```
<lc-exp> ::= <identificador>
              var-exp(id)
         ::= "(" "lambda" "(" <identificador>* ")" <lc-exp> ")"
              lambda-exp(lid, exp)
         ::= "(" <lc-exp> <lc-exp> ")"
              app-exp(rator, rand)
```

1. Dibujar los AST

a) (lambda (x y z) (lambda (x y) (x (lambda (x) y))))

b) (x (lambda (x a b) (lambda (x a b) (lambda (x a b) (x y)))))

c) (x (x (x (x (x (x (x (x (x lambda (a b c) (x (x (x (x y))))))))))))

2. Definir datatype

1. Hacer una función que transforme unparser AST -> listas

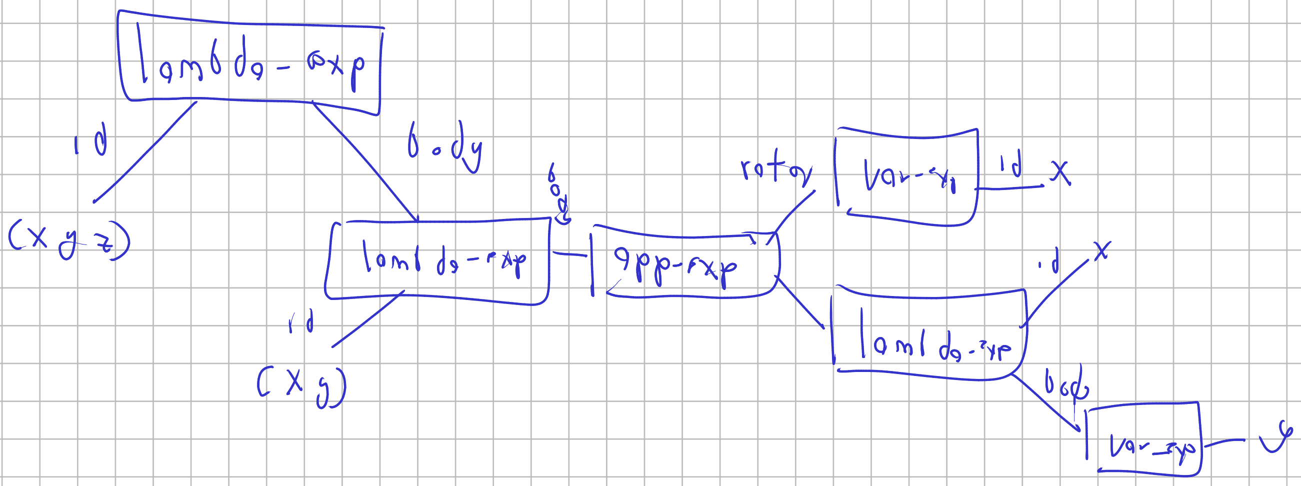
2. Hacer una función que transforme parser listas -> AST

```
   '(var-exp x) --> (var-exp x) ; '(lambda (x y) x)
                                  (lambda-exp '(x y) (var-exp 'x))
```

a) (lambda (x y z) (lambda (x y) (x (lambda (x) y))))

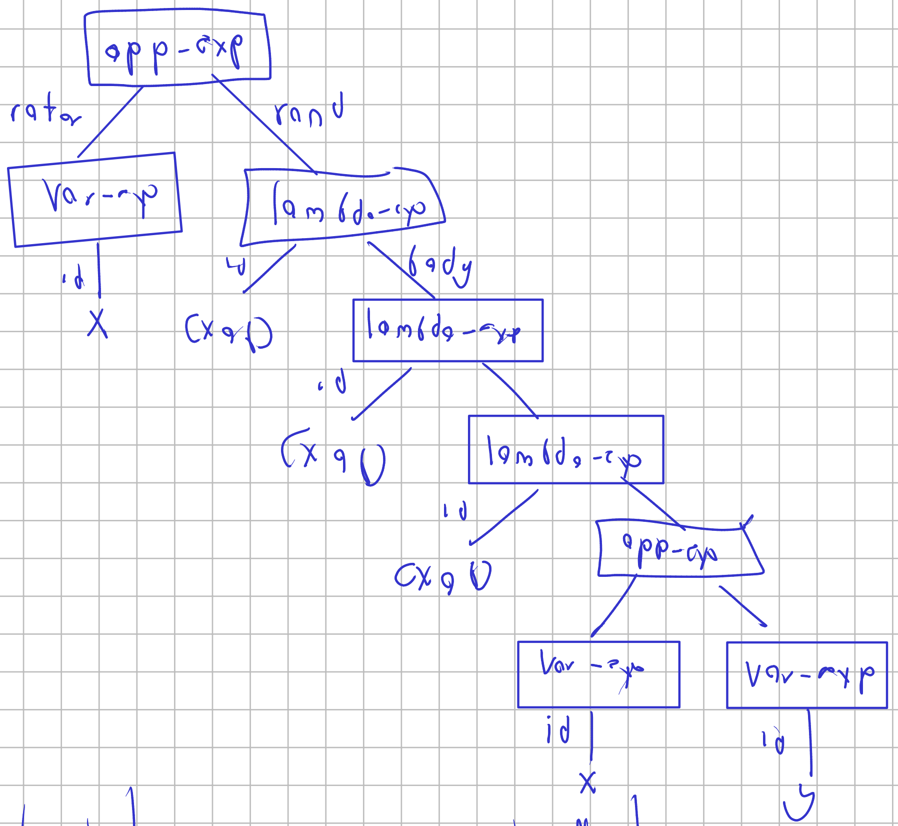
b) (x (lambda (x a b) (lambda (x a b) (lambda (x a b) (x y)))))

c) (x (x (x (x (x (x (x (x (x lambda (a b c) (x (x (x (x y))))))))))))

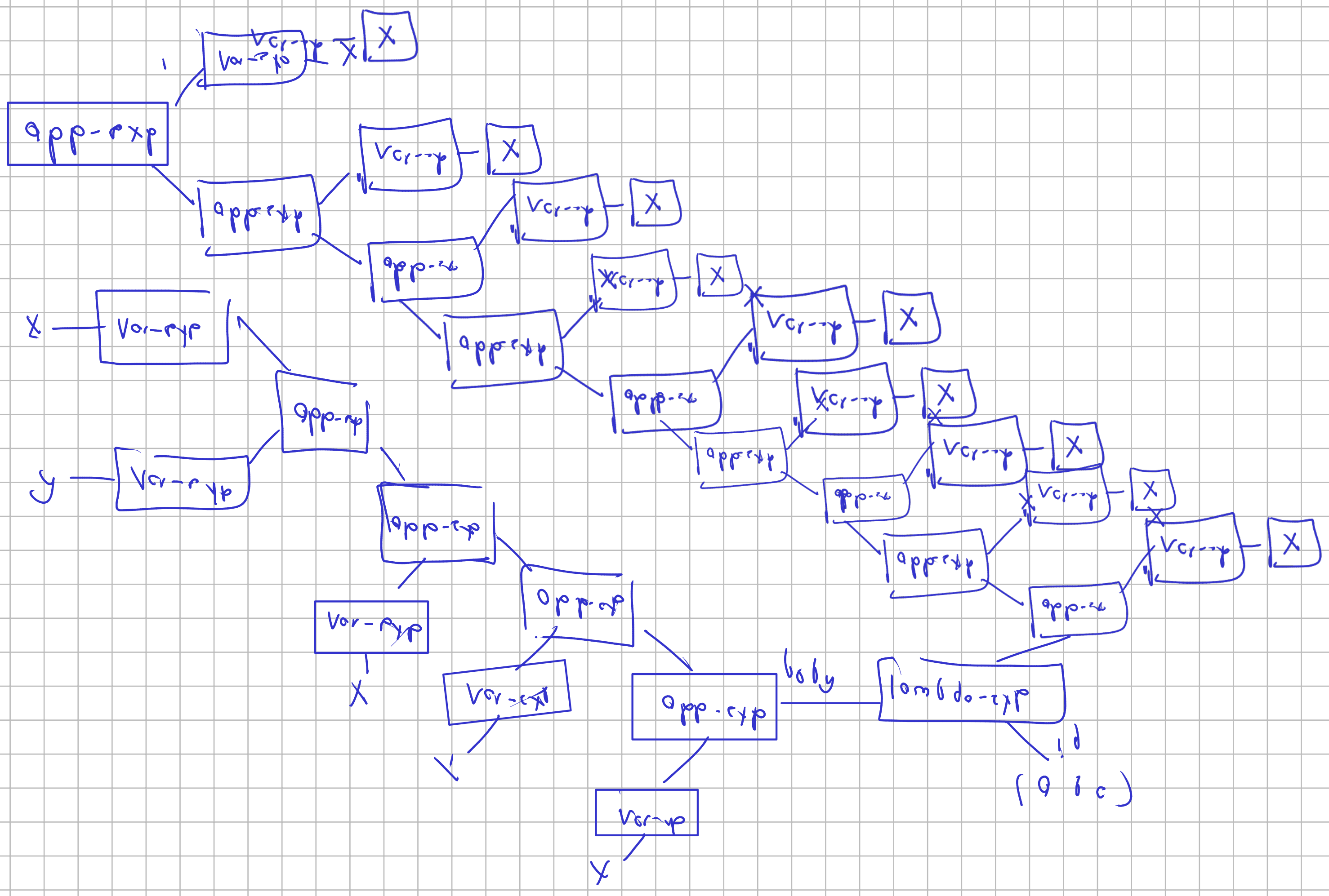


b) (x (lambda (x a b) (lambda (x a b) (lambda (x a b) (x y)))))

c) (x (x (x (x (x (x (x (x (x lambda (a b c) (x (x (x (x y))))))))))))



(x (x (x (x (x (x (x (x (x lambda (a b c) (x (x (x (x y))))))))))))



app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
body
lambda-exp
id
(a b c)
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
app-exp
var-exp
x
var-exp
y