

# Compiladores, 2024-1

## Práctica 6: Análisis Semántico

Manuel Soto Romero  
manu@ciencias.unam.mx

Javier Enríquez Mendoza  
javieem@ciencias.unam.mx

Pedro Ulises Cervantes González  
confundeme@ciencias.unam.mx

Braulio Aaron Santiago Carrillo  
braulioa124@ciencias.unam.mx

16 de Noviembre del 2023  
Fecha de entrega: 30 de Noviembre del 2023

### 1 Introducción

Considera el siguiente lenguaje conocido como WHILE:

$Comp\ C ::= L := E \mid \text{if } B \text{ then } C \text{ else } C \mid C; C \mid \text{while } B \text{ do } C \mid \text{skip}$   
 $Bool\ B ::= \text{true} \mid \text{false} \mid E = E \mid B \& B \mid \neg B$   
 $Arith\ E ::= L \mid n \mid E + E$

El objetivo de esta practica es profundizar en el proceso de análisis semántico. La práctica consiste en definir las siguientes funciones:

### 2 Análisis Semántico

Considera los siguientes tipos y sus juicios:

$$\begin{array}{c} \frac{}{L : Number} \quad \frac{}{n : Number} \quad \frac{}{true : Boolean} \quad \frac{}{false : Boolean} \quad \frac{}{skip : Void} \\[10pt] \frac{a_0 : Number \quad a_1 : Number}{a_0 + a_1 : Number} \\[10pt] \frac{b_0 : Boolean \quad b_1 : Boolean}{b_0 \& b_1 : Boolean} \quad \frac{b_0 : Boolean}{\neg b_0 : Boolean} \quad \frac{a_0 : Number \quad a_1 : Number}{a_0 = a_1 : Boolean} \\[10pt] \frac{a : Number}{L ::= a : Void} \quad \frac{b : Boolean \quad c_0 : Void \quad c_1 : Void}{\text{if } b \text{ then } c_0 \text{ else } c_1 : Void} \quad \frac{c_0 : Void \quad c_1 : Void}{c_0 ; c_1 : Void} \\[10pt] \frac{b : Boolean \quad c : Void}{\text{while } b \text{ do } c : Void} \end{array}$$

```
data ASA = Assign ASA ASA | IfThenElse ASA ASA ASA | Seq ASA ASA | WhileDo C C | Skip
         | Boolean Bool | Equal ASA ASA | And ASA ASA | Not ASA
         | Loc Int | Number Int | Sum ASA ASA deriving Show

data Type = Num | Bool | Void deriving Show
```

9 pts Define la función **TypeCheckerAux** que recibe un **ASA** y devuelve el tipo de la expresión únicamente si el tipado del programa es consistente. En otro caso arroja un error indicando el problema con el programa.

```

typeCheckerAux :: ASA -> Type
{ - Ejemplo -}
> typeCheckerAux (Sum (Loc 2) (Number 5))
> Num

> typeCheckerAux (IfThenElse (Number 2) (Not (Number 5)) (Skip))
> El tipo de (Number 2) no es el esperado

```

- 1 pto Define la función **TypeChecker** que recibe un **ASA** y devuelve dicho **ASA** si el tipado del programa es consistente y el programa es admisible.

```

typeChecker :: ASA -> ASA
{ - Ejemplo -}
> typeCheckerAux (Sum (Loc 2) (Number 5))
> error: Las expresiones aritmeticas no son programas validos en el lenguaje

> typeCheckerAux (IfThenElse (Number 2) (Not (Number 5)) (Skip))
> error: El tipo de (Number 2) no es el esperado

```

### 3 Entrega

- La practica será entregada en equipos de máximo 5 integrantes.
- La entrega será por Google Classroom.
- Únicamente un miembro del equipo sube la solución de la practica. El resto debe indicar los integrantes de su equipo en un comentario privado.
- Únicamente anexar el archivo con extensión **.hs** con la solución. El nombre del archivo debe ser el nombre del integrante que subió el archivo empezando por apellidos.
- Deberás entregar el ejercicio a más tardar a las 23:59 del día indicado. Después de esta hora, el ayudante rechazará el ejercicio.