### Universidad Nacional Autónoma de México Facultad de Ciencias Lenguajes de Programación

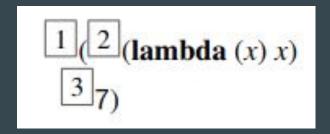
Karla Ramírez Pulido Algoritmo de Unificación

#### Algoritmo de Unificación

- 1. Si X e Y son constantes idénticas, no se hace nada.
- Si X es un identificador, reemplaza todas las ocurrencias de X por Y tanto en el stack como en la sustitución, y añade X → Y en la sustitución.
- Si Y es un identificador, reemplaza todas las ocurrencias de Y por X tanto en el stack como en la sustitución, y añade Y → X en la sustitución.
- 4. Si X es de la forma C(X1, ..., Xn) para algún constructor C, y Y es de la forma
   C(Y1, ..., Yn) (i.e. tienen el mismo constructor), entonces añade Xi=Yi para toda 1
   ≤ i ≤ n, al stack.
- 5. En cualquier otro caso, X e Y no se unifican. Se reporta un error.

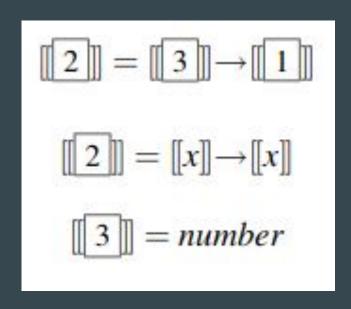
#### 1. Nombramos a las sub-expresiones

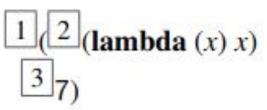
En la siguiente expresión (aplicación de función) tenemos las siguientes sub-expresiones:



2. Generamos las restricciones de tipo asociadas a:

Las restricciones son:





#### Ahora si ejecutamos el algoritmo:

Paso inicial introducimos en el stack las restricciones generadas:

| Action     | Stack  | Substitution |     |
|------------|--|--------------|-----|
| Initialize | $ \begin{bmatrix} 2 \end{bmatrix} = \begin{bmatrix} 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} 2 \end{bmatrix} = \begin{bmatrix} x \end{bmatrix} \rightarrow \begin{bmatrix} x \end{bmatrix} \\ \begin{bmatrix} 3 \end{bmatrix} = number $ | empty        | 175 |

#### Aplicamos paso 2 del algoritmo:

2. Si X es un identificador, reemplaza todas las ocurrencias de X por Y tanto en el stack como en la sustitución, y añade X → Y en la sustitución.

| Action     | Stack  | Substitution                      |
|------------|--|-----------------------------------|
| Initialize | $ \begin{bmatrix} 2 \end{bmatrix} = \begin{bmatrix} 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \end{bmatrix} \\ 2 \end{bmatrix} = \begin{bmatrix} x \end{bmatrix} \rightarrow \begin{bmatrix} x \end{bmatrix} \\ 3 \end{bmatrix} = number $ | empty                             |
| Step 2     | $ \begin{bmatrix} 3 \\ \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ \end{bmatrix} = \begin{bmatrix} x \\ \end{bmatrix} \rightarrow \begin{bmatrix} x \\ \end{bmatrix} $ $ \begin{bmatrix} 3 \\ \end{bmatrix} = number $                      | $[2] \mapsto [3] \rightarrow [1]$ |

#### Ver página 1 del Jamboard

## <u>Jamboard Lenguajes de</u> <u>Programación: tipos</u>

#### Aplicamos paso 4 del algoritmo:

4. Si X es de la forma C(X1, ..., Xn) para algún constructor C, y Y es de la forma C(Y1, ..., Yn) (i.e. tienen el mismo constructor), entonces añade Xi=Yi para toda  $1 \le i \le n$ , al stack.

| Action     | Stack   | Substitution   |
|------------|---|--|
| Initialize | $[\hspace{-0.04cm} 2\hspace{-0.04cm}] = [\hspace{-0.04cm} 3\hspace{-0.04cm}] \rightarrow [\hspace{-0.04cm} 1\hspace{-0.04cm}]$  | empty  |
|            | $[\![2]\!] = [\![x]\!] \rightarrow [\![x]\!]$   |  |
|            | [3] = number  |  |
| Step 2     | $[\hspace{-0.3em} [\hspace{-0.3em} 3\hspace{-0.3em}]\hspace{-0.3em}] \rightarrow [\hspace{-0.3em} [\hspace{-0.3em} 1\hspace{-0.3em}]\hspace{-0.3em}] = [\hspace{-0.3em} [\hspace{-0.3em} x\hspace{-0.3em}]\hspace{-0.3em}] \rightarrow [\hspace{-0.3em} [\hspace{-0.3em} x\hspace{-0.3em}]\hspace{-0.3em}]$ | $[\hspace{-0.2em} \begin{bmatrix} 2 \\ \end{bmatrix}] \mapsto [\hspace{-0.2em} \begin{bmatrix} 3 \\ \end{bmatrix}] \rightarrow [\hspace{-0.2em} \begin{bmatrix} 1 \\ \end{bmatrix}]$ |
|            | [ 3 ] = number  |  |
| Step 4     | $[\![3]\!] = [\![x]\!]$   | $[\hspace{-0.2em} \begin{bmatrix} 2 \\ \end{bmatrix}] \mapsto [\hspace{-0.2em} \begin{bmatrix} 3 \\ \end{bmatrix}] \rightarrow [\hspace{-0.2em} \begin{bmatrix} 1 \\ \end{bmatrix}]$ |
|            | $[\![1]\!] = [\![x]\!]$   |  |
|            | [ 3 ] = number  |  |

Ver página 2 del Jamboard

## <u>Jamboard Lenguajes de</u> <u>Programación: tipos</u>

#### Aplicamos paso 2 del algoritmo:

2. Si X es un identificador, reemplaza todas las ocurrencias de X por Y tanto en el stack como en la sustitución, y añade  $X \mapsto Y$  en la sustitución.

| Action     | Stack  | Substitution  |
|------------|--|---|
| Initialize | $[ \hspace{-0.6em} [\hspace{-0.6em} 2 \hspace{-0.6em} ] \hspace{-0.6em} ] = [ \hspace{-0.6em} [\hspace{-0.6em} 3 \hspace{-0.6em} ] \hspace{-0.6em} ] \rightarrow [ \hspace{-0.6em} [\hspace{-0.6em} 1 \hspace{-0.6em} ] \hspace{-0.6em} ]$ | empty   |
|            | $[\![2]\!] = [\![x]\!] \rightarrow [\![x]\!]$  |   |
|            | [ 3 ] = number   |   |
| Step 2     |  | $[\hspace{-0.04cm}[\hspace{.04cm}2\hspace{.04cm}]\hspace{-0.04cm}] \mapsto [\hspace{-0.04cm}[\hspace{.04cm}3\hspace{.04cm}]\hspace{-0.04cm}] \rightarrow [\hspace{-0.04cm}[\hspace{.04cm}1\hspace{.04cm}]\hspace{-0.04cm}]$ |
|            | [ 3 ] = number   |   |
| Step 4     | $[\hspace{-0.3em}[\hspace{-0.3em}3\hspace{-0.3em}]\hspace{-0.3em}] = [\hspace{-0.3em}[\hspace{-0.3em}x]\hspace{-0.3em}]$   | $[\hspace{-0.2em}[\hspace{.08em}2\hspace{.08em}]\hspace{.08em}] \mapsto [\hspace{-0.2em}[\hspace{.08em}3\hspace{.08em}]\hspace{.08em}] \rightarrow [\hspace{-0.2em}[\hspace{.08em}1\hspace{.08em}]\hspace{.08em}]$          |
|            | $[\![1]\!] = [\![x]\!]$  |   |
|            | [ 3 ] = number   |   |
| Step 2     | $[\![1]\!] = [\![x]\!]$  | $[\hspace{-0.04cm}[\hspace{.04cm}2\hspace{.04cm}]\hspace{.04cm}]\mapsto [\hspace{-0.04cm}[\hspace{.04cm}x]\hspace{.04cm}]\!\rightarrow [\hspace{-0.04cm}[\hspace{.04cm}1\hspace{.04cm}]\hspace{.04cm}]$                     |
|            | [x] = number   | $[\hspace{-1.5pt}[\hspace{.05em}3\hspace{.05em}]\hspace{.05em}]\mapsto\hspace{.05em}[\hspace{.05em}[\hspace{.05em}x\hspace{.05em}]\hspace{.05em}]$  |

#### Aplicamos paso 2 del algoritmo:

2. Si X es un identificador, reemplaza todas las ocurrencias de X por Y tanto en el stack como en la sustitución, y añade X → Y en la sustitución.

| Action     | Stack   | Substitution  |
|------------|---|---|
| Initialize |   | empty   |
|            | $\llbracket 2 \rrbracket = \llbracket x \rrbracket \rightarrow \llbracket x \rrbracket$ |   |
|            | [ 3 ] = number  |   |
| Step 2     |   | $[\hspace{-0.04cm}[\hspace{.04cm}2\hspace{.04cm}]\hspace{-0.04cm}] \mapsto [\hspace{-0.04cm}[\hspace{.04cm}3\hspace{.04cm}]\hspace{-0.04cm}] \rightarrow [\hspace{-0.04cm}[\hspace{.04cm}1\hspace{.04cm}]\hspace{-0.04cm}]$ |
|            | [ 3 ] = number  |   |
| Step 4     |   | $[\hspace{-0.04cm} 2\hspace{-0.04cm}]\hspace{-0.04cm} \mapsto [\hspace{-0.04cm} 3\hspace{-0.04cm}]\hspace{-0.04cm} \to [\hspace{-0.04cm} 1\hspace{-0.04cm}]\hspace{-0.04cm}]$   |
|            | $[\![\overline{1}]\!] = [\![x]\!]$  |   |
|            | $[ \overline{3} ] = number$   |   |
| Step 2     | $[\![1]\!] = [\![x]\!]$   | $[\hspace{-0.04cm}[\hspace{.04cm}2\hspace{.04cm}]\hspace{-0.04cm}]\mapsto [\hspace{-0.04cm}[\hspace{.04cm}x]\hspace{-0.04cm}]\!\rightarrow [\hspace{-0.04cm}[\hspace{.04cm}1\hspace{.04cm}]\hspace{-0.04cm}]$               |
|            | [x] = number  | $[\hspace{-1.5pt}[\hspace{.05em} 3\hspace{.05em}]\hspace{-1.5pt}]\mapsto [\hspace{-1.5pt}[\hspace{.05em} x]\hspace{-1.5pt}]$  |
| Step 2     | [x] = number  | $\llbracket 2 \rrbracket \mapsto \llbracket x \rrbracket \rightarrow \llbracket x \rrbracket$   |
|            |   | $[\![3]\!] \mapsto [\![x]\!]$   |
|            |   | $[\![1]\!] \mapsto [\![x]\!]$   |

#### Aplicando el paso 2:

2. Si X es un identificador, reemplaza todas las ocurrencias de X por Y tanto en el stack como en la sustitución, y añade X → Y en la sustitución.

| Action     | Stack   | Substitution  |
|------------|---|---|
| Initialize |   | empty   |
|            | $[\![2]\!] = [\![x]\!] \rightarrow [\![x]\!]$               |   |
|            | [3] = number  | 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8   |
| Step 2     |   | $[\hspace{-0.07em}[\hspace{.07em}2\hspace{.07em}]\hspace{-0.07em}] \mapsto [\hspace{-0.07em}[\hspace{.07em}3\hspace{.07em}]\hspace{-0.07em}] \rightarrow [\hspace{-0.07em}[\hspace{.07em}1\hspace{.07em}]\hspace{-0.07em}]$ |
|            | [ 3 ] = number  |   |
| Step 4     | $[\![3]\!] = [\![x]\!]$                                     | $[\hspace{-0.07em}[\hspace{.07em}2\hspace{.07em}]\hspace{-0.07em}] \mapsto [\hspace{-0.07em}[\hspace{.07em}3\hspace{.07em}]\hspace{-0.07em}] \rightarrow [\hspace{-0.07em}[\hspace{.07em}1\hspace{.07em}]\hspace{-0.07em}]$ |
|            | $\llbracket 1 \rrbracket = \llbracket x \rrbracket$         |   |
| 9          | [ 3 ] = number  |   |
| Step 2     | $\llbracket \boxed{1} \rrbracket = \llbracket x \rrbracket$ | $[\hspace{-0.2em}[\hspace{.08em}2\hspace{.08em}]\hspace{-0.2em}] \mapsto [\hspace{-0.2em}[\hspace{.08em}x\hspace{.08em}]\hspace{-0.2em}] \rightarrow [\hspace{-0.2em}[\hspace{.08em}1\hspace{.08em}]\hspace{-0.2em}]$       |
|            | [x] = number  | $[\hspace{-1.5pt}[\hspace{.05em} 3\hspace{.05em}]\hspace{.05em}] \mapsto [\hspace{-1.5pt}[\hspace{.05em} x\hspace{.05em}]\hspace{.05em}]$   |
| Step 2     | [x] = number  | $[\![2]\!] \mapsto [\![x]\!] \rightarrow [\![x]\!]$   |
|            |   | $[\![3]\!] \mapsto [\![x]\!]$   |
| 8          |   | $[\![1]\!] \mapsto [\![x]\!]$   |
| Step 2     | empty   | $[2] \mapsto number \rightarrow number$   |
|            | 1 1 1   | $[3] \mapsto number$  |
|            |   | $[1] \mapsto number$  |
|            |   | $[x] \mapsto number$  |

# ¿Dudas?

Gracias