2 ECUACIONES DE RECURRENCIA

Una ecuación de recurrencia es una expresión finita que define implícitamente una sucesión, en la cual un elemento de la sucesión se determina por medio de otros elementos más sencillos, que incluyen casos iniciales o básicos. La conexión con el análisis de algoritmos estriba en que la forma que se ha adoptado para medir las complejidades, utiliza funciones cuyo dominio son los números naturales, o en otras palabras, sucesiones. Si el algoritmo es recurrente, es de esperarse que las complejidades, como funciones que estiman la demanda de recursos a lo largo de la ejecución, sean sucesiones que satisfacen ciertas ecuaciones de recurrencia.

Solucionar una ecuación de recurrencia consiste en encontrar una expresión cerrada para una sucesión que satisfaga la ecuación, i.e., una expresión en la que los valores de los elementos de la sucesión no dependan de otros valores de la sucesión. Para el caso del análisis de algoritmos, es deseable contar con esta clase de expresiones cerradas, puesto que, a partir de formulaciones recurrentes para funciones de complejidad, resulta difícil establecer órdenes de crecimiento asintótico correspondientes.

La solución de ecuaciones de recurrencia es de suyo importante para la informática, si se piensa en que las matemáticas que esta disciplina utiliza son eminentemente constructivas. Muchos objetos informáticos son definidos de manera recurrente y los razonamientos inductivos son piedra angular para desarrollos teóricos y prácticos, utilizados de forma natural y, en ocasiones, incluso inconscientemente. En este capítulo se estudiarán métodos para solucionar algunas clases sencillas de ecuaciones de recurrencia, como son la mayoría que ocurren en el análisis de algoritmos.

Una observación, que puede simplificar en gran parte la comprensión de las ideas que aquí se exponen, es el paralelo existente de la teoría de ecuaciones de recurrencia con la de teoría clásica de ecuaciones diferenciales. Se puede notar que los desarrollos en ambas teorías guardan similitudes evidentes, v.gr., según se trate de ecuaciones lineales, homogéneas y no homogéneas, ecuaciones no lineales. Como es de esperarse, la consciencia de estas similitudes hará más sencillos el entendimiento y uso de una de las teorías si esto ya se tiene para la otra. Por otra parte, cabe advertir que, de igual manera que pueden encontrarse ecuaciones diferenciales muy difíciles de solucionar, i.e., para las que se desconoce un procedimiento sistemático de solución, es concebible encontrar ecuaciones de recurrencia cuya solución esté fuera del alcance de los métodos que aquí se exponen. O incluso, puede ser imposible encontrar soluciones por métodos algorítmicos.

2.1 DEFINICIONES BASICAS

Una *sucesión (de* X*)* es una función cuyo dominio son los números naturales, i.e., x: **nat** $\rightarrow$ **X**. Un *elemento (de una sucesión)* x es cualquier valor x(n), también denotado x_n [6]. El codominio X suele ser un conjunto numérico, típicamente **R**, el conjunto de los números reales. El conjunto de las sucesiones de X se denota $\langle X \rangle$. Para referirse a una sucesión como un todo se utiliza la notación $\langle x_n \rangle$.

La definición de una sucesión $\langle x_n \rangle$ debe ser finita y consistente. En consecuencia, una definición es una descripción finita de un método que permita determinar, después de un número finito de pasos, un único valor para cualquier elemento x_n, $n \in$ **nat**.

Una definición para una sucesión se llamará *cerrada* (o bien, *no recurrente*), si el valor de cada elemento x_n se determina independientemente de los valores de otros elementos. Por ejemplo, una definición cerrada para la sucesión de los cuadrados de los números naturales:

$$c_n = n^2, \text{ para } n \geq 0.$$

En cambio, una definición *recurrente* expresa los valores de algunos elementos en términos de los de otros, usualmente anteriores en el orden de **nat**. Por supuesto, debe haber elementos, llamados *iniciales* o *básicos* , cuya definición particular es no recurrente. Como ejemplo, la siguiente es una definición recurrente para la sucesión de los cuadrados de los números naturales:

$$c_0 = 0$$

$$c_n = c_{n-1} + 2n - 1 \quad , \text{ para } n \geq 1.$$

Una *ecuación de recurrencia* no es otra cosa que una definición recurrente de una sucesión (*incógnita*) que debe determinarse. En realidad, el concepto puede entenderse de manera algo más general, si se piensa que al plantear una ecuación es factible que varias sucesiones puedan satisfacer la recurrencia en cuestión. De este modo, una ecuación de recurrencia representa un conjunto de posibles sucesiones que la cumplen.

Una *solución (de tipo* X*)* para una ecuación de recurrencia es una definición cerrada de una sucesión de elementos en X que la satisface. Llamaremos soluciones naturales (resp. enteras, reales, complejas) aquellas de tipo **nat** (resp. **Z**, **R**, **C**). Como ya se anotó, es factible que una ecuación tenga más de una solución. Por ejemplo, la ecuación

$$x_n = x_{n-1} + 2n - 1 \quad , \text{ para } n \geq 1,$$

tiene infinitas soluciones. Si se buscan aquellas cuyos elementos sean números naturales, las soluciones son de la forma:

$$x_n = n^2 + a \qquad , \text{ para } n \geq 0,$$

donde a es cualquier número natural. En especial, cuando $a=0$, se obtiene la solución particular

$$x_n = n^2 \qquad , \text{ para } n \geq 0.$$

Por otra parte, nada garantiza que una ecuación de recurrencia deba tener al menos una solución, bien sea porque se está interesado en soluciones sobre un codominio X específico, o porque la recurrencia impone condiciones que ninguna sucesión puede satisfacer. Por ejemplo, la ecuación:

$$x_0 = 1$$

$$x_n^2 = -x_{n-1} \quad , \text{ para } n \geq 1$$

[6] En ocasiones se llama 'sucesión' a x(**nat**), el conjunto de sus elementos.

no tiene soluciones de valores reales (aunque sí de valores complejos). En cambio, la ecuación:

$$x_n = x_n + 1 \quad , \text{ para } n \geq 1$$

no tiene soluciones de ningún tipo.

Un *sistema de ecuaciones de recurrencia* es un conjunto de ecuaciones de recurrencia. Una *solución* de un sistema de ecuaciones es un conjunto de sucesiones que satisfacen las ecuaciones que lo conforman. De nuevo, un sistema puede tener ninguna, una o varias soluciones.

Por ejemplo, el sistema:

$$a_n = a_{n-1} + 2b_{n-1} \quad , \text{ para } n \geq 1$$

$$b_0 = 1$$

$$b_{n+1} = 2b_n \quad , \text{ para } n \geq 0$$

tiene como soluciones reales (A es cualquier constante) :

$$a_n = A + 2^{n+1} \quad , \text{ para } n \geq 0$$

$$b_n = 2^n \quad , \text{ para } n \geq 0.$$

2.2 EJEMPLOS EN ANÁLISIS DE ALGORITMOS

En general, la dificultad en el análisis de complejidad de un algoritmo es directamente proporcional a lo complicada que sea la definición misma del algoritmo. La misma idea, expresada de una manera positiva, lleva a la conclusión de que un algoritmo simple de expresar será, así mismo, simple de analizar.

Enseguida se presentan tres ejemplos sencillos, y en cierto modo representativos, de la clase de recurrencias que se pueden dar en el análisis de algoritmos.

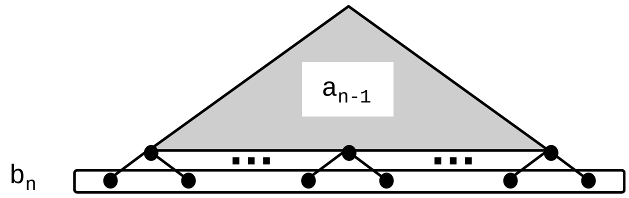
2.2.1 Conteo de nodos en árboles binarios

Los árboles son estructuras de datos cuya definición es eminentemente recurrente. Es de esperarse que las preguntas que se hagan sobre ellos puedan plantearse y resolverse, en forma natural, mediante algoritmos recurrentes.

Supóngase que se desea recorrer un árbol binario perfecto[7] de altura n, haciendo en cada nodo una visita de costo constante, y que el costo de viajar de un nodo a otro es despreciable. Nótese que el costo de un algoritmo que resuelva este problema está directamente relacionado con a_n, el número de nodos de un árbol binario perfecto de altura n.

La figura siguiente sugiere una ecuación de recurrencia para a_n:

[7] Es decir: un nodo tiene 0 ó 2 hijos y todas las hojas tienen igual altura.



Así, si b_n denota el número de nodos en el último nivel de un árbol binario perfecto de altura n, deberá valer el sistema de ecuaciones de recurrencia:

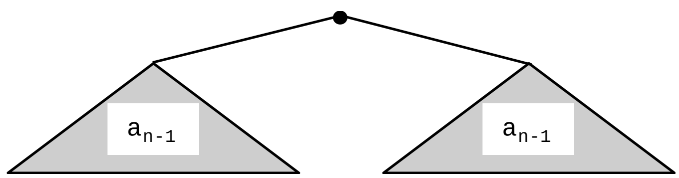
$$b_0 = 1$$

$$b_n = 2b_{n-1} \quad , \text{ para } n \geq 1$$

$$a_0 = 1$$

$$a_n = a_{n-1} + b_n \quad , \text{ para } n \geq 1.$$

Otra manera de visualizar el conteo de a_n, resulta de considerar un árbol binario perfecto de altura n como el nodo raíz que tiene como subárboles dos árboles binarios perfectos, cada uno de altura $n-1$. Gráficamente:



Entonces se llega a la ecuación de recurrencia:

$$a_0 = 1$$

$$a_n = 1 + 2a_{n-1} \quad , \text{ para } n \geq 1.$$

Siempre se puede intentar resolver una ecuación expandiendo la recurrencia, de la siguiente forma:

$$\begin{aligned} a_n &= 1 + 2a_{n-1} \\ &= 1 + 2(1 + 2a_{n-2}) = 1 + 2 + 4a_{n-2} \\ &= 1 + 2 + 4(1 + 2a_{n-3}) = 1 + 2 + 4 + 8a_{n-3} \\ &\ldots \\ &= 1 + 2 + 8 + \ldots + 2^n a_0 \\ &= \langle +: 0 \leq k \leq n: 2^k \rangle \\ &= 2^{n+1} - 1 \end{aligned}$$

La anterior relación se cumple si $n \geq 1$. Pero, como el resultado también vale para $n=0$, se tiene la solución:

$$a_n = 2^{n+1} - 1 \quad , \text{ para } n \geq 0.$$

Para este ejemplo se ha tenido la suerte de encontrar que la última expansión es una expresión conocida (una suma geométrica), de manera que la expresión cerrada para a_n se determina con facilidad.

2.2.2 La sucesión de Fibonacci

El siguiente problema, debido a Leonardo de Pisa (o Leonardo Fibonacci), pretende modelar la reproducción de conejos mediante un esquema sencillo de suposiciones:

Supóngase que los conejos se reproducen de acuerdo a las siguientes reglas:

- Cada pareja de conejos fértiles tiene una pareja de conejitos cada mes.
- Cada pareja de conejos comienza a ser fértil a partir del segundo mes de vida.
- Ninguna pareja de conejos muere ni deja de reproducirse.

Un granjero, que conoce estos hábitos de los conejos, decide construir una conejera y coloca en ella, en el primer mes, una pareja de conejos recién nacidos. ¿Cuántas parejas de conejos habrá en el mes n?

Para tratar de contestar la pregunta definimos las siguientes variables discretas, para $n \geq 0$:
F_n = población total de parejas de conejos en el mes n,
N_n = número de parejas de conejos nacidos en el mes n, y
V_n = número de parejas de conejos adultos en el mes n.

Del enunciado del problema se obtienen las siguientes relaciones, para $n \geq 0$:

$$(1) \quad F_n = N_n + V_n$$

$$(2) \quad N_{n+1} = V_n$$

$$(3) \quad V_{n+1} = N_n + V_n.$$

De (1) y (3) se obtiene, para $n \geq 0$:

$$(4) \quad V_{n+1} = F_n$$

De esta forma, para $n \geq 0$:

$$\begin{aligned} F_{n+2} &= N_{n+2} + V_{n+2} && \text{, por (1)} \\ &= V_{n+1} + V_{n+2} && \text{, por (2)} \\ &= F_n + F_{n+1} && \text{, por (4).} \end{aligned}$$

Así, para calcular el número total de parejas de conejos en un mes dado, basta conocer el mismo dato en los dos meses anteriores. Esto es cierto del segundo mes en adelante. Como en el mes 0 no hay conejos y en el mes 1 se coloca una pareja en la conejera, se tiene que:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-2} + F_{n-1} \quad \text{, para } n \geq 2.$$

La ecuación anterior define la llamada *sucesión de Fibonacci* . Una primera pregunta surge al tratar de encontrar una solución para esta recurrencia. Por otra parte, se puede pensar en escribir un algoritmo que, para $n \geq 0$, calcule el valor de F_n. Para este algoritmo, resulta interesante conocer su complejidad temporal.

Lo más sencillo es diseñar una función recurrente, cuya corrección está justamente garantizada por la ecuación que define $\langle F_n \rangle$:

```
function fib (n:nat):nat
{Pos:    fib = Fn }
```

```
⌈ if     n = 0    →   fib:= 0
  []     n = 1    →   fib:= 1
  []     n ≥ 2    →   fib:= fib(n-1) + fib(n-2)
  fi
⌋
```

Si sólo la suma se considera como operación básica, se tendrá la siguiente ecuación de recurencia para $T_{fib}(n)$:

$$
\begin{aligned}
T_{fib}(n) &= 0 && \text{, para } 0 \le n < 2\\
&= T_{fib}(n-1) + T_{fib}(n-2) + 1 && \text{, para } n \ge 2.
\end{aligned}
$$

Para tener una idea de una forma cerrada para $T_{fib}(n)$, se pueden tabular sus primeros valores y compararlos con los de F_n. De aquí se puede proponer y mostrar por inducción que

$$T_{fib}(n) = F_{n+1} - 1 \quad \text{, para } n \ge 0.$$

En otras palabras,

$$T_{fib}(n) = O(F_{n+1}).$$

Como una variante puede presentarse un algoritmo de agenda basado en `fib`, i.e., una versión iterativa de la función recurrente (cf. [Car91]):

```
function fibit (n:nat):nat
{Pos:    fibit = Fn }
⌈ var    agenda   : sequ nat;
         i, m : nat
⌈ m, agenda:= 0,⟨n⟩;
  {Inv: Fn = m + ⟨+ k: k∈agenda: Fk⟩ }
  do  agenda ≠ ⟨⟩   →   i,agenda:= head(agenda),tail(agenda);
                        if   i=0  →   skip
                         []  i=1  →   m:= m + 1;
                         []  i≥2  →   agenda:= agenda & ⟨i-1,i-2⟩
                        fi
  od;
  fibit:= m
⌋
```

En este caso, se tiene la recurrencia:

$$
\begin{aligned}
T_{fibit}(n) &= 0 && \text{, para } n=0\\
&= 1 && \text{, para } n=1\\
&= T_{fibit}(n-1) + T_{fibit}(n-2) && \text{, para } n \ge 2.
\end{aligned}
$$

Es decir, la sucesión $\langle T_{fibit}(n)\rangle$ soluciona la ecuación de Fibonacci, i.e.,

$$T_{fibit}(n) = F_n \ , \text{ para } n \geq 0.$$

De este modo, se obtiene que, también:

$$T_{fibit}(n) = O(F_n).$$

Más adelante se mostrará cómo se soluciona la ecuación de Fibonacci, lo que entonces completará el análisis de `fib` y `fibit` (V. 2.3) .

2.2.3 Ordenamiento por intercalamiento

El ordenamiento por intercalamiento (inglés: *mergesort*), consiste en partir una lista de entrada de `n` elementos en dos mitades, ordenar cada mitad e intercalar los resultados. Al hacer el análisis de complejidad en el peor caso, suponiendo que la operación de intercalamiento toma a lo sumo `n-1` comparaciones, se obtiene la ecuación

$$\begin{aligned} T_m(n) &= 0 && , \text{ si } n \leq 1 \\ &= T_m(\lfloor \tfrac{n}{2} \rfloor) + T_m(\lceil \tfrac{n}{2} \rceil) + n - 1 && , \text{ si } n > 1 \end{aligned}$$

Aunque en este momento no es formalmente fácil de demostrar, es intuitivamente plausible que:

$$T_m(n) = \theta(M(n))$$

donde

$$\begin{aligned} M(n) &= 0 && , \text{ si } n \leq 1 \\ &= 2\, M(\tfrac{n}{2}) + n - 1 && , \text{ si } n > 1 \end{aligned}$$

Una solución para esta última ecuación deberá ser una sucesión que cumpla las restricciones anotadas para los argumentos en que la división por 2 tenga sentido; obsérvese que, para otros argumentos no se afirma nada. Aparentemente, resolver de esta clase de ecuaciones parece una tarea complicada, pero más adelante se verá un método algebraico relativamente general para encontrar soluciones.

Más generalmente, al analizar algoritmos recurrentes se presentan con frecuencia ecuaciones de la forma:

$$\begin{aligned} T(n) &= f(n) && , \text{ si } n \leq n_0 \\ &= k\, T(\tfrac{n}{k}) + g(n) && , \text{ si } n > n_0. \end{aligned}$$

La situación se presenta cuando se utilizan técnicas de desarrollo de algoritmos de "dividir y conquistar". Un algoritmo, cuya complejidad temporal tenga la forma indicada, parte un problema de tamaño n en k problemas de tamaño $\frac{n}{k}$. El armado de la solución del todo a partir de las soluciones de las partes tiene como costo $g(n)$.

2.3 ECUACIONES DE RECURRENCIA LINEALES

Las ecuaciones de recurrencia más sencillas para las que se pueden encontrar soluciones de manera sistemática son las ecuaciones lineales. De los ejemplos de 2.2, tanto la sucesión de

Fibonacci como las funciones para contar nodos de árboles binarios perfectos son definidas por ecuaciones lineales.

Se considerarán sucesiones cuyos elementos están en un conjunto $X \in \{\mathbf{R}, \mathbf{C}\}$. Recuérdese que $\langle X \rangle$ es el conjunto de las sucesiones de X.

Una ecuación de recurrencia *lineal (de orden* k*)* es de la forma:

$$\text{Para } n \geq 0: \quad \sum_{i=0}^{k} p_i a_{n+i} = f_n$$

donde, para $i=0,\ldots,k$: $p_i \in X$, y $\langle f_n \rangle$ es una sucesión conocida. Cuando $f=0$, se dice que la ecuación es *homogénea*.

Con las definiciones:

$$\langle a_n \rangle + \langle b_n \rangle \quad := \quad \langle a_n + b_n \rangle$$

$$c \ \langle a_n \rangle \quad := \quad \langle c \ b_n \rangle$$

el conjunto $\langle X \rangle$ es un espacio lineal. La sucesión nula $0 := \langle 0 \rangle$ es el módulo de la suma de sucesiones.

2.3.1 Operadores lineales

Un operador

$$F: \langle X \rangle \to \langle X \rangle$$

es *lineal* si se cumple que:

$$F(c\langle a_n \rangle + \langle b_n \rangle) = c \ F(\langle a_n \rangle) + F(\langle b_n \rangle)$$

Los paréntesis se omiten, a menos que haya ambigüedades, es decir:

$$F\langle a_n \rangle = F(\langle a_n \rangle)$$

Sea $L\langle X \rangle$ el conjunto de los operadores lineales sobre $\langle X \rangle$. Con las operaciones

$$(F + G)\langle a_n \rangle \quad := \quad F\langle a_n \rangle + G\langle a_n \rangle$$

$$(cF)\langle a_n \rangle \quad := \quad c \ F\langle a_n \rangle$$

$L\langle X \rangle$ es un espacio vectorial. El operador 0 (función constante 0) es el módulo de la suma de operadores lineales.

Además, se puede definir una *multiplicación* o *composición* de operadores lineales:

$$(FG)\langle a_n \rangle \quad := \quad F(G\langle a_n \rangle)$$

y también la composición de dos operadores lineales es un operador lineal. Más aun, valen las distributividades:

$$F\ (G + H) = FG + FH$$

$$(F + G)\ H = FH + GH$$

Hay un módulo de la multiplicación: el operador identidad I, que por supuesto es operador lineal. Como notación, el operador cI se escribe también c.

Se pueden definir *potencias* enteras de operadores lineales:

$$F^0 = I$$

$$F^{r+1} = FF^r,\ r\geq 0.$$

Así, un polinomio sobre un operador lineal F es de nuevo un operador lineal.

De la teoría de espacios vectoriales, se sabe que el núcleo de un operador lineal es un espacio vectorial. Cuando se trata de un polinomio, se puede mostrar que el núcleo tiene como dimensión el grado del polinomio.

2.3.2 El operador E

El operador

$$\begin{array}{llll} E: & \langle X\rangle & \rightarrow & \langle X\rangle \\ & \langle a_n\rangle & \mapsto & \langle a_{n+1}\rangle \end{array}$$

es el *operador de adelanto*[8] . Es lineal, puesto que:

$$\begin{aligned} E(c\langle a_n\rangle + \langle b_n\rangle) &= E(\langle ca_n + b_n\rangle) \\ &= \langle ca_{n+1} + b_{n+1}\rangle \\ &= c\langle a_{n+1}\rangle + \langle b_{n+1}\rangle \\ &= c\ E\langle a_n\rangle + E\langle b_n\rangle \end{aligned}$$

Obsérvese que, para $i\geq 0$:

$$E^i\langle a_n\rangle = \langle a_{n+i}\rangle$$

Como ya se vio, tanto las potencias como los polinomios sobre E son también operadores lineales.

Con ayuda del operador de adelanto E, una ecuación de recurrencia lineal se puede escribir, en forma equivalente:

$$\sum_{i=0}^{k} p_i E^i\ \langle a_n\rangle = \langle f_n\rangle$$

o bien

$$P(E)\ a = f$$

donde $P(E)$ es el polinomio sobre el operador de adelanto E:

$$P(E) = \sum_{i=0}^{k} p_i E^i\ .$$

Cuando en la anterior expresión se cambia el operador E por una variable numérica x, se habla del *polinomio característico* de la ecuación, i.e.,

$$P(x) = \sum_{i=0}^{k} p_i x^i\ .$$

---

[8] El operador E corresponde al operador diferencial D en la teoría clásica de ecuaciones diferenciales.

2.3.3 Ecuaciones lineales homogéneas

Una ecuación homogénea tiene la forma

$$P(E)\ a = 0.$$

Supóngase una raíz r para el polinomio característico $P(x)$, i.e. $P(r)=0$. Considérese ahora la sucesión $\langle r^n \rangle$. Entonces:

$$\begin{aligned}
P(E)\langle r^n\rangle &= \sum_{i=0}^{k} p_i E^i \langle r^n \rangle \\
&= \sum_{i=0}^{k} p_i \langle r^{n+i} \rangle \\
&= \langle r^n \sum_{i=0}^{k} p_i r^i \rangle \\
&= \langle r^n\ P(r) \rangle \\
&= 0
\end{aligned}$$

Es decir, $\langle r^n \rangle$ es solución de la ecuación homogénea. El siguiente resultado es una generalización de la anterior observación:

Teorema A

a Para $0 \leq j < m$: $\quad (E-r)^m \langle n^j r^n \rangle = 0$

b Si r es una raíz de multiplicidad m para el polinomio $P(x)$, entonces las sucesiones $\langle n^j r^n \rangle$, con $j=0,\ldots,m-1$, son soluciones linealmente independientes de la ecuación homogénea

$$P(E)\ a = 0.$$

c Sea $P(x) = (x-r_1)^{m_1} \ldots (x-r_s)^{m_s}$ el polinomio característico de la ecuación de recurrencia homogénea

$$P(E)\ a = 0.$$

Entonces, la solución más general para esta ecuación es de la forma

$$\text{Para}\quad n \geq 0: \qquad a_n = \sum_{i=1}^{s} \sum_{j=0}^{m_i-1} c_{ij}\ n^j\ r_i^n$$

donde los coeficientes c_{ij} son constantes cualesquiera.

Demostración:
Se omite.

[]

La parte **a** del teorema resuelve una ecuación de recurrencia muy sencilla, como es:

$$(E-r)^m\ a = 0.$$

Se puede mostrar que la dimensión del espacio vectorial de las soluciones de una ecuación de grado m es, precisamente, m. La parte **b** muestra una base (m funciones independientes) para este espacio. Finalmente, la parte **c** revela un método para encontrar todas las soluciones que una ecuación lineal homogénea pueda tener: se deben encontrar todas las raíces del polinomio característico, con las multiplicidades que les correspondan. Para encontrar una solución particular, las constantes c_{ij} de la expresión general se pueden determinar considerando restricciones sobre algunos valores particulares de a_n.

Ejemplos

En los siguientes ejemplos se buscan soluciones reales y complejas para las ecuaciones de recurrencia planteadas.

a *Raíces reales diferentes*

Considérese la ecuación:

$a_0 = 0$

$a_1 = 1$

$a_n - 5a_{n-1} + 6a_{n-2} = 0 \quad , \text{ para } n \geq 2.$

Para poder utilizar el Teorema A, la recurrencia debe reescribirse de manera que el índice más pequeño sea n. De esta forma, la ecuación se plantea:

$a_0 = 0$

$a_1 = 1$

$a_{n+2} - 5a_{n+1} + 6a_n = 0 \quad , \text{ para } n \geq 0.$

O bien, utilizando el operador E:

$a_0 = 0$

$a_1 = 1$

$(E^2 - 5E + 6)\ a = 0.$

El polinomio característico es

$$x^2 - 5x + 6 = (x - 2)(x - 3),$$

de manera que la solución general tiene la forma:

$$a_n = A\ 2^n + B\ 3^n \quad , \text{ para } n \geq 0,$$

con A y B constantes determinadas por las condiciones iniciales, i.e., debe cumplirse que

$a_0 = A + B = 0$

$a_1 = 2A + 3B = 1.$

Así, se tiene como única solución:

$$a_n = 3^n - 2^n \quad , \text{ para } n \geq 0.$$

b *Raíces reales repetidas*

Considérese la ecuación:

$a_0 = 0$

$a_1 = 1$

$a_{n+2} - 4a_{n+1} + 4a_n = 0 \quad , \text{para } n \geq 0.$

Con ayuda del operador E, se plantea en forma equivalente:

$a_0 = 0$

$a_1 = 1$

$(E^2 - 4E + 4)\ a = (E - 2)^2\ a = 0.$

Así, el polinomio característico tiene una sola raíz, 2, de multiplicidad 2. Entonces la solución más general tiene la forma:

$$a_n = A\ 2^n + Bn\ 2^n \quad , \text{ para } n \geq 0.$$

Para determinar las constantes A y B, se observa que:

$a_0 = A = 0$

$a_1 = 2A + 2B = 1,$

de donde se concluye que:

$$a_n = n\ 2^{n-1} \quad , \text{ para } n \geq 0.$$

c *Raíces complejas*

En los problemas de análisis de algoritmos no es corriente encontrar ecuaciones cuyos polinomios característicos tengan una o varias raíces complejas. Sin embargo, la solución de tales recurrencias puede encontrarse, sin consideraciones adicionales, usando el Teorema A. Por ejemplo, para la ecuación (α, β números complejos arbitrarios):

$a_0 = \alpha$

$a_1 = \beta$

$a_{n+2} + a_n = 0 \quad ,\text{para } n \geq 0.$

En términos del operador E:

$a_0 = \alpha$

$a_1 = \beta$

$(E^2 + 1)\ a = 0$

El polinomio característico es:

$x^2 + 1 = 0,$

con raíces complejas i, $-i$. La solución más general es de la forma:

$$a_n = A\ i^n + B\ (-i)^n \text{ , para } n \geq 0.$$

Usando las condiciones de frontera:

$$a_n = \frac{a-ib}{2}\ i^n + \frac{a+ib}{2}(-i)^n \quad , \text{ para } n \geq 0.$$

□

2.3.4 Ecuaciones lineales no homogéneas

El siguiente teorema y su corolario proporcionan un método para resolver ecuaciones lineales no homogéneas en la forma más general:

Teorema B

Sea $P(E)$ un polinomio sobre el operador de adelanto E. Para la ecuación lineal no homogénea

$$P(E)\ a = f \qquad (*)$$

la solución más general es de la forma:

$$a = p + h,$$

donde p es una solución particular de $(*)$, i.e., p es una sucesión tal que

$$P(E)\ p = f,$$

y h es la solución más general de la ecuación homogénea correspondiente

$$P(E)\ h = 0.$$

Demostración:
Se omite.

[]

Corolario C

Supóngase que, en la ecuación $(*)$, el polinomio característico $P(x)$ es factorizable

$$P(x) = (x-r_1)^{m_1} \ldots (x-r_s)^{m_s}.$$

Entonces, la solución más general de $(*)$ es de la forma:

$$a_n = p_n + \sum_{i=1}^{s} \sum_{j=0}^{m_i-1} c_{ij}\, n^j\, r_i^n \qquad , \text{ para } n \geq 0,$$

donde $\langle p_n \rangle$ es una solución particular cualquiera de $(*)$.

Demostración:
Se omite.

[]

En otras palabras, el método sugerido para solucionar una ecuación lineal no homogénea consiste en encontrar una ecuación particular arbitraria a la que debe sumarse la solución más general de la ecuación lineal homogénea correspondiente. Como en la sección anterior ya se explicó un método para resolver, de la manera más general, una ecuación lineal homogénea, se plantea entonces el problema de encontrar una solución particular para la ecuación no homogénea.

El método del anulador

Para una ecuación de recurrencia lineal

$$P(E)\ a = f$$

supóngase conocido un polinomio $A(E)$, tal que $A(E)f = 0$. Entonces:

$$A(E)\ P(E)\ a = A(E)\ f = 0.$$

Ahora, $Q(E) = A(E)P(E)$ es un polinomio en E, para el que se debe resolver una ecuación homogénea. La forma general de las soluciones se establece con el Teorema A. El polinomio $A(E)$ se llama un *anulador* de f.

A continuación se presenta una lista de anuladores para algunas clases de sucesiones, así como la razón para que así suceda:

- $(E-r)^m$ es anulador para $\langle n^j r^n \rangle$, si $j = 0,\dots,m-1$.
  Es la misma afirmación del Teorema A, parte a.

- $(E-1)^m$ es anulador para $\langle n^j \rangle$, si $j = 0,\dots,m-1$.
  Caso particular del anterior: $r=1$.

- Si A es un anulador de $\langle a_n \rangle$, también lo es de $\langle k\ a_n \rangle$, para una constante k.
  Porque A es un operador lineal.

- Si A es un polinomio en E, anulador de $\langle a_n \rangle$, y B es un polinomio en E, anulador de $\langle b_n \rangle$, entonces la composición AB es un polinomio en E, anulador de $\langle a_n+b_n \rangle$.
  El hecho de que A y B sean polinomios en E permite afirmar que $AB = BA$. Entonces:

$$\begin{aligned} AB\langle a_n+b_n \rangle &= A(B\langle a_n+b_n \rangle) \\ &= AB\langle a_n \rangle \\ &= BA\langle a_n \rangle \\ &= B0 \\ &= 0 \end{aligned}$$

Ejemplos

a Considérese la ecuación:

$$a_0 = 1$$
$$a_n = 2a_{n-1} + 1\ ,\ \text{para } n\geq 1.$$

En términos del operador E:

$$a_0 = 1$$
$$(E - 2)\ a = \langle 1 \rangle$$

Un anulador para $\langle 1 \rangle$ es $(E - 1)$. Por esta razón:

$$(E - 1)(E - 2)\ a = (E - 1)\langle 1 \rangle = 0.$$

Por el Corolario C:

$$a_n = A + B\ 2^n\ ,\ \text{para } n\geq 0.$$

Además, se sabe que:

$$a_0 \quad = \quad 1 = A + B$$

$$a_1 \;=\; 2a_0 + 1 \;=\; 3 = A + 2B,$$

de donde se deduce que

$$a_n = 2^{n+1} - 1 \quad , \text{ para } n \geq 0.$$

b Sea la ecuación de recurrencia:

$$a_0 = 0$$
$$a_{n+1} + a_n = 3n + 2^n \; , \text{ para } n \geq 0.$$

Es decir:

$$a_0 = 0$$
$$(E + 1)\, a = \langle 3n + 2^n \rangle$$

Ahora, puesto que $(E - 1)^2$ anula a $\langle 3n \rangle$ y $(E - 2)$ anula a $\langle 2^n \rangle$, se tendrá que:

$$(E - 1)^2\, (E - 2)\, (E + 1)\, a = (E - 1)^2\, (E - 2)\langle 3n + 2^n \rangle = 0.$$

De este modo, la solución más general tiene la forma:

$$a_n = A + Bn + C\, 2^n + D(-1)^n \quad , \text{ para } n \geq 0.$$

A partir de la ecuación original se puede comprobar que:

$$a_0 = 0, \qquad a_1 = 1, \qquad a_2 = 4, \qquad a_3 = 6,$$

y, finalmente:

$$a_n = \frac{1}{12}\,(2^{n+2} + 18n - 9 + (-1)^n\, 5), \text{ para } n \geq 0.$$

[]

2.3.5 Ejercicios

1 Pruebe el Teorema A de 2.3.3.
AYUDA: (a) se muestra por inducción sobre j. Para (b), el polinomio característico es de la forma $A(E)(E-r)^m$, donde A es un polinomio para el que r no es raíz. Para (c), puede mostrarse una biyección entre el espacio vectorial de las soluciones y un espacio vectorial de dimensión $m = m_1 + \ldots + m_s$.

2 Sea $\langle F_n \rangle$ la sucesión de Fibonacci. Compruebe las siguientes afirmaciones, para $n \geq 0$:

a $F_n = O(\varphi^n)$, donde[9] $\varphi = \dfrac{1+\sqrt{5}}{2}$.

b $F_n = \mathrm{emc}(\varphi^n / \sqrt{5})$, donde $\mathrm{emc}(x)$ es el entero más cercano al número x.

---

[9] El número φ (del escultor griego Fidias) fue llamado por Euclides la "media y extrema razón". En el Renacimiento se le denominó la "divina proporción" y últimamente se le conoce como el "número de oro". En el arte clásico griego, la razón de ϕ a 1 se juzgó como "la más estética posible", de modo que aparece representada en las medidas de construcciones, esculturas, etc.

c Para $n>0$: $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$.

d $F_{n+1} F_{n-1} - F_n^2 = (-1)^n$.
AYUDA: Calcule determinantes a ambos lados en **c**.

e Para $k\geq0$: F_{nk} es un múltiplo de F_k.

3 Encuentre la solución más general para las ecuaciones de recurrencia:

a $a_0 = 1$
$a_1 = 5$
$a_{n+1} = 2a_n - a_{n-1}$, para $n\geq1$.

b $a_0 = 1$
$a_1 = 0$
$a_{n+1} = 4a_n - 4a_{n-1}$, para $n\geq1$.

c $a_0 = 1$
$a_1 = 1$
$a_{n+1} = a_n + a_{n-1}$, para $n\geq1$.

d $a_n = a_{n-1} + 2b_{n-1}$, para $n\geq0$
$b_0 = 1$
$b_{n+1} = 2b_n$, para $n\geq0$

4 Dé anuladores para cada una de las siguientes sucesiones ($n\geq0$):

a 12 **b** $12n$ **c** $12n^2$

d n^{m-1} **e** $4 - 4n + n^2$ **f** r^n

g $n^{m-1}r^n$ **h** $4 - 4n + n^2 + 5^n$

3 Encuentre la solución más general para las ecuaciones de recurrencia:

a $a_0 = 1$
$a_{n+1} = 3a_n + n$, para $n\geq0$.

b $x_n = x_{n-1} + 2^n - 1$, para $n\geq1$.

c $x_n = x_{n-1} + 2^n - 1$, para $n\geq1$.

2.4 ECUACIONES DE RECURRENCIA NO LINEALES

Al igual que en la teoría de ecuaciones diferenciales, los problemas serios comienzan cuando se trata de resolver ecuaciones de recurrencia no lineales. No hay métodos que funcionen para todos los casos, pero para ciertas familias de ecuaciones es posible efectuar transformaciones de los espacios que hacen las veces de dominio y rango de las posibles soluciones, de suerte que la ecuación transformada sí es lineal.

2.4.1 Cambio de dominio

Antes de intentar un método general, considérese la ecuación que se utilizó en el análisis del ordenamiento por intercalamiento en 2.2.3:

$$M(n) = 0 \quad , \text{ si } n \le 1$$
$$= 2\, M(\tfrac{n}{2}) + n - 1 \quad , \text{ si } n > 1$$

La recurrencia no es lineal y, más aun, debe entenderse que puede no tener sentido para valores impares de n. Sin embargo, considérese la sucesión $\langle x_k \rangle$, que describe los saltos que se encuentran en la recurrencia, ligada a la de los números naturales por las relaciones[10]:

$$x_0 = 1$$
$$x_{k-1} = \frac{n}{2} , \qquad x_k = n.$$

De las dos últimas ecuaciones, se obtiene una ecuación que debe satisfacer x:

$$x_k - 2x_{k-1} = 0.$$

De otra forma:

$$x_0 = 1$$
$$(E - 2)\, x = 0.$$

Entonces, se obtiene que

$$x_k = 2^k \quad , \text{ para } k \ge 0.$$

Recuérdese que, además, $n = 2^k$. Ahora, considérese la sucesión definida por

$$a_k = M(x_k) \quad , \text{ para } k \ge 0$$
$$= M(2^k) \quad , \text{ para } k \ge 0$$

Con estas transformaciones, se buscan soluciones para:

$$a_0 = T_1 = 0$$
$$a_k = 2a_{k-1} + 2^k - 1$$

Esta última es una ecuación lineal sobre k, y tiene como solución:

$$a_k = 2^k(k - 1) + 1 \quad , \text{ para } k \ge 0.$$

Finalmente, la solución se puede expresar en términos de n, la variable original:

$$M(n) = n \log n - n + 1 \quad , \text{ para } n \ge 1.$$

(El valor $M(0) = 0$ no está incluido en la fórmula anterior, precisamente porque $x_0 = 1$).

[]

El método

[10] Obsérvese cómo se evita decir que $x_0 = 0$, puesto que esto haría trivialmente nula la secuencia x.

Para tratar de generalizar el método utilizado en la solución de la anterior recurrencia, supóngase una ecuación de la forma

$$T_n = f(T_{\alpha(n)})$$

donde se espera que $\alpha(n)<n$. La técnica de *cambio de dominio* considera, en primer lugar, la sucesión $\langle x_k \rangle$, definida por la recurrencia

$$x_k = n$$
$$x_{k-1} = \alpha(n),$$

o bien,

$$x_{k-1} = \alpha(x_k).$$

Si esta última recurrencia se puede resolver para x, se define la sucesión

$$a_k = T_{x_k}$$

y se trata de resolver la ecuación original con el cambio de variable señalado, i.e., puesto que $T_{x_k} = f(T_{x_{k-1}})$, deberá tenerse que

$$a_k = f(a_{k-1}).$$

De lograrse una solución para esta última ecuación, puede refrasearse en términos de n, la variable original, ya que:

$$T_n = T_{x_k} = a_k.$$

[]

Ejemplo

Un ejemplo más complejo de transformación de dominio surge de analizar un algoritmo de multiplicación de números binarios de n bits, el cual realiza 3 multiplicaciones de $\frac{n}{2} + 1$ bits más un trabajo adicional de costo $O(n)$. Si se analiza recurrentemente el algoritmo, para estimar su complejidad temporal, se llega a una recurrencia de la forma (si $n\leq 3$ el algoritmo no recurre):

$$\begin{aligned} T(n) &= O(1) && \text{, para } n\leq 3 \\ &= 3\,T(\tfrac{n}{2} + 1) + O(n) && \text{, para } n>3 \end{aligned}$$

Como se busca una cota superior para estimar $T(n)$, basta considerar una función $T1(n)$, tal que (α, β son constantes adecuadas):

$$\begin{aligned} T1(n) &= \alpha && \text{, para } n\leq 3 \\ &= 3\,T1(\tfrac{n}{2} + 1) + \beta n && \text{, para } n>3 \end{aligned}$$

Naturalmente, $T = \theta(T1)$. Se buscará ahora una solución para $T1$, utilizando una transformación de dominio. En primer lugar, se establece la ecuación:

$$x_0 = 3$$
$$x_k = n$$

$$x_{k-1} = \frac{n}{2} + 1$$

De manera que:

$$x_0 = 3$$
$$x_{k+1} - 2x_k = -2.$$

Es decir:

$$x_k = 2^k + 2 \quad , \text{ para } k \geq 0.$$

Ahora se define

$$a_k = T1(x_k) \quad , \text{ para } k \geq 0$$

y se llega a la ecuación:

$$a_0 = T1(x_0) = T1(3) = \alpha$$
$$a_k = 3a_{k-1} + \beta 2^k + 2\beta \quad , \text{ para } k \geq 1,$$

que tiene como solución:

$$a_k = (\alpha + 3\beta)\, 3^k - \beta\, 2^{k+1} - \beta \,, \quad \text{para } k \geq 0.$$

Finalmente[11]:

$$T1(n) = (\alpha + 3\beta)\, (n - 2)^{\log 3} - 2\beta n + 3\beta \,, \text{ para } n \geq 3.$$

Asintóticamente:

$$\begin{aligned} T1(n) &= O(n^{\log 3}) &&, \text{ si } \alpha + 3\beta \neq 0 \\ &= O(n) &&, \text{ si } \alpha + 3\beta = 0,\ \beta \neq 0 \\ &= O(1) &&, \text{ si } \alpha = 0,\ \beta = 0. \end{aligned}$$

[]

2.4.2 Cambio de rango

Considérese la ecuación de recurrencia no lineal:

$$a_0 = 1$$
$$a_n = 3\, (a_{n-1})^2 \,, \text{ para } n \geq 1.$$

Si se toman logaritmos en ambos lados de las anteriores relaciones, se obtiene:

$$\log a_0 = 0$$
$$\log a_n = \log 3 + 2 \log a_{n-1} \quad , \text{ para } n \geq 1.$$

Ahora, llamando $b_n = \log a_n$, para $n \geq 0$, la recurrencia se puede refrasear en:

[11] Recuérdese que $a^x = b^{x \log_b a}$.

$$b_0 = 0$$
$$b_n = \log 3 + 2\, b_{n-1} \ , \text{ para } n \geq 1.$$

Esta última es una ecuación lineal, con solución

$$b_n = (\log 3)\,(2^n - 1) \quad , \text{ para } n \geq 0.$$

Por lo tanto:

$$a_n = 3^{2^n - 1} \quad , \text{ para } n \geq 0.$$

El método

Para una ecuación de recurrencia sobre una incógnita a_n, un *cambio de rango* es una transformación sobre la incógnita. Así, si la ecuación original es de la forma:

$$T_n = f(T_{\alpha(n)}),$$

con $\alpha(n) < n$, se propone una ecuación de la forma

$$b_n = g(T_n),$$

donde g es una función invertible. Para que el método tenga éxito, la función compuesta $g(f(.))$ debe ser manipulable algebraicamente. En otras palabras, se puede resolver

$$b_n = g(f(T_{\alpha(n)})),$$

para la incógnita b_n. En este caso:

$$T_n = g^{-1}(b_n).$$

[]

Ejemplo

Considérese la ecuación de recurrencia:

$$a_0 = 1$$
$$a_{n+1} = 5\sqrt{a_n} \quad , \text{ para } n \geq 0.$$

En un primer intento de cambio de rango se puede elevar al cuadrado para deshacer el radical:

$$a_0^2 = 1$$
$$a_{n+1}^2 = 5^2\, a_n \quad , \text{ para } n \geq 0.$$

Ahora pueden tomarse logaritmos:

$$2 \log a_0 = 0$$
$$2 \log a_{n+1} = 2 \log 5 + \log a_n \ , \text{ para } n \geq 0.$$

Con la transformación $b_n = \log a_n$, se consigue una formulación como:

$$b_0 = 0$$

$$b_{n+1} - \frac{1}{2} b_n = \log 5 \text{ , para } n \geq 0.$$

Resolviendo para b_n:

$$b_n = (\log 25)(1 - 2^{-n}) \quad \text{, para } n \geq 0,$$

de manera que:

$$a_n = 25^{(1-2^{-n})} \quad \text{, para } n \geq 0$$

[]

2.4.3 Ejercicios

1 Resuelva las siguientes ecuaciones de recurrencia (k es una constante positiva):

a
$$\begin{aligned} T(n) &= c_1 && \text{, si } n=1 \\ &= k\, T(\tfrac{n}{k}) + c_2 && \text{, si } n \geq 2 \end{aligned}$$

b
$$\begin{aligned} T(n) &= c_1 && \text{, si } n=1 \\ &= k\, T(\tfrac{n}{k}) + c_2\, n && \text{, si } n \geq 2 \end{aligned}$$

c
$$\begin{aligned} T(n) &= c_1 && \text{, si } n=1 \\ &= k\, T(\tfrac{n}{k}) + c_2\, n^m && \text{, si } n \geq 2 \end{aligned}$$

d
$$\begin{aligned} T(n) &= c_1 && \text{, si } n=1 \\ &= T(\tfrac{n}{2}) + c_2 && \text{, si } n \geq 2 \end{aligned}$$

e
$$\begin{aligned} T(n) &= c_1 && \text{, si } n=1 \\ &= 2\, T(\tfrac{n}{2}) + c_2 && \text{, si } n \geq 2 \end{aligned}$$

2 Considérese la ecuación de recurrencia (a, b son constantes, c es una función):

$$T(1) = c(1)$$
$$T(n) = a\, T(\tfrac{n}{b}) + c(n) \quad \text{, para } n \geq b$$

Muestre que, para $n = b^k$, con $k \geq 0$, se tiene que:

$$T(n) = \sum_{i=0}^{k} \sum_{i=0}^{k} a^i\, c(\tfrac{n}{b^i})$$

3 Para la ecuación de recurrencia (a, b, c son constantes):

$$T(1) = c$$
$$T(n) = a\, T(\tfrac{n}{b}) + c\, n \quad \text{, para } n \geq b$$

a Muestre que, para $n = b^k$, con $k \geq 0$, se tiene que:

$$\begin{aligned} T(n) &= c\, n\, \frac{1 - (a/b)^{k+1}}{1 - (a/b)} && \text{, si } a \neq b \\ &= c\, n\, (k+1) && \text{, si } a = b \end{aligned}$$

b Pruebe que:

$$\begin{aligned} T(n) &= \theta(n) &&, \text{ si } a<b \\ &= \theta(n \log n) &&, \text{ si } a=b \\ &= \theta(n^{\log_b a}) &&, \text{ si } a>b \end{aligned}$$

4 Use ecuaciones de recurrencia para hallar fórmulas compactas (que no involucran sumatorias) para:

a La suma de los n primeros cuadrados perfectos.

b La suma de los n primeros cuadrados perfectos.

5 **a** Calcule la suma $\langle +:\ 1\le k\le m:\ \frac{k}{2^k} \rangle$.

b Estime asintíoticamente la suma $\langle +:\ 1 \le k \le \lfloor \log n \rfloor:\ \frac{k}{2^k} \rangle$.

6 El análisis de un método rápido para multiplicar números de n bits lleva a la ecuación

$$T(n) = \log n + 2\ T(4\sqrt{n})$$

a Para qúe valores de n podría tener sentido esta ecuación de recurrencia, considerando que mide la complejidad de un algoritmo?

b Estime la complejidad asintótica de $T(n)$.

7 Una secuencia S de símbolos se dice *parentizada* si uno de los siguientes casos vale:

- La longitud de S es 1, i.e., S consta de un solo símbolo
- S es de la forma $(S_1)(S_2)$, donde S_1 y S_2 son secuencias parentizadas.

Si $a_1, a_2, \ldots, a_n$ son n símbolos diferentes y $p(n)$ es el número de secuencias parentizadas que contienen los elementos $a_1, a_2, \ldots, a_n$ dispuestos de izquierda a derecha exactamente en ese orden,

a Halle una ecuación de recurrencia que defina a $p(n)$, $n\ge 1$.

b Demuestre que $p(n+1) = \frac{1}{n+1}\binom{2n}{n}$ y que $p(n+1) \ge 2^{n-1}$, $n\ge 0$.

Bibliografía

[Apo69] Apostol, T.M., *Calculus*, Blaisdell Publ. Co., 1969.

[Car91] Cardoso, R., *Verificación y desarrollo de programas, Ediciones Uniandes -Ecoe,* 2a. impresión revisada, 1993.

[Mel77] Melhorn, K., *Effiiziente Algorithmen*, Teubner Studienbücher - Informatik, 1977.