INFSEN02-1 Sample exam

The INFDEV@HR Team

1 Exam

1.1 Question 1

Given the following lambda program, and a series of relevant delta rules, show the beta reductions for this program.

```
(TRUE ∨ FALSE)
```

1.2 Relevant delta rules

Boolean or:

```
(λp q→((p p) q))
```

True

```
(λt f→t)
```

False

```
(λt f→f)
```

1.3 Answer 1 (note: you do not need to write all this detail yourself, it is only included for completeness)

```
(TRUE ∨ FALSE)
```

```
((∨ TRUE) FALSE)
```

```
((λp q→((p p) q)) TRUE) FALSE)
```

```
(((λp q→((p p) q)) TRUE) FALSE)
```

```
(((λp q→((p p) q)) (λt f→t)) FALSE)
```

```
(((λp q→((p p) q)) (λt f→t)) FALSE)
```

```
(((λp q→((p p) q)) (λt f→t)) (λt f→f))
```

```
(((λp q→((p p) q)) (λt f→t)) (λt f→f))
```

```
((λq→(( (λt f→t) (λt f→t)) q)) (λt f→f))
```

```
((λq→(((λt f→t) (λt f→t)) q)) (λt f→f))
```

```
(((λt f→t) (λt f→t)) (λt f→f))
```

```
(((λt f→t) (λt f→t)) (λt f→f))
```

```
((λf t f→t) (λt f→f))
```

```
((λf t f→t) (λt f→f))
```

```
(λt f→t)
```

```
(λt f→t)
```

```
TRUE
```

1.4 Question 2

Given the following lambda calculus program, and a series of relevant delta rules, give the full typing derivation for the program.

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

1.5 Relevant delta rules

Boolean type:

```
(∀α ⇒(α→α→α))
```

1.6 Answer 2 (note: you do not need to write all this detail yourself, it is only included for completeness)

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean) Boolean) q))
```

```
(λ(p:Boolean)(q:Boolean)→(((Boolean Boolean) Boolean) q))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean) Boolean)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean) Boolean)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α)) Boolean)
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α)) Boolean) Boolean
    ) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α)) ) Boolean) Boolean))
```

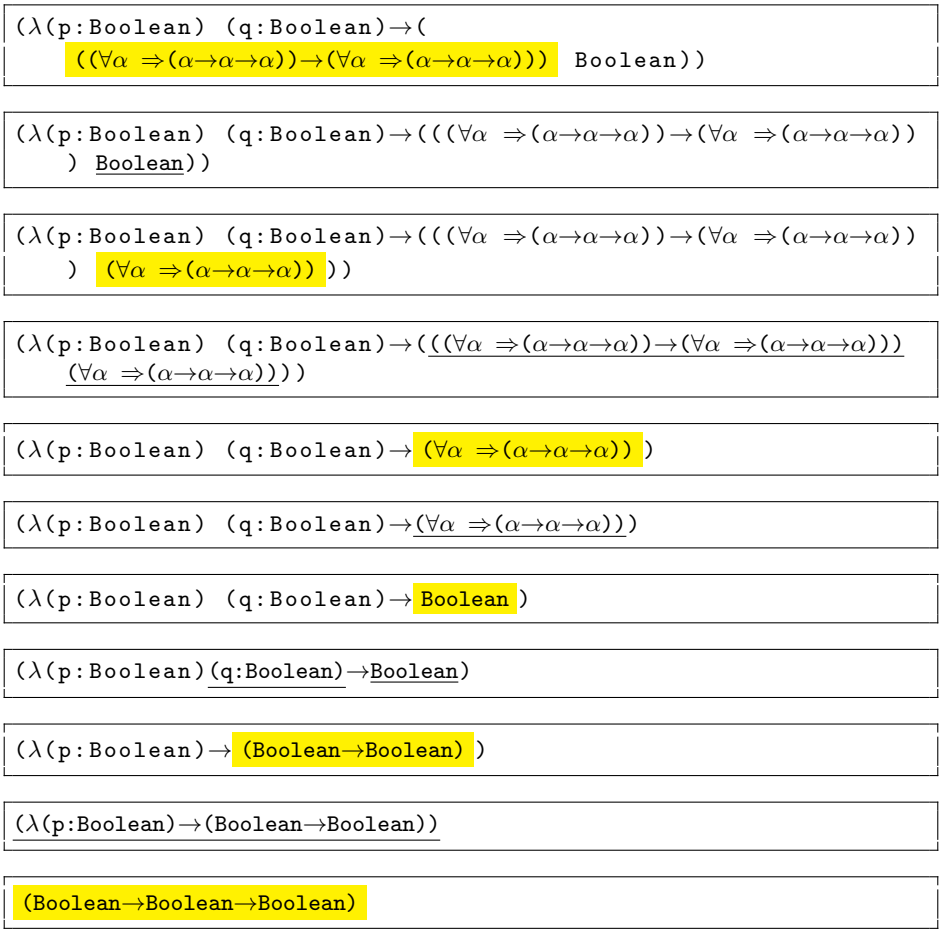
```
(λ(p:Boolean) (q:Boolean)→(((((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α)))
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean
    ) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    )→(∀α ⇒(α→α→α))) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    )→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α)) ) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    )→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α))) Boolean))
```


(λ(p:Boolean) (q:Boolean)→(
((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean))
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))
) Boolean))
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))
) (∀α ⇒(α→α→α))))
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)))
(∀α ⇒(α→α→α))))
(λ(p:Boolean) (q:Boolean)→ (∀α ⇒(α→α→α)))
(λ(p:Boolean) (q:Boolean)→(∀α ⇒(α→α→α)))
(λ(p:Boolean) (q:Boolean)→ Boolean)
(λ(p:Boolean)(q:Boolean)→Boolean)
(λ(p:Boolean)→ (Boolean→Boolean))
(λ(p:Boolean)→(Boolean→Boolean))
(Boolean→Boolean→Boolean)