Elementos de Teoría de la Computación
Clase 2: Definiciones Recursivas

Depart. de Teoría de la Computación - Facultad de Informática
Universidad Nacional del Comahue
Segundo Cuatrimestre de 2020

Gonzalo Heffesse
`gheffesse@fi.uncoma.edu.ar`

Contenido

¿Cuantos somos en la fila?

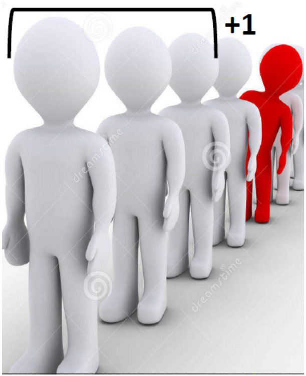
- Este es “Rogelio”.
- Rogelio fue a pagar sus facturas a último momento y muy cerca de la hora de cierre.
- Necesita saber si llegará a la caja antes de “quedarse sin sistema”, es decir necesita saber **cuantos son en la fila**...
- Pero sin poder ver mas allá que la persona de adelante y sin poder salir de la fila!

¿Cuantos somos en la fila?

- Solución:
- Preguntarle al de adelante cuantos son!...
- porque la "**fila** completa" hasta él está compuesta por:
  - la "**fila** de adelante" de Rogelio...
  - mas Rogelio!

¿Cuantos somos en la fila?

- Pero la persona de adelante tampoco puede ver mas allá de la persona que tiene adelante suyo...



- Solución?...
- Preguntarle al de adelante cuantos son!
- Porque la fila hasta ahí es:
  - la "**fila** de adelante" del "preguntado"
  - mas "el preguntado"!

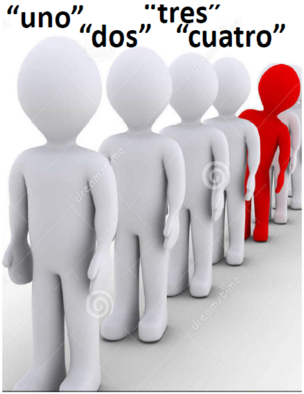
¿Cuantos somos en la fila?

- Eventualmente seguirán preguntando hacia adelante hasta llegar...
- al ***PRIMERO*** de la fila!

- Que es el único que **no tiene a quien preguntar**...
- y que ya **puede responder** la pregunta que le hicieron

¿Cuantos somos en la fila?

- Ahora que "alguien" respondió, cada uno de los que preguntaron puede sucesivamente responder a la pregunta que le hicieron...



¿Cuantos somos en la fila?

Ahora que "alguien" respondió, cada uno de los que preguntaron puede sucesivamente responder a la pregunta que le hicieron...

- hasta quien hizo la pregunta original...Rogelio!.

¿Cuántos somos?

Que hicimos?

- Hicimos una pregunta.
- Hicimos la misma pregunta a la persona de adelante, esto es, **repetimos el procedimiento** con una **muestra más chica** del **mismo problema**.
- Solo una persona de la fila no hizo la pregunta.

  EL PRIMERO

- Cuando obtuvimos la respuesta del de adelante, procesamos la información y luego respondimos.

¿Qué es una Definición Recursiva?

- Lo que definimos aparece como parte de la definición.
- La definición se aplica a **muestras cada vez más chicas.**
- Tiene que tener cota inferior. **Caso Base**.
- La solución se construye a partir de los casos anteriores.

Definición Recursiva de ¿Cuántos somos en la fila?

Queremos saber ¿Qué longitud tiene la fila hasta una persona determinada?

LongFila(Pers1) = 1
LongFila(PersN) = LongFila(PersN-1) +1

¿Qué es una Definición Recursiva?

Definición Recursiva

Llamamos Definición Recursiva o Definición Inductiva a aquella definición en la cual el ítem que está siendo definido aparece como parte de la definición.

¿Qué es una Definición Recursiva?

Partes de una Definición Recursiva

1. Una base, donde algunos casos simples del ítem que está siendo definido son dados explícitamente.
2. Un paso recursivo o inductivo, donde nuevos casos del ítem siendo definido pueden ser obtenidos en términos de los casos previos.

La Recursión es una idea muy importante que puede ser usada para definir secuencias de objetos, conjuntos de objetos y operaciones sobre objetos. Además, algunos algoritmos pueden ser recursivos.

Secuencias Definidas por Recursión

Secuencia

Una Secuencia S es una lista de objetos que están enumerados en algún orden.

Esto es, hay un primer elemento, un segundo, etc.

Mediante $S(k)$ se denota el k-ésimo elemento de la secuencia.

Secuencias Definidas por Recursión

Secuencia Definida Recursivamente

Una secuencia se define por recursión indicando explícitamente:

- el primer valor (o los primeros valores) de la secuencia y
- luego definiendo los valores posteriores en términos de los casos previos.

Ejemplo: Sea la secuencia S definida recursivamente por:

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

¿Qué valores están en la secuencia?

Secuencias Definidas por Recursión - Ejemplo

Ejemplo: Sea la secuencia S definida recursivamente por:

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

De acuerdo al punto 1, $S(1)$, el primer objeto de S es 2.

Por el punto 2, el segundo objeto de S es

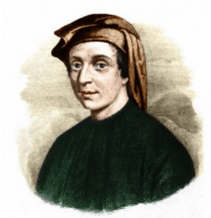
$$S(2) = 2S(1) = 2(2) = 4$$

Continuando de esta manera, podemos comprobar que S es la secuencia

$$2, 4, 8, 16, 32, \ldots$$

Una regla como la del punto (2) del Ejemplo, es denominada Relación de Recurrencia.

Repaso: Secuencia de Fibonacci



L. Fibonacci (1170-1250)

$F(1) = 1$
$F(2) = 1$
$F(n) = F(n-1) + F(n-2)$ para $n > 2$

$F(3) = F(2) + F(1) = 1 + 1 = 2$
$F(4) = F(3) + F(2) = 2 + 1 = 3$
...

Los primeros números de la Secuencia de Fibonacci son:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

Conjuntos Definidos por Recursión

Los objetos de una secuencia están ordenados (existe un primer elemento, un segundo, etc.).

Un conjunto es una colección de objetos sin un orden explícito. Algunos conjuntos también pueden ser definidos recursivamente.

Ejemplo: El conjunto de todas las fórmulas bien formadas (*fbf*) de la Lógica Proposicional.

1. Caso Base: toda letra de sentencia ($A, B, \ldots, Z$) es una *fbf*.
2. Caso Inductivo: si α y β son *fbfs*, entonces $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \rightarrow \beta$, $\sim\alpha$, y $\alpha \leftrightarrow \beta$ son también *fbfs*.

Conjunto FBF	α	β	α ∧ β	α ∨ β	α → β	~ α	α ↔ β
A,B,C,D,E,F….,Z	A	D	A ∧ D	A ∨ D	A → D	~A	A ↔ D
A,B,C,D,E….,Z, A∧D, A∨D, A→D, ~A, A↔D	~A	F	~A ∧ F	~A ∨ F	~A → F	~~A	~A ↔ F
A,B,C,D,E,F….,Z, A∧D, A∨D, A→D, ~A, A↔D ~A ∧ F, ~A ∨ F, ~A → F, ~~A, ~A ↔ F	A→D	~A∨F	A→D ∧ ~A∨F	A→D ∨ ~A∨F	A→D → ~A ∨F	~ A→D	A→D ↔ ~A∨F
A,B,C,D,E,F….,Z, A∧D, A∨D, A→D, ~A, A↔D ~A ∧ F, ~A ∨ F, ~A → F, ~~A, ~A ↔ F A→D ∧ ~A∨F, A→D ∨ ~A∨F, A→D → ~A ∨F, ~ A→D A→D ↔ ~A∨F							

Operaciones Definidas por Recursión

Ciertas operaciones sobre objetos también puede ser definidas recursivamente.

Ejemplo: la potencia a^n para un número real a distinto de cero y un entero positivo n:

- $2^4 = 2x2x2x2 = 2 \text{ x } (2x2x2) = 2 \text{ x } 2^3$
- $2^3 = 2x2x2 = 2 \text{ x } (2x2) = 2 \text{ x } 2^2$
- $2^2 = 2x2 = 2 \text{ x } 2^1$
- $2^1 = 2 = 2x1 = 2 \text{ x } 2^0$
- $2^0 = 1$

Operaciones Definidas por Recursión

Ciertas operaciones sobre objetos también puede ser definidas recursivamente.

Ejemplo 1: Dado un número real a distinto de cero y un entero positivo n, la definición recursiva de la potencia a^n es:

1. $a^0 = 1$
2. $a^n = (a^{n-1}) \cdot a$ para $n \geq 1$

Ejemplo 2: La definición recursiva del producto de dos enteros positivos m y n es:

1. $m \cdot 0 = 0$
2. $m \cdot n = m \cdot (n-1) + m$ para $n \geq 1$

Algoritmos Definidos por Recursión

En general, una relación de recurrencia puede ser implementada mediante un Algoritmo Iterativo o mediante un Algoritmo Recursivo.

Ejemplo: Sea la secuencia S definida recursivamente por:

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

Enfoque Iterativo

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

Algoritmo

S(entero n)
;función que iterativamente computa el valor de $S(n)$

Variables locales:
entero i, *ValorActual*

```
if n = 1 then
  Retornar 2
else
  i := 2
  ValorActual := 2
  while i ≤ n do
    ValorActual := 2 * ValorActual
    i := i + 1
  end while
  Retornar ValorActual
end if
end S
```

Enfoque Recursivo

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

Algoritmo

S(entero n)
;función que recursivamente computa el valor de $S(n)$

if $n = 1$ **then**
  Retornar 2
else
  Retornar 2 * $S(n-1)$
end if
end S

$$
\begin{array}{rclcl}
S(5) & = & 2 \cdot S(4) & = & 2 \cdot 2 \cdot S(3) & = \\
 & = & 2 \cdot 2 \cdot 2 \cdot S(2) & = & 2 \cdot 2 \cdot 2 \cdot 2 \cdot S(1) & = \\
 & = & 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 & = & 32 &
\end{array}
$$

Bibliografía

Mathematical Structures for Computer Science

J. Gersting

Sixth edition

2007

Desafíos YA :)

- Defina recursivamente las potencias de 2:
  1. Como SECUENCIA
  2. Como CONJUNTO