Taller 1

Semántica de lenguajes de programación Fundamentos de Interpretación y Compilación de Lenguajes de Programación

Integrantes:

2411023 - Juan Pablo Ospina Vanegas

2459375 – Diana Marcela Oviedo Murillo

2059817 – Carlos H Gutierrez Mejia

Docente:

Delgado Saavedra Carlos Andres

Septiembre 2024
Universidad Del Valle - Sede Tuluá
Facultad de Ingeniería de Sistemas de Información

Representación por Listas

Nombre del archivo: representacion-listas.rkt

- Ejemplo 1:

```
"representacion-listas.rkt"> a
'(complex-circuit
  (simple-circuit
   (m n o p)
   (e f)
   (comp-chip
    (INA INB INC IND)
    (OUTD OUTF)
    (complex-circuit
     (simple-circuit (a b) (e) (prim-chip (chip-and)))
     ((simple-circuit (c d) (f) (prim-chip (chip-and))))
     (a b c d)
     (e f))))
  ((simple-circuit
    (e f)
    (z)
    (comp-chip
     (INE INF)
     (OUTA)
     (simple-circuit (e f) (g) (prim-chip (chip-or))))))
  (m n o p)
  (z))
```

- Ejemplo 2:

```
"representacion-listas.rkt"> b
'(comp-chip
  (INA INB INC IND)
  (OUTA)
  (complex-circuit
   (simple-circuit (a b) (e) (prim-chip (chip-and)))
   ((simple-circuit (c d) (f) (prim-chip (chip-and)))
    (simple-circuit (e f) (g) (prim-chip (chip-or))))
   (a b c d)
   (g)))
```

- Ejemplo 3:

```
"representacion-listas.rkt"> c
'(simple-circuit (x y) (z) (prim-chip (chip-xor)))
```

- Ejemplo 4:

```
"representacion-listas.rkt"> d
'(complex-circuit
  (simple-circuit (a b) (c) (prim-chip (chip-nand)))
  ((simple-circuit (d e) (f) (prim-chip (chip-nor))))
  (a b d e)
  (c f))
```

- Ejemplo 5:

```
"representacion-listas.rkt"> e
'(comp-chip
  (IN1 IN2)
  (OUT1)
  (simple-circuit (g h) (i) (prim-chip (chip-xnor))))
```

- Ejemplo 6:

```
"representacion-listas.rkt"> f
'(complex-circuit
  (simple-circuit (j k) (l) (prim-chip (chip-not)))
  ((simple-circuit (m n) (o) (prim-chip (chip-or))))
  (j k m n)
  (l o))
```

- Ejemplo 7:

```
"representacion-listas.rkt"> g
'(comp-chip
  (INX INY)
  (OUTZ)
  (complex-circuit
   (simple-circuit (p q) (r) (prim-chip (chip-and)))
   ((simple-circuit (s t) (u) (prim-chip (chip-xor))))
   (p q s t)
   (r u)))
```

Universidad del Valle

Representación por Procedimientos

Nombre del archivo: representacion-procedimientos.rkt

- Ejemplo 1:

```
"representacion-procedimientos.rkt"> a
#<procedure:...-procedimientos.rkt:32:8>
"representacion-procedimientos.rkt"> (complex-circuit?
                                      (simple-circuit '(m n o p)
                                                      '(e f)
                                                      (comp-chip '(INA INB INC IND)
                                                                 '(OUTD OUTF)
                                                                 (complex-circuit
                                                                  (simple-circuit '(a b) '(e) (prim-chip (chip-and)))
                                                                  (list (simple-circuit '(c d) '(f) (prim-chip (chip-and))))
                                                                  '(a b c d)
                                                                  '(e f))))
                                      (list (simple-circuit
                                             '(e f)
                                             '(z)
                                             (comp-chip '(INE INF) '(OUTA) (simple-circuit '(e f) '(g) (prim-chip (chip-or))))))
                                      '(m n o p)
                                      '(z))
#<procedure:...-procedimientos.rkt:60:6>
```

- Ejemplo 2:

```
"representacion-procedimientos.rkt"> b
#<procedure:...-procedimientos.rkt:128:6>
"representacion-procedimientos.rkt"> (comp-chip? '(INA INB INC IND)
                                                 '(OUTA)
                                                 (complex-circuit (simple-circuit '(a b) '(e) (prim-chip (chip-and)))
                                                                  (list (simple-circuit '(c d) '(f) (prim-chip (chip-and)))
                                                                        (simple-circuit '(e f) '(g) (prim-chip (chip-or))))
                                                                  '(a b c d)
                                                                  '(g)))
#<procedure:...-procedimientos.rkt:150:6>
```

- Ejemplo 3:

```
"representacion-procedimientos.rkt"> c
#<procedure:...-procedimientos.rkt:32:8>
"representacion-procedimientos.rkt"> (complex-circuit?
                                      (simple-circuit '(a b c d)
                                                      '(x y)
                                                      (comp-chip '(INL INM INN INO)
                                                                 '(OUTP OUTQ)
                                                                 (complex-circuit
                                                                  (simple-circuit '(e f) '(g) (prim-chip (chip-nor)))
                                                                  (list (simple-circuit '(h i) '(j) (prim-chip (chip-and))))
                                                                  '(e f h i)
                                                                  '(g j))))
                                      (list (simple-circuit '(x y)
                                                            '(z)
                                                            (comp-chip '(INI INJ) '(OUTI) (simple-circuit '(x y) '(z) (prim-chip (chip-xor)))))
)
                                      '(a b c d)
                                      '(z))
#<procedure:...-procedimientos.rkt:60:6>
```

- Ejemplo 4:

```
"representacion-procedimientos.rkt"> d
#<procedure:...-procedimientos.rkt:128:6>
"representacion-procedimientos.rkt"> (comp-chip? '(INA INB INC IND)
                                                 '(OUTC)
                                                 (complex-circuit (simple-circuit '(m n) '(o) (prim-chip (chip-nand)))
                                                                  (list (simple-circuit '(p q) '(r) (prim-chip (chip-xnor)))
                                                                        (simple-circuit '(o r) '(s) (prim-chip (chip-or))))
                                                                  '(m n p q)
                                                                  '(s)))
#<procedure:...-procedimientos.rkt:150:6>
```

- Ejemplo 5:

```
"representacion-procedimientos.rkt"> e
#<procedure:...-procedimientos.rkt:128:6>
"representacion-procedimientos.rkt"> (comp-chip? '(IN1 IN2 IN3)
                                                 '(OUT1 OUT2)
                                                 (complex-circuit (simple-circuit '(p q) '(r) (prim-chip (chip-and)))
                                                                  (list (simple-circuit '(s t) '(u) (prim-chip (chip-nor)))
                                                                        (simple-circuit '(v w) '(x) (prim-chip (chip-xor))))
                                                                  '(p q s t v w)
                                                                  '(r u x)))
#<procedure:...-procedimientos.rkt:150:6>
```

- Ejemplo 6:

```
"representacion-procedimientos.rkt"> f
#<procedure:...-procedimientos.rkt:128:6>
"representacion-procedimientos.rkt"> (complex-circuit? (simple-circuit '(a b) '(c) (prim-chip (chip-or)))
                                                       (list(simple-circuit '(d e) '(f) (prim-chip (chip-and)))
                                                        (simple-circuit '(g h) '(i) (prim-chip (chip-xnor))))
                                                       '(a b d e g h)
                                                       '(c f i))
#<procedure:...-procedimientos.rkt:60:6>
```

- Ejemplo 7:

```
"representacion-procedimientos.rkt"> g
#<procedure:...-procedimientos.rkt:18:8>
"representacion-procedimientos.rkt"> (simple-circuit? '(x y) '(z) (prim-chip (chip-xor)))
#<procedure:...-procedimientos.rkt:48:6>
```

Representación con Datatypes

Nombre del archivo: representacion-datatype.rkt

- Ejemplo 1:

```
#(struct:complex-circuit #(struct:simple-circuit (m n o p) (e f) #(struct:comp-chip (INA INB INC IND) (OUTD OUTF) #(struct:complex-circuit #(struct:simple-circuit (a
b) (e) #(struct:prim-chip #(struct:chip-and))) (#(struct:simple-circuit (c d) (f) #(struct:prim-chip #(struct:chip-and)))) (a b c d) (e f)))) (#(struct:simple-circuit
 (e f) (z) #(struct:comp-chip (INE INF) (OUTA) #(struct:simple-circuit (e f) (q) #(struct:prim-chip #(struct:chip-or)))))) (m n o p) (z))
Fin Ejemplo a
```

- Ejemplo 2:

```
#(struct:comp-chip (INA INB INC IND) (OUTA) #(struct:complex-circuit #(struct:simple-circuit (a b) (e) #(struct:prim-chip #(struct:chip-and))) (#(struct:simple-circui
t (c d) (f) #(struct:prim-chip #(struct:chip-and))) #(struct:simple-circuit (e f) (q) #(struct:prim-chip #(struct:chip-or)))) (a b c d) (q)))
Fin Ejemplo b
```

- Ejemplo 3:

```
#(struct:complex-circuit #(struct:simple-circuit (a b c d) (x y) #(struct:comp-chip (INL INM INN INO) (OUTP OUTQ) #(struct:complex-circuit #(struct:simple-circuit (e
f) (q) #(struct:prim-chip #(struct:chip-nor))) (#(struct:simple-circuit (h i) (j) #(struct:prim-chip #(struct:chip-and)))) (e f h i) (q j)))) (#(struct:simple-circuit
 (x y) (z) #(struct:comp-chip (INI INJ) (OUTI) #(struct:simple-circuit (x y) (z) #(struct:prim-chip #(struct:chip-xor)))))) (a b c d) (z))
Fin Ejemplo c
```

- Ejemplo 4:

```
#(struct:comp-chip (INA INB INC IND) (OUTC) #(struct:complex-circuit #(struct:simple-circuit (m n) (o) #(struct:prim-chip #(struct:chip-nand))) (#(struct:simple-circu
it (p q) (r) #(struct:prim-chip #(struct:chip-xnor))) #(struct:simple-circuit (o r) (s) #(struct:prim-chip #(struct:chip-or)))) (m n p q) (s)))
Fin Ejemplo d
```

- Ejemplo 5:

```
#(struct:complex-circuit #(struct:simple-circuit (a b) (c) #(struct:prim-chip #(struct:chip-and))) (#(struct:simple-circuit (c d) (e) #(struct:prim-chip #(struct:chip
-or)))) (a b d) (e))
Fin Ejemplo e
```

- Ejemplo 6:

```
((comp-chip (IN1 IN2 IN3) (OUT1 OUT2) (complex-circuit (simple-circuit (p q) (r) (prim-chip (chip-and))) ((simple-circuit (s t) (u) (prim-chip (chip-nor))) (simple-cir
cuit (v w) (x) (prim-chip (chip-xor)))) (p q s t v w) (r u x)))
Fin Ejemplo g
```

- Ejemplo 7:

```
#(struct:simple-circuit (x y) (z) #(struct:prim-chip #(struct:chip-xor)))
Fin Ejemplo f
```

Universidad del Valle

Parser y Unparser

Nombre del archivo: parser-unparser.rkt

- Ejemplo 1:
  Parser:

```
----------------Parsers----------------
#(struct:complex-circuit #(struct:simple-circuit (m n o p) (e f) #(struct:comp-chip (INA INB INC IND) (OUTD OUTF) #(struct:
complex-circuit #(struct:simple-circuit (a b) (e) #(struct:prim-chip #(struct:chip-and))) (#(struct:simple-circuit (c d) (f
) #(struct:prim-chip #(struct:chip-and)))) (a b c d) (e f)))) (#(struct:simple-circuit (e f) (z) #(struct:comp-chip (INE IN
F) (OUTA) #(struct:simple-circuit (e f) (g) #(struct:prim-chip #(struct:chip-or)))))) (m n o p) (z))
fin parser a
```

  Unparser:

```
----------------Unparsers----------------
(complex-circuit (simple-circuit (m n o p) (e f) (comp-chip (INA INB INC IND) (OUTD OUTF) (complex-circuit (simple-circuit
(a b) (e) (prim-chip (chip-and))) ((simple-circuit (c d) (f) (prim-chip (chip-and)))) (a b c d) (e f)))) ((simple-circuit (
e f) (z) (comp-chip (INE INF) (OUTA) (simple-circuit (e f) (g) (prim-chip (chip-or)))))) (m n o p) (z))
fin unparser a
```

- Ejemplo 2:
  Parser

```
#(struct:comp-chip (INA INB INC IND) (OUTA) #(struct:complex-circuit #(struct:simple-circuit (a b) (e) #(struct:prim-chip #
(struct:chip-and))) (#(struct:simple-circuit (c d) (f) #(struct:prim-chip #(struct:chip-and))) #(struct:simple-circuit (e f
) (g) #(struct:prim-chip #(struct:chip-or)))) (a b c d) (g)))
fin parser b
```

  Unparser:

```
(comp-chip (INA INB INC IND) (OUTA) (complex-circuit (simple-circuit (a b) (e) (prim-chip (chip-and))) ((simple-circuit (c
d) (f) (prim-chip (chip-and))) (simple-circuit (e f) (g) (prim-chip (chip-or)))) (a b c d) (g)))
fin unparser b
```

- Ejemplo 3:
  Parser:

```
#(struct:simple-circuit (x y z w) (a b c) #(struct:comp-chip (IN1 IN2 IN3) (OUT1 OUT2) #(struct:complex-circuit #(struct:si
mple-circuit (p q) (r) #(struct:prim-chip #(struct:chip-xor))) (#(struct:simple-circuit (s t) (u) #(struct:prim-chip #(stru
ct:chip-nand)))) (p q s t) (r u))))
fin parser c
```

  Unparser:

```
(simple-circuit (x y z w) (a b c) (comp-chip (IN1 IN2 IN3) (OUT1 OUT2) (complex-circuit (simple-circuit (p q) (r) (prim-chi
p (chip-xor))) ((simple-circuit (s t) (u) (prim-chip (chip-nand)))) (p q s t) (r u))))
fin unparser c
```

- Ejemplo 4:
  Parser:

```
#(struct:complex-circuit #(struct:simple-circuit (a b c) (d e) #(struct:prim-chip #(struct:chip-nand))) (#(struct:simple-ci
rcuit (f g) (h) #(struct:prim-chip #(struct:chip-nor))) #(struct:simple-circuit (i j) (k) #(struct:prim-chip #(struct:chip-
xnor)))) (a b c f g i j) (d e h k))
fin parser d
```

  Unparser:

```
(complex-circuit (simple-circuit (a b c) (d e) (prim-chip (chip-nand))) ((simple-circuit (f g) (h) (prim-chip (chip-nor)))
(simple-circuit (i j) (k) (prim-chip (chip-xnor)))) (a b c f g i j) (d e h k))
fin unparser d
```

- Ejemplo 5:

  Parser:

```
#(struct:comp-chip (IN1 IN2 IN3) (OUT1 OUT2) #(struct:complex-circuit #(struct:simple-circuit (x y) (z) #(struct:prim-chip
#(struct:chip-xnor))) (#(struct:simple-circuit (a b) (c) #(struct:prim-chip #(struct:chip-or))) #(struct:simple-circuit (d
e) (f) #(struct:prim-chip #(struct:chip-and)))) (x y a b d e) (z c f)))
fin parser e
```

  Unparser:

```
(comp-chip (IN1 IN2 IN3) (OUT1 OUT2) (complex-circuit (simple-circuit (x y) (z) (prim-chip (chip-xnor))) ((simple-circuit (
a b) (c) (prim-chip (chip-or))) (simple-circuit (d e) (f) (prim-chip (chip-and)))) (x y a b d e) (z c f)))
fin unparser e
```

- Ejemplo 6:

  Parser:

```
#(struct:simple-circuit (p q r s) (t u v) #(struct:comp-chip (INP INQ INR) (OUTP OUTQ OUTR) #(struct:complex-circuit #(stru
ct:simple-circuit (w x) (y) #(struct:prim-chip #(struct:chip-or))) (#(struct:simple-circuit (z a) (b) #(struct:prim-chip #(
struct:chip-not)))) (w x z a) (y b))))
fin parser f
```

  Unparser:

```
(simple-circuit (p q r s) (t u v) (comp-chip (INP INQ INR) (OUTP OUTQ OUTR) (complex-circuit (simple-circuit (w x) (y) (pri
m-chip (chip-or))) ((simple-circuit (z a) (b) (prim-chip (chip-not)))) (w x z a) (y b))))
fin unparser f
```

- Ejemplo 7:

  Parser

```
#(struct:complex-circuit #(struct:simple-circuit (i j k) (l m) #(struct:prim-chip #(struct:chip-and))) (#(struct:simple-cir
cuit (n o) (p) #(struct:prim-chip #(struct:chip-not))) #(struct:simple-circuit (q r) (s) #(struct:prim-chip #(struct:chip-x
or))) #(struct:simple-circuit (t u) (v) #(struct:prim-chip #(struct:chip-nand)))) (i j k n o q r t u) (l m p s v))
fin parser g
```

  Unparser:

```
(complex-circuit (simple-circuit (i j k) (l m) (prim-chip (chip-and))) ((simple-circuit (n o) (p) (prim-chip (chip-not))) (
simple-circuit (q r) (s) (prim-chip (chip-xor))) (simple-circuit (t u) (v) (prim-chip (chip-nand)))) (i j k n o q r t u) (l
 m p s v))
fin unparser g
```