Functional and Logic Programming - Home Assignment 5 (Solution)

May 28, 2019

Exercises - λ-Calculus

1

$$
\begin{aligned}
&\textbf{a)} C\ I\ 10\ (\lambda z.*\ 5\ z)\\
&= (\lambda f.\lambda x.\lambda y.f\ y\ x)\ I\ 10\ (\lambda z.*\ 5\ z)\\
&\underset{\beta}{\longrightarrow} (\lambda x.\lambda y.I\ y\ x)\ 10\ (\lambda z.*\ 5\ z)\\
&\underset{\beta}{\longrightarrow} (\lambda y.I\ y\ 10)\ (\lambda z.*\ 5\ z)\\
&\underset{\beta}{\longrightarrow} I\ (\lambda z.*\ 5\ z)\ 10\\
&= (\lambda x.x)\ (\lambda z.*\ 5\ z)\ 10\\
&\underset{\beta}{\longrightarrow} (\lambda z.*\ 5\ z)\ 10\\
&\underset{\beta}{\longrightarrow} *\ 5\ 10\\
&\underset{\delta}{\longrightarrow} 50
\end{aligned}
$$

b) $S\ K\ K$

$$
\begin{aligned}
&= (\lambda x.\lambda y.\lambda z.\,(x\ z)\ (y\ z))\ K\ K \\
&\xrightarrow[\beta]{} (\lambda y.\lambda z.\,(K\ z)\ (y\ z))\ K \\
&\xrightarrow[\beta]{} \lambda z.\,(K\ z)\ (K\ z) \\
&= \lambda z.\,((\lambda x.\lambda y.x)\ z)\ (K\ z) \\
&\xrightarrow[\beta]{} \lambda z.\,(\lambda y.z)\ (K\ z) \\
&\xrightarrow[\beta]{} \lambda z.z
\end{aligned}
$$

c) $S\ (K\ S)\ K$

$$
\begin{aligned}
&= (\lambda x.\lambda y.\lambda z.\,(x\ z)\ (y\ z))\ (K\ S)\ K \\
&\xrightarrow[\beta]{} (\lambda x.\lambda y.\lambda z.\,(x\ z)\ (y\ z))\ (K\ S)\ K \\
&\xrightarrow[\beta]{} (\lambda y.\lambda z.\,((K\ S)\ z)\ (y\ z))\ K \\
&\xrightarrow[\beta]{} \lambda z.\,((K\ S)\ z)\ (K\ z) \\
&= \lambda z.\,(((\lambda x.\lambda y.x)\ S)\ z)\ (K\ z) \\
&\xrightarrow[\beta]{} \lambda z.\,((\lambda y.S)\ z)\ (K\ z) \\
&\xrightarrow[\beta]{} \lambda z.S\ (K\ z) \\
&\xrightarrow[\alpha]{} \lambda w.S\ (K\ w) \\
&= \lambda w.\,(\lambda x.\lambda y.\lambda z.\,(x\ z)\ (y\ z))\ (K\ w) \\
&\xrightarrow[\beta]{} \lambda w.\lambda y.\lambda z.\,((K\ w)\ z)\ (y\ z) \\
&\xrightarrow[\alpha]{} \lambda w.\lambda f.\lambda z.\,((K\ w)\ z)\ (f\ z) \\
&= \lambda w.\lambda f.\lambda z.\,(((\lambda x.\lambda y.x)\ w)\ z)\ (f\ z) \\
&\xrightarrow[\beta]{} \lambda w.\lambda f.\lambda z.\,((\lambda y.w)\ z)\ (f\ z) \\
&\xrightarrow[\beta]{} \lambda w.\lambda f.\lambda z.w\ (f\ z)
\end{aligned}
$$

2

a) $(\lambda f.\lambda x.f\ \ (f\ x))\,(\lambda y.+\ y\,1)\,(+\,2\,3)$

$\xrightarrow[\beta]{} (\lambda x.\,(\lambda y.+\ y\,1)\ \ ((\lambda y.+\ y\,1)\ \ x))\,(+\,2\,3)$

$\xrightarrow[\beta]{} (\lambda y.+\ y\,1)\ \ ((\lambda y.+\ y\,1)\ \ (+\,2\,3))$

$\xrightarrow[\beta]{} +\ \ ((\lambda y.+\ y\,1)\ \ (+\,2\,3))\ \ 1$

$\xrightarrow[\beta]{} +\ \ (+\ \ (+\,2\,3)\ \ 1)\ \ 1$

$\xrightarrow[\delta]{} +\ \ (+\,5\,1)\ \ 1$

$\xrightarrow[\delta]{} +\,6\,1$

$\xrightarrow[\delta]{} 7$

b) $(\lambda x.\,(\lambda z.z\ x)\ \ (\lambda x.x))\ \ y$

$\xrightarrow[\beta]{} (\lambda z.z\ y)\ \ (\lambda x.x)$

$\xrightarrow[\beta]{} (\lambda x.x)\ \ y$

$\xrightarrow[\beta]{} y$

c) $(\lambda x.+\ \ ((\lambda y.\,((\lambda x.*\ \ x\,y)\ \ 2))\ \ x)\ \ y)$

$\xrightarrow[\alpha]{} \lambda x.+\ \ ((\lambda y.\,((\lambda z.*\ \ z\,y)\ \ 2))\ \ x)\ \ y$

$\xrightarrow[\beta]{} \lambda x.+\ \ ((\lambda z.*\ \ z\,x)\ \ 2)\ \ y$

$\xrightarrow[\beta]{} \lambda x.+\ \ (*\,2\,x)\ \ y$

d) $(\lambda x.\,(\lambda y.+\ x\,y)\ 5)\ ((\lambda y.-\ y\,3)\ 7)$

$\xrightarrow[\beta]{} (\lambda y.+\ ((\lambda y.-\ y\,3)\ 7)\ y)\ 5$

$\xrightarrow[\beta]{} +\ ((\lambda y.-\ y\,3)\ 7)\ 5$

$\xrightarrow[\beta]{} +\ (-\,7\,3)\ 5$

$\xrightarrow[\delta]{} +\,4\,5$

$\xrightarrow[\delta]{} 9$

e) $(\lambda x.\lambda y.x)\ (\lambda f.f\ \ (f\ 1))\ ((\lambda x.x\ x\ x)\ (\lambda x.x\ x\ x))\ (\lambda y.*\ 2\,y)$

$\xrightarrow[\beta]{} (\lambda y.\,(\lambda f.f\ \ (f\ 1)))\ ((\lambda x.x\ x\ x)\ (\lambda x.x\ x\ x))\ (\lambda y.*\ 2\,y)$

$\xrightarrow[\beta]{} (\lambda f.f\ \ (f\ 1))\ (\lambda y.*\ 2\,y)$

$\xrightarrow[\beta]{} (\lambda y.*\ 2\,y)\ ((\lambda y.*\ 2\,y)\ 1)$

$\xrightarrow[\beta]{} *\,2\ ((\lambda y.*\ 2\,y)\ 1)$

$\xrightarrow[\beta]{} *\,2\ (*\,2\,1)$

$\xrightarrow[\delta]{} *\,2\,2$

$\xrightarrow[\delta]{} 4$

If we'll use **applicative order** evaluation on **e** the evaluation will **never terminate**:

since $(\lambda y.*\ 2\,y)$ is already in normal form the next reduction step will be on $(\lambda x.\lambda y.x)\ (\lambda f.f\ \ (f\ 1))\ ((\lambda x.x\ x\ x)\ (\lambda x.x\ x\ x))$ where $((\lambda x.x\ x\ x)\ (\lambda x.x\ x\ x))$ is the argument, so $((\lambda x.x\ x\ x)\ (\lambda x.x\ x\ x))$ will get evaluated first and will blow up the expression, because:

$$
\begin{aligned}
&(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\\
&\xrightarrow[\beta]{}(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\\
&\xrightarrow[\beta]{}(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\\
&\xrightarrow[\beta]{}(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\;(\lambda x.x\;x\;x)\\
&\xrightarrow[\beta]{}\ldots
\end{aligned}
$$

3

A non-recursive definition of the function **mcc**:

$$H = \lambda f.\lambda n.f\;\;(>\;\;n\,100)\;\;(-\,n\,10)\;\;(f\;\;(f\;\;(+\,n\,11)))$$

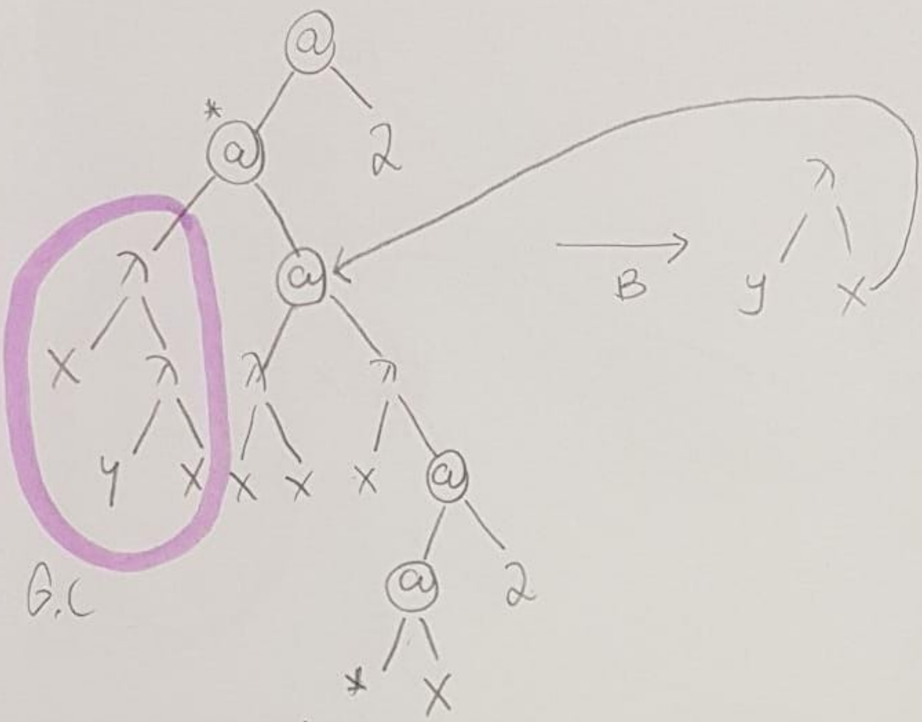
$$\mathbf{mcc} = Y\;H$$

where Y is the **Y combinator**:

$$Y = \lambda h.\,(\lambda x.h\;\;(x\,x))\;\;(\lambda x.h\;\;(x\,x))$$

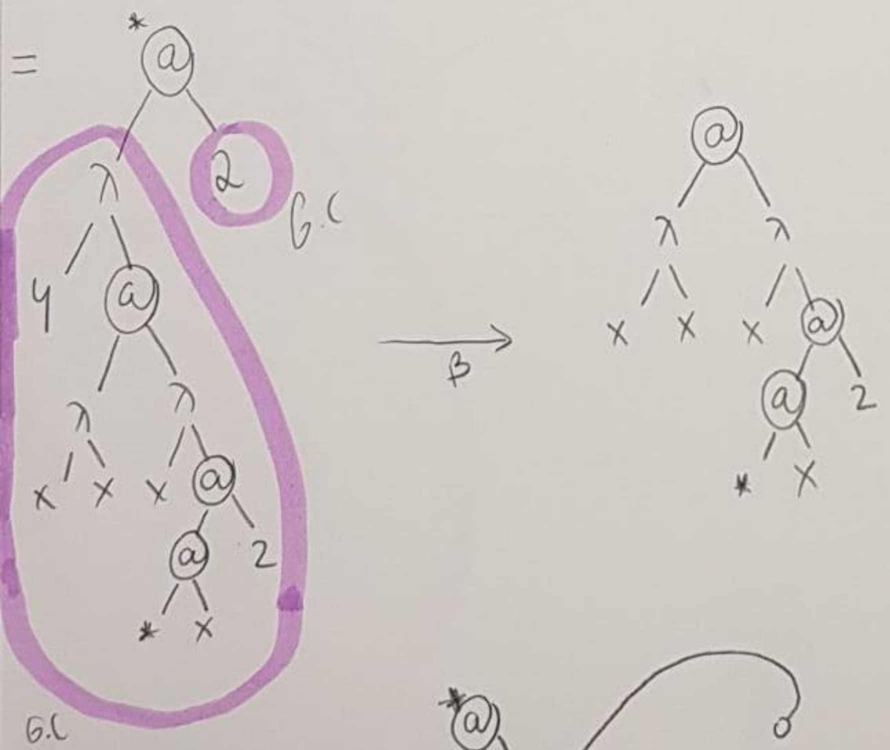
4a

$(\lambda x.\lambda y.x)\ ((\lambda x.x)(\lambda x.*\ x\ 2))\ 2$

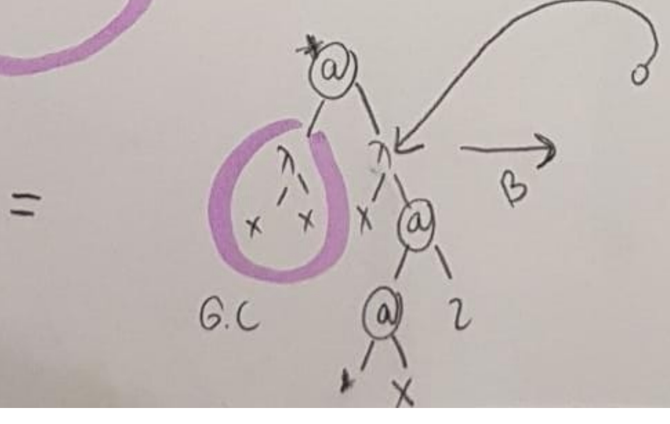


copy the argument to the @* node.

* Note: y is <u>not</u> used in the body of the function. 2 is also sent to g.c.

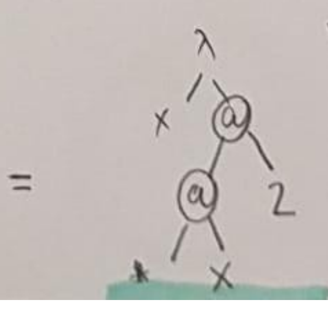


- projection function
- copy the argument to the @* node



WHNF.

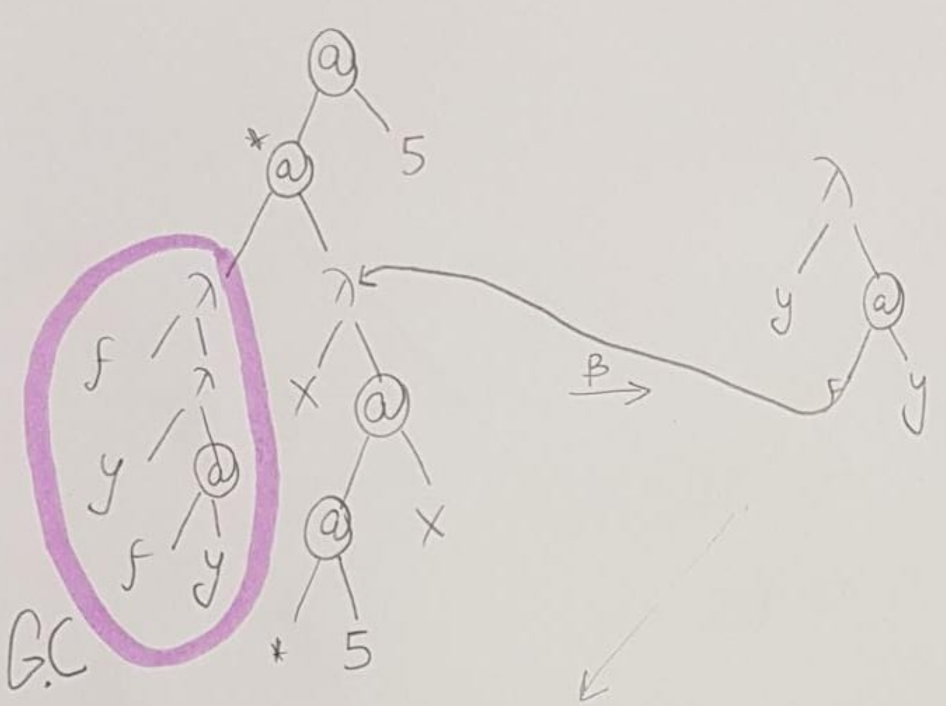
Done.



4 b

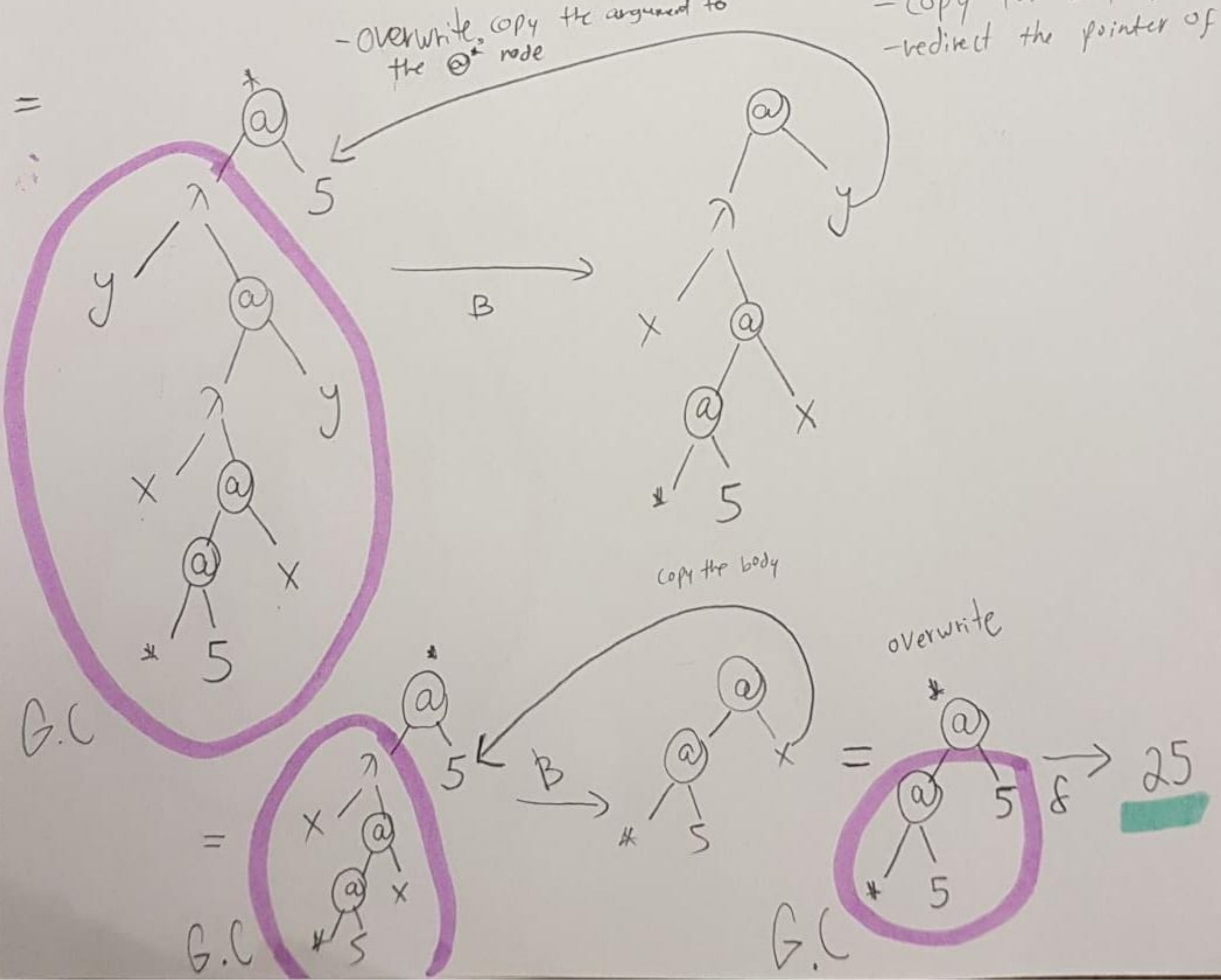
$(\lambda f.\lambda y.f\,y)\,(\lambda x.\,*\,5\;x)\;5$

- copy the body of the functi[on]
- redirect the pointers of f.



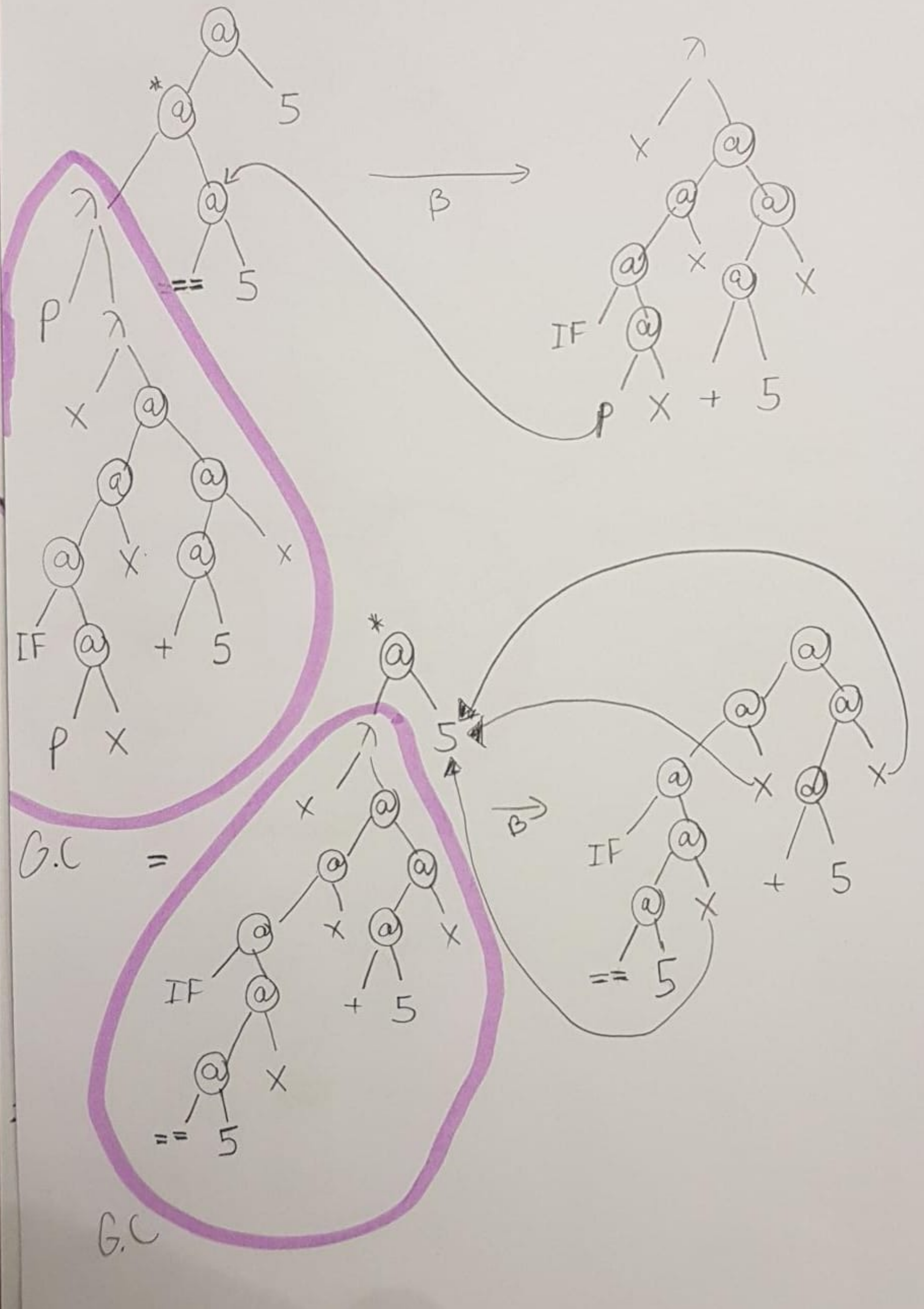
- overwrite, copy the argument to the @* node

- copy the body of the functi[on]
- redirect the pointer of y



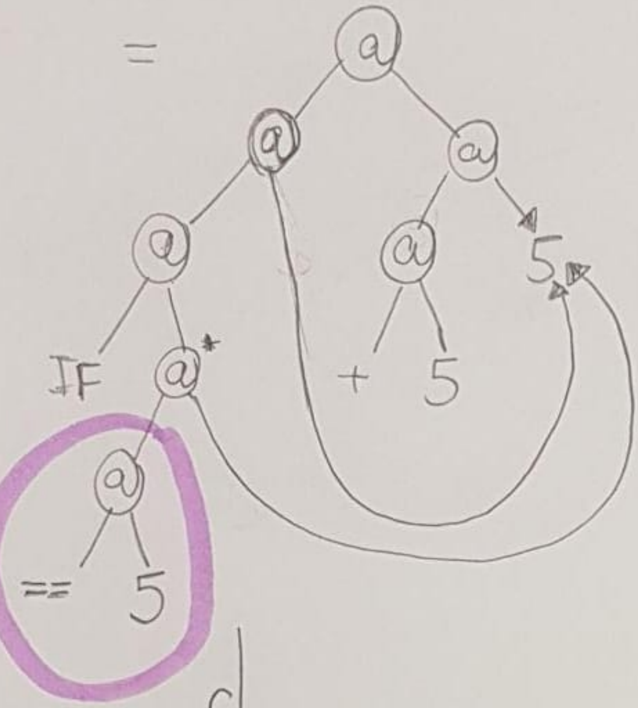
4c

$(\lambda p.\lambda x.\text{If}\ (p\ x)\ x\ (+\ 5\ x))\ (==\ 5)\ 5$



4c

=



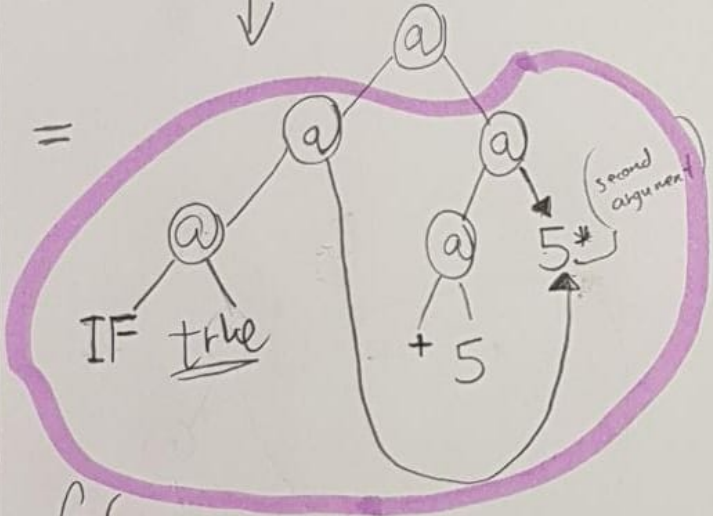
IF is Strict in its first argument.

((==5) 5) evaluate to true

G.C

δ

=



G.C

IF is a projection function
the first argument evaluate to true
so we'll evaluate than expression. (5)
(second argument) and then copy
to the

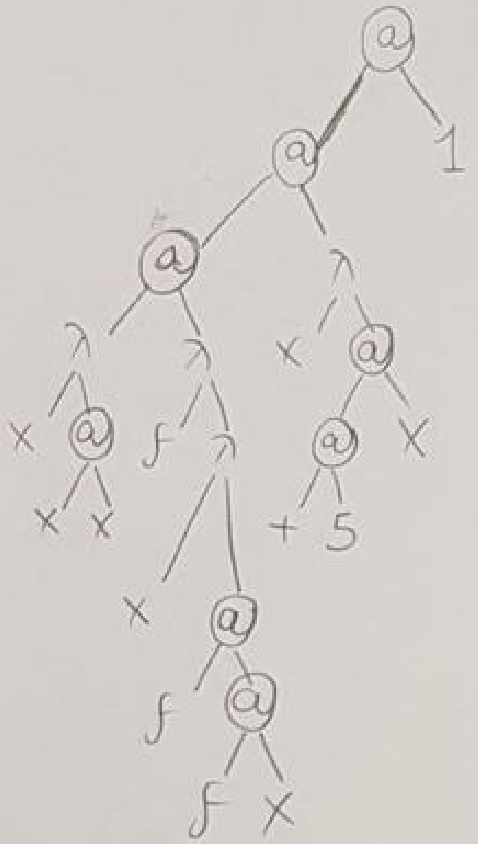
→ copy 5 to the root

5

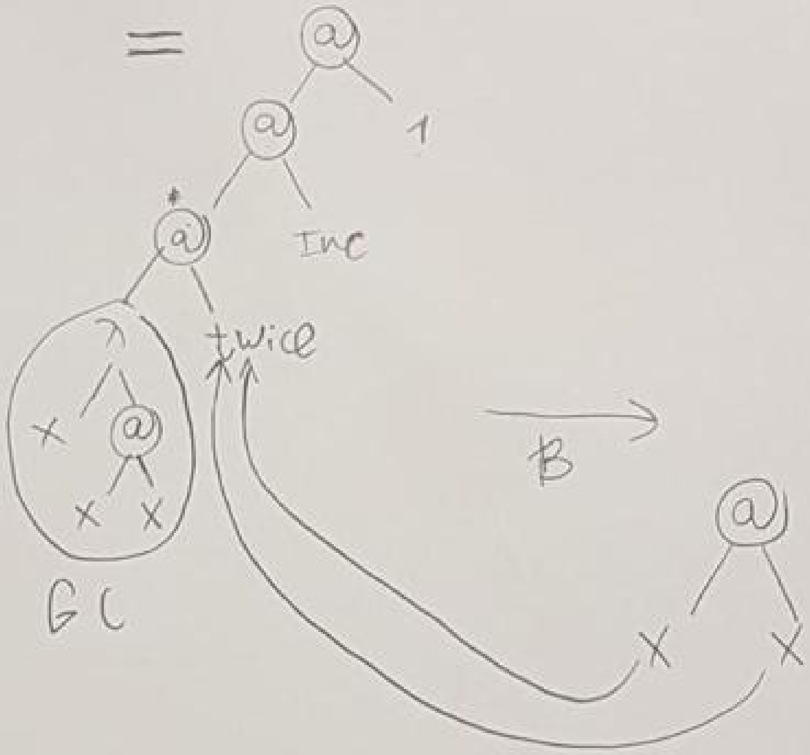
4d

$(\lambda x.xx)\ (\lambda f.\ \lambda x.\ f\ (f\ x))\ (\lambda x.+\ 5\ x)\ 1$

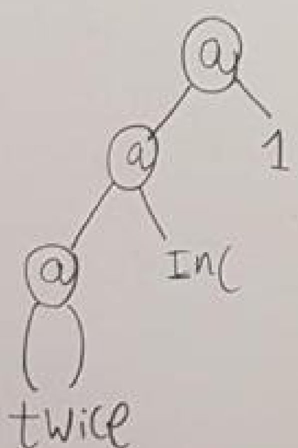
$$\begin{cases} Inc = \lambda x. + 5\ x \\ twice = \lambda f.\ \lambda x.\ f(f x) \end{cases}$$



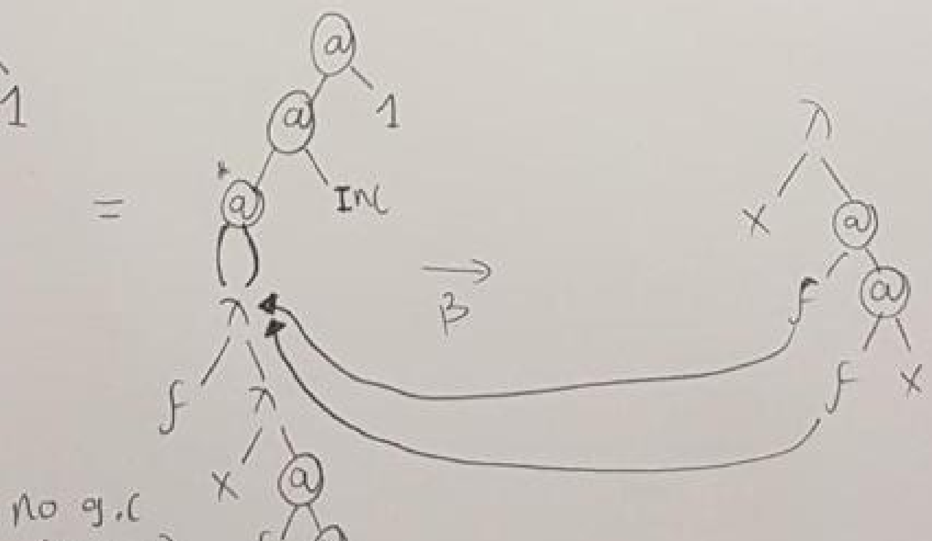
=



overwrite



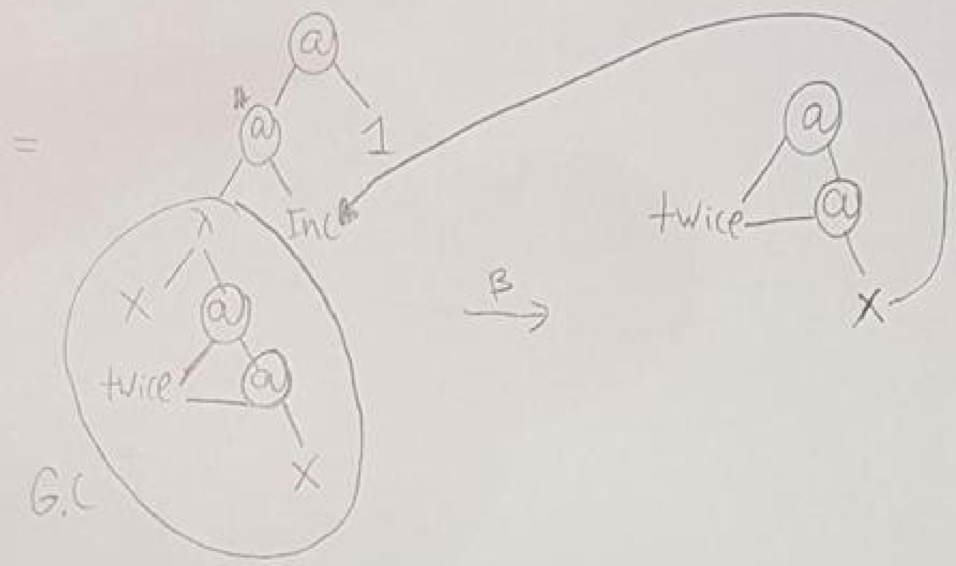
=

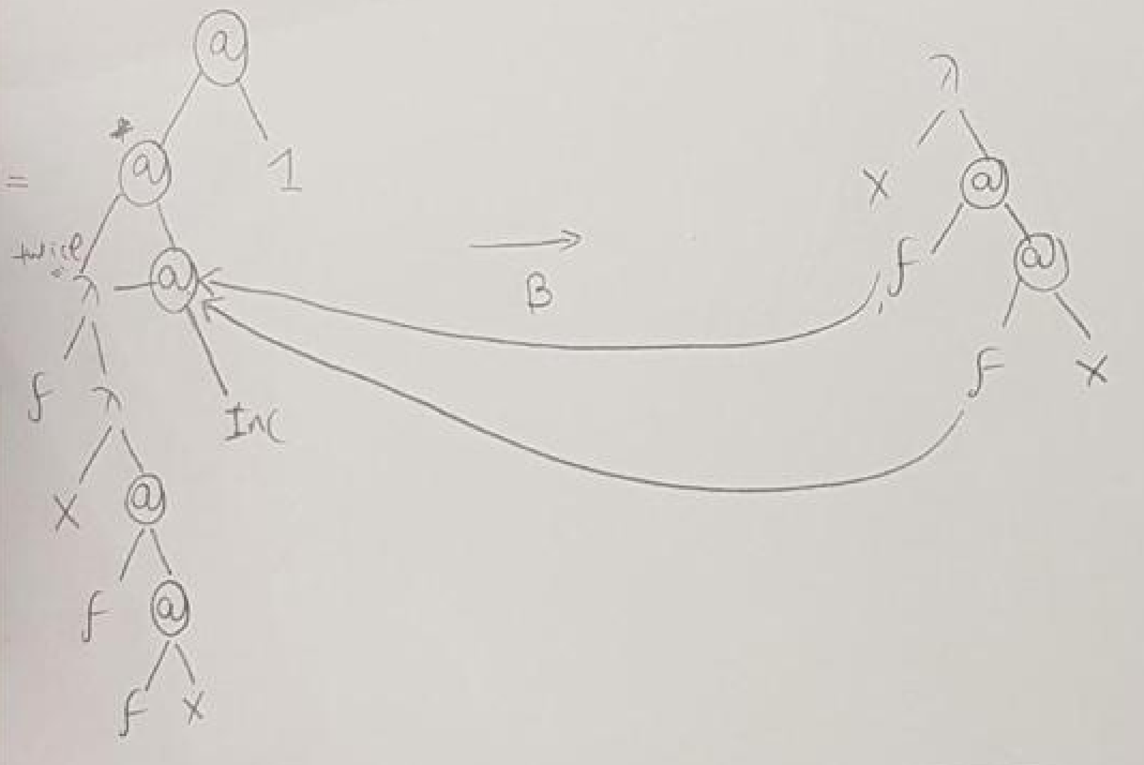


↓

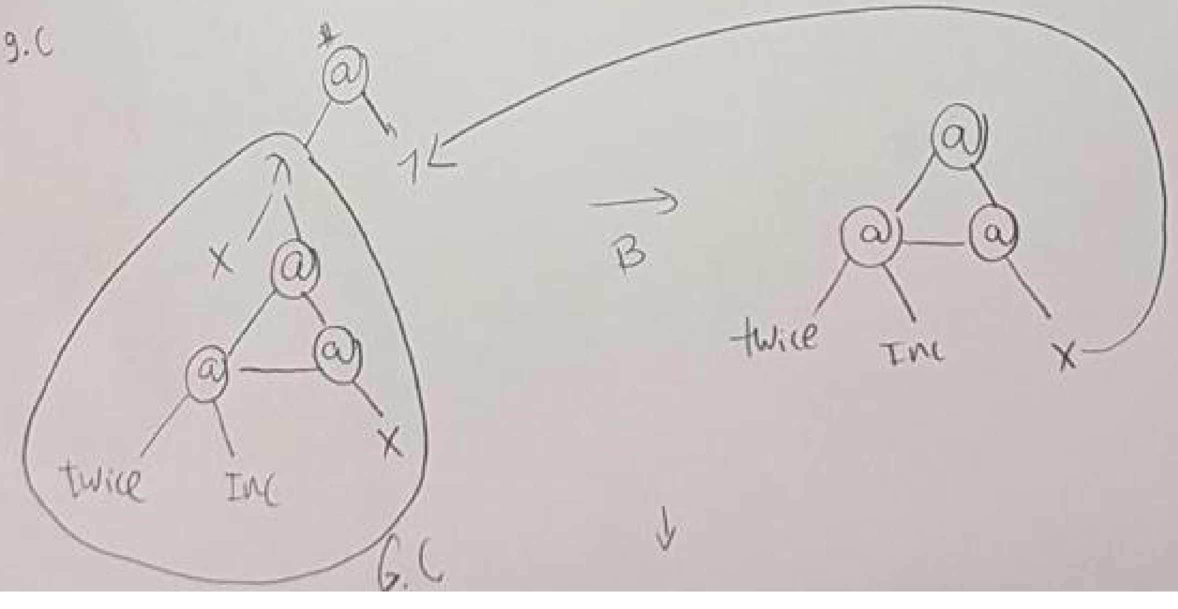
4d

overwrite

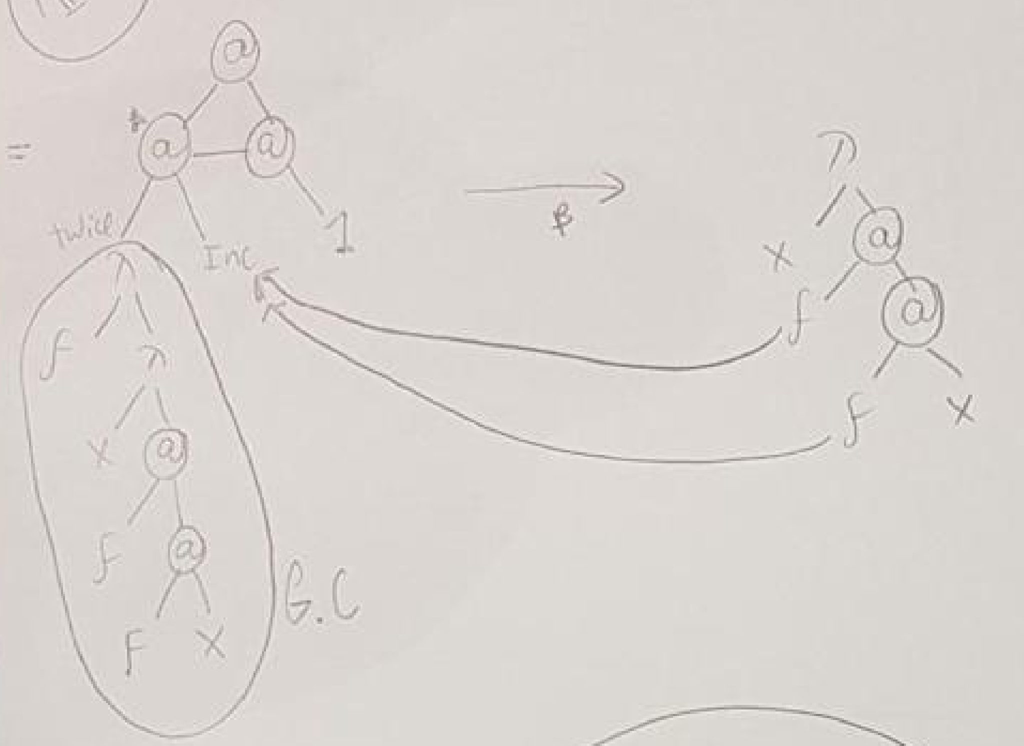


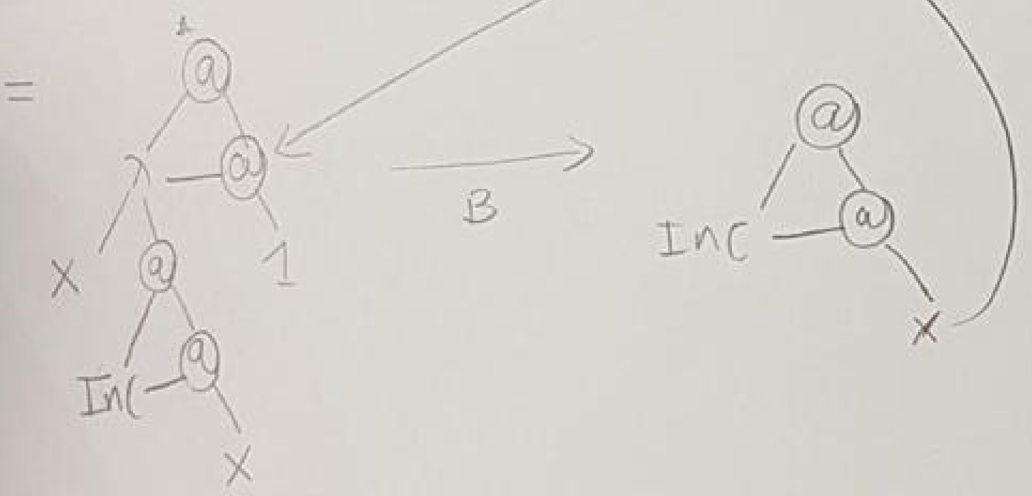


no g.c

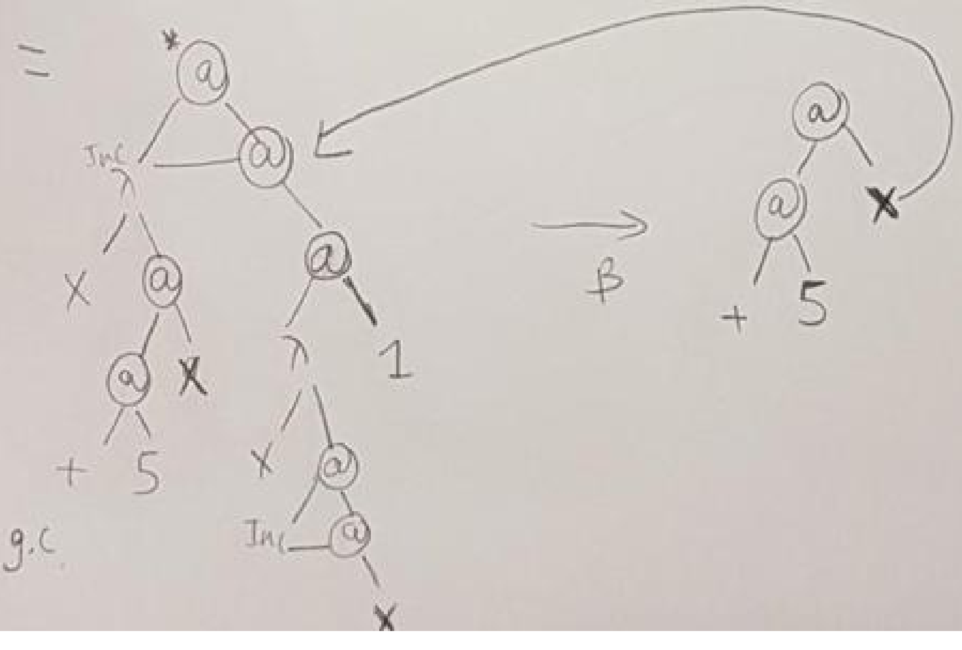


4d
part II



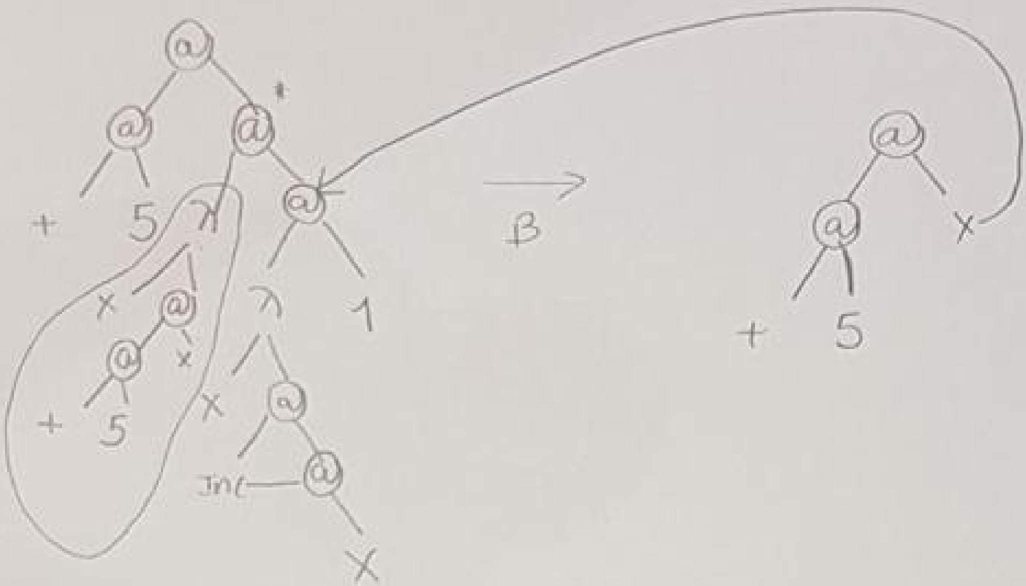


no g.c



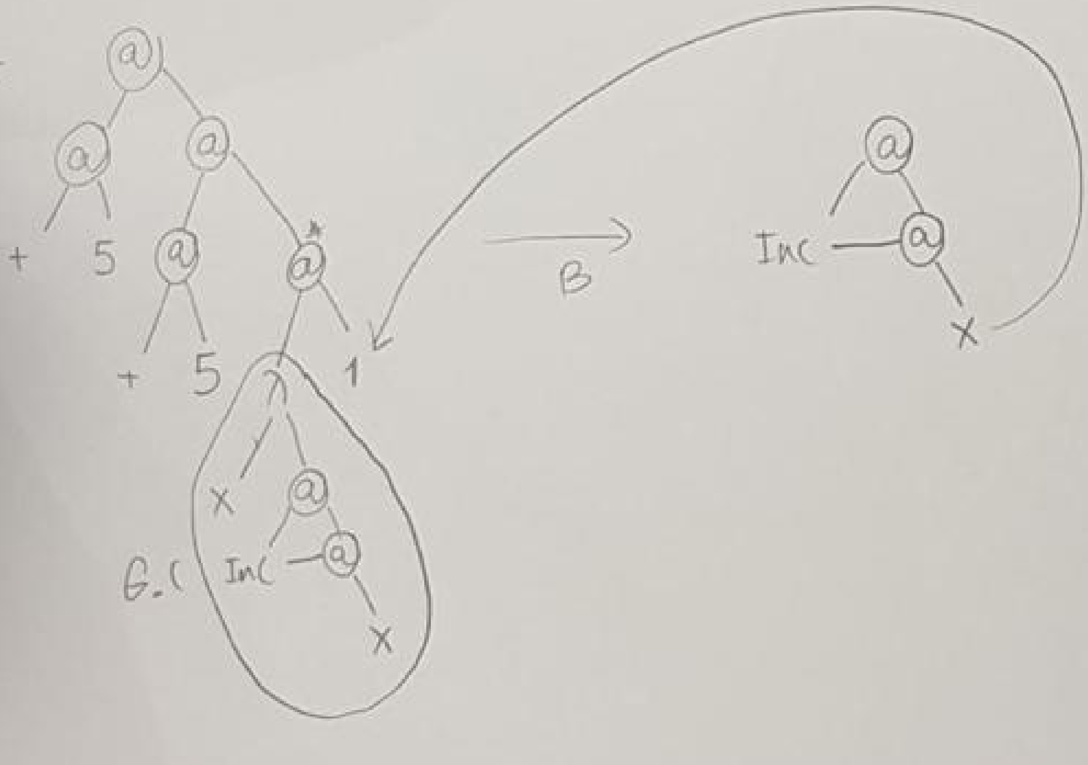
no g.c

=



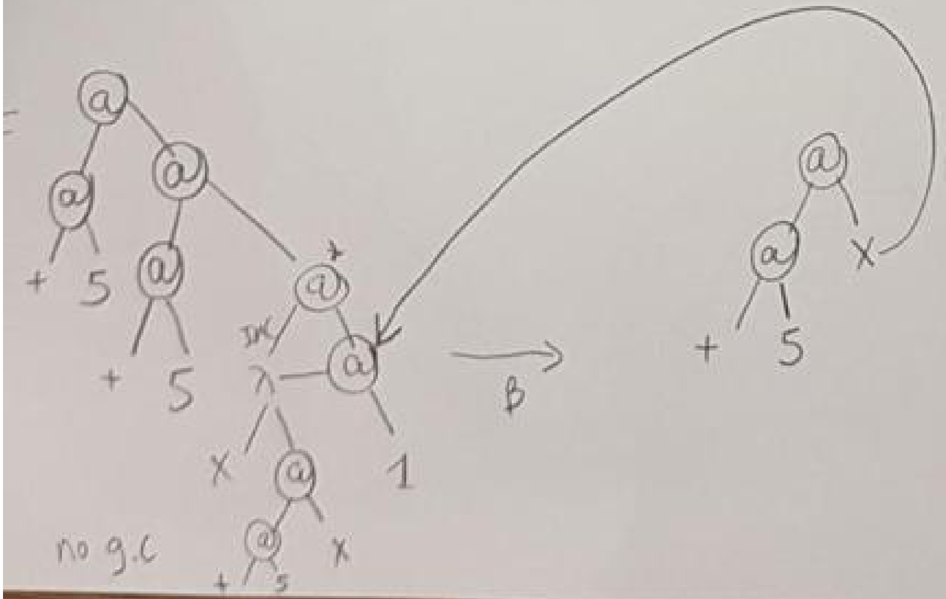
+
5
x
1
Inc
G.C
β

=

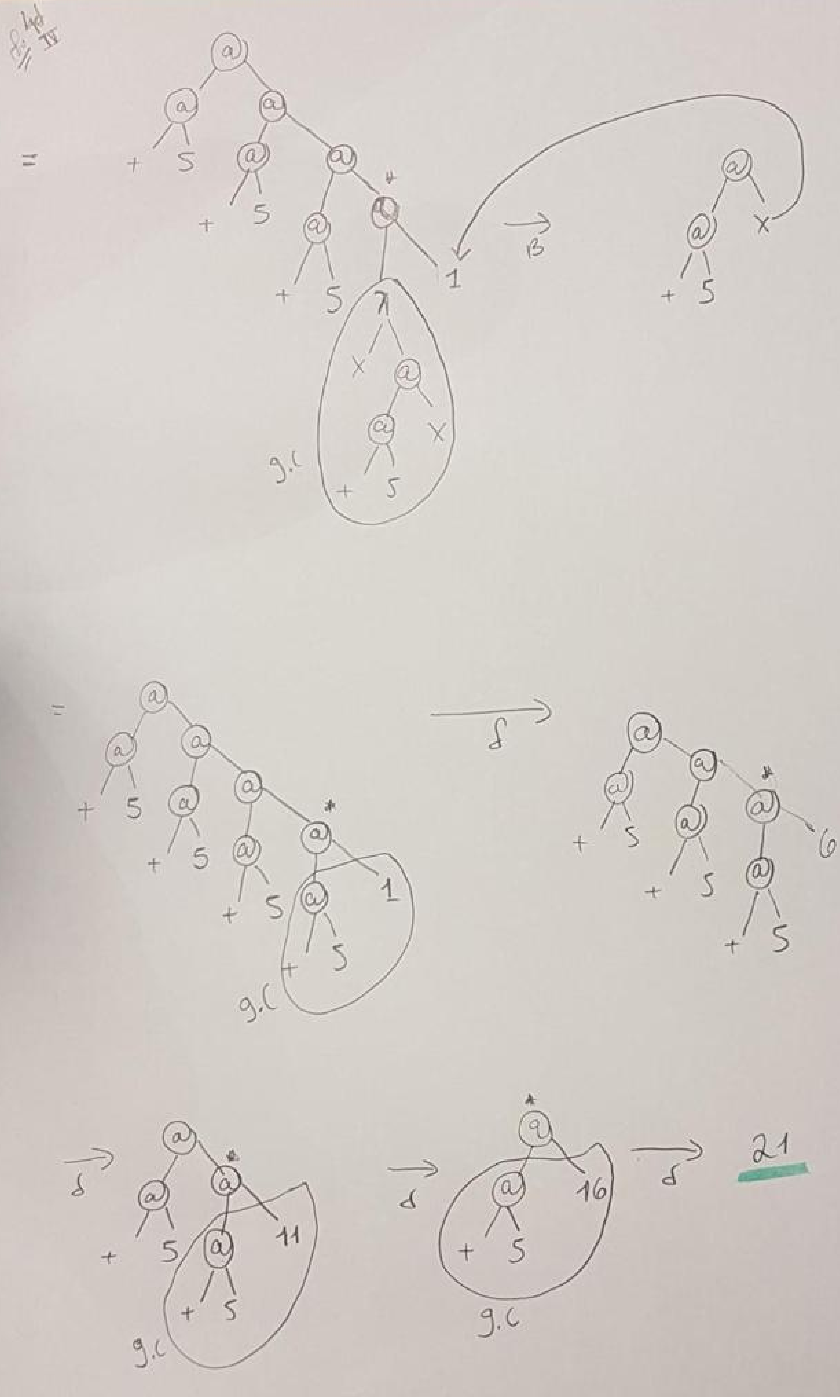


+
5
1
x
Inc
G.C
β

=



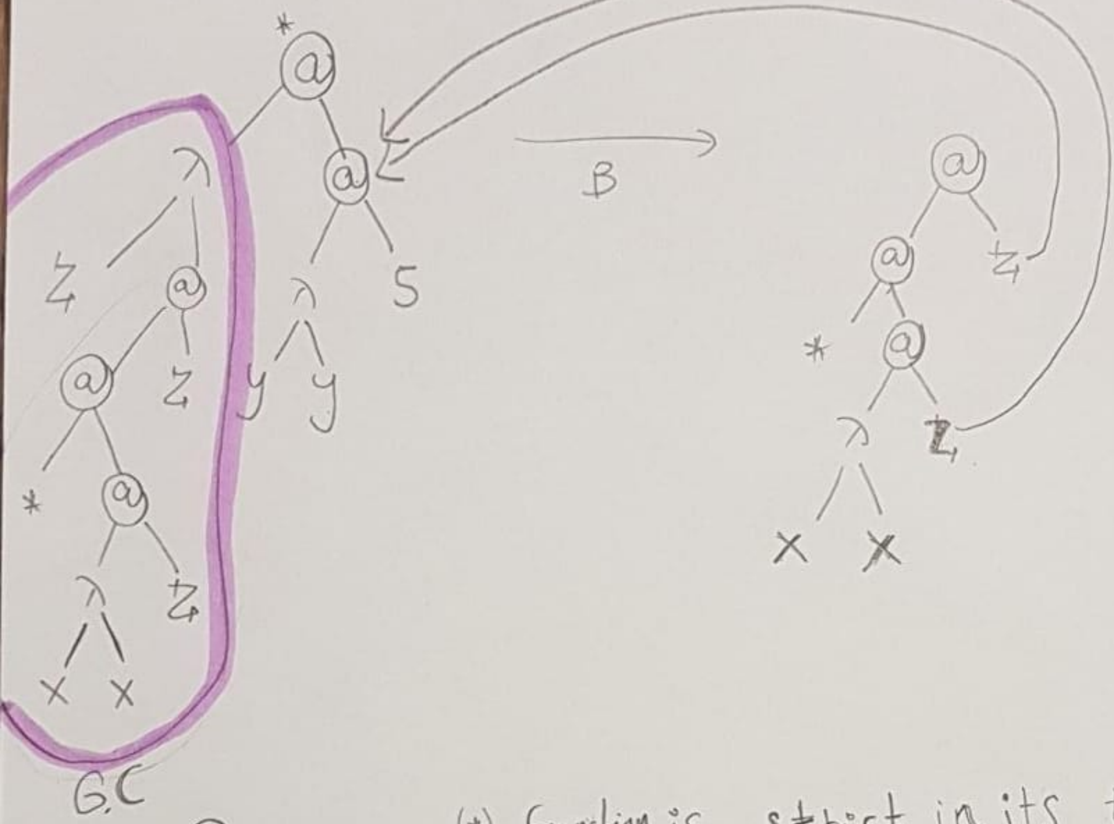
+
5
Inc
x
1
no g.c
β



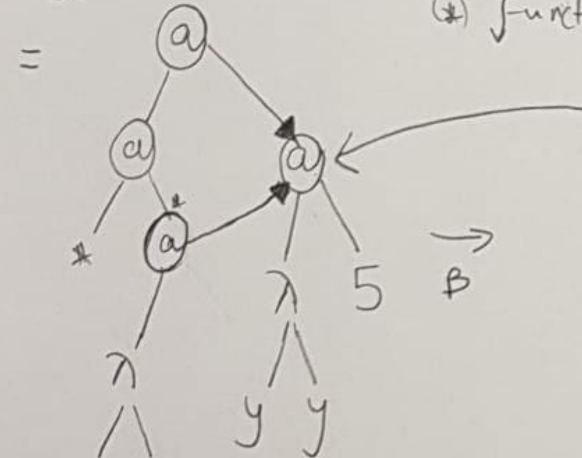
g.c
β
δ
21

4e

$$(\lambda z. * ((\lambda x.x)\, z)\, z)\ ((\lambda y.y)\, 5)$$



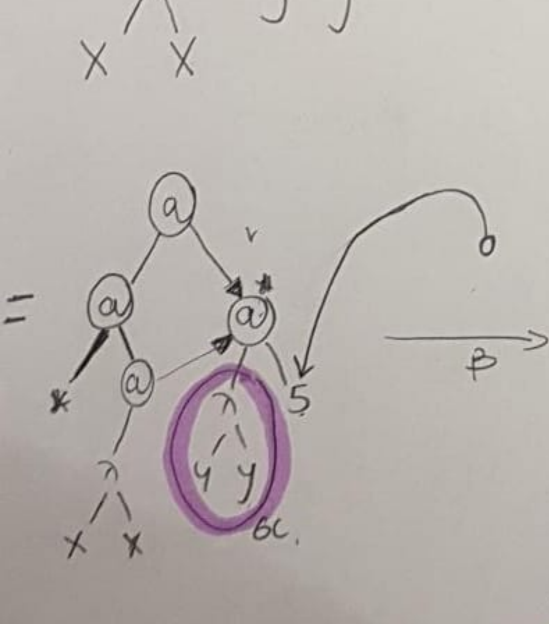
=



(*) function is strict in its two arguments.

- The identity function is a projection function. $(\lambda x.x)$
- Reduce the argument, the subgraph that ↙ is pointing at.
- once in WHNF - copy to root.

=



- another projection function. (identity)
- The argument (5) is in WHNF
- Safe to copy 5 to root