

# Lenguajes de Programación

## Tarea 2

Karla Ramírez Pulido Alan Alexis Martínez López

Semestre 2023-2 **Fecha de inicio:** 23 de febrero 2023  
Facultad de Ciencias UNAM **Fecha de entrega:** 02 de marzo 2023

### Integrantes:

Dania Paula Gongora Ramírez  
Salgado Tirado Diana Laura

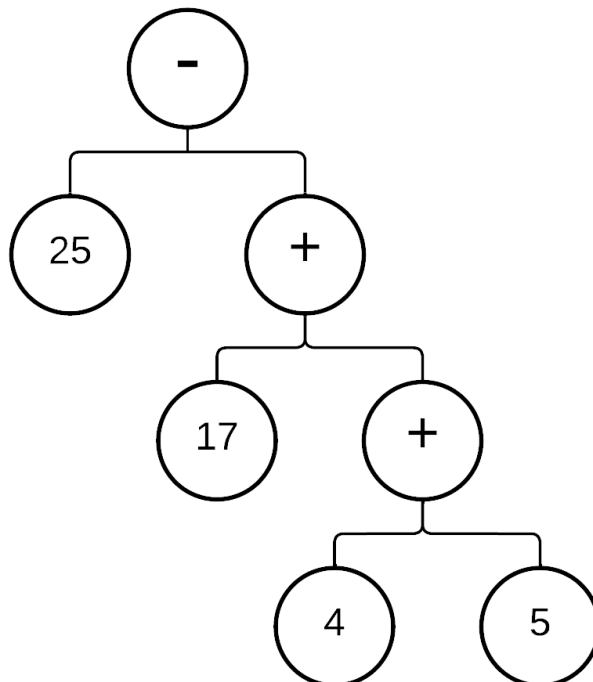
### Instrucciones

Resolver los siguientes ejercicios de forma clara y ordenada de acuerdo a los lineamientos de entrega de tareas disponibles en la página del curso.

### Ejercicios

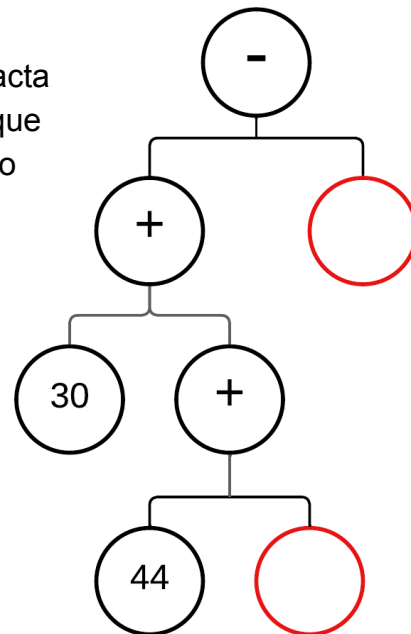
1. Dadas las siguientes expresiones en la gramática *WAE* dispuesta en sintaxis concreta, da la respectiva representación utilizando sintaxis abstracta por medio de los Árboles de Sintaxis Abstracta (ASA) correspondientes. En caso de no poder generar el árbol, justifica.

a.  $\{-\ 25\{+\ 17\{+\ 4\ 5\}\}\}$

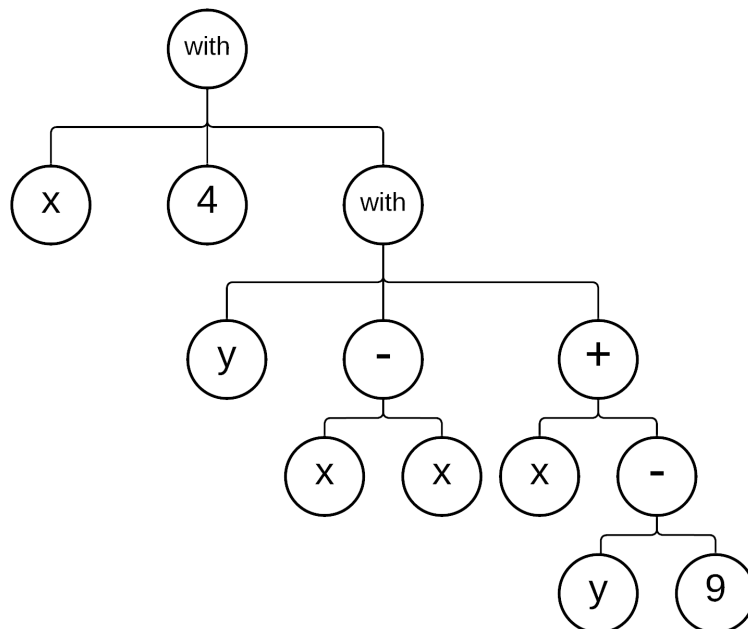


b.  $\{- \{+ 30 \{+ 44 \} \} \}$

No es posible construir un árbol de sintaxis abstracta ya que no es sintácticamente correcto, notemos que la suma tiene el lado derecho vacío por lo tanto no cumple con las reglas de la gramática WAE.



c.  $\{with \{x 4\}$   
 $\{with \{y \{- x x \} \}$   
 $\{+ x \{- y 9 \} \} \}$



2. Dadas las siguientes expresiones en la gramática *WAE* dispuesta en sintaxis concreta, da la sintaxis abstracta correspondiente y realiza la sustitución que se indica.

a.  $e = \{+ a \{+ b \{- 32 57\} \} \}$   
 (subst (parse e) 'a (add (num 3) (num 4)))

**Sintaxis abstracta.**

(add (id 'a) (add (id 'b) (sub(num -32)(num 57)) ) )

**Parse.**

(subst (parse e) 'a (add (num 3) (num 4))) con  $e = \{+ a \{+ b \{- 32 57\} \} \}$

subst (parse  $\{+ a \{+ b \{- 32 57\} \} \}$ ) 'a (add (num 3) (num 4)))

subst(add {parse a}{parse + b {-32 57} } ) 'a (add (num 3) (num 4)))

subst (add (id 'a){add {parse b}{parse {-32 57} } } 'a (add (num 3) (num 4)))

subst(add (id 'a)(add (id 'b))(sub {parse 32} {parse 57} )) 'a (add (num 3) (num 4)))

(add (id 'a) (add (id 'b) (sub((num -32)(num 57)) ) ) ) 'a (add (num 3) (num 4)))

**Sustitución.**

1.- (add (id 'a) (add (id 'b) (sub((num -32)(num 57)) ) ) ) 'a (add (num 3) (num 4)))

2.- (add (subst (id 'a) 'a (add (num 32)(num 4) ) ) (subst (add (id 'b) (sub (num 32)(num 57) ) )  
 a (add (num 3) (num 4) ) ) ) )

3.- (add (add (num 3)(num 4)) (add (subst (id 'b) 'a (add (num 3) (num 4))) (subst (sub (num  
 32) (num 57))) 'a (add (num 3) (num 4)))

4.- (add (add (num 3)(num 4))(sub (subst (num 32) 'a (add (num 3) (num 4)) (subst (num  
 57)'a (add (num 3)(num 4))))))

5.- (add (add (num 3)(num 4))(add (id b') (sub (num 32)(num 57)))

b.  $e = \{with \{y \{- 30 \{- y z\} \} \}$   
 $\{- 30 \{+ y z\} \} \}$   
 (subst (parse e) 'y (id 'w))

**Sintaxis abstracta.**

(with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z))))))

## Parse.

(parse e) 'a (add (num 3) (num 4))) con e={with {y {- 30 {- y z} }} {- 30 {+ y z} }}

(parse{with {y {- 30 {- y z} }} {- 30 {+ y z} }} )  
(with{parse y} {parse {-30{-y z}}}{parse {-30 {+y z}}}  
(with(id 'y)(sub {parse 30}{parse {-y z}}(sub {parse 30}{parse {+ y z}})))  
(with(id 'y)(sub (num 30)(sub {parse y}{parse z})(sub (num 30)(add {parse y} {parse z}))))  
(with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z)))))

## Sustitución:

Para e = (with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z))))) )

1.- (subst((with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z))))) 'y (id 'w))

2.- (with (subst 'y (id 'w)(id 'y)(subst 'y (id 'w) (sub (num 30)(sub (id 'y)(id 'z)) (subst 'y (id 'w) (sub (num 30)(add (id 'y) (id 'z))))) 'y (id 'w))))

3.- (with (id 'y)(sub (subst 'y (id 'w)(num 30))(subst 'y (id 'w)(sub (id 'y)(id 'z))(sub(subst 'y (id 'w)(num 30))(subst 'y (id 'w)(add (id 'y)(id 'z)))))

4.- (with (id 'y)(sub (num 30)(sub (subst 'y (id w)(id 'y)(subst 'y (id 'w) (id 'z))) (sub (num 30) (add(subst 'y (id 'w) (id 'y)(subst 'y (id w) (id 'z))

5.- (with (id 'y)(sub (num 30)(sub (id 'w)(id 'z))(sub (num 30)(add (id 'y)(id 'z)))))

## La sustitución ya hecha:

(with (id 'y)(sub (num 30)(sub (id 'w)(id 'z))(sub (num 30)(add (id 'y)(id 'z)))))

c. e = {with {y {- 30 {- y z} }}  
{- 30 {+ y z} }}  
(subst (parse e) 'z (id 'v))

## Sintaxis abstracta.

(with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z)))))

## Parse.

(parse e) 'a (add (num 3) (num 4))) con e={with {y {- 30 {- y z} }} {- 30 {+ y z} }}

(parse{with {y {- 30 {- y z} }} {- 30 {+ y z} }} )  
(with{parse y} {parse {-30{-y z}}}{parse {-30 {+y z}}}) (with(id 'y)(sub {parse 30}{parse {-y z}}(sub {parse 30}{parse {+ y z}})))  
(with(id 'y)(sub (num 30)(sub {parse y}{parse z})(sub (num 30)(add {parse y} {parse z}))))

(with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z))))))

### **Sustitución:**

Para e = (with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z))))))

1.- (subst((with(id 'y)(sub (num 30)(sub (id 'y)(id 'z))(sub (num 30)(add (id 'y) (id 'z)))))) 'z (id 'v))

2.- (with (subst 'z (id 'v)(id 'y)(subst 'z (id 'v) (sub (num 30)(sub (id 'y)(id 'z)) (subst 'z (id 'v) (sub (num 30)(add (id 'y) (id 'z)))))) 'y (id 'w))))

3.- (with (id 'y)(sub (subst 'z (id 'v)(num 30))(subst 'z (id 'v)(sub (id 'y)(id 'z)))(sub(subst 'z (id 'v)(num 30))(subst 'z (id 'v)(add (id 'y)(id 'z))))))

4.- (with (id 'y)(sub (num 30)(sub (subst 'z (id 'v)(id 'y)(subst 'z (id 'v)(id 'z)))) (sub (num 30) (add(subst 'z (id 'v) (id 'y)(subst 'z (id 'v)(id 'z))

5.- (with (id 'y)(sub (num 30)(sub (id 'y)(id 'v)))(sub (num 30)(add (id 'y)(id 'v))))

### **La sustitución ya hecha:**

(with (id 'y)(sub (num 30)(sub (id 'y)(id 'v)))(sub (num 30)(add (id 'y)(id 'v))))

3. Sea la siguiente expresión definida usando lenguaje *WAE*. Da la sintaxis abstracta esta expresión. Muestra el proceso de evaluación mediante la función *interp* y responde las siguientes preguntas.

```
{with {a 3}
  {with {b 9}
    {with {c 4}
      {with {d 11}
        {+ a {+ b {+ c d} } } } } }
```

**Sintaxis abstracta.**

```
(with (id a )(num 3)
  (with (id b)(num 9)
    (with (id c)(num 4)
      (with (id d) (num 11)
        (add (id a)(add (id b)(add (id c)(id d)))))))
```

**Evaluación mediante la función *interp*:**

```
(interp (with (id a )(num 3)
  (with (id b)(num 9)
    (with (id c)(num 4)
      (with (id d) (num 11)
        (add (id a)(add (id b)(add (id c)(id d)))))))
```

Primero entramos al caso del primer with que coincide con la sintaxis definida [( with id value body )]

```
(interp (subst (with (id b)(num 9)
  (with (id c)(num 4)
    (with (id d) (num 11)
      (add (id a)(add (id b)(add (id c)(id d)))))) (id a) (num 3) ))
```

Ahora aplicamos la función de subst

(if ( symbol =? id sub-id ) —> (if ( symbol =? b a ) —> false

```
(interp (with b (subst num 9 id a num 3)(subst (with (id c)(num 4)
  (with (id d) (num 11)
    (add (id a)(add (id b)(add (id c)(id d)))))) id a num 3 )))
```

Aplicamos la función en el primer subst (el azul)

```
(interp (with b (num 9)(subst (with (id c)(num 4)
                                (with (id d) (num 11)
                                      (add (id a)(add (id b)(add (id c)(id d)))))) id a num 3)))
```

Aplicamos la siguiente función subst

(if ( symbol =? id sub-id ) —> (if ( symbol =? c a ) —> false

```
(interp (with b (num 9)(with id c (subst num 4 id a num 3)(subst (with (id d) (num 11)
                                                                    (add (id a)(add (id b)(add (id c)(id d))))
                                                                    id a num 3))))
```

Aplicamos de nuevo la función subst y también subst

— (if ( symbol =? id sub-id ) —> (if ( symbol =? d a ) —> false —

```
(interp(with b (num 9)(with id c (num 4)(with id d (subst num 11 id a num 3)
(subst (add (id a)(add (id b)(add (id c)(id d))) id a num 3 ))))
```

Aplicamos de nuevo la función subst y también subst

```
(interp ((with b (num 9)(with id c (num 4)(with id d (num 11)
(add (subst (add (id b)(add (id c)(id d))) id a num 3 ) (subst (id a) id a num 3) )))))
```

Aplicamos de nuevo la función subst y también subst

```
(interp ((with b (num 9)(with id c (num 4)(with id d (num 11)
(add ((add (subst (add (id c)(id d)) id a num 3 )(subst (id b) id a num 3)) ) ( num 3) )))))
```

Aplicamos de nuevo la función subst y también subst

```
(interp ((with b (num 9)(with id c (num 4)(with id d (num 11)
(add ((add (add (subst (id d) id a num 3 )(subst (id c) id a num 3 )(id b)) ) ( num 3) )))))
```

Aplicamos de nuevo la función subst y también subst

```
(interp (with b (num 9)(with id c (num 4)(with id d (num 11)
(add ((add (add (id d)(id c))(id b)) ) ( num 3) )))))
```

Como ya no tenemos **subst** empezamos a evaluar **interp**

```
(interp (subst (with id c (num 4)(with id d (num 11)
(add ((add (add (id d )(id c ))(id b)) ) ( num 3) )) (b) (num 9)))
```

Aplicamos **subst** de nuevo

– (if ( symbol =? id sub-id ) → (if ( symbol =? c b) → false

```
(interp (with c (subst 4 b 9)(subst (with d 11 (+ (+ d c ) (b)) (3)) b 9 )))
```

Aplicamos los dos **subst** de nuevo

– (if ( symbol =? id sub-id ) → (if ( symbol =? d b) → false

```
(interp (with c (4) (with d (subst 11 b 9) (subst (+ (+ d c ) (b)) (3)) b 9 )))
```

Aplicamos los dos **subst** de nuevo

```
(interp (with c (4) (with d (11) (+ (subst (+ d c ) (b)) b 9) (subst 3 b 9) )))
```

Aplicamos los dos **subst** de nuevo

```
(interp (with c (4) (with d (11) (+ (+ (subst (+ d c ) (b)) (subst b b 9)) (3) )))
```

Aplicamos los dos **subst** de nuevo

```
(interp (with c (4) (with d (11) (+ (+ (+ (subst d b 9) (subst c b 9) ) (9)) (3) )))
```

Aplicamos los dos **subst** de nuevo

```
(interp (with c (4) (with d (11) (+ (+ (+ (d)(c) ) (9)) (3) )))
```

Aplicamos **interp** de nuevo

```
(interp (subst (with d (11) (+ (+ (+ (d)(c) ) (9)) (3)) ) c 4 )
```

Seguimos con **subst** de nuevo

— (if ( symbol =? id sub-id ) → (if ( symbol =? d c) → false —

```
(interp ( with d (subst 11 c 4)(subst (+ (+ (+ (d)(c) ) (9)) (3)) c 4) ))
```

Aplicamos los dos **subst** de nuevo

```
(interp ( with d 11 (+ (subst (+ (+ (d)(c) ) (9)) c 4 ) (subst 3 c 4) ))
```

Aplicamos los dos **subst** de nuevo

```
(interp ( with d 11 (+ (+ (subst + (d)(c) c 4)(subst 9 c 4) (3) ))
```

Aplicamos los dos **subst** de nuevo

```
(interp ( with d 11 (+ (+ (+ (subs d c 4)(subs c c 4) ) (9)) (3) ))
```



Aplicamos los dos **subst** de nuevo

(**interp** ( **with** d 11 ( + ( + ( + (d)(4) )(9) ) (3) ) ) )

Aplicamos **interp** de nuevo

(**interp** ( **subst** ( + ( + ( + (d)(4) )(9) ) (3) ) d 11 ) )

Seguimos con **subst** de nuevo

(**interp** ( ( + ( **subst** ( + ( + (d)(4) )(9) ) d 11 ) ( **subst** 3 d 11 ) ) ) )

Seguimos con **subst** de nuevo

(**interp** ( ( + ( ( + ( **subst** + (d)(4) d 11 )( **subst** 9 d 11 ) ) ) ( 3 ) ) ) )

Seguimos con **subst** de nuevo

(**interp** ( ( + ( ( + ( + ( **subst** d d 11 )( **subst** 4 d 11 ) )(9) ) ) ( 3 ) ) ) )

Aplicamos **interp**

(**interp** ( ( + ( ( + ( + (11)( 4 ) )( 9 ) ) ) ( 3 ) ) ) )

Aplicamos **interp** de nuevo

(**num** ( + ( **interp** ( ( + ( + (11)( 4 ) )( 9 ) ) ) ) ( **interp** 3 ) ) )

Aplicamos **interp** de nuevo

(**num** ( + ( ( + ( **interp** + (11)( 4 ) )( **interp** 9 ) ) ) ) ( 3 ) ) )

Aplicamos **interp** de nuevo

(**num** ( + ( ( + ( + ( **interp** 11 )( **interp** 4 ) )( 9 ) ) ) ) ( 3 ) ) )

Aplicamos **interp** de nuevo

(**num** ( + ( + ( + (11)( 4 ) )( 9 ) ) ( 3 ) ) )

Por ultimo evaluamos las sumas

= (**num** ( + ( + 15 9 ) ( 3 ) ) )

=(**num** ( + 24 3 ) )

=(**num** 27)

- a. ¿Cuántas veces se aplica el algoritmo de sustitución para evaluar esta expresión?

40 Veces

4. Convierte las siguientes expresiones usando la notación de índices de *De Bruijn*.

a. {with {v 2}  
     {with {w 3}  
         {with {x 4}  
             {with {y {+ v {- w x} } }  
                 {with {z {with {v {+ w x} } v } }  
                     {+ y {with {w {- y z} } {- w x} } } } } } }

#### Conversión.

{with {2}  
     {with {3}  
         {with {4}  
             {with { {+ <:2> {- <:1> <:0> } } }  
                 {with { {with {<:3> {+ <:2> <:1> } } <:0> } }  
                     {+ <:1> {with {<:3> {- <:1> <:0> } } {- <:3> <:2> } } } } } }

**Ya que en este ejercicio no se utiliza la posición, podemos omitir ese dato, además que esta notación es utilizada en las notas.**

5. Dadas las siguientes expresiones representadas mediante índices de *De Bruijn*, obtén su respectiva versión usando los nombres de los identificadores de variables, iniciando por "x", "y", "z", "v", "w".

a. {with {1 2 3}  
     {with {4 5 6}  
         {with { {with { {+ <:0 1> <:1 2> } {- <:1 1> <:0 0> } } 3 } }  
             {with {<: 0 0>}  
                 {+ <:3 2> {+ <:2 1> {+ <:1 0> <:0 0> } } } } }

**Respuesta:**

{with {{x 1} {y 2} {z 3} }  
     {with {{v 4} {w 5} {u 6}}  
         {with { t {with { {+ w z} {- y v} } 3 } }  
             {with { t }  
                 { z {+ w { + t t } } } } }

6. Dada la siguiente expresión.

{with {w 2}  
     {with {x 3}  
         {with {y {+ w x} } }

$\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ y y \} \} \} \}$

a. Evalúa la expresión expresión. Muestra los pasos que se deben de hacer en cada una de sus derivaciones intermedias.

$\{with \{w 2\}$   
 $\{with \{x 3\}$   
 $\{with \{y \{+ w x \} \}$   
 $\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ y y \} \} \} \}$

$= \{with \{y \{+ 2 3 \} \}$   
 $\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ y y \} \} \} \}$  *[sustitución]*

$= \{with \{y \{5 \} \}$   
 $\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ y y \} \} \} \}$  *[+ operación]*

$= \{with \{y 5 \}$   
 $\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ y y \} \} \} \}$  *[sustitución]*

$= \{with \{y 5 \}$   
 $\{with \{w -2\}$   
 $\{with \{x -3\}$   
 $\{+ 5 5 \} \} \} \}$  *[sustitución]*

```
= {with {y 5 }
  {with {w -2}
    {with {x -3 }
      {+ 5 5 } } } } } [+ operación]
```

=10                      [resultado]

b. ¿Pueden haber otros resultados? ¿Por qué?

No, ya que si utilizamos alcance dinámico (el otro tipo de alcance) nos quedan variables libres, por lo que no hay forma de evaluar otro resultado.

Esto suceder ya que si utilizamos alcance dinámico comenzamos tratando de evaluar  $\{+ y y\}$  buscaríamos y nos toparemos con  $\{with \{w -2\} \{with \{x -3 \}$  pero como no las ocupamos en lo que queremos evaluar las botamos de la pila, de esta forma no accedemos a estos valores para reemplazarlos en  $\{with \{y \{+ w x \} \}$  por lo que nos quedamos con variables libres, de esta forma no podemos evaluar.

c. ¿Cuál es el resultado correcto en dado caso de haber más de un posible resultado? ¿Por qué?

Ambos resultados son correctos, solo que dependen del tipo de alcance que se quiera utilizar, aunque el más legible para el programador y más usado es el estático.



