# Questions

1. For each term *A* of the terms below, give its translation *ω*(*A*) from M to Λ showing all the steps, their number and underlining all the parts you are working on, just like we did in the above example:
   1. (*λxz.xz*). (1)

*ω*(*λxz.xz*) =0

ω(λxz.xz) = 3

λω[X](λz.xz) = 3

λλω[Z,X](xz) = 2

λλω[Z,X](x)ω[Z,X](z) = 1

λλ2ω[Z,X](z) = 1

λλ21

* 1. (*λxy.xy*). (1)

*ω*(*λxy.xy*) =0

*ω*(*λxy.xy*) =3

*λω[X]*(*λy.xy*) =3

*λλω[Y,X]*(*xy*) =2

*λλω[Y,X]*(*x*)*ω[Y,X]*(*y*) =1

*λλ2ω[Y,X]*(*y*) =1

*λλ21*

* 1. *xz*(*λxy.xy*). (1)

*ω*(*xz*(*λxy.xy*)) =0

*ω[x,y,z]*(*xz*(*λxy.xy*)) =3

*λω[x,x,y,z]*(*xz*(*λy.xy*)) =3

*λλω[y,x,x,y,z]*(*xz*(*xy*)) =2

*λλω[y,x,x,y,z]*(*x*) *ω[y,x,x,y,z]*(*x z*(*xy*) =2

*λλ2ω[y,x,x,y,z]*(*z*) *ω[y,x,x,y,z]*(*xy*) =2

*λλ25ω[y,x,x,y,z]*(*x*) *ω[y,x,x,y,z]*(*y*) =1

*λλ252ω[y,x,x,y,z]*(*y*) =1

*λλ2521*

* 1. (*λxy.xy*)*xz*. (1)

*ω*((*λxy.xy*)*xz*) =0

*ω[x,y,z]*((*λxy.xy*)*xz*) =3

*λω[x,x,y,z]*((*λy.xy*)*xz*) =3

*λλω[y,x,x,y,z]*((*xy*)*xz*) =2

*λλω[y,x,x,y,z]*(*xy*)*ω[y,x,x,y,z]*(*xz*) =2

*λλω[y,x,x,y,z]*(*x*)*ω[y,x,x,y,z]*(*y*)*ω[y,x,x,y,z]*(*xz*) =2

*λλ2ω[y,x,x,y,z]*(*y*)*ω[y,x,x,y,z]*(*xz*) =2

*λλ21ω[y,x,x,y,z]*(*x*)*ω[y,x,x,y,z]*(*z*) =1

*λλ212ω[y,x,x,y,z]*(*z*) =1

*λλ2125*

1. Give a translation function *f* from M to M that will translate terms in M to terms in M so for example:

*f* ((*λx.x*)*y*) = *y* [*x*]*x*

*f* ((*λx.*(*λy.xy*)*z*)(*λz.z*)) = [*z*]*z* [*x*] *z* [*y*] *y x*. (1)

*f* (*v*) =v

*f* (*λv.A*) =[v]A|

*f* (*AB*) = <B|> A|

1. Use your translation function *f* of Question 2, to translate all the terms in Question 1 above into terms of M . That is, give *f* (*λxz.xz*) and *f* (*λxy.xy*) and *f* (*xz*(*λxy.xy*)) and *f* ((*λxy.xy*)*xz*) showing all the steps. (2)

* *f* (*λxz.xz*) = *..* = *..* = *........*
* *f* (*λxy.xy*) = *..* = *..* = *........*
* *f* (*xz*(*λxy.xy*)) = *..* = *..* = *........*
* *f* ((*λxy.xy*)*xz*) = *..* = *..* = *........*

1. For each of BEXP and IEXP write a printing function printBEXP (respectively printIEXP) that prints its elements nicely just like we wrote printLEXP which prints nicely the elements of LEXP. (2)

(\*Prints a term in item lambda calculus\*) fun printIEXP (IID v) =

fun printIEXP (IID v) =

    print v

  | printIEXP (ILAM (v,e)) =

    (print "[";

     print v;

     print "]";

     printIEXP e)

  | printIEXP (IAPP(e1,e2)) =

    (print "<";

     printIEXP e2;

     print ">";

     printIEXP e1);

(\*Prints a term in classical lambda calculus with de Bruijn indices\*) fun printBEXP (BID n) =

fun printBEXP (BID n) =

    print (Int .toString (n))

  | printBEXP (BLAM (e)) =

    (print "(\\";

     printBEXP e;

     print ")")

  | printBEXP (BAPP(e1,e2)) =

    (print "(";

     printBEXP e1;

     print " ";

     printBEXP e2;

     print ")");

1. For each term below, write it in LEXP and print it using printLEXP, write its translation by *f* into IEXP and print it using printIEXP, and its translation by *ω* into BEXP and print it using printBEXP,

(a) (*λxz.xz*). (1.5)

printLEXP at101 gives

(\x.(\z.(x z)))val it = () : unit

printIEXP aIt101 gives

[x][z]<z>xval it = () : unit

printBEXP aBt101 gives

(\(\(2 1)))val it = () : unit

(b) (*λxy.xy*). (1.5)

printLEXP bt101 gives

(\x.(\y.(x y)))val it = () : unit

printIEXP bIt101 gives

[x][y]<x>yval it = () : unit

printBEXP bBt101 gives

(\(\(2 1)))val it = () : unit

1. *xz*(*λxy.xy*). (1.5)

printLEXP ct101 gives

((x z) (\x.(\y.(x y))))val it = () : unit

1. (*λxy.xy*)*xz*. (1.5)

......