

CS314 Problem Set 3

Lambda calculus Make all parentheses explicit in the following lambda-expressions

- a. $\lambda x.xz \lambda y.xy$
- b. $(\lambda x.xz) \lambda y.w \lambda w.wyzx$
- c. $\lambda x.xy \lambda x.yx$

Find all free (unbound) variables in the following lambda-expressions

- d. $\lambda x.x z \lambda y.x y$
- e. $(\lambda x.x z) \lambda y.w \lambda w.w y z x$
- f. $\lambda x.x y \lambda x.y x$

0.

```
C = [39.2, 36.5, 37.3, 38, 37.8]
F = list(map(lambda x: (float(9)/5)*x + 32, C))
>>> print(F)
C = list(map(lambda x: (float(5)/9)*(x-32), F))
>>> print(C)
```

1.

```
a = [1, 2, 3, 4]
b = [17, 12, 11, 10]
c = [-1, -4, 5, 9]
>>> list(map(lambda x, y : x+y, a, b))

>>> list(map(lambda x, y, z : x+y+z, a, b, c))

>>> list(map(lambda x, y, z : 2.5*x + 2*y - z, a, b, c))
```

2.

```
a = [1, 2, 3]
b = [17, 12, 11, 10]
c = [-1, -4, 5, 9]
>>> list(map(lambda x, y, z : 2.5*x + 2*y - z, a, b, c))
```

4.

```
fibonacci = [0,1,1,2,3,5,8,13,21,34,55]
odd_numbers = list(filter(lambda x: x % 2, fibonacci))
>>> print(odd_numbers)
even_numbers = list(filter(lambda x: x % 2 == 0, fibonacci))
>>> print(even_numbers)
```

5.

```
import functools
>>> functools.reduce(lambda x,y: x+y, [47,11,42,13])
```

6.

```
from functools import reduce
f = lambda a,b: a if (a > b) else b
>>> reduce(f, [47,11,42,102,13])
```

7.

```
from functools import reduce
>>> reduce(lambda x, y: x+y, range(1,101))
```

8.

```
orders = [ ["34587", "Learning Python, Mark Lutz", 4, 40.95],
            ["98762", "Programming Python, Mark Lutz", 5, 56.80],
            ["77226", "Head First Python, Paul Barry", 3, 32.95],
            ["88112", "Einführung in Python3, Bernd Klein", 3, 24.99]]

min_order = 100
invoice_totals = list(map(lambda x: x if x[1] >= min_order else (x[0], x[1]
+ 10),
                           map(lambda x: (x[0],x[2] * x[3]), orders)))

print(invoice_totals)
```

Solution

```
1. >>> print(F)
[102.56, 97.7, 99.14, 100.4, 100.03999999999999]
>>> C = list(map(lambda x: (float(5)/9)*(x-32), F))
>>> print(C)
[39.2, 36.5, 37.300000000000004, 38.000000000000001, 37.8]

2. >>> list(map(lambda x, y : x+y, a, b))
[18, 14, 14, 14]
>>> list(map(lambda x, y, z : x+y+z, a, b, c))
[17, 10, 19, 23]
>>> list(map(lambda x, y, z : 2.5*x + 2*y - z, a, b, c))
[37.5, 33.0, 24.5, 21.0]

3. [37.5, 33.0, 24.5]

4. [1, 1, 3, 5, 13, 21, 55]
   [0, 2, 8, 34]

5. 113

6. 102

7. 5050

8. [('34587', 163.8), ('98762', 284.0), ('77226', 108.85000000000001),
    ('88112', 84.97)]
```

https://www.python-course.eu/python3_lambda.php

a. $\lambda x.xz \lambda y.xy$	$\rightarrow (\lambda x.((x\ z)\ (\lambda y.(x\ y))))$
b. $(\lambda x.xz) \lambda y.w \lambda w.wyzx$	$\rightarrow (((\lambda x.(x\ z))\ (\lambda y.(w\ (\lambda w.(((w\ y)\ z)\ x))))))$
c. $\lambda x.xy \lambda x.yx$	$\rightarrow (\lambda x.((x\ y)\ (\lambda x.(y\ x))))$
d. $\lambda x.x\ z \lambda y.x\ y$	$\rightarrow (\lambda x.((x\ \underline{z})\ (\lambda y.(x\ y))))$
e. $(\lambda x.x\ z) \lambda y.w \lambda w.w\ y\ z\ x$	$\rightarrow (((\lambda x.(x\ \underline{z}))\ (\lambda y.(\underline{w}\ (\lambda w.(((w\ y)\ \underline{z})\ \underline{x}))))))$
f. $\lambda x.x\ y \lambda x.y\ x$	$\rightarrow (\lambda x.((x\ \underline{y})\ (\lambda x.(\underline{y}\ x))))$