Tarea 3:
Lenguajes de Programación

Araujo Chavez Mauricio
312210047

Carmona Mendoza Martín
313075977

1. Escribe las reglas de tipado (semántica estática) para cada una de las expresiones anteriores.

Considera la gramática:

$$e \ ::= \ x|n|true|false|e+e|if\ e\ then\ e\ else\ e|iszero\ e|let\ x=e\ in\ e\ end|e<e|e=e|\neg e$$

con $n \in \mathbb{Z}$. La extensión al paradigma imperativo se hace de la siguiente manera:

$$l_n|e_1 \ := \ e_2|ref\ e|!e|e_1;e_2|while\ e_1\ do\ e_2|()$$

- x

$$\frac{}{\Gamma, x : T \vdash x : T}$$

- n

$$\frac{}{\Gamma \vdash num[n] : Nat}$$

- $true$

$$\frac{}{\Gamma \vdash bool[true] : Bool}$$

- $false$

$$\frac{}{\Gamma \vdash bool[false] : Bool}$$

- $e_1 + e_2$

$$\frac{\Gamma \vdash e_1 : Nat \quad \Gamma \vdash e_2 : Nat}{\Gamma \vdash e_1 + e_2 : Nat}$$

- $if\ e_1\ then\ e_2\ else\ e_3$

$$\frac{\Gamma \vdash e_1 : Bool \ \ \Gamma \vdash e_2 : T \ \ \Gamma \vdash e_3 : T}{\Gamma \vdash if(e_1, e_2, e_3) : T}$$

- $iszero\ e$

$$\frac{\Gamma \vdash e : Nat}{\Gamma \vdash iszero\ e : Bool}$$

- $let\ x = e_1\ in\ e_2\ end$

$$\frac{\Gamma \vdash e_1 : T \quad \Gamma, x : T \vdash e_2 : S}{\Gamma \vdash let(e_1, x.e_2) : S}$$

- $e_1 < e_2$

$$\frac{\Gamma \vdash e_1 : Nat \quad \Gamma \vdash e_2 : Nat}{\Gamma \vdash e_1 < e_2 : Bool}$$

- $e_1 = e_2$

$$\frac{\Gamma \vdash e_1 : T \quad \Gamma, e_1 : T \vdash e_2 : T}{\Gamma \vdash e_1 = e_2 : T}$$

- $\neg\ e$

$$\frac{\Gamma \vdash e : Bool}{\Gamma \vdash \neg\ e : Bool}$$

- l_n

$$\frac{\Sigma(l) = T}{\Gamma \mid \Sigma \vdash l : Ref\ T}$$

- $e_1 := \ e_2$

$$\frac{\Gamma \mid \Sigma \vdash e_1 : Ref\ T \quad \Gamma \mid \Sigma \vdash e_2 : T}{\Gamma \mid \Sigma \vdash e_1 := e_2 : Void}$$

- $ref\ e$

$$\frac{\Gamma \mid \Sigma \vdash e : T}{\Gamma \mid \Sigma \vdash ref\ e : Ref\ T}$$

- $!e$

$$\frac{\Gamma \mid \Sigma \vdash e : Ref\ T}{\Gamma \mid \Sigma \vdash !e : T}$$

- $e_1; e_2$

$$\frac{\Gamma \vdash e_1 : Void \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1; e_2 : T}$$

- *while* e_1 *do* e_2

$$\frac{\Gamma \mid \Sigma \vdash e_1 : Bool \quad \Gamma \mid \Sigma \vdash e_2 : Void}{\Gamma \mid \Sigma \vdash e_1 : while(e_1, e_2) : Void}$$

- $()$

$$\frac{}{\Gamma \mid \Sigma \vdash () : Void}$$

2. Considera los siguientes programas. Escribe la derivación de tipos de los tres programas descritos.

- $p_1 \rightleftharpoons$

```
let x = ref (iszero(3+4)) in
        let y = ref (if !x then 3 else 4) in
                let z = if !y < 10 then !y + 6 else 7+!y
        end
end
```

Denotamos e como let y = ref (if !x then 3 else 4) in let z = if !y ¡ 10 then !y + 6 else 7+!y end

$$\emptyset \mid \emptyset \vdash let\ x = ref(iszero(3+4))\ in\ e\ end : S$$

$$\emptyset \mid \emptyset \vdash ref(iszero(3+4)) : Ref\ Bool$$
$$\emptyset \mid \emptyset \vdash iszero(3+4) : Bool$$
$$\emptyset \mid \emptyset \vdash 3+4 : Nat$$
$$\emptyset \mid \emptyset \vdash 3 : Nat$$
$$\emptyset \mid \emptyset \vdash 4 : Nat$$
$$x : Bool \mid \emptyset \vdash e : S$$

Equivalentemente definimos e_1 como let z = if !y < 10 then !y + 6 else 7+!y

$$x : Bool \mid \emptyset \vdash let\ y\ =\ ref\ (if!xthen3else4)\ in\ e_1\ end : S$$

$$x : Bool \mid \emptyset \vdash ref\ (if!xthen3else4) : Ref\ Nat$$
$$x : Bool \mid \emptyset \vdash if\ !x\ then\ 3\ else\ 4 : Nat$$
$$x : Bool \mid \emptyset \vdash !x : Bool$$
$$x : Bool \mid l_x : Bool \vdash l_x : Ref\ Bool$$
$$x : Bool \mid l_x : Ref\ Bool \vdash l_x : Bool$$
$$x : Bool \mid l_x : Bool \vdash 3 : Nat$$
$$x : Bool \mid l_x : Bool \vdash 4 : Nat$$

Continuamos con e_1 y suponemos que si tiene end para ejecutar bien el tipado.
e_1 = let z = if !y < 10 then !y + 6 else 7+!y end

$$x : Bool, y : Nat \mid \emptyset \vdash let\ z\ =\ if\ !y\ <\ 10\ then\ !y\ +\ 6\ else\ 7{+}!y\ end : S$$

$$x : Bool, y : Nat \mid \emptyset \vdash !y : Nat$$
$$x : Bool, y : Nat \mid l_y : Nat \vdash l_y : Ref\ Nat$$

$$x : Bool, y : Nat \mid l_y : Ref\ Nat \vdash l_y : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash 10 : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash 6 : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash l_y\ <\ 10 : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash l_y\ +\ 6 : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash 7 : Nat$$

$$x : Bool, y : Nat \mid l_y : Nat \vdash 7\ +\ l_y : Nat$$

$$x : Bool, y : Nat, z : Nat \mid \emptyset \vdash \emptyset$$

- $p_2 \rightleftharpoons$

```
let z = ref 5 in
        let w = ref 3 in
            while (0 < !w)
                z := 5+3;
                w := !w-1
        end
        !z
end
```

Definimos e como let w = ref 3 in while (0 < !w) z := 5+3; w := !w-1 end

$$\emptyset \mid \emptyset \vdash let\ z\ =\ ref\ 5\ in\ e\ !z\ end : S$$

$$\emptyset \mid \emptyset \vdash ref\ 5 : Nat$$

$$\emptyset \mid \emptyset \vdash 5 : Nat$$

$$z : Nat \mid \emptyset \vdash e : S$$

Definimos e_1 como while (0 < !w) z := 5+3; w := !w-1

$$z : Nat \mid \emptyset \vdash let\ w\ ref\ 3\ in\ e_1\ end\ !z : S$$

$$z : Nat \mid \emptyset \vdash ref\ 3 : Nat$$

$$z : Nat \mid \emptyset \vdash 3 : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash e_1 : S$$

$$z : Nat, w : Nat \mid \emptyset \vdash while\ (0\ < !w)\ z\ :=\ 5 + 3;\ w\ := !w - 1 end : S$$

$$z : Nat, w : Nat \mid \emptyset \vdash (0\ < !w) : Bool$$

$$z : Nat, w : Nat \mid \emptyset \vdash l_w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash l_w : Ref\ Nat$$

$$z : Nat, w : Nat \mid l_w : Ref\ Nat \vdash l_w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 0 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z\ :=\ 5 + 3;\ w\ := !w - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z\ :=\ 5 + 3 : Void$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 5 + 3 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 5 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 3 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash w\ := !w - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash !w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 1 : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash !z : Nat$$

$$z : Nat, w : Nat \mid l_z : Nat \vdash l_z : Ref\ Nat$$

$$z : Nat, w : Nat \mid l_z : Ref\ Nat \vdash l_z : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash \emptyset$$

- $p_3 \rightleftharpoons$

```
let z = ref 10 in
     let w = ref 7 in
          while (0 < !w)
               z := !z-1;
               w := !w-1
     end
     if !z = 3 then true else false
end
```

Definimos e como let w = ref 7 in while (0 < !w) z := !z-1; w := !w-1 end

$$\emptyset \mid \emptyset \vdash let\ z\ =\ ref\ 10\ in\ e\ if\ !z\ =\ 3\ then\ true\ else\ false\ end : S$$

$$\emptyset \mid \emptyset \vdash ref\ 10 : Nat$$

$$\emptyset \mid \emptyset \vdash 10 : Nat$$

$$z : Nat \mid \emptyset \vdash e : S$$

Definimos e_1 como while (0 < !w) z := !z-1; w := !w-1

$$z : Nat \mid \emptyset \vdash let\ w\ =\ ref\ 7\ in\ e_1\ end$$

$$z : Nat \mid \emptyset \vdash ref\ 7 : Nat$$

$$z : Nat \mid \emptyset \vdash 7 : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash e_1 : S$$

$$z : Nat, w : Nat \mid \emptyset \vdash while\ (0\ < !w)\ z\ := !z - 1;\ w\ := !w - 1\ end : S$$

$$z : Nat, w : Nat \mid \emptyset \vdash (0\ < !w) : Bool$$

$$z : Nat, w : Nat \mid \emptyset \vdash 0 : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash !w : Nat$$

$$z : Nat, w : Nat \mid \emptyset \vdash l_w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z\ := !z - 1; w\ := !w - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z\ := !z - 1 : Void$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z : Ref\ Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash !z - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash !z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat \vdash l_z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash w\ := !w - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash w : Ref\ Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash !w - 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash 1 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash !w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash l_w : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash if\ !z\ =\ 3\ then\ true\ else\ false\ end$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash !z\ =\ 3 : Bool$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash !z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash l_z : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash 3 : Nat$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash true : Bool$$

$$z : Nat, w : Nat \mid l_w : Nat, l_z : Nat \vdash false : Bool$$

$$z : Nat, w : Nat \mid \emptyset \vdash \emptyset$$

3. Define listas ligadas como estructuras de datos efímeras y define una función que obtenga el último elemento de dichas listas (puedes combinar sintaxis de Haskell como se vio en clase).

data Tlist = Void | Nodo (Elem, Ref Tlist)

–Función
getLast :: Ref Tlis → Elem
getLast lista = case !(lista) of
Void → error "Lista vacia"
Nodo(Elem, Ref TList) → if !(RefList) == Void
then Elem
else getLast (Ref TList)