Fundamentos de lenguajes de programación
Duración 2.5 horas

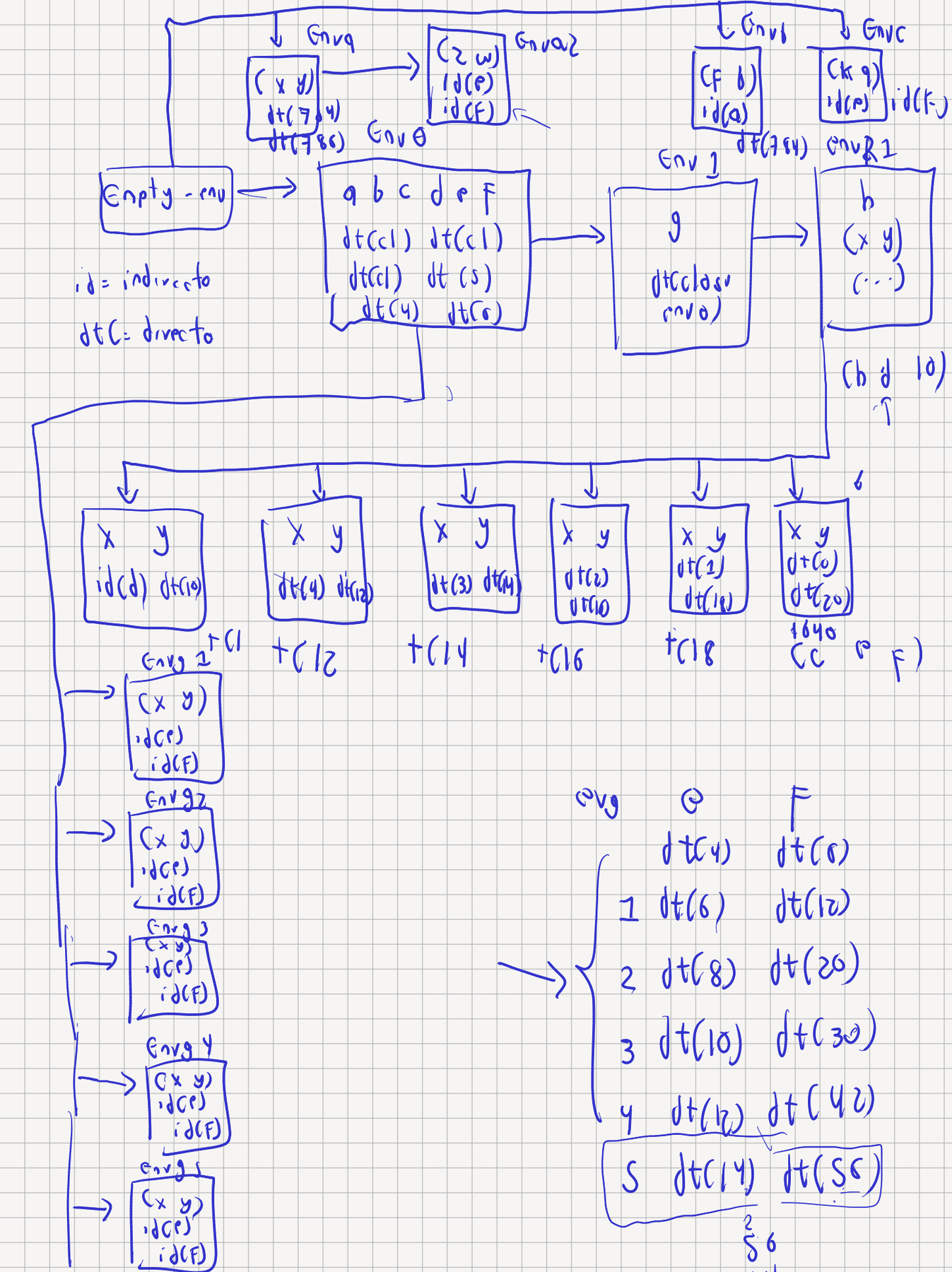
Carlos Andres Delgado S, Msc
carlos.andres.delgado@correounivalle.edu.co

13 de Octubre de 2022

1. (50 puntos) Considerando el ambiente inicial vacío y considerando para por referencia,

```
let
 a = proc (x,y) proc(z,w) +(x,y,z,w)
 b = proc (f,b) (f b +(b,2))
 c = proc (k,q) *(k,q)
 d = 5
 e = 4
 f = 6
 in
    let
        g = proc(x,y)
              begin
                set x = +(x,2);
                set y = +(x,y);
                +(x,y)
              end
    in
        letrec
              h(x,y) =
                   if >(x,0)
                     then
                       begin
                         (g e f);
                         +(y, (h -(x,1) +(y,2)))
                       end
                     else
                         ( (b a (c e f)) e f)
  in
    (h d +(e,f))
```

El resultado es 1710. Dibuje los ambientes asociados a esta expresión Recuerde indicar donde hay targets directos e indirectos.


Env a
(x y)
dt(784)
dt(786)
Env a2
(z w)
id(e)
id(F)
Env 0
Empty-env
a b c d e f
dt(c1) dt(c1)
dt(c1) dt(s)
dt(4) dt(6)
Env 1
g
dt(closure
env 0)
Env b1
b
(x y)
(...)
Env b
(F b)
id(a)
Env c
(k g)
id(e)
id(F)
dt(784)
id = indirecto
dt = directo
(b d 10)
x y
id(d) dt(10)
x y
dt(4) dt(12)
x y
dt(3) dt(14)
x y
dt(2)
dt(16)
x y
dt(1)
dt(18)
x y
dt(0)
dt(20)
1640
(c e f)
+C1
+12
+14
+16
+18
Env 1
(x y)
id(e)
id(F)
Env 2
Env 3
Env 4
Env 5
env e f
0 dt(4) dt(6)
1 dt(6) dt(12)
2 dt(8) dt(20)
3 dt(10) dt(30)
4 dt(12) dt(42)
5 dt(14) dt(56)
56
14
224
56
784
+(784, 786, 14, 56)
1640
(closure
(z, w)
env a)
+(10, 12, 14, 16, 18, 1640)
1710

2. (50 puntos) Añada los diccionarios al interpretador de asignación (disponible en el campus), estos corresponden a la siguiente gramática:

```
<expresion> ::= "{" (<identificador>:<expresion> (,))* ")"
            ::= dict-exp(lids lexps)
```

Por ejemplo un diccionario puede ser:

```
{altura:10, peso:200, edad:300}
```

a) (20 puntos) Genere la representación de los diccionarios dentro su lenguaje. Para esto debe agregar el caso a la expresión gramátical y generar una representación dentro evaluar-expresion.

b) (15 puntos) Genere la expresión:

```
<expresion> ::= "access" "(" <expresion> "," <identificador> ")"
            ::= access-exp(dict id)
```

access la cual recibe un diccionario y una llave, este permite retornar el valor almacenado con la llave correspondiente.

```
let
   q= {altura:10, peso:200, edad:300}
   in
     access(q,edad)
```

Debe retornar

```
300
```

c) (15 puntos) Cree la primitiva $dict- > keys$ este retorna una lista con lids asociados al diccionario. Ejemplo

```
let
   q= {altura:10, peso:200, edad:300}
   in
     dict->keys(q)
```

Debe retornar

```
(altura ,peso ,edad)
```