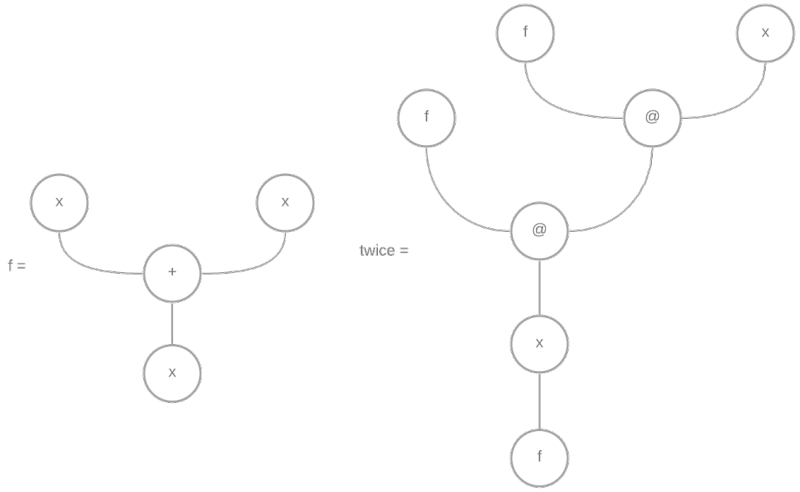
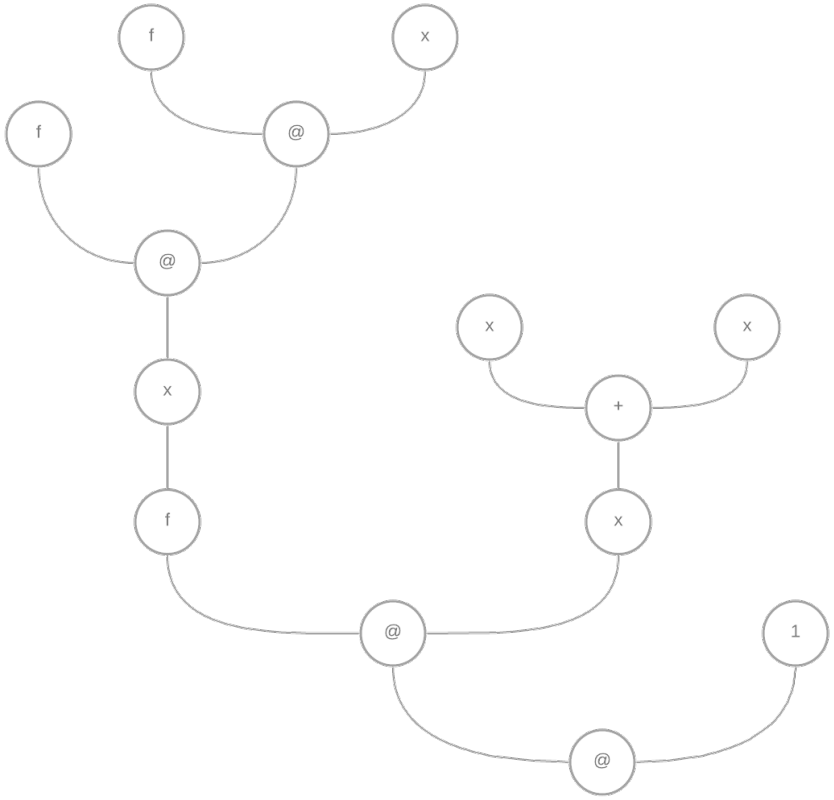
PLPDI Compilers: Assignment 2

Question A

```
func f x = x + x
func twice f x = f(f(x))
twice f 1
```

1. Draw the AST





twice f 1

2. Draw the ASG

I created the ASG of this program using the online ASG-based abstraction machine.

The following *SPARTAN* statement:

```
bind f = LAMBDA(;x. PLUS(x,x)) in
bind twice = LAMBDA(; f. LAMBDA(; x. APP(f,APP(f,x))))
in
APP(APP(twice,f),1)
```

Created the following ASG:

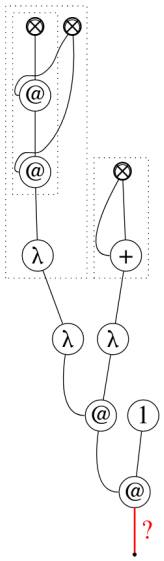
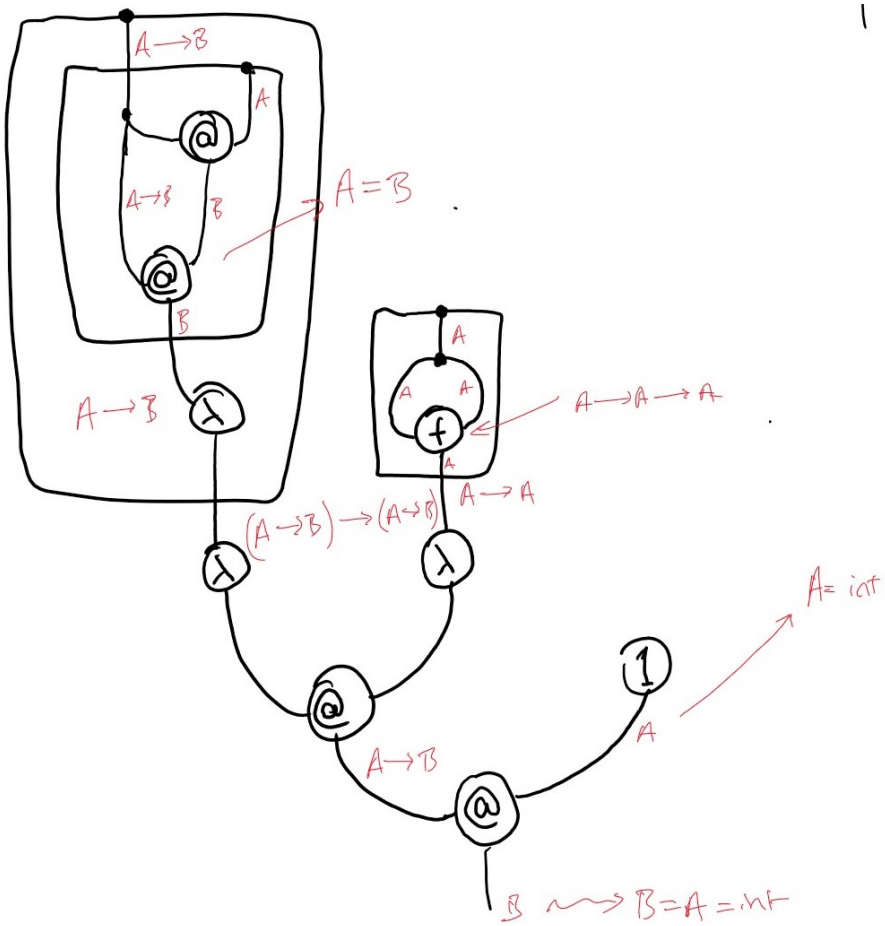


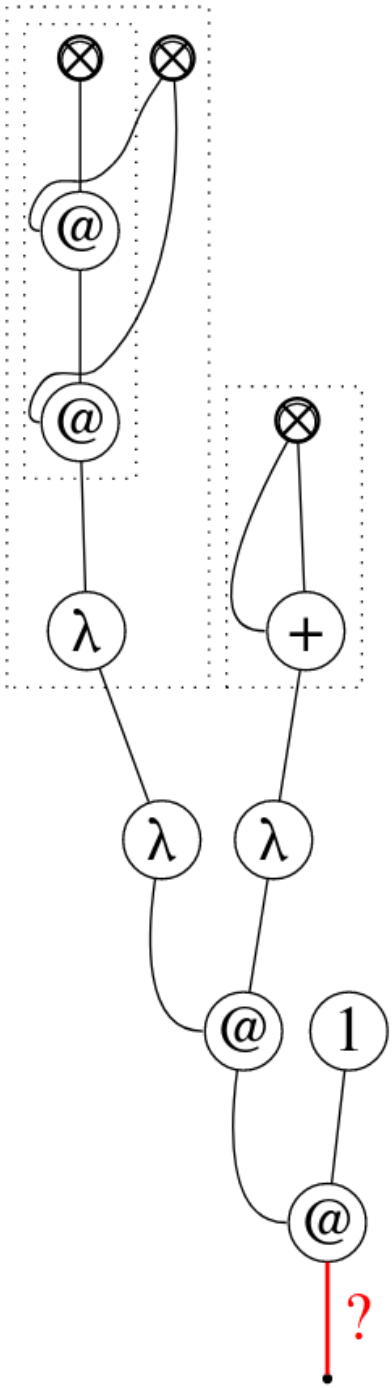
Figure 1: Initial ASG

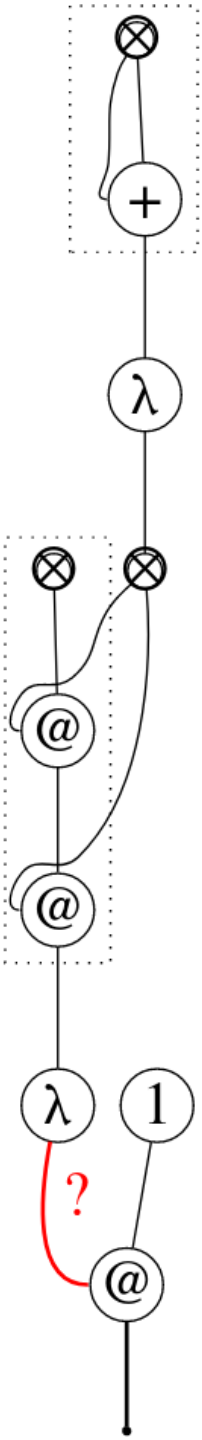
3. Perform type inference on this program

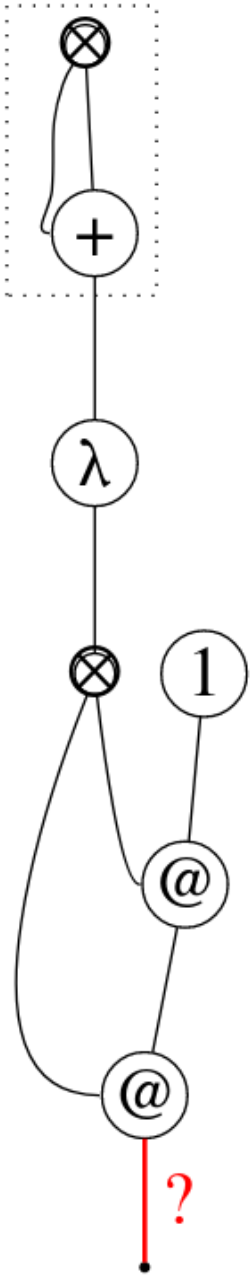


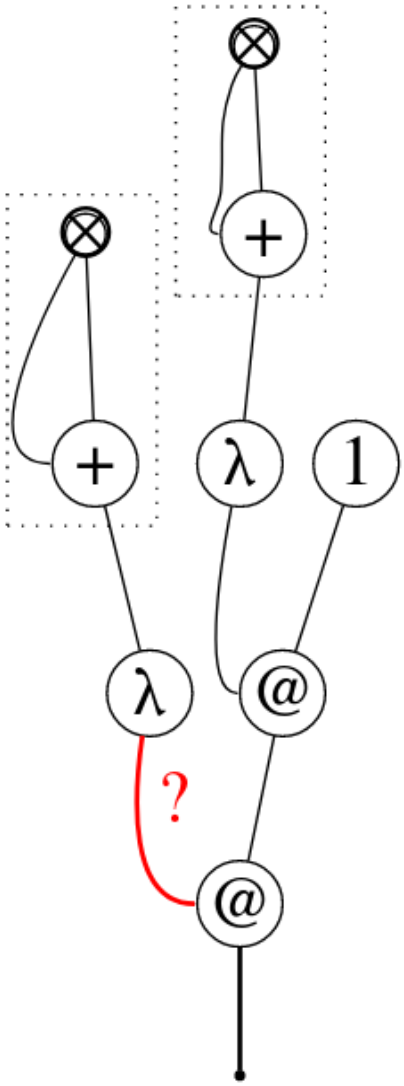
4. Draw the intermediate ASGs in the evaluation of this program

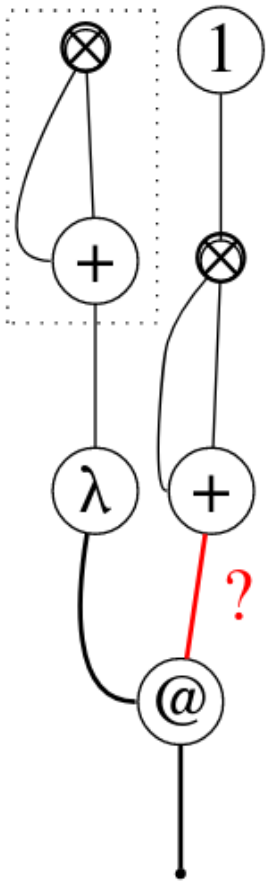
Note Diagrams read from left to right on the page

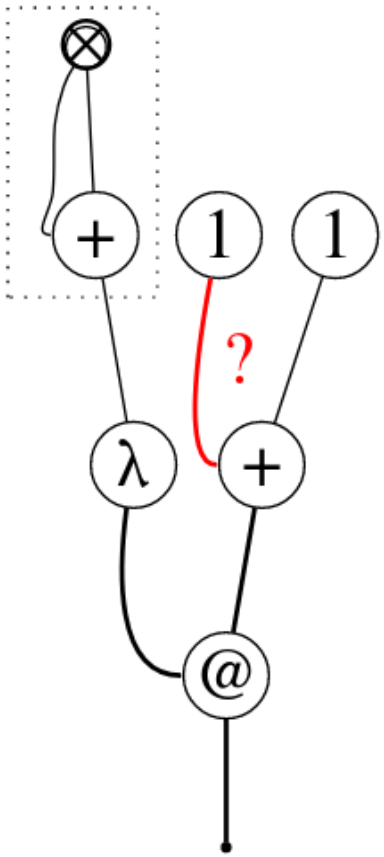

@
@
λ
+
λ
λ
@
1
@
?

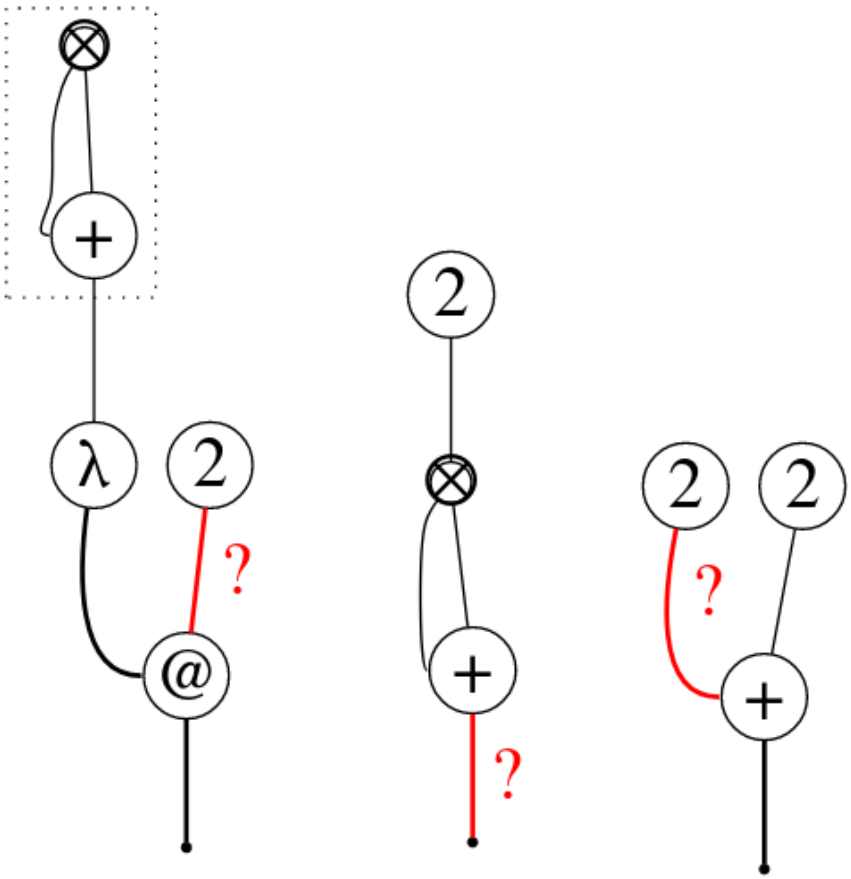

+
λ
@
@
λ
1
?
@


+
λ
1
@
@
?


+
+
λ
1
λ
@
?
@


1
+
λ
+
?
@


+
1
1
?
λ
+
@

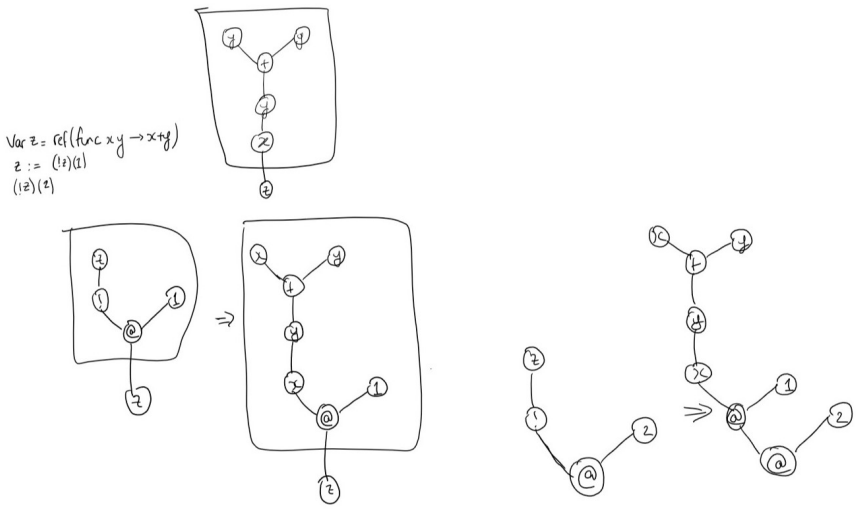

+
λ
2
?
@
2
?
+
2
2
?
+


4

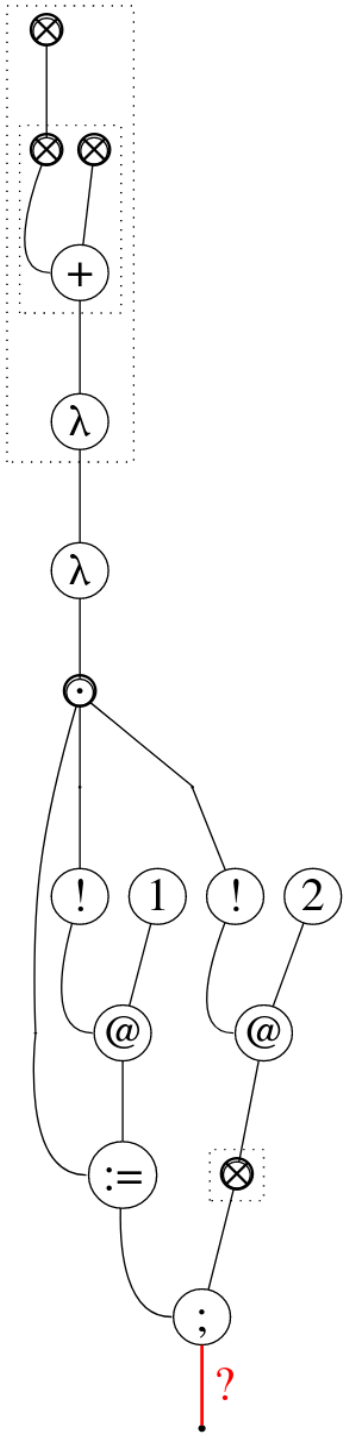
Question B

```
var z = ref(func x y -> x + y)
z := (!z)(1)
(!z)(2)
```

1. Draw the AST of this program

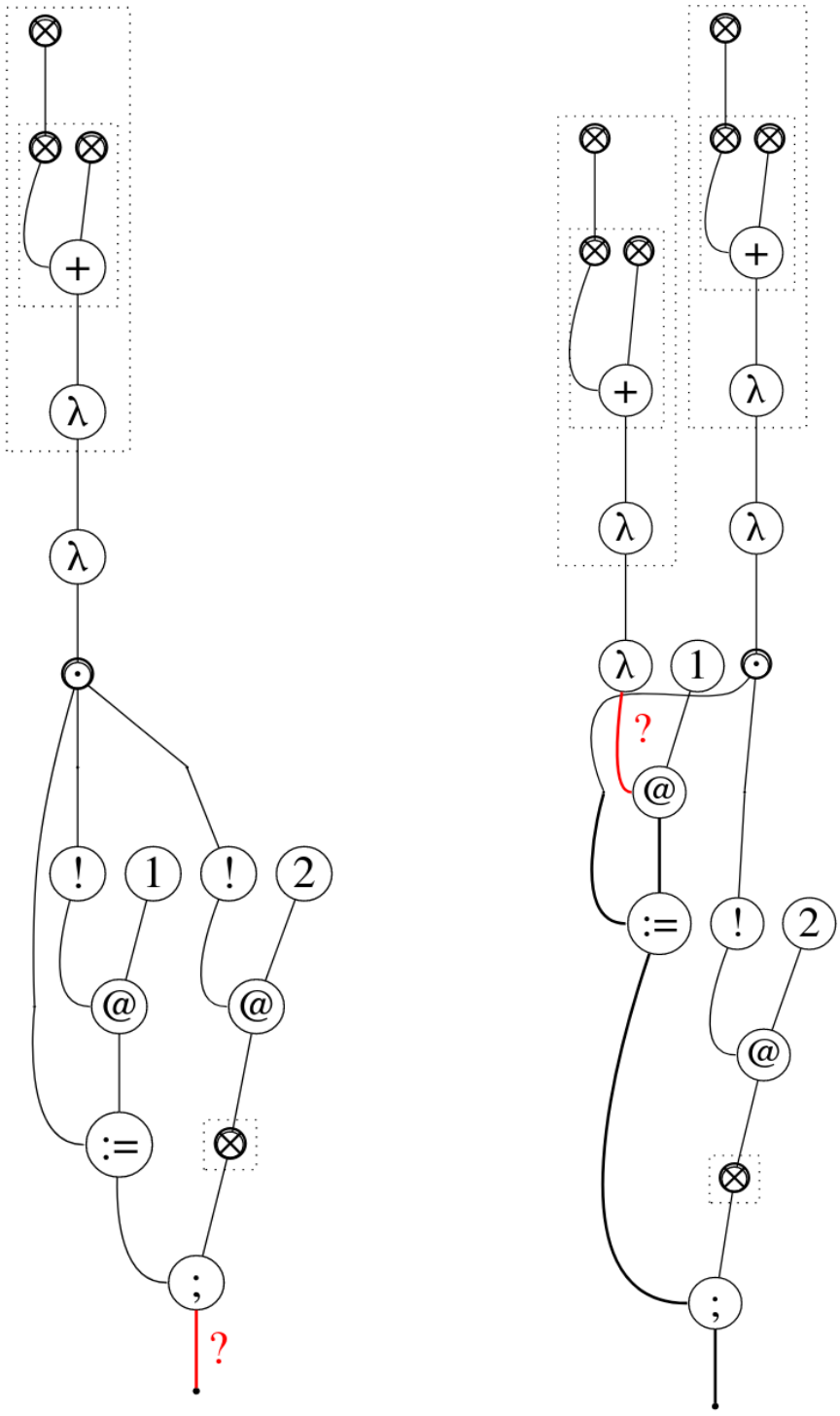


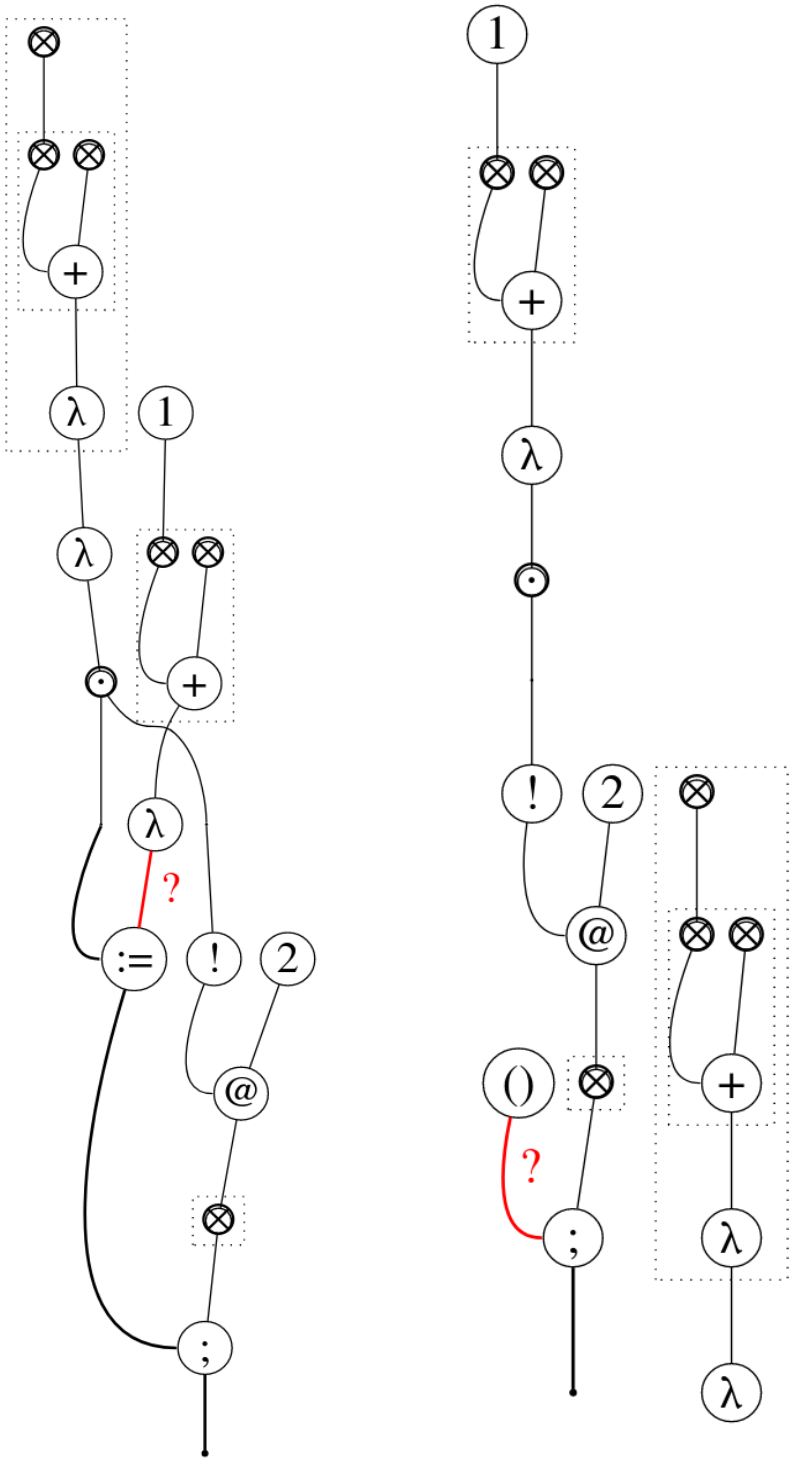
2. Draw the ASG of this program

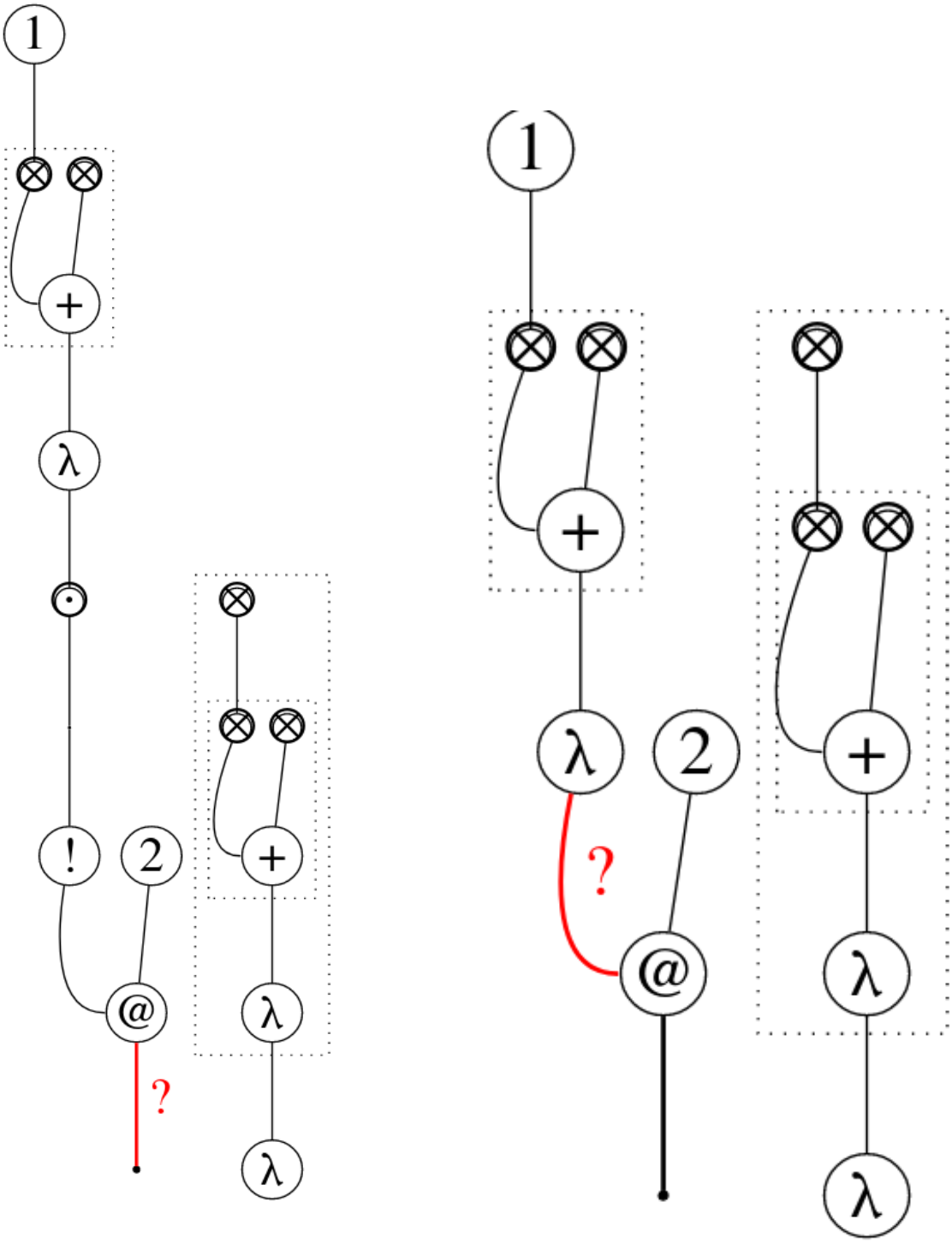

+
λ
λ
!
1
!
2
@
@
:=
;
?

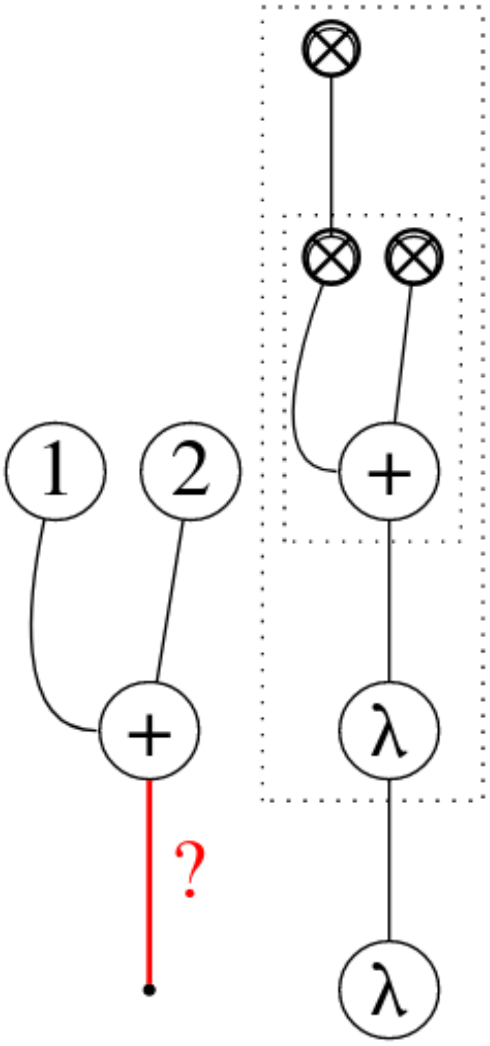
3. Draw the intermediate ASGs in the evaluation of this program

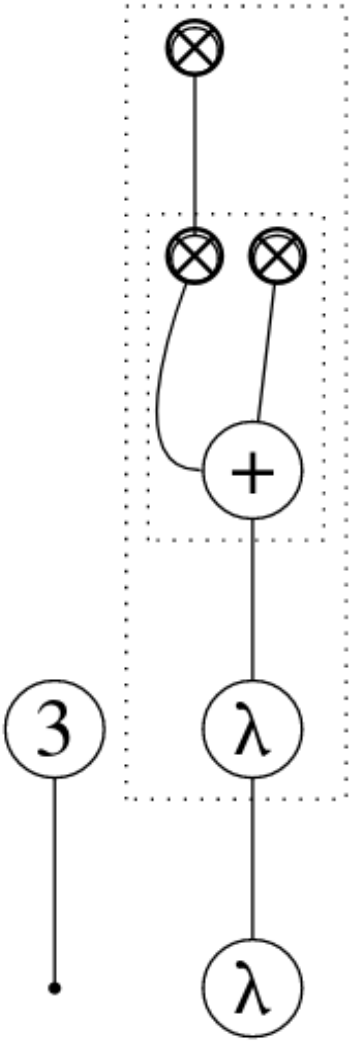
Note Diagrams read from left to right on the page


+
λ
λ
!
1
!
2
@
@
:=
;
?
+
λ
λ
λ
1
?
@
:=
!
2
@
;


1
+
λ
1
λ
+
λ
?
:=
!
2
@
;
1
+
λ
!
2
@
()
?
;
+
λ
λ


1
+
λ
!
2
@
?
+
λ
1
+
λ
2
?
@
+
λ


1
2
+
λ
?
λ


+
3
λ
λ

4. Perform type inference on this program

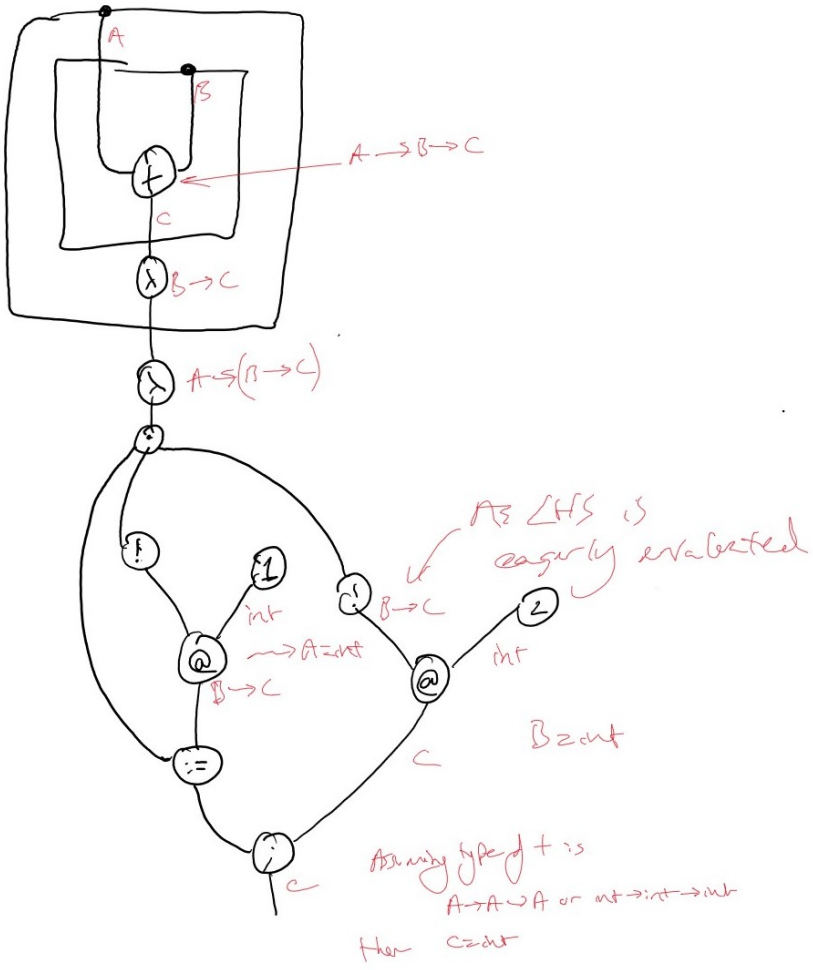


Figure 2: Type inference