

Paradigmas Fundamentales de Programación

Técnicas de programación perezosa

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación
Escuela de Ingeniería de Sistemas y Computación,
home page: <http://eisc.univalle.edu.co>
Universidad del Valle - Cali, Colombia

Plan

- 1 Técnicas de programación perezosa
 - Flujos perezosos
 - Ejemplo: el problema de Hamming
 - Operaciones perezosas sobre listas
 - Criba de Eratóstenes perezosa

Plan

- 1 Técnicas de programación perezosa
 - Flujos perezosos
 - Ejemplo: el problema de Hamming
 - Operaciones perezosas sobre listas
 - Criba de Eratóstenes perezosa

Flujos perezosos

Productor/consumidor revisitado

```

fun lazy {Generar N}
  N|{Generar N+1}
end
fun {Suma Xs A Límite}
  if Límite>0 then
    case Xs of
      X|Xr then
        {Suma Xr
          A+X
          Límite-1}
    end
  else A end
end

```

Diferencias ansioso vs. perezoso

```

local Xs S in
  Xs={Generar 0}
  S={Suma Xs 0 150}
  {Browse S}
end

```

- La invocación a *Generar* no necesita colocarse en un hilo propio.
- El consumidor decide cuántos elementos de la lista se deben generar.

Flujos perezosos

Productor/consumidor revisitado

```

fun lazy {Generar N}
  N|{Generar N+1}
end
fun {Suma Xs A Límite}
  if Límite>0 then
    case Xs of
      X|Xr then
        {Suma Xr
          A+X
          Límite-1}
    end
  else A end
end

```

Diferencias ansioso vs. perezoso

```

local Xs S in
  Xs={Generar 0}
  S={Suma Xs 0 150}
  {Browse S}
end

```

- La invocación a `Generar` no necesita colocarse en un hilo propio.
- El consumidor decide cuántos elementos de la lista se deben generar.

Flujos perezosos

Productor/consumidor revisitado

```

fun lazy {Generar N}
  N|{Generar N+1}
end
fun {Suma Xs A Límite}
  if Límite>0 then
    case Xs of
      X|Xr then
        {Suma Xr
          A+X
          Límite-1}
    end
  else A end
end

```

Diferencias ansioso vs. perezoso

```

local Xs S in
  Xs={Generar 0}
  S={Suma Xs 0 150}
  {Browse S}
end

```

- La invocación a `Generar` no necesita colocarse en un hilo propio.
- El consumidor decide cuántos elementos de la lista se deben generar.

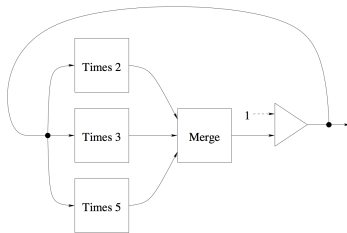
Plan

- 1 Técnicas de programación perezosa
 - Flujos perezosos
 - Ejemplo: el problema de Hamming
 - Operaciones perezosas sobre listas
 - Criba de Eratóstenes perezosa

El problema de Hamming (1)

El problema

- Generar los primeros n enteros de la forma $2^a 3^b 5^c$ con $a, b, c \geq 0$.
- Idea: generarlos en orden ascendente en un flujo potencialmente infinito.
- Sea h el flujo parcial generado. El siguiente elemento es $\min(2x, 3y, 5z)$ donde x (resp. y, z) es el menor elemento de h tal que $2x$ (resp. $3y, 5z$) es más grande que el último elemento de h .
- Inicialmente 1 es el único elemento de h .



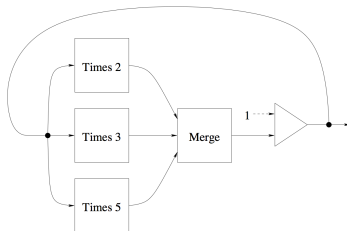
Solución en Oz

```
H=1 | {Mezclar
      {MultEscalar 2 H}
      {Mezclar
        {MultEscalar 3 H}
        {MultEscalar 5 H}
      }
    }
{Browse H}
```


El problema de Hamming (1)

El problema

- Generar los primeros n enteros de la forma $2^a 3^b 5^c$ con $a, b, c \geq 0$.
- Idea: generarlos en orden ascendente en un flujo potencialmente infinito.
- Sea h el flujo parcial generado. El siguiente elemento es $\min(2x, 3y, 5z)$ donde x (resp. y, z) es el menor elemento de h tal que $2x$ (resp. $3y, 5z$) es más grande que el último elemento de h .
- Inicialmente 1 es el único elemento de h .



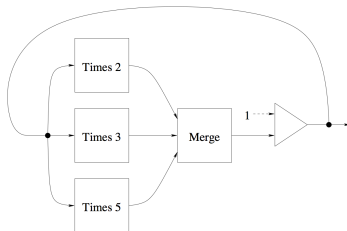
Solución en Oz

```
H=1 | {Mezclar
      {MultEscalar 2
                H}
      {Mezclar
        {MultEscalar 3
                  H}
        {MultEscalar 5
                  H}}
      }
{Browse H}
```

El problema de Hamming (1)

El problema

- Generar los primeros n enteros de la forma $2^a 3^b 5^c$ con $a, b, c \geq 0$.
- Idea: generarlos en orden ascendente en un flujo potencialmente infinito.
- Sea h el flujo parcial generado. El siguiente elemento es $\min(2x, 3y, 5z)$ donde x (resp. y, z) es el menor elemento de h tal que $2x$ (resp. $3y, 5z$) es más grande que el último elemento de h .
- Inicialmente 1 es el único elemento de h .



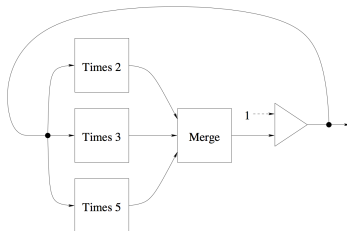
Solución en Oz

```
H=1 | {Mezclar
      {MultEscalar 2
                  H}
      {Mezclar
      {MultEscalar 3
                  H}
      {MultEscalar 5
                  H}}
      }
{Browse H}
```

El problema de Hamming (1)

El problema

- Generar los primeros n enteros de la forma $2^a 3^b 5^c$ con $a, b, c \geq 0$.
- Idea: generarlos en orden ascendente en un flujo potencialmente infinito.
- Sea h el flujo parcial generado. El siguiente elemento es $\min(2x, 3y, 5z)$ donde x (resp. y, z) es el menor elemento de h tal que $2x$ (resp. $3y, 5z$) es más grande que el último elemento de h .
- Inicialmente 1 es el único elemento de h .



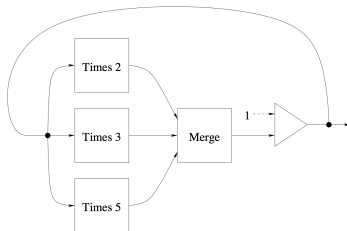
Solución en Oz

```
H=1 | {Mezclar
      {MultEscalar 2
                  H}
      {Mezclar
      {MultEscalar 3
                  H}
      {MultEscalar 5
                  H}}
      }
{Browse H}
```

El problema de Hamming (1)

El problema

- Generar los primeros n enteros de la forma $2^a 3^b 5^c$ con $a, b, c \geq 0$.
- Idea: generarlos en orden ascendente en un flujo potencialmente infinito.
- Sea h el flujo parcial generado. El siguiente elemento es $\min(2x, 3y, 5z)$ donde x (resp. y, z) es el menor elemento de h tal que $2x$ (resp. $3y, 5z$) es más grande que el último elemento de h .
- Inicialmente 1 es el único elemento de h .



Solución en Oz

```

H=1 | {Mezclar
      {MultEscalar 2
                H}

      {Mezclar
        {MultEscalar 3
                  H}

        {MultEscalar 5
                  H} }

      }
{Browse H}

```

El problema de Hamming (2)

```

fun lazy {MultEscalar N H}
  case H of X|H2 then N*X|{MultEscalar N H2} end
end
fun lazy {Mezclar Xs Ys}
  case Xs#Ys of (X|Xr)#(Y|Yr) then
    if X<Y then X|{Mezclar Xr Ys}
    elseif X>Y then Y|{Mezclar Xs Yr}
    else X|{Mezclar Xr Yr}
    end
  end
end
proc {Tocar N H}
  if N>0 then {Tocar N-1 H.2} else skip end
end

```

Plan

- 1 Técnicas de programación perezosa
 - Flujos perezosos
 - Ejemplo: el problema de Hamming
 - Operaciones perezosas sobre listas
 - Criba de Eratóstenes perezosa

Operaciones perezosas sobre listas

Map perezoso

```

fun lazy {LMap Xs F}
  case Xs
  of nil then nil
  [] X|Xr then
    {F X}|{LMap Xr F}
  end
end

```

Listas perezosas de enteros

```

fun lazy {LEntre I J}
  if I>J then nil
  else
    I|{LEntre I+1 J}
  end
end

```

Filtro perezoso

```

fun lazy {LFilter L F}
  case L
  of nil then
    nil
  [] X|L2 then
    if {F X} then
      X|{LFilter L2 F}
    else {LFilter L2 F}
    end
  end
end

```

Operaciones perezosas sobre listas

Map perezoso

```

fun lazy {LMap Xs F}
  case Xs
  of nil then nil
  [] X|Xr then
    {F X}|{LMap Xr F}
  end
end

```

Listas perezosas de enteros

```

fun lazy {LEntre I J}
  if I>J then nil
  else
    I|{LEntre I+1 J}
  end
end

```

Filtro perezoso

```

fun lazy {LFilter L F}
  case L
  of nil then
    nil
  [] X|L2 then
    if {F X} then
      X|{LFilter L2 F}
    else {LFilter L2 F}
    end
  end
end

```


Operaciones perezosas sobre listas

Map perezoso

```

fun lazy {LMap Xs F}
  case Xs
  of nil then nil
  [] X|Xr then
    {F X}|{LMap Xr F}
  end
end

```

Listas perezosas de enteros

```

fun lazy {LEntre I J}
  if I>J then nil
  else
    I|{LEntre I+1 J}
  end
end

```

Filtro perezoso

```

fun lazy {LFilter L F}
  case L
  of nil then
    nil
  [] X|L2 then
    if {F X} then
      X|{LFilter L2 F}
    else {LFilter L2 F}
    end
  end
end

```

Plan

- 1 Técnicas de programación perezosa
 - Flujos perezosos
 - Ejemplo: el problema de Hamming
 - Operaciones perezosas sobre listas
 - Criba de Eratóstenes perezosa

Criba de Eratóstenes perezosa

```
declare  
fun lazy {IntsFrom N}  
  N|{IntsFrom N+1}  
end  
fun lazy {Sieve Xs}  
  X|Xr=Xs in  
  X | {Sieve  
      {LFilter Xr  
        fun {$ Y} Y mod X \=0 end }  
      }  
end  
Primes={Sieve {IntsFrom 2}}  
{Browse Primes}  
{Browse {List.nth Primes 16}}
```