**Лабораторная работа № 1 (Олейник Е.И.)**

**Задание 1.**

Две абстракции называются альфа-эквивалентными, если результаты их бета-редукции на одинаковых значениях аргументов совпадают (в качестве аргументов возьмём 42 и 422):

1. λxy. xz

(λxy. xz)42 = (λx. (λy. xz))42

x := 4 => (λy. 4z)2

y:= 2 => **4z**

1. λxz. xz

(λxz. xz)42 = (λx. (λz. xz))42

x := 4 => (λz. 4z)2

z := 2 => 42 => не эквивалентны

1. λmn. mz

(λmn. mz)42 = (λm. (λn. mz))42

m := 4 => (λn. 4z)2

n := 2 => 4z => **эквивалентны**

1. λz(λx). xz

(λz(λx). xz)42

z := 4 => (λx. x4)2

x := 2 => 24 => не эквивалентны

2. λxy. xxy

(λx. (λy. xxy))42

x := 4 => (λy. 44y)2

y := 2 =>  **442**

1. λmn. mnp

(λm. (λn. mnp))42

m := 4 => (λn. 4np)2

n := 2 => 42p => не эквивалентны

1. λx(λy). xy

(λx(λy).xy)42

x := 4 => (λy. 4y)2

y := 2 => 42 => не эквивалентны

1. λa(λb). aab

(λa(λb). aab)42

a := 4 => (λb. 44b)2

b := 2 => 442 => **эквивалентны**

1. λ xyz. zx

( λx. λy. λz. zx)422

x := 4 => (λy. λz. z4)22

y := 2 => (λz. z4)2

z := 2 => **24**

a) λx. (λy). (λz) (нет аргумента для функции)

b) λtos. st

(λt. λo. λs. st)422

t := 4 => (λo. λs. s4)22

o := 2 => (λs. s4)2

s := 2 => **24** => **эквивалентны**

c) λmnp. mn

(λm. λn. λp. mn)422

m := 4 => (λn. λp. 4n)22

n := 2 => (λp.42)2

p := 2 => 42=> не эквивалентны

**Задание 2**

Комбинатором называется абстракция, в которой отсутствуют свободные переменные (переменные, не встречающиеся в голове).

1. λx. xxx => комбинатор (все переменные связные)

2. λxy. zx => не комбинатор (есть свободная переменная z)

3. λxyz. xy(zx) => комбинатор (все переменные связные)

4. λxyz. xy(zxy) => комбинатор (все переменные связные)

5. λxy. xy(zxy) => не комбинатор (есть свободная переменная z)

**Задание 3**

1. λx. xxx - бета-редукцию применить нельзя, так как выражение уже в бета-нормальной форме.

2. (λz. zz)(λy. yy) - процесс расходится, так как z ≔ λy. yy => (λy. yy) (λy. yy) и далее бета-редукцию применить нельзя.

3. (λx. xxx)z - процесс сходится, так как (λx. xxx)z => x ≔ z => zzz .

**Задание 4**

1. (λabc.cba)zz(λwv.w)

a := z => (λbc.cbz)z(λwv.w)

b := z => (λc.czz)(λwv.w)

c := (λwv.w) => (λwv.w)zz

w := z => (λv.z)z

v := z => z

1. (λx.λy.xyy)(λa.a)b

x := (λa.a)b => λy.(λa.a)byy

y := b => (λa.a)bb

a := b => bb

1. (λy.y)(λx.xx)(λz.zq)

y := (λz.zq) => (λz.zq) (λx.xx)

z := (λx.xx) => (λx.xx)q  
x := q => qq

1. (λz.z)(λz.zz)(λz.zy) ⬄ (λm.m)(λn.nn)(λz.zy)

m := (λn.nn) => (λn.nn)(λz.zy)

n := (λz.zy) => (λz.zy)(λz.zy)

z := (λz.zy) => (λz.zy)y

z :=y => yy

1. (λx.λy.xyy)(λy.y)y ⬄ (λx.λy.xyy)(λa.a)y

x := (λa.a) => (λy.(λa.a)yy)y = (λa.a)yy

a := y => yy

1. (λa.aa)(λb.ba)c

a := (λb.ba) => (λb.ba)(λb.ba)c

b := (λb.ba) => (λb.ba)ac

b := a => aac

1. (λxyz.xz(yz))(λx.z)(λx.a) ⬄ (λxyz.xz(yz))(λm.n)(λp.a)

x := (λm.n) => (λyz.(λm.n)z(yz))(λp.a)

y := (λp.a) => λz.(λm.n)z((λp.a)z)

p := z => λz.(λm.n)za

m := z => λz.na