Practica 3

LPM - Laboratorio de programación multiparadigma

MITSS

Sergi Sanz Carreres

Índice:

[4.1. Monty Python Example 2](#_Toc38823175)

[1.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(X) =>\* true .? 2](#_Toc38823176)

[2.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(lola) =>\* true .? 2](#_Toc38823177)

[3.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(lola) ~>\* true . si cambiamos la segunda regla de mismopeso para que compare pato y lola? 3](#_Toc38823178)

[4.2. Listas 4](#_Toc38823179)

[1.- ¿Qué devuelve la ejecución de la expresión rewrite length(nil) .? 4](#_Toc38823180)

[2. ¿Qué devuelve la ejecución de la expresión vu-narrow length(1 : 2 : 3 : 4 : nil) =>! N .? 4](#_Toc38823181)

[3. ¿Qué devuelve la ejecución de la expresión vu-narrow [1] length(NL) =>\* 0 .? 5](#_Toc38823182)

[4. ¿Qué devuelve la ejecución de la expresión vu-narrow [1] length(NL) =>\* 10 .? 5](#_Toc38823183)

[4.3. Suma y comparación 6](#_Toc38823184)

[1.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(X,Y),1) =>\* true .? 6](#_Toc38823185)

[2.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(0,0),s(Y)) =>\* true .? 6](#_Toc38823186)

[3.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(X,Y),s(X)) =>\* true .? 7](#_Toc38823187)

[4.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] sum(X,Y) =:= 1 and leq(X,Y) =>\* true .? 7](#_Toc38823188)

[4.4. Resta 8](#_Toc38823189)

[1. ¿Qué devuelve la expresión vu-narrow [1] minus(s(0),s(s(0))) =>\* X:Nat .? 8](#_Toc38823190)

[2. ¿Qué devuelve la expresión vu-narrow [1] minus(s(s(0)),s(0)) =>\* X:Nat .? 8](#_Toc38823191)

[3. ¿Qué devuelve la expresión vu-narrow [1] minus(X:Nat,s(s(0))) =>\* Y:Nat .? 8](#_Toc38823192)

# **4.1. Monty Python Example**

## 1.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(X) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing knife

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **bruja(X)** a una instancia del término **true**, mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con X, que en este caso no hay.

## 2.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(lola) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing knife

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **bruja(lola)** a una instancia del término **true**, mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con lola, que en este caso no hay.

## 3.- ¿Qué devuelve la ejecución de la expresión vu-narrow bruja(lola) ~>\* true . si cambiamos la segunda regla de mismopeso para que compare pato y lola?

Para la realización de este ejercicio se ha tenido que sustituir la regla:

rl mismopeso(pato,jamon) => true [narrowing] .

Por la regla:

rl mismopeso(pato,lola) => true [narrowing] .

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A close up of text on a white background

Description automatically generated

Donde se observa como =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con true.

# **4.2. Listas**

## 1.- ¿Qué devuelve la ejecución de la expresión rewrite length(nil) .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing knife

Description automatically generated

Al aplicar reescritura se realiza un paso donde obtenemos la largaría de nil que es cero.

## 2. ¿Qué devuelve la ejecución de la expresión vu-narrow length(1 : 2 : 3 : 4 : nil) =>! N .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird, tree

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **length(1:2:3:4:nil)** a una instancia del término **N**, mientras que =>! devuelve todos los posibles estados finales a los que se llega reescribiendo el término inicial que le pasamos y que hacen matching con el patrón de la derecha. Por tanto, obtendremos una solución en la que se nos indica que 4 estados hacen matching, obteniendo como resultado N -- > 4.

## 3. ¿Qué devuelve la ejecución de la expresión vu-narrow [1] length(NL) =>\* 0 .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **length(NL)** a una instancia del término **0**, en este caso se encuentra limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con NL, que en este caso el resultado es nil.

## 4. ¿Qué devuelve la ejecución de la expresión vu-narrow [1] length(NL) =>\* 10 .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A screenshot of a cell phone

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **length(NL)** a una instancia del término **10**, en este caso se encuentra limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con NL, siendo en este caso el resultado es **NL -> @1:Nat : @2:Nat : @3:Nat : @4:Nat : @5:Nat : @6:Nat : @7:Nat : @8:Nat : @9:Nat : @10:Nat : nil**.

# **4.3. Suma y comparación**

## 1.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(X,Y),1) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **sum((X,Y), 1)** a una instancia del término **true**, en este caso se encuentra limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con sum(X,Y),1), siendo en este caso el resultado X -> 0 e Y -> 0.

## 2.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(0,0),s(Y)) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **sum((0,0), s(Y))** a una instancia del término **true**, en este caso se encuentra limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con **sum((0,0), s(Y)),** siendo en este caso el resultado Y -> 0.

## 3.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] leq(sum(X,Y),s(X)) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A screenshot of a tree

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **sum((X,Y), s(X))** a una instancia del término **true**, en este caso se encuentra limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con **sum((X,Y), s(X)),** siendo en este caso el resultado X -> 0 e Y -> 0.

## 4.- ¿Qué devuelve la ejecución de la expresión vu-narrow [1] sum(X,Y) =:= 1 and leq(X,Y) =>\* true .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A screenshot of a cell phone

Description automatically generated

En este caso la expresión **vu-narrow** se encarga de buscar algún camino de ejecución que lleve de una instancia del término **sum(X,Y) =:= 1 and leq(X,Y)**  a una instancia del término **true**, estando limitado a [1] solución. Mientras que =>\* devuelve todos los posibles estados a los que se llega con cero o más pasos de reescritura desde el estado inicial y que hagan matching con el patrón. Por tanto, se devolverá todos los posibles estados que hagan matching con **sum(X,Y) =:= 1 and leq(X,Y),** siendo en este caso el resultado X -> 0 e Y -> 1.

# **4.4. Resta**

El código añadido para posibilitar esta funcionalidad es el siguiente:

op minus : Nat Nat -> [Nat] .

rl minus(X, 0) => X [narrowing] .

rl minus(s(X), s(Y) ) => minus(X, Y) [narrowing] .

## 1. ¿Qué devuelve la expresión vu-narrow [1] minus(s(0),s(s(0))) =>\* X:Nat .?

A picture containing knife

Description automatically generated

Debido a que el s(s(0)) es 2 y el s(0) es 1, nos encontramos con un valor negativo al aplicar la resta, pero ese motivo no se devuelve solución ya que incumple la propiedad de que tienen que ser naturales.

## 2. ¿Qué devuelve la expresión vu-narrow [1] minus(s(s(0)),s(0)) =>\* X:Nat .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird, table

Description automatically generated

En este caso, se aprecia como s(s(0)) obtiene el valor 2, mientras que s(0) obtiene el valor 1, pudiéndose aplicar la resta satisfaciblemente y obteniendo un valor resultante de Nat 1.

## 3. ¿Qué devuelve la expresión vu-narrow [1] minus(X:Nat,s(s(0))) =>\* Y:Nat .?

El resultado obtenido al realizar la ejecución de la expresión es la siguiente:

A picture containing bird

Description automatically generated

En este caso, obtenemos s\_^2(@1:Nat) que significa lo mismo que s(s(@1:Nat) siendo @1:Nat una nueva variable generada por maude, por tanto eso significa que se producirá la resta entre 4 y 3, dando como resultado 1, por ese motivo Y tiene como resultado Y:Nat -> %1:Nat