

# Compiladores, 2024-1

## Práctica 4: Análisis Sintáctico

Manuel Soto Romero  
manu@ciencias.unam.mx

Javier Enríquez Mendoza  
javciem@ciencias.unam.mx

Pedro Ulises Cervantes González  
confundeme@ciencias.unam.mx

Braulio Aaron Santiago Carrillo  
braulioa124@ciencias.unam.mx

3 de Octubre del 2023  
Fecha de entrega: 17 de Octubre del 2023

### 1 Introducción

Considera el siguiente lenguaje conocido como WHILE:

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

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

### 2 Análisis Sintáctico

Considera la definición de Tokens de la practica 3 y la siguiente definiciones:

```
data Content = T Token | S | C | B | E deriving Show

type Input = [Token]
type Stack = [Content]
```

4.9 pts Define la función **parserAux** que recibe una lista de tokens, el stack y devuelve verdadero si y solo si la lista de tokens pertenece al lenguaje con el stack actual. Esta función es un parser LL(1) ad hoc del lenguaje.

```
parserAux :: Input -> Stack -> Bool
{ - Ejemplo -}
> parserAux [LP, Loc 2, Assign, Number 1, Seq, LP, Loc 3, Assign, Number
  0, Seq, While, Not, Loc 2, Equal, Loc 2, Do, LP, Loc 2, Assign, LP, Loc 2, Sum, Number
  1, RP, Seq, Loc 3, Assign, LP, Loc 3, Sum, Number 1, RP, RP, RP, RP] [S]
> True

> parserAux [If, Not, And, Boolean True, Boolean False, Then, Skip, Else, Skip] [S]
> True
```

0.1 pts Utilizando la función **parserAux**, define la función **parser** que recibe una lista de tokens **WHILE** y devuelve verdadero si y solo si la lista de tokens pertenece al lenguaje.

```

parser :: Input -> Bool
{ - Ejemplo -}
> parser [LP,Loc 2,Assign,Number 1,Seq,LP,Loc 3,Assign,Number 0,Seq,While,Not,Loc
  2,Equal,Loc 2,Do,LP,Loc 2,Assign,LP,Loc 2,Sum,Number 1,RP,Seq,Loc
  3,Assign,LP,Loc 3,Sum,Number 1,RP,RP,RP,RP]
> True

> parser parser [If, Not, And, Boolean True, Boolean False, Then, Skip, Else, Skip]
> True

```

Considera la definición de Tokens de la practica 3 y la siguiente definiciones:

```

data C = AssignASA E E | IfThenElse B C C | SeqASA C C | WhileDo B C | SkipASA deriving
  Show

data B = BoolASA Bool | EqualASA E E | AndASA B B | NotASA B deriving Show

data E = LocASA Int | NumberASA Int | SumASA E E deriving Show

```

5 pts Utilizando la herramienta **Happy**, vuelve a definir un **parser** que recibe una cadena del lenguaje **WHILE** y devuelve su árbol de sintaxis abstracta.

### 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. Indicar los integrantes del equipo como encabezado en el archivo con la solución.
- 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.