Exam for Programming Language Principals, Design and Implementation (Extended)

ID 1803086

After inserting your student ID and the module title in the preamble, write your answers below.

Question 1

(a)

$$M = \lambda f : \mathbb{B} \to \mathbb{B}.\lambda g : \mathbb{B} \to \mathbb{B}.\lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } fy \texttt{ else } gy$$

(i)

$$\frac{\frac{\frac{\frac{\frac{}{\Gamma \vdash x : \mathbb{B}}\,\texttt{VAR} \qquad \frac{\frac{}{\Gamma \vdash f : \mathbb{B} \to \mathbb{B}}\,\texttt{VAR} \qquad \frac{}{\Gamma \vdash y : \mathbb{B}}\,\texttt{VAR}}{\Gamma \vdash fy : \mathbb{B}}\,\texttt{APP} \qquad \Pi_1}{\Gamma, y : \mathbb{B} \vdash \texttt{if } x \texttt{ then } fy \texttt{ else } gy : \mathbb{B}}}{\Gamma, x : \mathbb{B} \vdash \lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } fy \texttt{ else } gy : \mathbb{B} \to \mathbb{B}}\,\texttt{ABS}}{\Gamma, g : \mathbb{B} \to \mathbb{B} \vdash \lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } fy \texttt{ else } gy : \mathbb{B} \to \mathbb{B} \to \mathbb{B}}\,\texttt{ABS}}{f : \mathbb{B} \to \mathbb{B} \vdash M_1 : (\mathbb{B} \to \mathbb{B}) \to \mathbb{B} \to \mathbb{B} \to \mathbb{B}}\,\texttt{ABS}}{\{\} \vdash M : (\mathbb{B} \to \mathbb{B}) \to (\mathbb{B} \to \mathbb{B}) \to \mathbb{B} \to \mathbb{B} \to \mathbb{B}}\,\texttt{ABS}$$

Where:

- $M_1 = \lambda g : \mathbb{B} \to \mathbb{B}.\lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } fy \texttt{ else } gy$
- $\Pi_1 =$

$$\frac{\frac{}{\Gamma \vdash g : \mathbb{B} \to \mathbb{B}}\,\texttt{VAR} \qquad \frac{}{\Gamma \vdash y : \mathbb{B}}\,\texttt{VAR}}{\Gamma \vdash gy : \mathbb{B}}\,\texttt{APP}$$

(ii) To produce the exclusive function from M we can define the first order parameters F and G as follows:

$$F = \lambda y : \mathbb{B}.\texttt{if } y \texttt{ then false else true} : \mathbb{B} \to \mathbb{B}$$

$$G = \lambda y : \mathbb{B}.\texttt{if } y \texttt{ then true else false} : \mathbb{B} \to \mathbb{B}$$

Alternatively, G can simply be defined as the boolean identity function $\lambda y : \mathbb{B}.y : \mathbb{B} \to \mathbb{B}$. This is the definition I will use in later parts of the question.

(iii) The fact that this expression is well-typed under the Simply-typed λ-Calculus means

Finish this question

(iv)

$$\dfrac{\dfrac{}{\lambda f : \mathbb{B} \to \mathbb{B}.M_1 F \to_v \lambda g : \mathbb{B} \to \mathbb{B}\lambda x : \mathbb{B}\texttt{if } x \texttt{ then } Fy \texttt{ else } gy}\,\beta}{\dfrac{\lambda f : \mathbb{B} \to \mathbb{B}.M_1 FG \to_v \\ (\lambda g : \mathbb{B} \to \mathbb{B}\lambda x : \mathbb{B}\texttt{if } x \texttt{ then } Fy \texttt{ else } gy)G}{(MFG)\texttt{false true} \to_v \\ ((\lambda g : \mathbb{B} \to \mathbb{B}\lambda x : \mathbb{B}\texttt{if } x \texttt{ then } Fy \texttt{ else } gy)G)\texttt{false true}}\,\texttt{CTX}_{(\bullet)\texttt{false true}}}\,\texttt{CTX}_{\bullet G} \tag{1}$$

$$\dfrac{\dfrac{}{(\lambda g : \mathbb{B} \to \mathbb{B}\lambda x : \mathbb{B}\texttt{if } x \texttt{ then } Fy \texttt{ else } gy)G \to_v \\ \lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } Fy \texttt{ else } Gy}\,\beta}{((\lambda g : \mathbb{B} \to \mathbb{B}\lambda x : \mathbb{B}\texttt{if } x \texttt{ then } Fy \texttt{ else } gy)G)\texttt{false true} \to_v \\ (\lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } Fy \texttt{ else } Gy)\texttt{false true}}\,\texttt{CTX}_{(\bullet)\texttt{false true}} \tag{2}$$

$$\dfrac{\dfrac{}{(\lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } Fy \texttt{ else } Gy)\texttt{false} \to_v \\ \lambda y : \mathbb{B}\texttt{if false then } Fy \texttt{ else } Gy}\,\beta}{(\lambda x : \mathbb{B}.\lambda y : \mathbb{B}.\texttt{if } x \texttt{ then } Fy \texttt{ else } gy)\texttt{false true} \to_v \\ (\lambda y : \mathbb{B}\texttt{if false then } Fy \texttt{ else } Gy)\texttt{true}}\,\texttt{CTX}_{\bullet\texttt{true}} \tag{3}$$

$$\dfrac{}{(\lambda y : \mathbb{B}\texttt{if false then } Fy \texttt{ else } Gy)\texttt{true} \to_v \\ \texttt{if false then } F\texttt{true} \texttt{ else } G\texttt{true}}\,\beta \tag{4}$$

$$\dfrac{}{\texttt{if false then } F\texttt{true} \texttt{ else } G\texttt{true} \to_v G\texttt{true}}\,\texttt{IteF} \tag{5}$$

$$\dfrac{}{(\lambda y : \mathbb{B}.y)\texttt{true} \to_v \texttt{true}}\,\beta \tag{6}$$

Where:

- M_1 is as defined the same as above.

$\texttt{true} \in V$, $\therefore$ $(MFG)\texttt{false true}$ computes to a value.

(b)

$$\texttt{Stack} = \forall\alpha.(\mathbb{N} \to \alpha \to \alpha) \to \alpha \to \alpha$$

(i)

$$\lambda\alpha.\lambda f : \mathbb{N} \to \alpha \to \alpha.\lambda x : \alpha.f0(f0(f1x))$$

(ii)

$$\texttt{peek} = \lambda d : \mathbb{N}.\lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId$$

Where:

- $G = \lambda n : \mathbb{N}.\lambda g : \mathbb{N} \to \mathbb{N}.\lambda x : \mathbb{N}.gn$
- $I = \lambda x : \mathbb{N}.x$

$$\cfrac{\cfrac{\cfrac{\Pi_1 \quad \Pi_2}{\Gamma \vdash s\{\mathbb{N} \to \mathbb{N}\}G : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})}\,\texttt{APP} \qquad \cfrac{\cfrac{}{\Gamma, x : \mathbb{N} \vdash x : \mathbb{N}}\,\texttt{VAR}}{\Gamma \vdash I : \mathbb{N} \to \mathbb{N}}\,\texttt{ABS}}{\Gamma \vdash ((s\{\mathbb{N} \to \mathbb{N}\}G)I) : \mathbb{N} \to \mathbb{N}}\,\texttt{APP} \qquad \cfrac{}{\Gamma \vdash d : \mathbb{N}}\,\texttt{VAR}}{\cfrac{\Gamma, s : \texttt{Stack} \vdash ((s\{\mathbb{N} \to \mathbb{N}\}G)I)d : \mathbb{N} \to \texttt{Stack} \to \mathbb{N} : \mathbb{N}}{\cfrac{d : \mathbb{N} \vdash \lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId : \mathbb{N} \to \texttt{Stack} \to \mathbb{N} : \texttt{Stack} \to \mathbb{N}}{\{\} \vdash \lambda d : \mathbb{N}.\lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId : \mathbb{N} \to \texttt{Stack} \to \mathbb{N}}\,\texttt{ABS}}\,\texttt{ABS}}\,\texttt{APP}$$

Where:

- $\Pi_1 =$

$$\cfrac{\cfrac{}{\Gamma \vdash s : \texttt{State}}\,\texttt{VAR}}{\Gamma \vdash s\{\mathbb{N} \to \mathbb{N}\} : (\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})) \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})}\,\texttt{TAPP}$$

- $\Pi_2 =$

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma \vdash g : \mathbb{N} \to \mathbb{N}}\,\texttt{VAR} \qquad \cfrac{}{\Gamma \vdash n : \mathbb{N}}\,\texttt{VAR}}{\Gamma, x : \mathbb{N} \vdash gn : \mathbb{N}}\,\texttt{APP}}{\cfrac{\Gamma, g : \mathbb{N} \to \mathbb{N} \vdash \lambda x : \mathbb{N}.gn : \mathbb{N} \to \mathbb{N}}{\Gamma, n : \mathbb{N} \vdash \lambda g : \mathbb{N} \to \mathbb{N}.\lambda x : \mathbb{N}.gn : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}}\,\texttt{ABS}}\,\texttt{ABS}}{\Gamma \vdash G : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}}\,\texttt{ABS}$$

(iii) Prove:

$$\texttt{peek}\ \ ds_2 \to_v^* m$$

$$\cfrac{\cfrac{}{(\lambda d : \mathbb{N}.\lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId)d \to_v \lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId}\,\beta}{\texttt{peek}\ \ ds_2 \to_v (\lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId)s_2}\,\texttt{CTX}_{\bullet s_2}$$

$$\cfrac{}{(\lambda s : \texttt{Stack}.s\{\mathbb{N} \to \mathbb{N}\}GId)s_2 \to_v s_2\{\mathbb{N} \to \mathbb{N}\}GId}\,\beta$$

$$\cfrac{\cfrac{}{\begin{gathered}\lambda\alpha.\lambda f : \mathbb{N} \to \alpha \to \alpha.\lambda x : \alpha.fn(fmx)\{\mathbb{N} \to \mathbb{N}\} \to_v \\ \lambda f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}).\lambda x : (\mathbb{N} \to \mathbb{N}).fn(fmx)\end{gathered}}\,T_\beta}{\begin{gathered}((s_2\{\mathbb{N} \to \mathbb{N}\}G)I)d \to_v \\ (((\lambda f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}).\lambda x : \mathbb{N} \to \mathbb{N}.fn(fmx))G)I)d\end{gathered}}\,\texttt{CTX}_{(\bullet G)I)d}$$

$$\cfrac{\cfrac{}{\begin{gathered}(\lambda f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}).\lambda x : \mathbb{N} \to \mathbb{N}.fn(fmx))G \to_v \\ \lambda x : \mathbb{N} \to \mathbb{N}.Gn(Gmx)\end{gathered}}\,\beta}{\begin{gathered}(((\lambda f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}).\lambda x : \mathbb{N} \to \mathbb{N}.fn(fmx))G)I)d \to_v \\ ((\lambda x : \mathbb{N} \to \mathbb{N}.Gn(Gmx))I)d\end{gathered}}\,\texttt{CTX}_{((\bullet)I)d}$$

$$\cfrac{\cfrac{}{(\lambda x : \mathbb{N} \to \mathbb{N}.Gn(Gmx))I \to_v Gn(GmI)}\,\beta}{((\lambda x : \mathbb{N} \to \mathbb{N}.Gn(Gmx))I)d \to_v (Gn(GmI)d)}\,\texttt{CTX}_{(\bullet)d}$$

$$\cfrac{\cfrac{}{(\lambda n : \mathbb{N}.\lambda g : \mathbb{N} \to \mathbb{N}.\lambda x : \mathbb{N}.gn)n \to_v \lambda g : \mathbb{N} \to \mathbb{N}.\lambda x : \mathbb{N}.gn}\,\beta}{(Gn(GmI)d) \to_v ((\lambda g : \mathbb{N} \to \mathbb{N}.\lambda x : \mathbb{N}.gn)(GmI))d}\,\texttt{CTX}_{(\bullet(GmI))d}$$

$$\frac{\dfrac{}{(\lambda n:\mathbb{N}\lambda g:\mathbb{N}\to\mathbb{N}\lambda x:\mathbb{N}.gn)m \to_v \lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gm}\,\beta}{\begin{array}{c}((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(Gml))d \to_v \\ ((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gm)I))d\end{array}}\,\mathtt{CTX}_{((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(\bullet I))d}$$

$$\frac{\dfrac{}{(\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gm)I \to_v \lambda x:\mathbb{N}.Im}\,\beta}{\begin{array}{c}((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gm)I))d \to_v \\ ((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(\lambda x:\mathbb{N}.Im))d\end{array}}\,\mathtt{CTX}_{((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(\bullet))d}$$

$$\frac{\dfrac{}{(\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(\lambda x:\mathbb{N}.Im) \to_v \lambda x:\mathbb{N}.(\lambda x:\mathbb{N}Im)n}\,\beta}{\begin{array}{c}((\lambda g:\mathbb{N}\to\mathbb{N}.\lambda x:\mathbb{N}.gn)(\lambda x:\mathbb{N}.Im))d \to_v \\ (\lambda x:\mathbb{N}.(\lambda x:\mathbb{N}.Im)n)d\end{array}}\,\mathtt{CTX}_{(\bullet)d}$$

$$\frac{}{(\lambda x:\mathbb{N}.(\lambda x:\mathbb{N}.Im)n)d \to_v (\lambda x:\mathbb{N}.Im)n}\,\beta$$

$$\frac{}{(\lambda x:\mathbb{N}.Im)n \to_v Im}\,\beta$$

$$\frac{}{Im \equiv (\lambda x:\mathbb{N}.x)n \to_v m}\,\beta$$

(iv) an abstract *stack* datatype can be defined as follows:

$$\begin{aligned}\mathtt{pack}\ &\langle \mathtt{Stack}, \langle s_0, \langle \mathtt{push}, \langle \mathtt{peek}, \mathtt{pop}\rangle\rangle\rangle\rangle \\ &\ \mathtt{as} \\ &\exists \mathtt{stack}.\mathtt{stack} \times (\mathbb{N} \to \mathtt{stack} \to \mathtt{stack}\times \\ &(\mathtt{stack} \to \mathbb{N} \to \mathbb{N} \times \mathtt{stack} \to \mathtt{stack}))\end{aligned}$$

Check type of "pop"

Where:

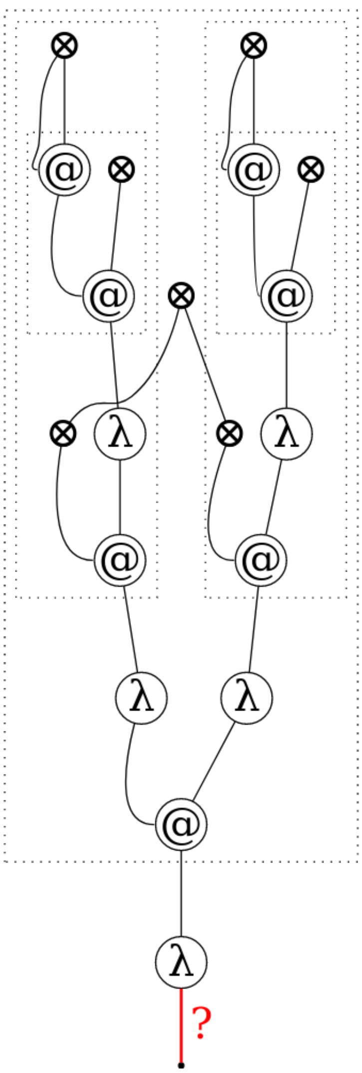
- $s_0 = \lambda\alpha.\lambda f:\mathbb{N}\to\alpha\to\alpha.\lambda x:\alpha.x$
- `Stack` $= \forall\alpha.(\mathbb{N}\to\alpha\to\alpha)\to\alpha\to\alpha$
- `push`, `peek` and `pop` are defined the same as in the exam booklet, over (concrete) the `Stack` type.

Question 2

$$Y = \lambda y.(\lambda f.t(\lambda z.ffz))(\lambda f.t(\lambda z.ffz))$$

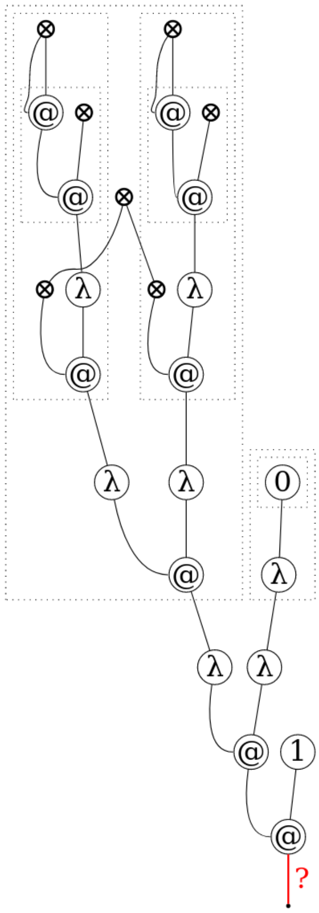
(a) The following ASG was generated using SPARTAN and the following code:

```
            LAMBDA(;t.APP(
        LAMBDA(; f.
          APP(t, LAMBDA(; z.
                          APP(APP(f,f),z))
              )
        ),
        LAMBDA(; f.
          APP(t, LAMBDA(; z.
                          APP(APP(f,f),z))
              )
        )
    ))
```

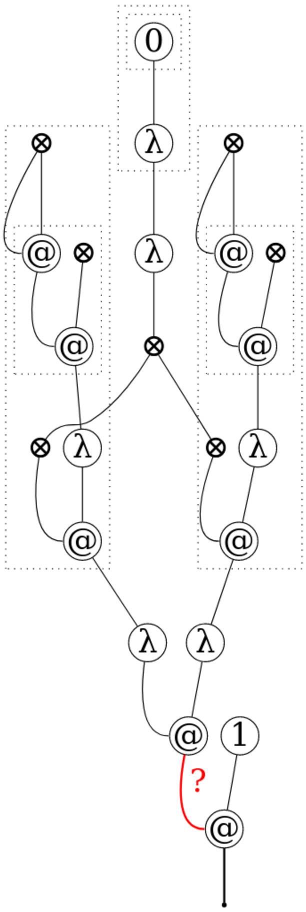


(b) (i)

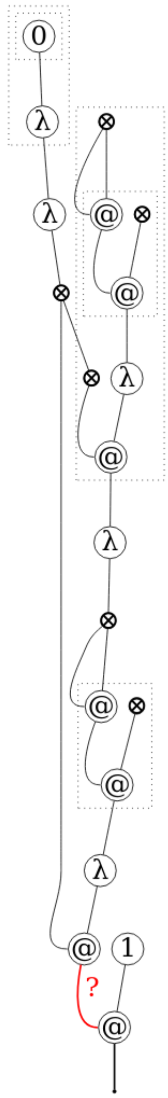
$$Y(\lambda f.\lambda x.0)1$$



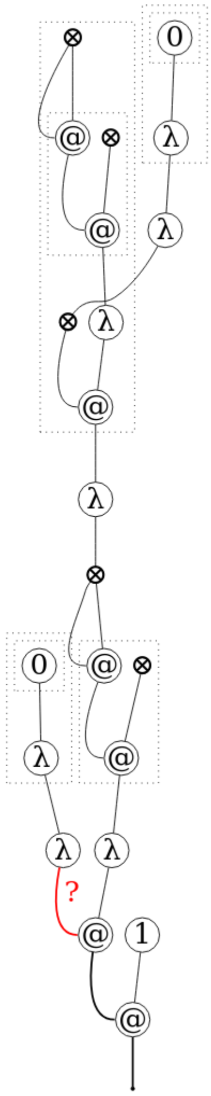
Firstly, the ASG evaluate the LHS, finding values on either side of the first application, it performs a reduction, replacing $\lambda t.$ with $\lambda f.\lambda x.0$:



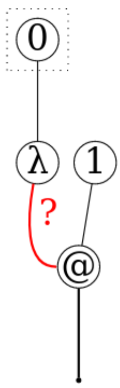
It goes on to attempt to further evaluate the LHS, again finding two values either side of an application, the LH thunk is expanded attaching the RH value as its parameter:



Next, the machine performs a rewrite of a shared reference of $\lambda f.\lambda x.0$:



Again finding two values on the LHS application, a reduction is performed, stripping the outer λ from $\lambda f.\lambda x.0$, discarding the Y combinator:



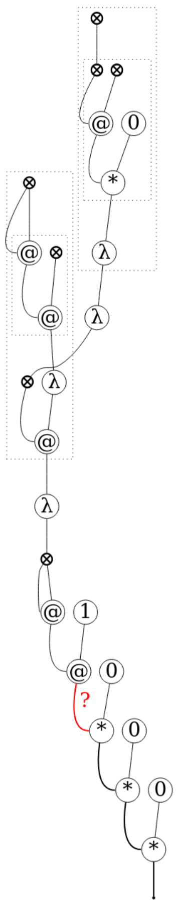
Finally, finding 2 values either side of our application the system removes the next λ, leaving just 0, our final result from this computation:

(ii)

$$Y(\lambda f.\lambda x.f(x) * 0)1$$

This expression will diverge. This occurs as the second operand of the $*$ operator (1) is is never evaluated as our Y combinator will infinitely expand over the first function argument $\lambda f.\lambda x.f(x) * 0$

After 91 steps of execution the ASG abstract machine will be in the following state:



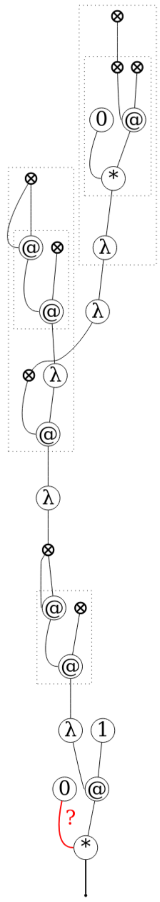
Intuitively, you can see that the second operand (1) is being pushed up as the Y combinator duplicates the operation and first operand (0) infinitely.

(iii) If we were to swap our operands to form an expression:

$$Y(\lambda f.\lambda x.0 * f(x))1$$

This would terminate, due to the nature of our $*$ *shortcut* operator which does not need to evaluate a second argument in the case where the first operand is 0.

After 26 steps of execution we reach the following state:



Here you can see that we have a state where we are attempting to evaluate $0 * @$, with a traditional (eager) multiplication operator, we would find a value on the left and an expression on the right, forcing us to evaluate the RHS until we reach a value. However, our *shortcut* operator allows us to instead skip this evaluate and return 0, eliminating the hang the Y combinator would otherwise cause.

Question 3

Add more questions if necessary.

Statement of good academic conduct
By submitting this assignment, I understand that I am agreeing to the following statement of good academic conduct.

- I confirm that this assignment is my own work and I have not worked with others in preparing this assignment.
- I confirm this assignment was written by me and is in my own words, except for any materials from published or other sources which are clearly indicated and acknowledged as such by appropriate referencing.
- I confirm that this work is not copied from any other person's work (published or unpublished), web site, book or other source, and has not previously been submitted for assessment either at the University of Birmingham or elsewhere.
- I confirm that I have not asked, or paid, others to prepare any part of this work for me.
- I confirm that I have read and understood the University regulations on plagiarism.