

# I217E: Functional Programming

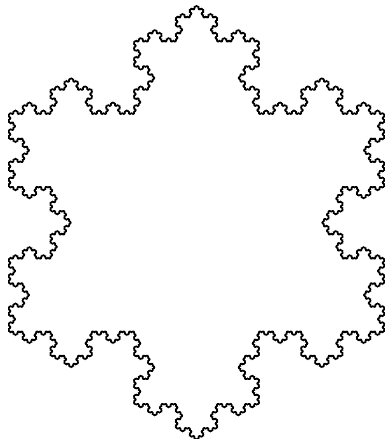
Nao Hirokawa

JAIST

Term 2-1, 2022

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

## Exercise I: Draw Koch's Snowflake



## 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	applications	11/18	confluence
10/26	program reasoning	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)

## Step 1: Turtle Graphics

given angle  $\delta \in \mathbb{R}$

$\text{eval } (x, y, \alpha) [] = [(x, y)]$   
 $\text{eval } (x, y, \alpha) ('+' : cs) = \text{eval } (x, y, \alpha + \delta) cs$   
 $\text{eval } (x, y, \alpha) ('-' : cs) = \text{eval } (x, y, \alpha - \delta) cs$   
 $\text{eval } (x, y, \alpha) (c : cs) = (x, y) : \text{eval } (x', y', \alpha) cs$

where  $x' = x + \cos \alpha$  and  $y' = y + \sin \alpha$

## Exercise

- depict  $\text{eval } (0, 0, 0) \text{ "F+F--F+F"}$ , assuming  $\delta = \frac{\pi}{3}$
- implement  $\text{eval}$

## Step 2: String Rewriting

1 `rewrite` replaces all occurrences of F by F+F--F+F

```
rewrite "F--F--F" =  
  "F+F--F+F--F+F--F+F--F+F--F+F"
```

2 `power`  $f\ n\ x = f^n(x)$

3 `snowflake` = `power rewrite 6 "F--F--F"`

## Exercise II: Generate HTML Table

```
records = [  
  (2000, [7.6, 6.0, 9.4, ..., 8.8]),  
  (2001, [4.9, 6.6, 9.8, ..., 8.4]),  
  (2002, [7.4, 7.9, 12.2, ..., 7.2]),  
  (2003, [5.5, 6.4, 8.7, ..., 9.2]),  
  (2004, [6.3, 8.5, 9.8, ..., 9.9]),  
  (2005, [6.1, 6.2, 9.0, ..., 6.4]),  
  (2006, [5.1, 6.7, 9.8, ..., 9.5]),  
  (2007, [7.6, 8.6, 10.8, ..., 9.0]),  
  (2008, [5.9, 5.5, 10.7, ..., 9.8]),  
  (2009, [6.8, 7.8, 10.0, ..., 9.0]),  
  (2010, [7.0, 6.5, 9.1, ..., 9.9]),  
  (2011, [5.1, 7.0, 8.1, ..., 7.5]),  
  (2012, [4.8, 5.4, 8.8, ..., 7.3]),  
  (2013, [5.5, 6.2, 12.1, ..., 8.3]),  
  (2014, [6.3, 5.9, 10.4, ..., 6.7]),  
  (2015, [5.8, 5.7, 10.3, ..., 9.3]),  
  (2016, [6.1, 7.2, 10.1, ..., 8.9]),  
  (2017, [5.8, 6.9, 8.5, ..., 6.6]) ]
```

Temperatures in Tokyo

-	1	2	3	4	5	6	7	8	9	10	11	12
2000	7.6	6.0	9.4	14.5	19.8	22.5	27.7	28.3	25.6	18.8	13.3	8.8
2001	4.9	6.6	9.8	15.7	19.5	23.1	28.5	26.4	23.2	18.7	13.1	8.4
2002	7.4	7.9	12.2	16.1	18.4	21.6	28.0	28.0	23.1	19.0	11.6	7.2
2003	5.5	6.4	8.7	15.1	18.8	23.2	22.8	26.0	24.2	17.8	14.4	9.2
2004	6.3	8.5	9.8	16.4	19.6	23.7	28.5	27.2	25.1	17.5	15.6	9.9
2005	6.1	6.2	9.0	15.1	17.7	23.2	25.6	28.1	24.7	19.2	13.3	6.4
2006	5.1	6.7	9.8	13.6	19.0	22.5	25.6	27.5	23.5	19.5	14.4	9.5
2007	7.6	8.6	10.8	13.7	19.8	23.2	24.4	29.0	25.2	19.0	13.3	9.0
2008	5.9	5.5	10.7	14.7	18.5	21.3	27.0	26.8	24.4	19.4	13.1	9.8
2009	6.8	7.8	10.0	15.7	20.1	22.5	26.3	26.6	23.0	19.0	13.5	9.0
2010	7.0	6.5	9.1	12.4	19.0	23.6	28.0	29.6	25.1	18.9	13.5	9.9
2011	5.1	7.0	8.1	14.5	18.5	22.8	27.3	27.5	25.1	19.5	14.9	7.5
2012	4.8	5.4	8.8	14.5	19.6	21.4	26.4	29.1	26.2	19.4	12.7	7.3
2013	5.5	6.2	12.1	15.2	19.8	22.9	27.3	29.2	25.2	19.8	13.5	8.3
2014	6.3	5.9	10.4	15.0	20.3	23.4	26.8	27.7	23.2	19.1	14.2	6.7
2015	5.8	5.7	10.3	14.5	21.1	22.1	26.2	26.7	22.6	18.4	13.9	9.3
2016	6.1	7.2	10.1	15.4	20.2	22.4	25.4	27.1	24.1	18.7	11.4	8.9
2017	5.8	6.9	8.5	14.7	20.0	22.0	27.3	26.4	22.8	16.8	11.9	6.6

source of data is Japan Meteorological agency:

[http://www.data.jma.go.jp/obd/stats/etrn/view/monthly\\_s3.php?prec\\_no=44&block\\_no=47662](http://www.data.jma.go.jp/obd/stats/etrn/view/monthly_s3.php?prec_no=44&block_no=47662)

## Step 1: Generate Table

File: table1.html

```
<!DOCTYPE html>  
  
<h1>Temperatures in Tokyo</h1>  
<table>  
  <tr>  
    <th> -  
    <th> 1  
    <th> 2  
    <th> 3  
  </tr>  
  <tr>  
    <td> 2000  
    <td> 9.1  
    <td> 14.2  
    <td> 19.3  
  </tr>  
  <tr>  
    <td> 2001  
    <td> 18.4  
    <td> 23.5  
    <td> 27.6  
  </tr>  
</table>
```

Browser: a.html

Temperatures in Tokyo

-	1	2	3
2000	9.1	14.2	19.3
2001	18.4	23.5	27.6

## Step 2: Generate Colored Table

File: table2.html

```
<!DOCTYPE html>  
  
<style>  
table {  
  border-top: 1px solid;  
  border-bottom: 1px solid;  
}  
.cold { background: #aaf; } /* -10 */  
.cool { background: #ccf; } /* 10-15 */  
.mild { background: #fef; } /* 15-20 */  
.warm { background: #fcc; } /* 20-25 */  
.hot { background: #faa; } /* 25- */  
</style>  
  
<h1>Temperatures in Tokyo</h1>  
<table>  
  <tr>  
    <th> -  
    <th> 1  
    <th> 2  
    <th> 3  
  </tr>  
  <tr>  
    <td> 2000  
    <td class="cold"> 9.1  
    <td class="cool"> 14.2  
    <td class="mild"> 19.3  
  </tr>  
  <tr>  
    <td> 2001  
    <td class="mild"> 18.4  
    <td class="warm"> 23.5  
    <td class="hot"> 27.6  
  </tr>  
</table>
```

## Step 3: Replace Dummy Data

File: Temperature.hs

```
...
records = [
  (2000, [7.6, 6.0, 9.4, ..., 8.8]),
  (2001, [4.9, 6.6, 9.8, ..., 8.4]),
  ...,
  (2017, [5.8, 6.9, 8.5, ..., 6.6]) ]

-- Dummy data
{-
records = [
  (2000, [9.1,14.2,19.3]),
  (2001, [18.4,23.5,27.6]) ]
-}
```

I217E: Functional Programming

9/16

Browser: a.html

### Temperatures in Tokyo

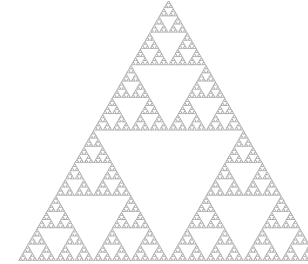
-	1	2	3	4	5	6	7	8	9	10	11	12
2000	7.6	6.0	9.4	14.5	19.8	22.5	27.7	28.3	25.6	18.8	13.3	8.8
2001	4.9	6.6	9.8	15.7	19.5	23.1	28.5	26.4	23.2	18.7	13.1	8.4
2002	7.4	7.9	12.2	16.1	18.4	21.6	28.0	28.0	23.1	19.0	11.6	7.2
2003	5.5	6.4	8.7	15.1	18.8	23.2	22.8	26.0	24.2	17.8	14.4	9.2
2004	6.3	8.5	9.8	16.4	19.6	23.7	28.5	27.2	25.1	17.5	15.6	9.9
2005	6.1	6.2	9.0	15.1	17.7	23.2	25.6	28.1	24.7	19.2	13.3	6.4
2006	5.1	6.7	9.8	13.6	19.0	22.5	25.6	27.5	23.5	19.5	14.4	9.5
2007	7.6	8.6	10.8	13.7	19.8	23.2	24.4	29.0	25.2	19.0	13.3	9.0
2008	5.9	5.5	10.7	14.7	18.5	21.3	27.0	26.8	24.4	19.4	13.1	9.8
2009	6.8	7.8	10.0	15.7	20.1	22.5	26.3	26.6	23.0	19.0	13.5	9.0
2010	7.0	6.5	9.1	12.4	19.0	23.6	28.0	29.6	25.1	18.9	13.5	9.9
2011	5.1	7.0	8.1	14.5	18.5	22.8	27.3	27.5	25.1	19.5	14.9	7.5
2012	4.8	5.4	8.8	14.5	19.6	21.4	26.4	29.1	26.2	19.4	12.7	7.3
2013	5.5	6.2	12.1	15.2	19.8	22.9	27.3	29.2	25.2	19.8	13.5	8.3
2014	6.3	5.9	10.4	15.0	20.3	23.4	26.8	27.7	23.2	19.1	14.2	6.7
2015	5.8	5.7	10.3	14.5	21.1	22.1	26.2	26.7	22.6	18.4	13.9	9.3
2016	6.1	7.2	10.1	15.4	20.2	22.4	25.4	27.1	24.4	18.7	11.4	8.9
2017	5.8	6.9	8.5	14.7	20.0	22.0	27.3	26.4	22.8	16.8	11.9	6.6

## Homework

- 1 Extend `rewrite` so that it takes rewriting rules. For instance,
 

```
rules = [( 'A', "B-A-B"), ( 'B', "A+B+A")]
```

```
rewrite rules "B-A-B" = "A+B+A-B-A-B-A+B+A"
```
- 2 Draw Sierpinski's triangle:



See: <https://en.wikipedia.org/wiki/L-system>

I217E: Functional Programming

10/16

## Report Assignment I

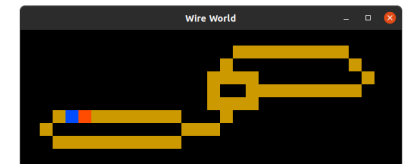
I217E: Functional Programming

11/16

## Report Assignment I: Wire World

xor.txt

```
.....
.....#####.
.....#.....#.
.....####. ....#.
.....#.....#####.
.....####. ....
..#ht#####. ....#
..#.....###.....
..#####. ....
.....
```



### Notation

'.' empty      '#' conductor      'h' head      't' tail

I217E: Functional Programming

12/16

[illegible]

## Use Haskell and Gloss to implement the Wire World.



- submit by email
  - Subject: report1
  - To: hirokawa@jaist.ac.jp
  - attaching **only one** file: *WyourStudentID.hs*.
- code must start from information
  - name: **your full name**
  - id: **your student ID**
  - acknowledgements: **name** if anybody has assisted you
- deadline: **Nov 7 (Mon) 12:00 JST**

- The program name must be *WyourStudentID.hs*, like *W2110999.hs*.
- The program must take a text file of an initial configuration.
- On my environment your program will be executed by:

```
ghc Wxxxxxxxxx.hs
./Wxxxxxxxxx sample1.txt
./Wxxxxxxxxx sample2.txt
```

- 20 points if the program runs correctly (at least 1 frame per second)
- 0 points, otherwise

- deadline is strict (no deadline extension)
- submission style is strict, and do not submit any additional file
- use GHC 9.\*.\* (or 8.\*.\*) and Gloss
- do not import any module except System.Environment and Graphics.Gloss
- do not change program specification
- do not ask me to test your code
- **no plagiarism**  
investigate by yourself what is regarded as **plagiarism**