# Capítulo 2

# Práctica 2. Recurrencia.

## 2.1. Sucesiones predefinidas en maxima

Algunas sucesiones están predefinidas en maxima. Para usar las siguientes funciones es necesario cargar el paquete "functs"

**arithmetic (a, d, n)** Devuelve el n-ésimo término de la progresión aritmética $a$, $a+d$, $a+2d$, $\ldots$, $a+(n-1)d$.

**geometric (a, r, n)** Devuelve el n-ésimo término de la progresión geométrica $a$, $ar$, $ar^2$, $\ldots$ , $ar^{n-1}$.

**harmonic (a, b, c, n)** Devuelve el n-ésimo término de la progresión armónica $a/b$, $a/(b+c)$, $a/(b+2c)$, $\ldots$, $a/(b+(n-1)c)$.

**arithsum (a, d, n)** Devuelve la suma de la progresión aritmética hasta el n-ésimo término.

**geosum (a, r, n)** Devuelve la suma de la sucesión geométrica hasta el n-ésimo término. Si n es infinito (inf) la suma será finita sólo si el valor absoluto de r es menor que 1.

Para calcular los 10 primeros términos de geosum (sucesión cuyo elemento n-ésimo es la suma de los n primeros términos de una sucesión geométrica) cuando la razón de la sucesión geométrica es 1/4:

```
(%ixx) load(functs);

 Warning - you are redefining the Maxima function lcm
```

(%oxx) $/usr/share/maxima/5{,}20{,}1/share/simplification/functs.mac$

```
(%ixx) makelist(geosum(1,1/4,i),i,1,10);
```

(%oxx) $[1, \frac{5}{4}, \frac{21}{16}, \frac{85}{64}, \frac{341}{256}, \frac{1365}{1024}, \frac{5461}{4096}, \frac{21845}{16384}, \frac{87381}{65536}, \frac{349525}{262144}]$

## 2.2. Recurrencia

Una sucesión está definida por recurrencia si la descripción de un término, $a_n$, viene dada como función de un número determinado de términos anteriores: $a_{n-1}, \ldots a_{n-t}$. Entonces tendremos una ecuación de la forma:

$$a[n] = f_1 * a[n-1] + ... + f_t * a[n-t] + f_{t+1}$$

Si $f_{t+1} = 0$ decimos que es una **relación de recurrencia homogénea**.

Al número t le llamamos **grado de la relación de recurrencia**.

Si cada $f_i$ es una constante, decimos que la relación de recurrencia tiene **coeficientes constantes**. Para que una sucesión dada por una relación de recurrencia de grado t esté completamente determinada es necesario dar t **valores iniciales**, es decir, t valores consecutivos de la sucesión, que nos permitan calcular el siguiente elemento utilizando la relación.

Un ejemplo típico de sucesión definida por recurrencia es la sucesión de Fibonacci:

$$a_n = a_{n-1} + a_{n-2}$$
$$a_0 = 0;$$
$$a_1 = 1;$$

que viene predefinida en maxima con estos valores iniciales.
Observamos que es una relación de recurrencia **homogénea**, de **grado 2** (por eso requiere 2 valores iniciales) y tiene **coeficientes constantes**.

```
(%ixx) fib(6);
 (%oxx) 8

(%ixx) makelist(fib(i),i,0,10);
 (%oxx) [0,1,1,2,3,5,8,13,21,34,55]
```

Podemos definir sucesiones por recurrencia en maxima. La siguiente es un ejemplo que tiene grado 1, es no homogénea y no tiene coeficientes constantes:

```
   (%ixx) A0 : 4; /* valor inicial */
 A [n] := if n > 0 then F (n)*A [n-1]+G(n) else A0; /* relación de grado 1 */
 F (n) := n^2; /* no es de coeficientes constantes */
G (n) := 2*n+1; /* no es homogénea*/
```

Observamos que mientras que en la definición de F(n) usamos paréntesis, en la de A[n] se utilizan corchetes. Esto indica a maxima que tiene que memorizar los valores que ya han sido calculados para obtener el siguiente (se usa el término **memoizing** para referirse a este hecho en programación), es decir, que A es una función definida por recurrencia, mientras que F o G son funciones explícitas.

```
(%ixx) makelist(A[n],n,0,4);

(%oxx) [4,7,33,304,4873]
```

**Ejercicio:** *Halla una relación de recurrencia que genere la siguiente sucesión:*

$$\{1, 2, 5, 12, 29, 70, 169, \dots\}$$

*y define la correspondiente función recursiva en maxima. Calcula los primeros 7 términos y comprueba que tu propuesta es acertada.*

## 2.3. Resolución de relaciones de recurrencia

Una relación de recurrencia, o ecuación de recurrencia, es una igualdad en la que se relaciona un término de una sucesión $a_n$ con algunos de los términos que lo preceden. Una solución de una relación de recurrencia es una sucesión que verifique dicha relación.

Maxima utiliza el paquete solve_rec para la resolución de ecuaciones de recurrencia. Supongamos que tenemos la relación

$$a_n = a_{n-1} + a_{n-2}$$
$$a_1 = 1$$
$$a_2 = 1$$

entonces ejecutamos:

(%ixx) load(solve_rec);
(%oxx) /usr/share/maxima/5.20.1/share/contrib/solve_rec/solve_rec.mac
(%ixx) solve_rec(a[k]=a[k-1]+a[k-2], a[k], a[1]=1,a[2]=1);

(%oxx) $a[k] = \frac{(\sqrt{5}+1)^k}{\sqrt{5}2^k} - \frac{(\sqrt{5}-1)^k(-1)^k}{\sqrt{5}2^k}$

también puede utilizarse para calcular una solución general, es decir, cuando no se fijan los valores iniciales

```
(%ixx) solve_rec(a[k]=c*a[k-1],a[k]);
```

(%oxx) $a[k] = \%k_1 * c^k$

donde $\%k_1$ es un parámetro que puede ser determinado imponiendo condiciones iniciales. La que acabamos de obtener es la solución general para una relación de recurrencia de grado 1, homogénea y con coeficientes constantes.

**Ejercicio:** *Halla la solución para la relación de recurrencia que has propuesto en el ejercicio de la sección anterior, con las condiciones iniciales correspondientes.*

**Ejercicios:** *Decide si son lineales, homogéneas y de coeficientes constantes las siguientes relaciones. Resuelvelas con los valores iniciales que se proporcionan*

1. $a_n = 2na_{n-1}$; $a_1 = 1$;
2. $a_n = 7a_{n-2} - 6a_{n-3}$; $a_1 = 0; a2 = 1; a_3 = 1$
3. $a_n = 2na_{n-1}$; $a_1 = 1$

**Ejercicio:** A continuación obtenemos las soluciones de otros tipos de relaciones con coeficientes constantes. Decide de qué tipo son.

```
(%ixx) solve_rec(a[k]=c*a[k-1]+b,a[k]);
(%ixx) solve_rec(a[k]=c*a[k-2]+b*a[k-1], a[k]);
(%ixx) solve_rec(a[k]=c*a[k-2]+b*a[k-1]+d, a[k]);
(%ixx) solve_rec(a[k]=k*a[k-1], a[k]);
(%ixx) solve_rec(a[k]=a[k-1]+(k-1), a[k]);
(%ixx) solve_rec(a[k]*a[k-1]=c, a[k]);
(%ixx) solve_rec(a[k]=a[k-1]*a[k-2], a[k]);
```

**Ejercicio:** Para las relaciones anteriores que sean lineales y homogéneas, calcula el polinomio asociado, las raíces de éste (usa el comando solve) y observa la relación entre estas raíces y la solución general.