell. Do not use any extra module except `Data.List`.

[10] **Q1.** Implement the function `f` :: $[a] \to [[a]]$ given by the equation:

$$\texttt{f}\ [x_1, x_2, \ldots, x_n] = \left[\, [x_k, x_{k+1}, \ldots, x_m] \mid 1 \leqslant k \leqslant m \leqslant n \,\right]$$

For instance, we have:

$$\texttt{f}\ [1,2,3] = \left[\, [1], [1,2], [1,2,3], [2], [2,3], [3] \,\right]$$
$$\texttt{f}\ \texttt{"ab"} = [\texttt{"a"}, \texttt{"ab"}, \texttt{"b"}]$$

Note that the order of sublists is unimportant.

⋮

Goal of This Course

Goal

to become familiar with

- **functional programming**
- **program reasoning**
- **computational model** (programming language theory)

Ultimate Goal

to understand

math is **not your enemy**

Let's Begin Functional Programming

Functions and Types

Mathematical Definition

$$\texttt{squareSum} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$
$$\texttt{squareSum}(x, y) = x^2 + y^2$$

$$\texttt{identity} : A \to A \qquad (A : \text{arbitrary set})$$
$$\texttt{identity}(x) = x$$

Definition in Haskell

```
squareSum :: Int -> Int -> Int
squareSum x y = x * x + y * y

identity :: a -> a
identity x = x
```

Recursion

Mathematical Definition

$$\texttt{sum}(n) = 0 + 1 + 2 \underbrace{+ \cdots +}_{\text{not expressible}} n$$

$$\begin{aligned}
\texttt{sum}(0) &= 0 \\
\texttt{sum}(1) &= 0 + 1 &&= \texttt{sum}(0) + 1 \\
\texttt{sum}(2) &= 0 + 1 + 2 &&= \texttt{sum}(1) + 2 \\
\texttt{sum}(3) &= 0 + 1 + 2 + 3 &&= \texttt{sum}(2) + 3
\end{aligned}$$

Recursive Definition

$$\texttt{sum}(n) = \begin{cases} 0 & \text{if } n = 0 \\ \texttt{sum}(n-1) + n & \text{otherwise} \end{cases}$$

Sum

$$\texttt{sum} : \mathbb{N} \to \mathbb{N}$$

$$\texttt{sum}(n) = \begin{cases} 0 & \text{if } n = 0 \\ \texttt{sum}(n-1) + n & \text{otherwise} \end{cases}$$

```
sum1 :: Int -> Int
sum1 n =
  if n == 0 then 0 else n + sum1 (n-1)

sum2 :: Int -> Int
sum2 n | n == 0    = 0
       | otherwise = n + sum2 (n - 1)

sum3 :: Int -> Int
sum3 0 = 0
sum3 n = n + sum3 (n - 1)
```


Exercise: Implement!

$$\texttt{factorial} : \mathbb{N} \to \mathbb{N}$$
$$\texttt{factorial}(n) = n!$$

$$\texttt{fib} : \mathbb{N} \to \mathbb{N}$$

$$\texttt{fib}(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ \texttt{fib}(n-1) + \texttt{fib}(n-2) & \text{otherwise} \end{cases}$$

Lists

Definition

- **lists** are of form $x_1 : (x_2 : \cdots (x_n : [\,]))$ where $x_1, \ldots, x_n$ are of same type
- abbreviated to $[x_1, x_2, \cdots, x_n]$

Example

1. **OK:** $[\,]$ **empty list**
2. **OK:** $1 : (2 : (3 : [\,]))$, $1 : 2 : 3 : [\,]$, $[1, 2, 3]$
3. **OK:** $[\texttt{"abc"}, \texttt{"def"}]$, $[\text{True}, \text{False}]$
4. **OK:** $[[1, 2], [3], [\,]]$ **nested list**
5. **NG:** $1 : (2 : 3)$
6. **NG:** $[1, \texttt{"abc"}]$ heterogeneous list

Length (same as `length`)

$\texttt{myLength}\ [x_1, \ldots, x_n] = n$

$$\begin{aligned}
\texttt{myLength}\ (\qquad\qquad\qquad\qquad [\,]) &= 0\\
\texttt{myLength}\ (\qquad\qquad \texttt{"a"} : [\,]) &= 1\\
\texttt{myLength}\ (\qquad \texttt{"b"} : \texttt{"a"} : [\,]) &= 2\\
\texttt{myLength}\ (\texttt{"c"} : \texttt{"b"} : \texttt{"a"} : [\,]) &= 3
\end{aligned}$$

```
myLength :: [a] -> Int
myLength []       = ...
myLength (x : xs) = ...
```


Append (same as ++)

$\texttt{append}\ [x_1, \ldots, x_m]\ [y_1, \ldots, y_n] = [x_1, \ldots, x_m, y_1, \ldots, y_n]$

$$\begin{aligned}
\texttt{append}\ \qquad\qquad\quad [\,]\ \ (4:5:[\,]) &= \qquad\qquad 4:5:[\,]\\
\texttt{append}\ (\qquad\quad 3:[\,])\ (4:5:[\,]) &= \qquad\quad 3:4:5:[\,]\\
\texttt{append}\ (\quad 2:3:[\,])\ (4:5:[\,]) &= \quad 2:3:4:5:[\,]\\
\texttt{append}\ (1:2:3:[\,])\ (4:5:[\,]) &= 1:2:3:4:5:[\,]
\end{aligned}$$

```
append :: [a] -> [a] -> [a]
append []       ys = ...
append (x : xs) ys = ...
```


Exercise: Implement!

$\texttt{sumList} : [\mathsf{Int}] \to \mathsf{Int}$

$\texttt{sumList}\ [x_1, x_2, \ldots, x_n] = x_1 + x_2 + \cdots + x_n$

$\texttt{evens} : [a] \to [a]$

$\texttt{evens}\ [x_0, x_1, x_2, \ldots, x_n] = [x_0, x_2, x_4, \ldots, x_k]$

where $k = \lfloor n/2 \rfloor$

for example,

$$\texttt{sumList}\ [10, 20, 30] = 60$$

$$\texttt{evens}\ [0, 10, 20, 30, 40] = [0, 20, 40]$$

$$\texttt{evens}\ [0, 10, 20, 30, 40, 50] = [0, 20, 40]$$

Homework 1/3

Remark

- do homework exercises; they are not part of evaluation but important
- solutions are discussed during next lecture

0. Set up a proper programming environment on your PC
1. Implement `myGcd` : Int → Int → Int that corresponds to:

$$\texttt{myGcd} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$\texttt{myGcd}(x, y) = \begin{cases} x & \text{if } y = 0 \\ \texttt{myGcd}(y, x) & \text{if } y > x \\ \texttt{myGcd}(x - y, y) & \text{otherwise} \end{cases}$$

For example, $\texttt{myGcd}(12, 32) = 4$.

Homework 2/3

$\boxed{2}$ Implement `range` : Int → Int → [Int] given by

$$\texttt{range}\ m\ n = [m, m+1, m+2, \ldots, n]$$

For instance,

$$\texttt{range}\ 10\ 15 = [10, 11, 12, 13, 14, 15]$$
$$\texttt{range}\ 10\ 9 = [\,]$$

$\boxed{3}$ Implement `insert` : Int → [Int] → [Int] that inserts an integer into a sorted list over integers:

$$\texttt{insert}\ 5\ [2, 2, 4, 6] = [2, 2, 4, 5, 6]$$
$$\texttt{insert}\ 7\ [2, 2, 4, 6] = [2, 2, 4, 6, 7]$$

Homework 3/3

$\boxed{4}$ Implement the **insertion sort**

$$\texttt{isort} : [\mathrm{Int}] \to [\mathrm{Int}]$$

whose behavior is illustrated by the following calculation:

$$\begin{aligned}
& \texttt{isort}\ [5, 2, 3, 2] \\
= {} & \texttt{insert}\ 5\ (\texttt{insert}\ 2\ (\texttt{insert}\ 3\ (\texttt{insert}\ 2\ [\,]))) \\
= {} & \texttt{insert}\ 5\ (\texttt{insert}\ 2\ (\texttt{insert}\ 3\ ([2]))) \\
= {} & \texttt{insert}\ 5\ (\texttt{insert}\ 2\ [2, 3]) \\
= {} & \texttt{insert}\ 5\ [2, 2, 3] \\
= {} & [2, 2, 3, 5]
\end{aligned}$$