

Ejercicio 2.

La función *infer* retorna un valor de tipo *Either String Type* y no un valor de tipo *Type* porque de esta forma cuando la expresión que se analiza tipa se devuelve un *Type*, representando el tipo inferido, pero si no tipa puede devolver un mensaje de error.

La función *>>=* recibe un valor *v* (de tipo *Either String Type*) y una función *f* (que toma un valor de tipo *Type* y devuelve uno de tipo *Either String Type*). Y hace lo siguiente:

Si *v* es un *Left s* devuelve *v*. Y así, se propaga el error.

Si *v* es un *Right t* devuelve el valor que se obtiene de aplicar *f* a *t*.

Ejercicio 11.

Tenemos:

$Ack : N \rightarrow N \rightarrow N$

$Ack\ 0\ n = n + 1$

$Ack\ m\ 0 = Ack\ (m - 1)\ 1$

$Ack\ m\ n = Ack\ (m - 1)\ (Ack\ m\ (n - 1))$

Ahora definimos una función equivalente:

$Ack' : N \rightarrow N \rightarrow N$

$Ack'\ 0 = succ$

$Ack'\ m = Iter\ (Ack'\ (m-1))$

$Iter : (N \rightarrow N) \rightarrow N \rightarrow N$

$Iter\ (f)\ 0 = f(1)$

$Iter\ (f)\ n = f(Iter(f)\ (n-1))$

ACK.lam

```
def Iter = \f:Nat -> Nat . \n:Nat. R (f (suc 0)) (\g:Nat->Nat.\x:Nat. f g) n
```

```
def ack = \m:Nat. (R (\n:Nat. suc n) (\g:Nat->Nat.\x:Nat. \n:Nat. Iter g n ) m)
```