

CECS 424
Assignment 5
Total: 40 Points

General Instruction

- I recommend that you type your answers to exercise questions by using a word processor (Microsoft word, LibreOffice writer, L^AT_EX, etc.).
 - Submit a PDF file (not a zip file) via BeachBoard (Not email or in class).
-

1. (10 points) Evaluate the following λ expressions:

- (a) (2 points) $((\lambda x. \lambda y. (y\ x)\ \lambda p. \lambda q. p)\ \lambda i. i)$
- (b) (2 points) $((((\lambda x. \lambda y. \lambda z. ((x\ y)\ z)\ \lambda f. \lambda a. (f\ a))\ \lambda i. i)\ \lambda j. j)$
- (c) (2 points) $(\lambda h. ((\lambda a. \lambda f. (f\ a)\ h)\ h)\ \lambda f. (f\ f))$
- (d) (2 points) $((\lambda p. \lambda q. (p\ q)\ (\lambda x. x\ \lambda a. \lambda b. a))\ \lambda k. k)$
- (e) (2 points) $((((\lambda f. \lambda g. \lambda x. (f\ (g\ x))\ \lambda s. (s\ s))\ \lambda a. \lambda b. b)\ \lambda x. \lambda y. x)$

2. (5 points) Define a function:

```
def make_triplet = ...
```

which is like `make_pair` but constructs a triplet from a sequence of three arguments so that any one of the arguments may be selected by the subsequent application of a triplet to a selector function.

Define selector functions:

```
def triplet_first = ...  
def triplet_second = ...  
def triplet_third = ...
```

which will select the first, second or third item from a triplet respectively.

```
make_triplet <item1> <item2> <item3> triplet_first => ... => <item1>  
make_triplet <item1> <item2> <item3> triplet_second => ... => <item2>  
make_triplet <item1> <item2> <item3> triplet_third => ... => <item3>
```

for the arbitrary arguments: `<item1> <item2> <item3>`

3. (10 points) Use α conversion to ensure unique names in the expressions in each of the following λ expressions:

- (a) (2 points) $\lambda x. \lambda y. (\lambda x. y\ \lambda y. x)$

- (b) (2 points) $\lambda x. (x \ (\lambda y. (\lambda x. x \ y) \ x))$
- (c) (2 points) $\lambda a. (\lambda b. a \ \lambda b. (\lambda a. a \ b))$
- (d) (2 points) $(\lambda free. bound \ \lambda bound. (\lambda free. free \ bound))$
- (e) (2 points) $\lambda p. \lambda q. (\lambda r. (p \ (\lambda q. (\lambda p. (r \ q)))) \ (q \ p))$
4. (5 points) The boolean operation **implication** is defined by the following truth table:

X	Y	X IMPLIES Y
F	F	T
F	T	T
T	F	F
T	T	T

Define a λ calculus representation for **implication**:

`def implies = $\lambda x. \lambda y. \dots$`

5. (5 points) The boolean operation **equivalence** is defined by the following truth table:

X	Y	X EQUIV Y
F	F	T
F	T	F
T	F	F
T	T	T

Define a λ calculus representation for **equivalence**:

`def equiv = $\lambda x. \lambda y. \dots$`

6. (5 points) Write a function that finds the product of the numbers between **n** and **one**:

`rec prod n = \dots`

in λ calculus is equivalent to:

$n * n-1 * n-2 * \dots * 1$

in normal arithmetic. Assume the function **isone n** is defined.