

Universidad Nacional Autónoma de México

Facultad de Ciencias

Lenguajes de Programación
Karla Ramírez Pulido

Sustitución

Sustitución

Variables

Operación de Asignación

$$f(x, y) = x^3 + y^3$$

$$f(12, 1) = 12^3 + 1^3 = 1728 + 1 = 1729$$

$$f(10, 9) = 10^3 + 9^3 = 1000 + 729 = 1729$$

Extendemos el lenguaje

```
<WAE> ::= <num>
        | {+ <WAE> <WAE>}
        | {- <WAE> <WAE>}
        | {with {<id> <WAE>} <WAE>}
        | <id>
```

Constructor

(define-type WAE

[num (n number?)]

[add (lhs WAE?) (rhs WAE?)]

[sub (lhs WAE?) (rhs WAE?)]

[with (name symbol?) (named-expr WAE?) (body WAE?)]

[id (name symbol?)])

EJEMPLO 1: {with {x 5} {+ x x}}

{with { x 5 } {+ x x} }

id valor-id cuerpo-with

Vamos a sustituir el valor-id de x en el cuerpo del with

{+ 5 5}

Evalúamos

= 10

EJEMPLO 2: {with {x {+ 5 5}} {with {y {- x 3}} {{+ y y}}}

{with {x {+ 5 5}}}

{with {y {- x 3}}}

{+ y y}}

[+ operation]

= {with {x 10}}

{with {y {- x 3}} {+ y y}}

[substitution]

= {with {y {- 10 3}} {+ y y}}

[descent]

EJEMPLO: {with {x {+ 5 5}} {with {y {- x 3}} {{+ y y}}}

= {with {y {- 10 3}} {+ y y}} [- operation]

= {with {y 7} {+ y y}} [substitution]

= {with {y 7} {+ 7 7}} [descent]

= {+ 7 7} [+ operation]

= 14

Ejemplo 3: {with {x 5} {+ 10 4}}

{with {x 5} {+ 10 4}}
id valor-id cuerpo-with

¿Hay algo que sustituir? NO, pero se tendrá que recorrer toda la expresión
with-expr o cuerpo-with

Sustitución es vacía

= {with {x 5} {+ 10 4}}

Recuerdan:

Expr [var := valor]

$(+ x x)[x := 0]$

$= (+ 0 0)$

$= 0$

Contexto: introducción

Contexto de una variable

Ambiente de una variable

The diagram illustrates variable scope using three nested rectangular regions. The outermost region is light blue and contains the code `{ int x=0;` at the top and a closing brace `}` at the bottom. Inside this is a medium purple region containing `{ int x=0;` at the top and a closing brace `}` at the bottom. The innermost region is light orange and contains `{` at the top, `int x=0;` in the middle, and `}` at the bottom. This visualizes how a variable declared in an outer scope is visible in inner scopes, while a variable declared in an inner scope is only visible within that specific inner scope.

```
{  
  int x=0;  
  
  {  
    int x=0;  
  
    {  
      int x=0;  
    }  
  }  
}
```

Ejemplo 4: {with {x 5} {+ x {with {x 3} 10}}} }

{with {x 5} {+ x {with {x 3} 10}}} }

id valor-id cuerpo-with

= {+ 5 {with {x 3} 10}}} }

SOLO AQUI porque lo que sigue está dentro de OTRO AMBIENTE

= {+ 5 {with {x 3} 10}}} }

Aquí tomo la x = 3 y la sustituimos en el cuerpo de ese with

Ejemplo 4: $\{\text{with } \{x \ 5\} \ \{+ \ x \ \{\text{with } \{x \ 3\} \ 10\}\} \}$

$= \{+ \ 5 \ 10\}$

Aquí tomo la $x = 3$ y la sustituimos en el cuerpo de ese with

La expresión final sustituida es:

$\{+ \ 5 \ 10\}$

Evaluación de esa expresión es:

15

Ejemplo 4: {with {x 5} {+ x {with {x 3} 10}}} }

Ejemplo 5: {with {x 5} {+ x {with {x 3} x}}} }

Asignaciones de valores
(inicializando)

{with {**x** **5**} {+ x {with {**x** **3**} 10}}}



Asignaciones de valores
(inicializando)

{with {**x** **5**} {+ **x** {with {**x** **3**} **x**}}}



Ejemplo 5: {with {x 5} {+ x {with {x 3} x}} }

{with {x 5} {+ x {with {x 3} x}} }

id valor-id cuerpo-with

= {+ 5 {with {x 3} x}} }

SOLO AQUI porque lo que sigue está dentro de OTRO AMBIENTE

= {+ 5 {with {x 3} x}} }

Aquí tomo la x = 3 y la sustituimos en el cuerpo de ese with

Ejemplo 5: $\{\text{with } \{x \ 5\} \ \{+ \ x \ \{\text{with } \{x \ 3\} \ x\}\} \}$

$= \{+ \ 5 \ 3\}$

Aquí tomo la $x = 3$ y la sustituimos en el cuerpo de ese with

La expresión final sustituida es:

$\{+ \ 5 \ 3\}$

Evaluación de esa expresión es:

8

Ejemplo 6: {with {x 5} {+ x {with {y 3} x}}}

{with {x 5}

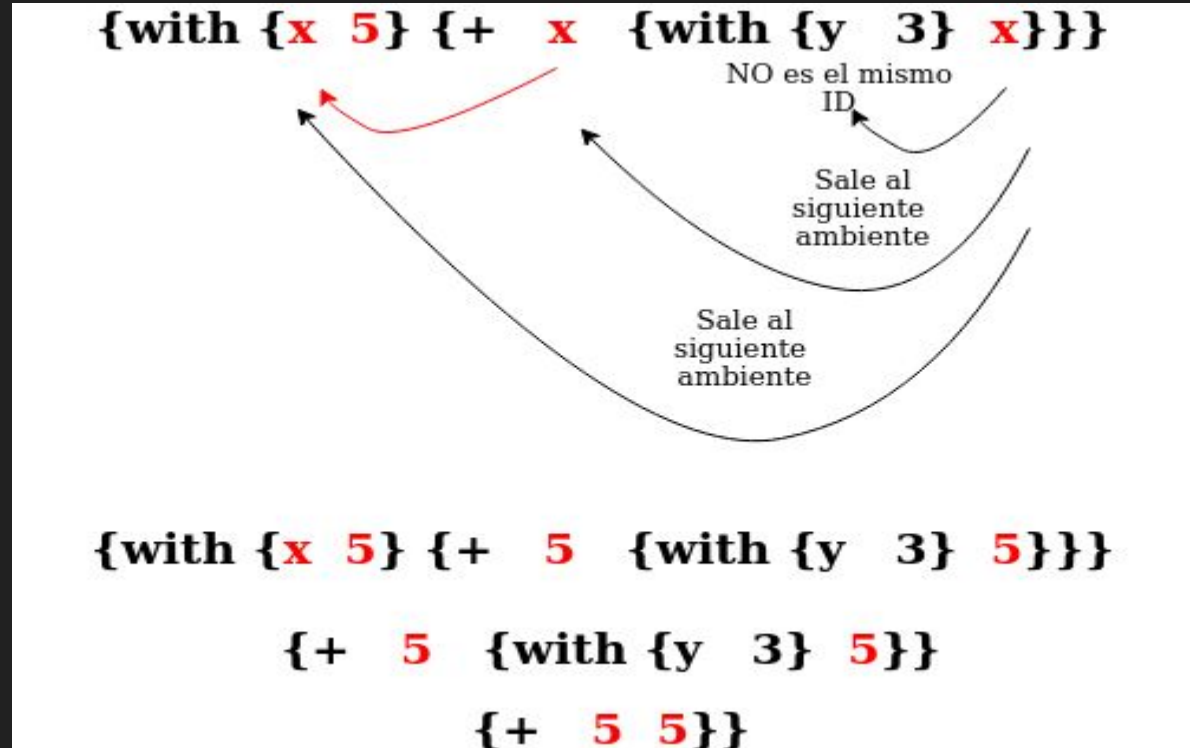
{+ x {with {y 3} x}}}

En Amarillo las inicializaciones

Las que no están en color amarillo son las variables que alcanzan su valor de alguna que debe de estar inicializada

Ejemplo 6: {with {x 5} {+ x {with {y 3} x}}}

¿Alcanzan su valor
el id x ?



Conceptos relevantes

Instancia ligada o acotada: instancia que alcanza su valor de una instancia de ligado, es decir que está contenida en su alcance.

Instancia de ligado: instancia que es inicializada con un valor o que se le da un valor, y éste puede ser alcanzado por alguna instancia ligada si se encuentra en su alcance.

Instancia libre: instancia que no alcanza su valor en una expresión (bajo ningún alcance).

Alcance: región del programa donde una variable ligada alcanza su valor de una de ligado.

Ejemplo 7: $\{\text{with } \{ x \ \{+ \ 2 \ 2\} \} \ \{+ \ x \ y\} \}$

$\{\text{with } \{ \textcolor{violet}{x} \ \textcolor{yellow}{\{+ \ 2 \ 2\}} \} \ \textcolor{green}{\{+ \ x \ y\}} \}$

$= \ \textcolor{green}{\{+ \ \textcolor{green}{\{+ \ 2 \ 2\}} \ \textcolor{orange}{y} \}}$

x es instancia de ligado

x en la suma es la instancia ligada a una de ligado

y es una instancia libre

En tiempo de ejecución

Todas las instancias tienen que tener un valor.

Las únicas que pueden quedar libres son los parámetros formales de las funciones, métodos, procedimientos.

Ejemplo 8: evaluar la expresión

$$\{\text{with } \{x \ +\ 5\ 5\}\} = \{\text{with } \{y \ -\ 10\ 3\}\}$$

$$\{\text{with } \{y \ -\ x\ 3\}\} \quad \{+\ y\ y\}$$

$$\{+\ y\ y\}} = \{\text{with } \{y\ 7\}$$

$$= \{\text{with } \{x\ 10\} \quad \{+\ y\ y\}$$

$$\{\text{with } \{y \ -\ x\ 3\}\} = \{+\ 7\ 7\}$$

$$\{+\ y\ y\}} = 14$$

Ejemplo 9: $\{\text{with } \{x\ 5\} \{\text{with } \{y\ x\} y\}\}$

$\{\text{with } \{x\ 5\}$ $= \{\text{with } \{y\ 5\}$

$\{\text{with } \{y\ x\}$ $y\}\}$

$y\}\}$

$= \{\text{with } \{y\ 5\}$

$y\}\}$

$= y$ es un id. ligado a y
con valor de 5

$= 5$

Ejemplo 10: {with {x x} x}

{with {x x} x}

La asignación:

x con valor de x

La sustitución:

x ligada a la de ligado, i.e. x

{with {x x} x}

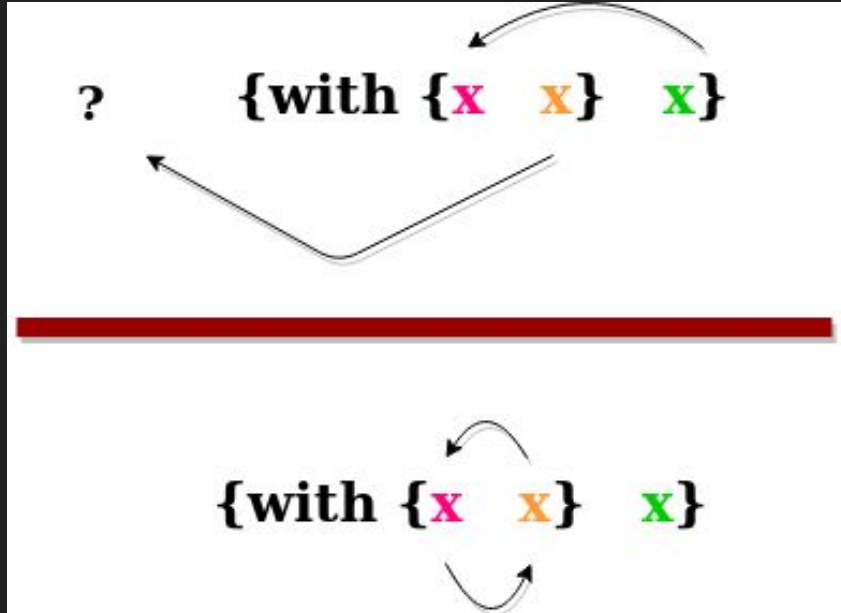


x: id de ligado

x: id. libre

x: id. ligado

{with {x x} x}



{with {x x} x}

¿Qué regresaría?

- a. Error de sintaxis i.e. “id. libre”
- a. Error de semántica

Más ejemplos

De la siguiente expresión ¿qué variables son de ligado, ligadas o libres?

```
{with { x 2}
  {with { x 3}
    {+ x 4}}}
```

x tiene el valor de 2 es de ligado

x tiene el valor de 3 es de ligado

x en el cuerpo de la expresión
with es ligada

No hay variables libres en esta
expresión

Más ejemplos

De la siguiente expresión ¿qué variables son de ligado, ligadas o libres?

```
{with { x 2}
  {with { x z}
    {+ x 4}}}}
```

x tiene el valor de 2 es de ligado

x (de ligado) tiene el valor del id.
z

x en el cuerpo de la expresión
with es ligada a la variable x = z

Variable libre: z

Siguiendo el ejemplo anterior

Evaluación de la expresión es:

$$\begin{aligned} &\{\text{with } \{ \mathbf{x} \ 2 \} \\ &\quad \{\text{with } \{ \mathbf{x} \ \mathbf{z} \} \\ &\quad \quad \{+ \ \mathbf{x} \ 4\}\}\} \end{aligned}$$
$$\{\text{with } \{ \mathbf{x} \ 2 \}$$
$$\quad \{\text{with } \{ \mathbf{x} \ \mathbf{z} \}$$
$$\quad \quad \{+ \ \mathbf{x} \ 4\}\}\}$$
$$= \{\text{with } \{ \mathbf{x} \ \mathbf{z} \}$$
$$\quad \{+ \ \mathbf{x} \ 4\}\}\}$$
$$= \{+ \ \mathbf{x} \ 4\} = \{+ \ \mathbf{z} \ 4\}$$

Función: subst

`:: subst : WAE symbol WAE \rightarrow WAE`

`:: substitutes second argument with third argument in first argument,`

`:: as per the rules of substitution; the resulting expression contains`

`:: no free instances of the second argument`

`(define (subst expr sub-id val)`

`(type-case WAE expr`

(define (subst expr sub-id val)

(type-case WAE expr

[num (n) expr]

[add (l r) (add (subst l sub-id val) (subst r sub-id val))]

[sub (l r) (sub (subst l sub-id val) (subst r sub-id val))]

[id (v) (if (symbol=? v sub-id) val expr)]

[with (bound-id named-expr bound-body)

```
[with (bound-id named-expr bound-body)
```

```
(if (symbol=? bound-id sub-id)
```

```
(with bound-id
```

```
(subst named-expr sub-id val )
```

```
bound-body )
```

```
(with bound-id
```

```
( subst named-expr sub-id val )
```

```
( subst bound-body sub-id val ))]] ))
```

Algoritmo de sustitución

```
(define (subst expr sub-id val)
```

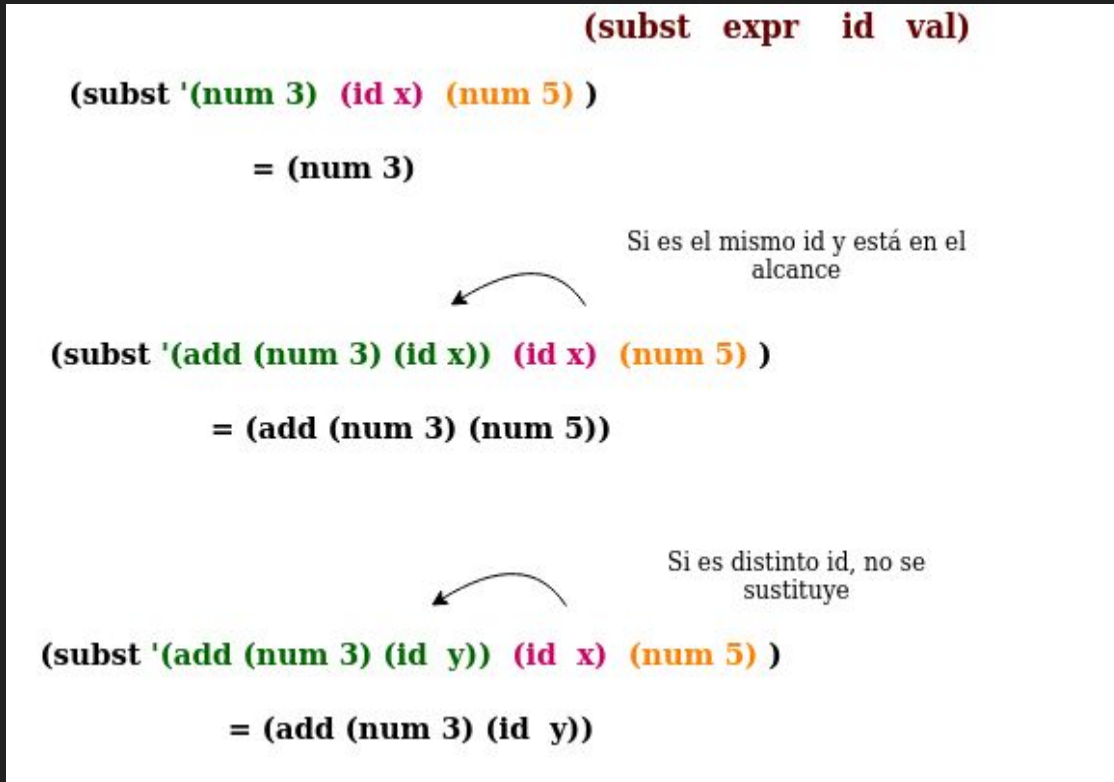
```
  (type-case WAE expr
```

```
    [num ( n ) expr]
```

```
    [add ( l r )
```

```
      (add (subst l sub-id val)
```

```
            (subst r sub-id val))])
```



Algoritmo de sustitución

(define (subst expr sub-id val)

[id (v) (if (symbol=? v sub-id) val expr)]

(subst '(id x) '(id y) '(num 4))

i.e. (subst 'x 'y 4)

v = x
sub-id = y
val = 4

= (if (symbol=? v sub-id) val expr)
 (if (symbol=? x y)
 4
 x)
= x

(subst '(id y) '(id y) '(num 4))

i.e. (subst 'y 'y 4)

v = y
sub-id = y
val = 4

= (if (symbol=? v sub-id) val expr)
 (if (symbol=? y y)
 4
 x)
= 4


```
(define (subst expr sub-id val)
```

```
  [with (bound-id named-expr bound-body)
```

```
    (if (symbol=? bound-id sub-id)
```

```
        (with bound-id
```

```
            (subst named-expr sub-id val )
```

```
            bound-body )
```

```
        (with bound-id
```

```
            ( subst named-expr sub-id val)
```

```
            ( subst bound-body sub-id val )))] )
```

Algoritmo de sustitución

(subst expr id val)



Si el id de la expr. es el mismo identificador con el que se llama subst NO podemos sustituir ¿por qué?

(subst '(with ((id x) (num 1)) (add (id x) (num 2))) (id x) (num 5))

(subst '{(with (x 1) (+ x 2))} x 5)

El id del with opaca al anterior

= { with (x 1) (+ x 2) }

En la evaluación sería = (+ 1 2)

Algoritmo de sustitución

(subst expr id val)

Si el id de la expr. es otro identificador con el que se llama subst, tenemos que seguir revisando el cuerpo del with

(subst '(with ((id y) (num 1)) (add (id y) (num 2))) (id x) (num 5))

(subst '{(with (y 1) (+ y 2))} x 5)

= { with (y 1) (+ y 2) }

Evaluación sería = (+ 1 2)

Algoritmo de sustitución

(subst expr id val)

Si el id de la expr. es otro identificador con el que se llama subst, tenemos que seguir revisando el cuerpo del with

(subst '(with ((id y) (num 1)) (add (id y) (id x))) (id x) (num 5))

(subst '{(with (y 1) (+ y x))} x 5)

= { with (y 1) (+ y 5) }

En la evaluación sería = {+ 1 5}

Índices de Bruijn

Representación por medio de índices donde $i = 0, \dots, n$.

< : profundidad posición >


Números representan el alcance (profundidad) y la posición (que ocupan en ese alcance)

Índices de Bruijn (Nicolaas de Bruijn)

{ with {x 5} {+ x x} }

=> {with 5 {+ <: 0 0> <: 0 0>}}

{ with {x 5} {+ x x} }



Representación por índices

<: profundidad posición>

Ejemplo 2: Índices de Bruijn

{ with {x 5}

{ with {y 3}

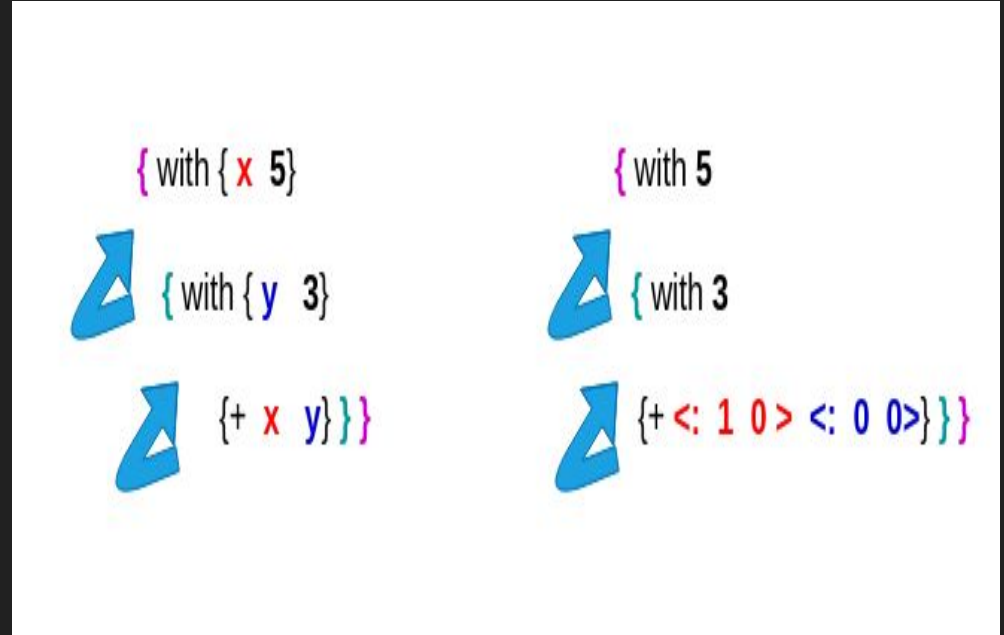
{+ x y} } }

=>

{ with 5

{with 3

{+ <: 1 0> <: 0 0>} } }



Ejemplo 2: Índices de Bruijn

{with { x 5 }

{with 5

{with { y {+ x 3 } }

{with {+ <: 0 0> 3 }

{+ x y} } }


{+ <: 1 0> <: 0 0> } } }


{with { x 5 }

{with { x 5 }

 {with { y {+ x 3 } }

 {with { y {+ <: 0 0> 3 } }

 {+ x y} } }

 {+ <: 1 0> <: 0 0> } } }

Gracias