Universidad Nacional del Sur
Departamento de Cs. e Ingeniería de la Computación
Lic. en Cs. de la Computación

Algoritmos y Complejidad

Revisión de Conceptos y Técnicas Matemáticas

Primer Cuatrimestre de 2017

En el estudio de los algoritmos muchas veces es esencial establecer formalmente sus propiedades, y para ello es necesario tener práctica en el uso de varias herramientas matemáticas. En este apunte se repasan algunas técnicas de demostración y conceptos que van a ser usados a lo largo de la materia. Estos conocimientos fueron impartidos en materias previas, por lo que solo se pretende realizar un repaso informal de los mismos. La primera parte de técnicas de demostración contiene elementos que deben ser manejados en forma fluida por todos aquellos que pretendan encarar el estudio formal de las propiedades de un algoritmo. La segunda parte de conceptos y propiedades sirve como referencia para la resolución de problemas matemáticos; no es necesario tenerlos presente de memoria, sino conocer su existencia para el caso que en lleguen a necesitarse.

1. Técnicas de demostración

1.1. Pruebas por contradicción

Una *prueba por contradicción*, o *prueba por el absurdo*, consiste en demostrar la validez de una hipótesis probando que su negación conduce a una contradicción. En otras palabras, supongamos que se quiere probar S. Por ejemplo, S puede ser "existen infinitos números primos". Para dar una prueba por contradicción de S se empieza suponiendo que S es falso, o lo que es equivalente que *no* S es cierto. Luego, siguiendo un razonamiento matemático válido, se deduce como verdadera alguna otra proposición previamente conocida como falsa. Se concluye por lo tanto que la suposición inicial era incorrecta.

Mostraremos dos ejemplos de este método de demostración.

Ejemplo 1.1 El siguiente teorema era conocido ya en la época de la Grecia clásica, y es generalmente atribuido a Euclides.

Teorema *Existen infinitos números primos.*

Demostración Sea P el conjunto de todos los números primos. Supongamos *por el absurdo* que P es un conjunto finito. Sabemos que P no es vacío, ya que por ejemplo $2 \in P$. Dado que P es finito y no vacío, tiene sentido multiplicar todos sus elementos. Sea entonces x este producto, y y su sucesor inmediato $x + 1$. Consideremos entonces el menor natural d que es mayor que 1 y divisor de y, es decir $y = k \times d, k \geq 1$. Tal natural existe dado que $y > 1$ y no pedimos que d sea distinto de y. Primero notemos que d es primo, de lo contrario existiría un divisor propio de d, menor que d y que también dividiría a y. (Nota: la oración anterior es en sí misma una prueba por contradicción anidada en la demostración general.)

Luego, $d \in P$ y entonces también d es divisor de x el producto de todos los primos que incluyen a d. Es decir, $x = k' \times d, k' \geq 1$. Hemos concluido entonces que d divide exactamente tanto a x como a y. Pero esto es imposible ya que $y = x + 1$, luego $k \times d = k' \times d + 1$ y $1 = (k - k') \times d$ siendo $d > 1$. Entonces nuestra suposición inicial es falsa, y por lo tanto existen infinitos números primos. ∎

Es posible extraer de la prueba anterior un algoritmo que, aunque no muy eficiente, calcula un nuevo número primo a partir de un conjunto finito de primos dados.

```
función NuevoPrimo ( P : conjunto de primos )
  {El parámetro debe ser un conjunto de primos no vacío.}
  x := el producto de los elementos de P;
  y := x+1;
  d := 1;
  repetir
     d := d + 1;
  hasta que d divide a y;
retornar d;
```

La prueba del teorema de Euclides asegura que el valor retornado por `NuevoPrimo(P)` es un número primo no perteneciente a `P`. Sin embargo, alguien podría cuestionar la utilidad del algoritmo anterior existiendo el siguiente algoritmo, mucho más simple, para el mismo problema.

```
función Simple ( P : conjunto de primos )
  {El parámetro debe ser un conjunto de primos no vacío.}
  x := el mayor elemento de P;
  repetir
     x := x+1;
  hasta que x sea primo;
retornar x;
```

Es claro que este segundo algoritmo da como resultado un primo que no pertenece a `P`, *siempre que nos aseguremos que el algoritmo termina*. El problema está en el hecho de que el ciclo de `Simple(P)` nunca terminaría si `P` contuviera el primo más grande. Naturalmente esto no puede pasar porque no existe el primo más grande, pero se necesita la demostración de Euclides para asegurarlo. En resumen, `Simple(P)` funciona, pero la prueba de su terminación no es inmediata. Al contrario, el hecho de que `NuevoPrimo(P)` siempre termina es claro (en el peor caso terminaría cuando `d` alcance a `y`) pero el hecho de que devuelve un primo requiere una prueba. Estos algoritmos muestran dos propiedades que siempre son importantes de conocer para todo algoritmo: su correctitud, y su terminación en todos los casos. ∎

Hemos visto en el ejemplo anterior que a veces es posible convertir una demostración matemática en un algoritmo. Desafortunadamente esto no siempre es así cuando se trata de una prueba por contradicción, como lo muestra el siguiente ejemplo.

Ejemplo 1.2
Teorema *Existen dos números irracionales x e y tales que x^y es un número racional.*

Demostración Suponemos *por el absurdo* que x^y es necesariamente irracional, para todos los números irracionales x e y. Es conocido que $\sqrt{2}$ es irracional (esto era ya conocido en la época de Pitágoras, que vivió incluso antes que Euclides), y sea $z = (\sqrt{2})^{\sqrt{2}}$. Por nuestra suposición, z

debe ser irracional dado que es el resultado de elevar un número irracional ($\sqrt{2}$) a una potencia también irracional ($\sqrt{2}$). Consideremos ahora $w = z^{\sqrt{2}}$. De nuevo, w es irracional por nuestra suposición dado que z y $\sqrt{2}$ son irracionales. Pero

$$w = z^{\sqrt{2}} = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^{(\sqrt{2}\times\sqrt{2})} = (\sqrt{2})^2 = 2$$

Hemos llegado a la conclusión de que 2 es irracional, por lo que debemos suponer que nuestra proposición inicial es falsa y por lo tanto elevar un número irracional a una potencia irracional puede resultar en un número racional. ∎

No es posible extraer de la demostración anterior un algoritmo para encontrar dos irracionales x e y tales que x^y sea racional. En principio, uno estaría tentado a decir que el algoritmo simplemente debe devolver $x = z = (\sqrt{2})^{\sqrt{2}}$ e $y = \sqrt{2}$ dado que la demostración prueba que z es irracional y $z^{\sqrt{2}}$ es racional. Pero sin embargo, esta "prueba" de que z es irracional fue hecha basada en la suposición (luego demostrada falsa) que todo irracional elevado a una potencia irracional es irracional. En realidad, es solamente la "prueba" la que es incorrecta porque de hecho z es irracional, pero es mucho más difícil demostrarlo. Debemos por lo tanto siempre ser cuidadosos de no usar resultados intermedios "'probados" en el medio de una prueba por contradicción.

No existe manera directa de encontrar un par de números x e y en las condiciones del teorema a partir de esta demostración. Esta prueba se denomia entonces *no constructiva*, y no es extraño que las pruebas por contradicción lo sean. Aunque ciertos matemáticos no aceptan las demostraciones no constructivas, la mayoría las consideran como pruebas perfectamente válidas. En cualquier caso, trataremos de evitarlas lo máximo posible en el contexto del estudio de los algoritmos. ∎

1.2. Pruebas por inducción

De todas las herramientas matemáticas útiles en el algorítmica, la *inducción matemática* quizás sea la más importante. No solo permite demostrar importantes propiedades sobre la correctitud, terminación y eficiencia de los algoritmos, sino que además que puede ser usada para *determinar* las características particulares de lo que se quiere probar. Esto se tratará con el nombre de *inducción constructiva*.

Antes de entrar en detalles sobre la técnica de prueba, es importante disgregar un comentario sobre la naturaleza del descubrimiento científico. Existen dos formas fundamentales de descubrir nuevo conocimiento en la ciencia: *inducción* y *deducción*. La inducción consiste en la "inferencia de una ley general a partir de instancias particulares", mientras que la deducción es "una inferencia particular a partir de una ley general". Mostraremos que, aunque con inducción se puede arribar a conclusiones falsas, es imprescindible tenerla en cuenta dentro del ámbito adecuado. La deducción, por otra parte, siempre produce conclusiones válidas si es aplicada correctamente.

La razón principal por la que no se puede confiar en el resultado de razonamiento inductivo es que es posible que haya casos particulares que todavía no han sido considerados. Por ejemplo, la experiencia cotidiana muestra que "siempre es posible que una persona más suba al colectivo". Pero aplicar inducción en este caso resultaría en la regla totalmente absurda "el colectivo tiene capacidad infinita de pasajeros". Para un ejemplo más matemático basta con considerar el polinomio $p(n) = n^2 + n + 41$. Si se computan $p(0), p(1), p(2), \ldots, p(10)$ se obtiene $41, 43, 47, 53, 61, 71, 83, 97, 113, 131$ y 151. Es fácil verficar que todos estos números naturales

son primos. Por lo tanto, se podría inferir por inducción que "$p(n)$ es un número primo para todos los naturales n". Pero de hecho $p(40) = 1681 = 41^2$ es compuesto.

Un ejemplo más sorprendente de mal uso de la inducción está dado por la conjetura de Euler, que fue formulada en 1769: ¿es posible que la suma de tres cuartas potencias sea una cuarta potencia? Formalmente, la conjetura piden encontrar naturales a, b, c y d tales que

$$a^4 + b^4 + c^4 = d^4$$

Después de no encontrar ningún ejemplo de estos números, Euler conjeturó que esta ecuación nunca puede ser satisfecha (esta conjetura está relacionada con el más conocido Último Teorema de Fermat). Más de doscientos años pasaron hasta que se encontró en 1987 el primer contraejemplo a esta conjetura, que involucra números de siete y ocho dígitos. Ha sido demostrado, usando cientos de horas de tiempo de computación en varias Connection Machines, que el único contraejemplo en el que d es menor que un millón es

$$95800^4 + 217519^4 + 414560^4 = 422481^4$$

(sin tener en cuenta la solución obtenida de multiplicar cada uno de estos números por 2). Observar que 422481^4 es un número de 23 dígitos.

Por el contrario, el razonamiento deductivo no está sujeto a errores de este tipo. Siempre que la regla invocada sea correcta y aplicable a la situación en consideración, la conclusión alcanzada es necesariamente correcta. Matemáticamente, si es verdadero que una proposición $P(x)$ vale para todos los x en un conjunto X, y se tiene un valor y que pertenece a X, entonces el hecho $P(y)$ puede ser afirmado. Esto no es lo mismo que decir que no se puede inferir algo falso usando razonamiento deductivo. A partir de una premisa falsa, se puede llegar deductivamente a una conclusión falsa. De hecho, este es el principio en el que se basan las pruebas por contradicción. Por ejemplo, si es correcto que $P(x)$ vale para todo x en X, pero descuidadamente aplicamos la regla a algún y que no pertenece a X entonces podemos erróneamente creer que vale $P(y)$. En forma similar, si nuestra creencia en que $P(x)$ es verdadero para todo x en X está basada en razonamiento inductivo no apropiado, entonces $P(y)$ puede ser falso aún si y pertenece a X. En conclusión, el razonamiento deductivo puede llevar a resultados erróneos, pero *sólo* en el caso que las reglas que se siguen son incorrectas o no se siguen apropiadamente.

Es entonces razonable preguntarse en este punto por qué usar la poco confiable regla de inducción en lugar de la muy segura regla de deducción. Hay dos razones básicas para usar inducción en el proceso del descubrimiento científico. Un físico que trata de descubrir las leyes fundamentales que gobiernan el Universo, debe obligadamente usar un enfoque inductivo: las reglas que infiere deben reflejar mediciones experimentales reales. Aún un físico teórico, tal como Einstein, todavía necesita de resultados de experimentos reales llevados a cabo por otros. Por ejemplo, fue el razonamiento inductivo el que llevó a Halley predecir el retorno de su cometa epónimo, y el que usó Mendeleev para predecir no solo la existencia de elementos químicos aún no descubiertos, sino también sus propiedades químicas.

Pero seguramente, parecería que solo deducción es legítimo usar en ciencias rigurosas tales como la matemática y las ciencias de la computación. Después de todo, proposiciones matemáticas tales como la existencia de infinitos números primos o la correctitud del algoritmo de Euclides (vistas a continuación) solo pueden ser probadas en una manera rigurosa y deductiva, sin necesidad de recurrir a datos experimentales. Aparentemente el razonamiento inductivo estaría prohibido en la matemática. A pesar de toda esta evidencia, esta conclusión es equivocada. En realidad, la matemática es frecuentemente una ciencia experimental. No es extraño que un matemático descubra una verdad matemática a partir de considerar varios casos especiales e

inferir de ellos *por inducción* una regla general que le parece plausible. Por ejemplo, si se observa que

$$\begin{aligned} 1^3 &= 1 = 1^2 \\ 1^3 + 2^3 &= 9 = 3^2 \\ 1^3 + 2^3 + 3^3 &= 36 = 6^2 \\ 1^3 + 2^3 + 3^3 + 4^3 &= 100 = 10^2 \\ 1^3 + 2^3 + 3^3 + 4^3 + 5^3 &= 225 = 15^2 \end{aligned}$$

se puede sospechar que la suma de los cubos de los n primeros enteros positivos es siempre un cuadrado perfecto. Resulta que en este caso el razonamiento inductivo produce una ley válida. Si se es un poco más perspicaz, se puede observar que la suma de los cubos de los n primeros enteros positivos es precisamente el cuadrado de la suma de los n primeros enteros positivos..

Sin embargo, no importa cuan atrayente la evidencia sea cuando más, y más valores de n son testeados, una regla general de esta clase no puede ser aseverada solo en base a evidencia inductiva. La diferencia entre la matemática y las ciencias inherentemente experimentales radica en que una vez que una ley matemática general ha sido descubierta por inducción, se debe a continuación probarla rigurosamente aplicando el enfoque deductivo. No obstante la inducción tiene su lugar en el proceso matemático, dado que por ejemplo no se podría esperar probar rigurosamente un teorema que ni siquiera ha sido formulado. La inducción es necesaria para la formulación de conjeturas y la deducción es igualmente necesaria para demostrarlas, o algunas veces para refutarlas. Ninguna técnica puede tomar el lugar de la otra.

El mismo enfoque cabe aplicar a las ciencias de la computación. Un razonamiento inductivo, la observación de los resultados en varias corridas de un programa, puede llevar a suponer que el algoritmo de dicho programa tiene determinadas propiedades, por ejemplo que es eficiente en el espacio de memoria que usa. Sin embargo, esta propiedad no deja de ser más que una sospecha si no se usa un razonamiento deductivo para demostrarla rigurosamente. Por esta razón es que el estudio del algorítmica requiere práctica de varias herramientas matemáticas. Tenemos la suerte, o la desgracia, que los elementos que son objeto de nuestro estudio (programas, algoritmos, computadoras, etc.) están formalmente caracterizados, y por lo tanto sus propiedades se pueden investigar sin temor a los "errores de medición".

Por último, el objetivo final de este comentario: una de las más útiles herramientas *deductivas* disponibles en matemática tiene la mala fortuna de ser llamada *inducción matemática*. Esta terminología es confusa, pero es necesario acostumbrarse a ella.

1.2.1. El principio de la inducción matemática

Consideremos el siguiente algoritmo:

```
función sq ( n )
  si n=0 entonces retornar 0;
         sino retornar 2*n+sq(n-1)-1;
```

Si se corre en algunas entradas, se puede observar que

$$\mathtt{sq}(0) = 0, \mathtt{sq}(1) = 1, \mathtt{sq}(2) = 4, \mathtt{sq}(3) = 9, \mathtt{sq}(4) = 16$$

Por inducción, parecería obvio que $\mathtt{sq}(n) = n^2$ para todo $n \geq 0$. Pero, ¿cómo puede probarse esto rigurosamente?, ¿es realmente verdadero? Digamos que el algoritmo *tiene éxito* en n siempre que $\mathtt{sq}(n) = n^2$, y que *falla* de otro modo.

Consideremos cualquier natural $m \geq 1$ y supongamos por el momento que el algoritmo tiene éxito en $m-1$. Por definición del algoritmo, $\texttt{sq}(m) = 2 \times m + \texttt{sq}(m-1) - 1$. Por nuestra suposición $\texttt{sq}(m-1) = (m-1)^2$. Por lo tanto

$$\texttt{sq}(m) = 2 \times m + (m-1)^2 - 1 = 2 \times m + m^2 - 2 \times m + 1 - 1 = m^2$$

Entonces hemos probado que el algoritmo tiene éxito en m siempre que tiene éxito en $m-1$, para todo $m \geq 1$. Además, claramente el algoritmo tiene éxito en 0.

El *principio de inducción matemática*, descripto a continuación, nos permite inferir de lo anterior que el algoritmo tiene éxito para todo $n \geq 0$. Existen dos maneras de entender el por qué se obtiene la conclusión: constructivamente y por contradicción. Sea n cualquier entero positivo en el cual se quiere probar que el algoritmo tiene éxito. Supongamos además que $n \geq 7$ (los valores menores se pueden probar fácilmente en forma individual). Sabemos, de acuerdo a los ejemplos, que el algoritmo tiene éxito en 4. Por la regla general que acabamos de ver que dice que si se tiene éxito en $m-1$ entonces también se tiene éxito en m, inferimos que el algoritmo tiene éxito en 5. Aplicando reiteradamente la regla sabemos que tiene éxito en 6, en 7, y así siguiendo hasta llegar a $n-1$. Finalmente, una aplicación adicional prueba que el algoritmo tiene éxito en n. Es claro que se puede aplicar este algoritmo explícitamente (sin necesidad de "así siguiendo") para cualquier valor positivo fijo de n.

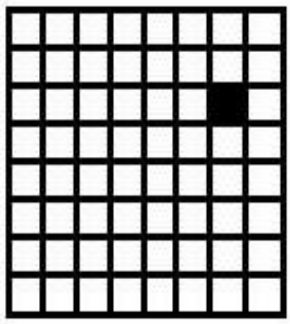
La prueba anterior es necesaria aplicarla para todos los valores de n. Si se prefiere una única prueba que funcione para todos los $n \geq 0$ y que no contiene "asi siguiendo", se debe aceptar el *axioma del menor natural* que dice que todo conjunto no vacío de enteros positivos contiene un elemento más chico que todos los demás. Este axioma permite usar este número más chico como base para probar los teoremas.

Ahora, para probar la correctitud del algoritmo supongamos *por contradicción* que existe al menos un entero positivo en el cual el algoritmo falla. Luego el conjunto de enteros positivos en el cual el algoritmo falla es un conjunto no vacío. Aplicando el axioma podemos obtener n como el menor natural en el cual el algoritmo falla. Observemos que $n \geq 5$ dado que hemos comprobado explícitamente el éxito del algoritmo para $n \leq 4$. Además, el algoritmo debe tener éxito en $n-1$, de otra forma n no sería el menor natural en el que falla. Pero entonces aplicando nuestra regla general se obtiene que el algoritmo tiene éxito en n, lo cual contradice nuestra suposición inicial. Por lo tanto este n no puede existir, lo que implica que el algoritmo siempre tiene éxito. Dado que conocemos que el algoritmo tiene éxito en 0, entonces concluimos que $\texttt{sq}(n) = n^2$ para todo natural $n \geq 0$.

Enunciaremos a continuación una versión del principo de inducción matemática que es suficiente para la mayoría de los casos. Una versión más poderosa se define en 1.2.3. Sea una propiedad P de los números naturales. Por ejemplo, $P(n)$ podría ser "$\texttt{sq}(n) = n^2$", o "la suma de los cubos de los n primeros naturales es igual al cuadrado de la suma de los n primeros naturales", o "$n^3 < 2^n$", o "$n < 0$". Las primeras dos propiedades son ciertas para todo $n \geq 0$, mientras que la tercera vale siempre que $n \geq 10$ y la cuarta es siempre falsa. Sea entonces a un natural denominado *base*. Entonces si

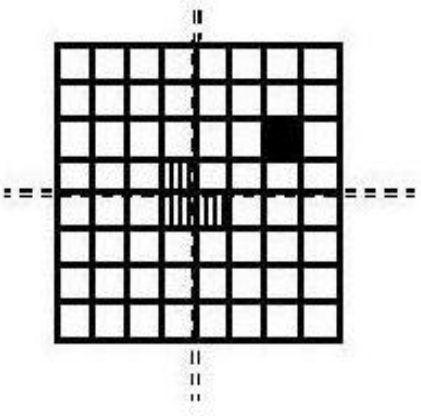
1. $P(a)$ vale y,
2. $P(n)$ debe valer siempre que $P(n-1)$ vale, para todo natural $n > a$.

entonces la propiedad $P(n)$ vale para todos los naturales $n \geq a$. Usando este principio se puede afirmar que $\texttt{sq}(n) = n^2$ para todo natural n mostrando que $\texttt{sq}(0) = 0$ y que siempre que vale $\texttt{sq}(m-1) = (m-1)^2$ entonces vale $\texttt{sq}(m) = m^2$, para todo $m \geq 1$.

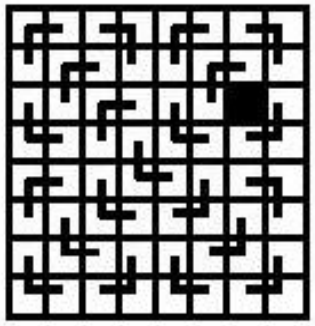


(a) Tablero con un casillero destacado

(b) Ficha



(c) Ubicación de la primer ficha



(d) Solución

Figura 1: El problema del cubrimiento.

Ejemplo 1.3 Consideremos el siguiente problema de cubrimiento. Se tiene un tablero dividido en casilleros cuadrados iguales. Hay k casilleros en cada fila y k casilleros en cada columna, donde k es una potencia de 2. Un casillero arbitrario es distinguido como el *casillero especial*; ver figura 1(a). También se dispone de un conjunto de fichas, cada una de las cuales se parece a un tablero de 2×2 con un casillero removido; ver figura 1(b). El problema consiste en cubrir el tablero con estas fichas de manera que todo casillero esté cubierto exactemente por una ficha, con la excepción del casillero especial que no debe ser cubierto. La figura 1(d) muestra una solución para la instancia de la figura 1(a).

Teorema *El problema del cubrimiento siempre tiene solución.*

Demostración La prueba es por inducción matemática en el natural n tal que $k = 2^n$.

- **Base:** El caso $n = 0$ es trivialmente satisfecho ya que $k = 1$, y el tablero de 1×1 consiste de un único casillero que necesariamente es el casillero especial. Este tablero es cubierto simplemente no haciendo nada. (Si este argumento no es convincente, basta con controlar el siguiente caso: si $n = 1$ entonces $k = 2$, y un tablero de 2×2 es exactamente una ficha a la que se le agrega el casillero especial.)

- **Paso inductivo:** Consideremos cualquier $m \geq 1$ y sea $k = 2^m$. Supongamos la *hipótesis inductiva* que el teorema es válido para tableros de $2^{m-1} \times 2^{m-1}$. Consideremos un tablero de $m \times m$ que contiene un casillero especial arbitrario. Se divide el tablero en cuatro subtableros iguales (esto es posible ya que k es potencia de 2 y $m \geq 1$). El casillero especial pertenece ahora a sólo uno de los cuatro subtableros. Se ubica una ficha en el centro del tablero original de forma de cubrir exactamente un casillero de cada uno de los tres restantes subtableros, como indica la figura 1(c). Cada uno de estos tres casilleros cubiertos serán los casilleros especiales para los correspondientes subtableros. Se dispone entonces de cuatro subtableros de $2^{m-1} \times 2^{m-1}$, cada uno de los cuales tiene un casillero especial. Por nuestra hipótesis inductiva, cada uno de ellos puede ser cubierto, y la solución final se obtiene combinando el cubrimiento de los subtableros con la ficha ubicada en el centro del tablero original.

Dado que el teorema es válido cuando $k = 2^0$, y que su validez para $k = 2^m$ se concluye a partir de la validez para $k = 2^{m-1}$, para todo $m \geq 1$, entonces el principio de inducción matemática permite afirmar que el teorema es verdadero para todos los n que son potencia de 2. ∎

No es difícil transformar la prueba de este teorema en un algoritmo para realizar efectivamente el cubrimiento (tal vez no un algoritmo computacional, pero por lo menos un algoritmo para realizarlo "a mano"). Este algoritmo de cubrimiento sigue el patrón general conocido como *dividir y conquistar*, que se verá en la materia. ∎

Identificaremos ahora en detalle todos los elementos de una prueba por inducción bien formada. Consideremos de nuevo una propiedad abstracta P de los naturales, un natural a y supongamos que queremos probar que $P(n)$ vale para todos los $n \geq a$. Se debe empezar por la *base inductiva*, que consiste en probar que $P(a)$ vale. Este caso base es usualmente fácil, algunas veces incluso trivial, pero es fundamental que sea llevado a cabo apropiadamente. De otra forma, la "prueba" completa carecería de fundamento.

La base inductiva es seguida por el *paso inductivo*, el cual es generalmente el más sustancial. Este debe empezar con "consideremos cualquier $m > a$", o lo que es equivalente $m \geq a + 1$. Debe continuar con una mención explícita de la *hipótesis inductiva* que esencialmente establece que se suponde que vale $P(m-1)$. En este punto resta inferir que $P(m)$ es válido suponiendo la hipótesis inductiva. Finalmente, una oración adicional tal como la que finaliza el teorema anterior puede ser insertada para concluir el razonamiento, aunque generalmente es innecesaria.

Con respecto a la hipótesis inductiva, es importante entender que cuando se supone $P(m-1)$ esto se hace con un caracter *provisional*. En realidad no se sabe si vale o no hasta que el teorema ha sido probado. En otras palabras, el objetivo en el paso inductivo es probar que la verdad de $P(m)$ se concluye lógicamente de la verdad de $P(m-1)$, sin importar si realmente $P(m-1)$ es verdadero o no. De hecho, si $P(m-1)$ no vale el paso inductivo no permite afirmar nada sobre la validez de $P(m)$.

Por ejemplo, consideremos la propiedad "$n^3 < 2^n$", que denotaremos $P(n)$. Para todo entero positivo m es fácil mostrar que $m^3 < 2 \times (m-1)^3$ si y solo si $m \geq 5$. Consideremos cualquier $m \geq 5$ y supogamos provisionalmente que $P(m-1)$ vale. Luego

$$\begin{array}{rcll} m^3 & < & 2 \times (m-1)^3 & \text{porque } m \geq 5 \\ & < & 2 \times 2^{m-1} & \text{por la suposición de que } P(m-1) \text{ vale} \\ & = & 2^m & \end{array}$$

Se ve que $P(m)$ se infiere lógicamente de $P(m-1)$ siempre que $m \geq 5$. No obstante, $P(4)$ *no vale* (esto es $4^3 < 2^4$, o sea $64 < 16$) y entonces nada se puede afirmar sobre $P(5)$. A través de prueba

y error, se puede encontrar sin embargo que $P(10)$ vale (esto es $10^3 = 1000 < 2^{10} = 1024$). Entonces es legítimo afirmar que $P(11)$ vale, y de la validez de $P(11)$ surge la de $P(12)$, y así siguiendo. Por el principio de inducción matemática , dado que $P(10)$ vale y dado que $P(m)$ vale siempre que vale $P(m-1)$ para todo $m \geq 5$, se puede concluir que $n^3 < 2^n$ para todos los $n \geq 10$. Es instructivo notar que $P(n)$ vale también para $n = 0$ y para $n = 1$, pero que no podemos usar esos puntos como base inductiva porque el paso inductivo no se aplica a valores tan chicos de n.

Hay un aspecto de las pruebas por inducción matemática que en principio parece desconcertante, e incluso paradójico: *es a veces más fácil probar una proposición más fuerte que una más débil*. Por ejemplo vimos que es fácil conjeturar que la suma de los cubos de los n primeros números naturales es un cuadrado perfecto. Pero probar por inducción esto no es sencillo. La dificultad radica en que hipótesis inductivas como "la suma de los cubos de los $m-1$ primeros naturales es siempre un cuadrado perfecto" no son de mucha ayuda en probar que lo mismo vale para los m primeros naturales, porque no se especifica *cual* es el cuadrado perfecto. Por el contrario, es mucho más fácil probar por inducción matemática que la suma de los cubos de los n primeros naturales es precisamente el cuadrado de las suma de los primeros n naturales. En este caso la hipótesis inductiva es mucho más significativa.

1.2.2. Todos los alumnos aprueban "Algoritmos y Complejidad"

El error más común en el desarrollo de pruebas por inducción matemática merece una sección aparte. Consideremos el siguiente, absurdo, "teorema".

TEOREMA *Todos los alumnos pertenecientes a algún conjunto de alumnos de "Algoritmos y Complejidad" obtienen la misma nota final.*

Demostración Sea $\mathcal{C}$ un conjunto de alumnos de "Algoritmos y Complejidad". Probaremos por inducción sobre la cantidad n de alumnos en $\mathcal{C}$ que todos obtienen la misma nota final.

- **Base:** el caso $n = 0$ es trivialmente verdadero: si no hay alumnos en el conjunto, entonces seguramente todos tienen la misma nota. (Si el argumento anterior no es convincente, basta con controlar el siguiente caso más simple: si $n = 1$, entonces existe un solo alumno en $\mathcal{C}$, y es claro que "todos" obtienen la misma nota final.)

- **Paso inductivo:** Supongamos que hay m alumnos en C denominados $a_1, a_2, \ldots, a_m$. Supongamos por hipótesis inductiva que cualquier conjunto de $m-1$ alumnos es tal que todos sacan la misma nota final en "Algoritmos y Complejidad" (pero por supuesto, todos los alumnos de un conjunto pueden sacarse una nota distinta que todos los alumnos de otro conjunto). Sea $\mathcal{C}_1$ el conjunto de alumnos $\{a_2, a_3, \ldots, a_m\}$, y $\mathcal{C}_2$ el conjunto de alumnos $\{a_1, a_3, \ldots, a_m\}$. Tanto $\mathcal{C}_1$ como $\mathcal{C}_2$ tienen $m-1$ alumnos, y entonces se les puede aplicar la hipótesis inductiva. En particular todos los alumnos de $\mathcal{C}_1$ sacan nota p, y todos los alumnos de $\mathcal{C}_2$ sacan nota q, posiblemente diferente de p. Pero como $a_m \in \mathcal{C}_1$, el alumno a_m tiene nota p; pero también $a_m \in \mathcal{C}_2$, entonces a_m tiene nota q. Como un alumno no puede tener notas distintas en la misma materia, entonces $p = q$ y todos los alumnos de $\mathcal{C}$ tienen la misma nota $p = q$.

Antes de continuar es interesante tratar de averiguar por uno mismo dónde está la falacia de esta "prueba". Como ayuda digamos que el problema no está en la formulación de la hipótesis inductiva "cualquier conjunto de $m-1$ alumnos es tal que todos sacan la misma nota final en "Algoritmos y Complejidad" ". ∎

Solución: El problema está en que $a_m \in \mathcal{C}_1$ y $a_m \in \mathcal{C}_2$ *no* es cierto si $m = 2$. El razonamiento es impecable para los casos $m = 0$ y $m = 1$. Más aún, es verdadero que el teorema vale para conjuntos de m alumnos siempre que vale para conjuntos de $m - 1$, pero *solo cuando* $m \geq 3$. Se puede pasar de 2 a 3, de 3 a 4, de 4 a 5, y así siguiendo, pero *no* se puede pasar de 1 a 2. Dado que el caso base solo contiene 0 y 1, y no se puede pasar de 1 a 2, el paso inductivo nunca puede comenzar a aplicarse. Este pequeño eslabón faltante en la prueba es suficiente para invalidarla completamente. Es algo similar a lo que pasaba cuando probamos que $n^3 < 2^n$: el paso inductivo no se aplica para $n < 5$, y por lo tanto que la propiedad sea válida para $n = 0$ y $n = 1$ es irrelevante. La diferencia importante está en que $n^3 < 2^n$ es verdadero para $n = 10$, y por lo tanto para todos los valores de n mayores que 10.

1.2.3. Inducción matemática generalizada

El principio de inducción matemática descripto hasta el momento es apropiado para probar varias propiedades interesantes. Sin embargo, existen casos en donde una variante un poco más poderosa es preferible. Esta variante es conocida como *inducción matemática generalizada*, y se ilustra a continuación.

Supongamos que se desea probar que todos los naturales compuestos pueden ser expresados como productos de números primos (el *teorema fundamental de la aritmética* dice que esta descomposición es única; esto no es lo que probaremos acá). No nos preocuparemos todavía por la base de la inducción, sino que pasaremos directamente al paso inductivo. Cuando se trata de probar que m puede ser expresado como producto de números primos (suponiendo que m es compuesto), la hipótesis inductiva "natural" sería que $m - 1$ también puede ser descompuesto. Sin embargo, no existe nada en la descomposición en primos de $m - 1$ que sirva para la descomposición en primos de m. Lo que en realidad se necesita es la hipótesis inductiva más fuerte que *todos* los números compuestos menores que m pueden ser descompuestos como productos de primos. La prueba correcta del teorema se muestra más adelante, después que se haya enunciado formalmente el principio de inducción matemática generalizada.

Otra generalización muchas veces útil tiene que ver con la base. A veces es necesario probar una *base extendida*, esto es probar la base en más de un punto. Notar que se han probado bases extendidas para la correctitud de `sq` y para el problema del cubrimiento, pero eso fue un lujo: el paso inductivo pudo haber sido aplicado para probar el caso de $n = 1$ a partir de la base $n = 0$. Pero este no es siempre el caso; muchas veces se debe probar independientemente la validez de varias bases inductivas antes de atacar el paso inductivo.

Estamos en condiciones entonces de formular un principio de inducción matemática más general. Sea P cualquier propiedad de los naturales, y a y b dos naturales tales que $a \leq b$. Si

1. $P(n)$ vale para todo $a \leq n < b$ y,

2. para cualquier natural $n \geq b$, el hecho de que $P(n)$ vale es consecuencia de la suposición que $P(m)$ vale para todos los m tales que $a \leq m < n$,

entonces la propiedad $P(n)$ vale para todos los naturales $n \geq a$.

Aún es posible una generalización más del principio de inducción matemática conveniente cuando no estamos interesados en probar una propiedad para *todos* los naturales mayores que la base. Es frecuente el caso en el que se desea probar que alguna propiedad P vale, pero solo para los naturales para los cuales vale también otra propiedad Q. Hemos visto dos ejemplos de esta situación: el problema del cubrimiento se aplica solo a los k que son potencias de 2, y nuestra propiedad sobre la descomposición en primos se aplica solo a números compuestos (aunque se puede extender naturalmente a números primos). Cuando esto ocurre, es suficiente

con mencionar Q explícitamente en el enunciado del teorema a ser probado, probar la base (posiblemente extendida) solo para puntos que cumplen Q, y probar el paso inductivo solo también en estos puntos. Por supuesto, la hipótesis inductiva tiene que ser debilitada en la misma forma. Consideremos cualquier n mayor que la base tal que $Q(n)$ vale. Para probar que $P(n)$ vale, solamente es posible suponer que $P(m)$ vale siempre que $a \leq m < n$ y que $Q(m)$ también vale. En el problema del cubrimiento, solamente podemos usar la hipótesis inductiva para cubrir tableros de 4×4 cuando lo estamos probando para el cubrimiento 8×8, pero *no* se puede asumir lo mismo para un tablero de 5×5.

Antes de ilustrar este principio, observemos que se permite que la base sea vacía. Esto ocurre cuando $a = b$ porque en este caso no existen naturales n tales que $a \leq n < b$. También puede pasar cuando $a < b$ si $Q(n)$ nunca vale cuando $a \leq n < b$. Esto no invalida la prueba porque en este caso la validez de $P(n)$ para el menor n en el cual se aplica el paso inductivo se prueba bajo una hipótesis de inducción vacía, esto es se prueba sin ninguna suposición. El siguiente ejemplo muestra este caso.

Ejemplo 1.4
Teorema *Todo entero positivo compuesto puede ser expresado como el producto de números primos.*

Demostración La prueba es por inducción matemática generalizada. En este caso no hay necesidad de una base.

- **Paso inductivo:** Sea cualquier natural compuesto $n \geq 4$ (observar que 4 es el menor compuesto positivo, por lo que no tiene sentido considerar valores menores que n). Supongamos por hipótesis inductiva que cualquier compuesto positivo menor que n puede ser expresado como producto de números primos (en el caso más chico cuando $n = 4$, esta hipótesis inductiva es vacía). Consideremos el menor natural d que es mayor que 1 y que es divisor de n. Como se argumentó en la prueba del teorema de Euclides, necesariamente d es primo. Sea $m = n/d$. Notar que $1 < m < n$ porque n es compuesto y $d > 1$. Se presentan dos casos

  - si m es primo, se ha descompuesto n como el producto de dos primos $n = d \times m$.
  - si m es compuesto, entonces también es positivo y menor que n, y por lo tanto se aplica la hipótesis inductiva: m puede ser expresado como producto de primos, digamos $m = p_1 \times p_2 \times \ldots \times p_k$. Por lo tanto, $n = d \times p_1 \times p_2 \times \ldots \times p_k$ producto de primos.

En cualquiera de ambos casos, se completó la prueba del paso inductivo, y por lo tanto del teorema. ∎

El ejemplo que veremos a continuación no solo muestra una prueba por inducción matemática generalizada, sino que también ilustra cómo combinar distintos resultados parciales para la demostración de un teorema general. Esta técnica, similar a la definición de procedimientos dentro de un programa, es comunmente usada para facilitar la legibilidad y el entendimiento de la demostración.

Ejemplo 1.5 Sean m y n dos números enteros positivos. El *máximo común divisor* de m y n, denotado $\gcd(m, n)$, es el más grande natural que divide exactamente tanto a m como a n. Por ejemplo, $\gcd(6, 15) = 3$ y $\gcd(10, 21) = 1$. Existen varios algoritmos para calcular el máximo

común divisor entre dos números, en particular el generalmente enseñado en la escuela primaria de encontrar los factores primos de ambos números y luego multiplicar aquellos factores comunes entre los dos.

Sin embargo, existe un algoritmo mucho más eficiente para calcular un máximo común divisor. Este es el famoso *algoritmo de Euclides*, también atribuido al matemático griego, y uno de los algoritmos no triviales más antiguos conocidos.

```
función gcd( m,n )
  mientras m>0
    temp := n mod m;
    n := m;
    m := temp;
  retornar n;
```

Para ser históricamente exactos, el algoritmo original de Euclides trabaja con restas sucesivas en lugar de calcular el módulo.

A continuación analizaremos la eficiencia de la versión enunciada del algoritmo a través de calcular una cota superior para la cantidad de veces que se repite el ciclo `mientras`, para dos valores cualesquiera de m y n. Antes de pasar directamente a probar el teorema, probaremos primero dos lemas auxiliares. En general, se puede decir que un lema es una proposición matemática que se usa en la demostración de un teorema. Es decir, el resultado del lema no tiene sentido por sí mismo, sino que solamente se usa para incluirlo en la prueba de un teorema. En este caso, el segundo lema usará inducción generalizada ya que para probar el paso inductivo para m necesita aplicar la hipótesis inductiva a $m-2$ en lugar de a $m-1$.

LEMA 1 *Sean n y m dos enteros positivos tal que $n \geq m$, entonces siempre es válido que $n \bmod m < n/2$.*

Demostración Sabemos que $n \bmod m < m$ por definición de módulo (ver sección 2.4). Entonces si $m \leq n/2$ el teorema ya está probado; resta considerar el caso que $n/2 < m < n$. Si $m > n/2$ entonces $1 \leq n/m < 2$, y por lo tanto $\lfloor n/m \rfloor = 1$. Esto implica que $n \bmod m = n - m < n - n/2 = n/2$. Luego en ambos casos vale el enunciado. ∎

LEMA 2 *Sea m_i el valor de la variable m al final de la i-ésima iteración del ciclo* `mientras` *del algoritmo de Euclides. Entonces $m_i < m_0/2^{\lfloor i/2 \rfloor}$ para todo $i \geq 1$.*

Demostración Análogamente a m_i, notemos con n_i el valor de la variable n al final de la i-ésima iteración del ciclo. La prueba es por inducción generalizada sobre i.

- **Base:** si $i = 1$, por el programa es claro que $m_1 = n_0 \bmod m_0 < m_0 = m_0/2^{\lfloor 1/2 \rfloor}$.
- **Paso inductivo:** sea $k \geq 2$, supongamos por hipótesis inductiva que $m_j < m_0/2^{\lfloor j/2 \rfloor}$ para todo $1 \leq j < k$, y tratemos de probar la propiedad para m_k. Por el programa sabemos que $m_k = n_{k-1} \bmod m_{k-1}$, y claramente $n_i > m_i$ para todo $i \geq 1$ (para ser totalmente formal, también habría que probar por inducción esta última propiedad). Entonces, aplicando el lema 1 tenemos que $m_k < n_{k-1}/2$. También por el programa podemos determinar que $n_i = m_{i-1}$ para todo $i \geq 1$ (otra prueba por inducción más). Luego $m_k < m_{k-2}/2$, y aplicando la hipótesis inductiva ya que $k - 2 < k$ obtenemos

$$m_k < m_{k-2}/2 < m_0/(2^{\lfloor (k-2)/2 \rfloor} \times 2) = m_0/2^{\lfloor (k-2)/2 \rfloor + 1} = m_0/2^{\lfloor k/2 \rfloor}$$

  dado que $\lfloor (k-2)/2 \rfloor + 1 = \lfloor ((k-2)/2) + 1 \rfloor = \lfloor k/2 \rfloor$.

TEOREMA *El ciclo* `mientras` *del algoritmo de Euclides se ejecuta menos de* $1+2\times \lg m_0$ *veces.*

Demostración Claramente la cantidad k de iteraciones del ciclo está determinada cuando $m_k = 0$, y entonces $m_{k-1} \geq 1$. Pero por el lema 2, sabemos que $m_{k-1} < m_0/2^{\lfloor (k-1)/2 \rfloor}$ y entonces $2^{\lfloor (k-1)/2 \rfloor} < m_0$ y $k \leq 1 + 2 \times \lg m_0$. ∎

Hemos probado entonces que el ciclo `mientras` del algoritmo se repite menos de $1 + 2 \times \lg m_0$ veces, siend m_0 el valor inicial de m. Intuitivamente, podemos afirmar que esto nos da una idea del tiempo de ejecución del algoritmo, ya que el ciclo constituye la parte principal del mismo y el contenido del ciclo son asignaciones y operaciones aritméticas comunes. Informalmente, también podemos observar que esta cantidad de iteraciones no es grande. Cuando veamos las técnicas para analizar el tiempo de ejecución de algoritmos, retomaremos en particular este resultado para un análisis más formal. ∎

1.2.4. Inducción constructiva

La inducción matemática es usada principalmente como una técnica de prueba. Frecuentemente, es empleada para probar aserciones que parecen haber sido sacadas de la galera de un mago. Mientras que la validez de estas aserciones queda demostrada, su origen permanece misterioso. Sin embargo, la inducción matemática es una herramienta suficientemente poderosa para permitirnos descubrir no solamente la validez de un teorema, sino también su enunciado preciso. Aplicando la técnica de *inducción constructiva* descripta a continuación, se puede simultáneamente probar la verdad de una propiedad parcialmente especificada y descubrir las especificaciones faltantes gracias a las cuales el teorema es correcto. Ilustraremos esta técnica con dos ejemplos sobre la *secuencia de Fibonacci*, definida a continuación. El segundo ejemplo muestra cómo la técnica puede ser aplicada al análisis de algoritmos.

La secuencia cuyo nombre es dado por Fibonacci, matemático italiano del siglo XII, tiene la siguiente definición formal:

$$F_n = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ F_{n-1} + F_{n-2} & \text{si } n \geq 2 \end{cases}$$

La secuencia empieza 0,1,1,2,3,5,8,13,21,34,... Cuenta con numerosas aplicaciones en ciencias de la computación, en matemática, y en teoría de juegos. De Moivre produjo la siguiente fórmula, que es fácil de probar usando inducción matemática:

$$F_n = \frac{1}{\sqrt{5}} \times [\phi^n - (-\phi)^{-n}]$$

donde $\phi = (1 + \sqrt{5})/2$ es el denominado *número de oro*. Dado que $0 < \phi^{-1} < 1$, el término $(-\phi)^{-n}$ sólo puede ser ignorado cuando n es grande. Luego el valor de F_n es aproximadamente $\phi^n/\sqrt{5}$, que es exponencial en n.

Pero, ¿cuál es el origen de la fórmula de De Moivre? En la materia se verá una técnica general para resolver recurrencias como esta de Fibonacci. Mientras tanto, supongamos que no conocemos estas técnicas ni la fórmula de De Moivre, y que necesitamos tener una idea del comportamiento de la secuencia de Fibonacci. Si se calculan algunos valores de la secuencia,

enseguida se descubre que crece muy rápidamente (F_{100} es un número de 21 dígitos). Entonces, es razonable la siguiente conjetura "la secuencia de Fibonacci crece en forma exponencial". ¿Cómo se hace para probar esto? La dificultad radica en que la conjetura es demasiado vaga para ser probada en forma directa por inducción matemática: recordemos que frecuentemente es más fácil probar una proposición fuerte que una débil. Entonces podemos adivinar que existe un número real $x > 1$ tal que $F_n \geq x^n$ para todo entero n suficientemente grande (esta proposición es evidentemente falsa para $n \leq 2$). En símbolos

$$\text{Conjetura: } (\exists x)(\forall n \in \mathbf{N}, n \text{ suficientemente grande})(F_n \geq x^n)$$

Existen dos incógnitas en el enunciado de la conjetura que queremos probar: el valor de x y el significado de "suficientemente grande". Olvidemos esta última por el momento. Sea $P_x(n)$ la proposición "$F_n \geq x^n$" y consideremos un entero suficientemente grande n. El enfoque por inducción constructiva consiste en preguntarnos para cuáles valores de x $P_x(n)$ es consecuencia de la *hipótesis inductiva parcialmente especificada* que dice que $P_x(m)$ vale para todo entero m que es menor que n pero áun suficientemente grande. Usando la definición de secuencia de Fibonacci y esta hipótesis, y suponiendo que $n-1$ y $n-2$ son todavía suficientemente grandes, entonces

$$F_n = F_{n-1} + F_{n-2} \geq x^{n-1} + x^{n-2} = (x^{-1} + x^{-2}) \times x^n$$

Para concluir que $F_n \geq x^n$, necesitamos que $x^{-1}+x^{-2} \geq 1$, o lo que es equivalente $x^2-x-1 \leq 0$. Aplicando álgebra elemental (ver sección 2.9) y considerando que sólo nos interesa el caso que $x > 1$, resolver esta inecuación cuadrática implica que $1 < x \leq \phi = (1+\sqrt{5})/2$.

Hemos probado que $P_x(n)$ es consecuencia de $P_x(n-1)$ y $P_x(n-2)$ siempre que $1 < x \leq \phi$. Esto corresponde a probar el paso inductivo en una prueba por inducción matemática. Para aplicar el principio de inducción matemática y concluir que la secuencia de Fibonacci crece exponencialmente, debemos atacar el problema de la base inductiva. En este caso, dado que la validez de $P_x(n)$ depende en la validez de $P_x(n-1)$ y $P_x(n-2)$, debemos probar que la propiedad P_x vale en dos enteros positivos consecutivos para asegurar que vale de ahí en adelante.

Resulta que no existe un natural n tal que $F_n \geq \phi^n$. Sin embargo, es fácil encontrar dos naturales consecutivos para los cuales la propiedad P vale para x estrictamente menor que ϕ. Por ejemplo, $P_x(11)$ y $P_x(12)$ valen para $x = 3/2$. Por lo tanto, $F_n \geq (3/2)^n$ para todo $n \geq 11$. Esto completa la prueba de que la secuencia de Fibonacci crece por lo menos exponencialmente. El mismo proceso puede ser usado para probar que no crece más rápido que exponencialmente: $F_n \leq y^n$ para todo entero positivo n siempre que $y \geq \phi$. Aquí, de nuevo, la condición sobre y no es herencia divina: se obtiene de nuevo por inducción constructiva cuando se trata de encontrar condiciones sobre y de forma que el paso inductivo pueda seguir adelante. Juntando estas observaciones, podemos llegar a la conclusión que F_n crece exponencialmente; más precisamente como una potencia de un número cercano a ϕ. Lo remarcable es que pudimos llegar a esta conclusión sin necesidad de una fórmula explícita como la fórmula de De Moivre.

En resumen, el proceso de inducción constructiva consiste en establecer la conjetura inicial, que incluye una o más incógnitas. A través del desarrollo de la prueba del paso inductivo con una hipótesis inductiva parcialmente especificada, y a continuación de la base, se descubren restricciones que estas incógnitas deben satisfacer. Si existen elementos que satisfacen las restricciones encontradas para las incógnitas, entonces se eligen valores que las satisfacen y a continuación se está en condiciones de realizar una prueba formal por inducción de la conjetura en donde las incógnitas fueron reemplazadas por los valores elegidos. Si no existen elementos que satisfacen las restricciones encontradas, entonces nuestra conjetura inicial no puede ser probada por inducción. Ante esta alternativa existen tres opciones: la conjetura inicial es falsa, y por lo tanto no puede ser probada; se elige algún otro método de prueba para la conjetura;

o se modifica la conjetura, tal vez introduciendo nuevas incógnitas y fortaleciendo la hipótesis inductiva parcialmente especificada, de forma de probar inducción constructiva otra vez.

Nuestro segundo ejemplo de inducción constructiva muestra este último caso. Trata sobre el análisis del algoritmo obvio para calcular la secuencia de Fibonacci:

```
función Fibonacci ( n )
  si n<2 entonces retornar n;
        sino retornar Fibonacci(n-1)+Fibonacci(n-2);   (*)
```

Sea $g(n)$ el número de veces que la instrucción marcada (*) es ejecutada cuando se llama a `Fibonacci`(n) (contando las instrucciones ejecutada en llamadas recursivas). Esta función es interesante porque $g(n)$ da una cota para el tiempo requerido por la llamada a `Fibonacci`(n).

Claramente, $g(0) = g(1) = 0$. Cuando $n \geq 2$ la instrucción (*) es ejecutada una vez en el nivel más alto, y $g(n-1)$ y $g(n-2)$ veces la primera y la segunda llamada recursiva respectivamente. Por lo tanto,

$$g(n) = \begin{cases} 0 & \text{si } n = 0 \\ 0 & \text{si } n = 1 \\ g(n-1) + g(n-2) + 1 & \text{si } n \geq 2 \end{cases}$$

Esta fórmula es similar a la recurrencia que define la misma secuencia de Fibonacci. Es por lo tanto razonable conjeturar la existencia de constantes reales positivas a, b tales que $a \times F_n \leq g(n) \leq b \times F_n$ para todos los naturales n suficientemente grandes. Usando inducción constructiva, es directo encontrar que $a \times F_n \leq g(n)$ para n suficientemente grandes dado que vale en dos naturales consecutivos sin importar el valor de a. Por ejemplo, tomando $a = 1$, $F_n \leq g(n)$ vale para todo $n \geq 2$.

Sin embargo cuando tratamos de probar la otra parte de nuestra conjetura, es decir que existe b real positivo tal que $b \times F_n \geq g(n)$ para n suficientemente grande, se presenta un problema. Para ver lo que pasa realmente, sea $P_b(n)$ la proposición "$g(n) \leq b \times F_n$" y consideremos un n suficientemente grande. Queremos determinar condiciones sobre el valor b de forma de hacer que $P_b(n)$ se concluya de la hipótesis que $P_b(m)$ vale para los $m < n$ suficientemente grandes. Usando la definición de la secuencia de Fibonacci y esta hipótesis inductiva parcialmente especificada, y considerando que $n-1$ y $n-2$ son suficientemente grandes,

$$g(n) = g(n-1) + g(n-2) + 1 \leq b \times F_{n-1} + b \times F_{n-2} + 1 = b \times F_n + 1$$

Por lo que se infiere que $g(n) \leq b \times F_n + 1$, pero no que $g(n) \leq b \times F_n$. Cualquiera sea el valor de b, es imposible probar el paso inductivo dado que b es positivo.

¿Significa esto que la conjetura original es falsa, o simplemente que la inducción constructiva no es lo suficientemente potente para probarla? Ninguna de los dos. El truco consiste en usar inducción constructiva para probar que existen constantes reales b, c tales que $g(n) \leq b \times F_n - c$ para todos los n suficientemente grandes. Esto puede parecer extraño, dado que $g(n) \leq b \times F_n - c$ es una proposición más fuerte que $g(n) \leq b \times F_n$, la cual fuimos incapaces de probar. Podemos tener esperanzas de éxito, sin embargo, considerando que si la proposición a ser probada es más fuerte, entonces también lo es la hipótesis inductiva que está permitido usar.

Sea entonces un natural n suficientemente grande. Debemos determinar para cuáles valores de b y c la verdad de $g(n) \leq b \times F_n - c$ es consecuencia de la hipótesis inductiva parcialmente especificada que $g(m) \leq b \times F_m - c$ para todos los $m < n$ suficientemente grandes. Usando la definición de la secuencia de Fibonacci y esta hipótesis, y siendo $n-1$ y $n-2$ suficientemente grandes,

$$g(n) = g(n-1) + g(n-2) + 1 \leq b \times F_{n-1} - c + b \times F_{n-2} - c + 1 = b \times F_n - 2 \times c + 1$$

Para concluir que $g(n) \leq b \times F_n - c$ es suficiente con que $-2 \times c + 1 \leq -c$, o lo que es equivalente que $c \geq 1$. Hemos probado entonces que la verdad de nuestra conjetura en un dado natural n es consecuencia de la verdad en sus dos inmediatos predecesores siempre que $c \geq 1$, sin importar los valores de b. Antes de afirmar que se ha probado el teorema, debemos determinar dos valores de b y c que hagan que la conjetura se verdadera en dos naturales consecutivos. Por ejemplo, tomando $b = 2$ y $c = 1$ hacen que sea verdadera en $n = 1$ y $n = 2$, y por lo tanto $g(n) \leq 2 \times F_n - 1$ para todo $n \geq 1$.

La idea central del fortalecimiento de la proposición no totalmente especificada cuando la inducción constructiva es insuficiente parece una técnica mágica. No obstante, es una idea que aparece muy naturalmente con la experiencia del uso de la inducción constructiva.

2. Fórmulas y propiedades útiles

2.1. Desigualdades

Sean a, b, c, d números reales.

- si $a < b$ y $b < c$, entonces $a < c$ (transitividad)
- si $a < b$, entonces $a + c < b + c$ para todo c (monotonicidad de la suma)
- si $a < b$ y $0 < c$, entonces $a \times c < b \times c$ (monotonicidad del producto positivo)
- si $a < b$ y $c < 0$, entonces $b \times c < a \times c$ (antimonotonicidad del producto negativo)
- si $a < b$ y $c < d$, entonces $a + c < b + d$

Estas propiedades también valen en el caso de $\leq$.

2.2. Valor absoluto

Sean r, s, a, b números reales.

- $|r| = 0$ si y solo si $r = 0$
- $|r| = |-r|$
- $|a - b| = |b - a|$
- $|r \times s| = |r| \times |s|$
- $|r/s| = |r|/|s|$, para $s \neq 0$
- $-|r| \leq r \leq |r|$
- $|a + b| \leq |a| + |b|$ (desigualdad triangular)

2.3. Pisos y techos

Sea x, a, b, n números enteros, $a, b \neq 0$.

- $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
- $\lceil \lceil x/a \rceil / b \rceil = \lceil x/(a \times b) \rceil$
- $\lfloor \lfloor x/a \rfloor / b \rfloor = \lfloor x/(a \times b) \rfloor$
- $\lceil x/a \rceil \leq (x + (a - 1))/a$
- $\lfloor x/a \rfloor \geq (x - (a - 1))/a$

2.4. Aritmética modular

Sean a, b, c, n enteros positivos. Recordar que, por definición, $a \bmod b = c$ si existe natural k tal que $a = b \times k + c$; que b divide a a si $a \bmod b = 0$; y que $a \equiv b \pmod{c}$ si y solo si c divide $a - b$. Esta relación entre enteros es reflexiva, simétrica y transitiva. Si $a \equiv b \pmod{n}$, entonces valen:

- $a + c \equiv b + c \pmod{n}$
- $a \times c \equiv b \times c \pmod{n}$

2.5. Exponentes

Sean x, a, b números reales, $x \neq 0$.

- $x^0 = 1$, $x^1 = x$, $x^{-1} = 1/x$
- $x^a \times x^b = x^{a+b}$
- $x^a / x^b = x^{a-b}$
- $(x^a)^b = x^{a \times b} = (x^b)^a$
- $x^a + x^a = 2x^a \neq x^{2a}$
- $2^a + 2^a = 2^{a+1}$

2.6. Logaritmos

Sean a, b, c reales positivos, y n real. Indicaremos los logaritmos en base 2 como lg, los logaritmos en base e como ln, y los logaritmos en base 10 como log. Recordar que, por definición, $a^b = c$ si y solo si $\log_a c = b$. Además, se usa la notación $\log_b^n a = (log_b a)^n$, $\log_b \log_c a = \log_b(\log_c a)$ y $\log_b a + n = (\log_b a) + n \neq \log_b(a + n)$.

- $\log_b a = \log_c a / \log_c b$, para $c > 0$
- $\log_b a = 1/(\log_a b)$
- $\log_c(a \times b) = \log_c a + \log_c b$
- $\log_c(a/b) = \log_c a - \log_c b$
- $\log_c(a^n) = n \times \log_c a$
- $a^{\log_b c} = c^{\log_b a}$
- $\log_b(1/a) = -\log_b a$
- $\log_c a < a$
- $\log_c 1 = 0$
- $\lg 2 = 1$
- $\lg 1024 = 10$
- $\lg 1048576 = 20$

2.7. Factoriales

Sea n un número natural. La primer propiedad puede usarse para demostrar las demás.

- aproximación de Stirling:

$$n! = \sqrt{2 \times \pi \times n} \times (n/e)^n \times (1 + \Theta(\frac{1}{n}))$$

- $n! \in o(n^n)$
- $n! \in \omega(2^n)$
- $\log(n!) \in \Theta(n \times \log n)$

2.8. Números combinatorios

Sea k un entero positivo, y $k \leq n$. La k combinación de n es un posible subconjunto de k elementos tomados de un conjunto de n elementos. El número $\binom{n}{k}$ es la cantidad de todas las posibles k combinaciones de n, y se define

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Otra definición equivalente, recursiva, es la siguiente

$$\binom{n}{k} = \begin{cases} 1 & \text{si } k = 0 \text{ o } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{si } 0 < k < n \\ 0 & \text{de otra forma} \end{cases}$$

- $\binom{n}{k} = \binom{n}{n-k}$
- $\binom{m+1}{n} = \binom{m}{n} + \binom{m}{n-1}$
- $(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \times a^i \times b^{n-i}$
- $\sum_{r=0}^{n} \binom{n}{r} = 2^n$

2.9. Raíces de polinomios

El polinomio $a \times x^2 + b \times x + c$ tiene dos raíces dadas por la fórmula:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 \times a \times c}}{2 \times a}.$$

Recordar además que k es raíz de un polinomio $p(x)$ si y solo si $p(x)$ es divisible por el polinomio $x - k$, y que un polinomio de grado n tiene exactamente n raíces (no necesariamente todas distintas ni reales).

2.10. Límites de sucesiones

Definición 2.1 $\lim_{n\to\infty} a_n = L$ si para cada $\epsilon > 0$ existe un número positivo N tal que $|a_n - L| < \epsilon$ siempre que $n > N$. ∎

Definición 2.2 $\lim_{n\to\infty} a_n = \infty$ significa que para todo número real positivo P existe un número N tal que si $n > N$, entonces $a_n > P$. ∎

- $\lim_{n\to\infty} r^n = 0 \quad \text{si} \quad |r| < 1$
- $\lim_{n\to\infty} |r^n| = \infty \quad \text{si} \quad |r| > 1$
- $\lim_{n\to\infty}(a_n + b_n) = \lim_{n\to\infty} a_n + \lim_{n\to\infty} b_n$
- $\lim_{n\to\infty}(a_n - b_n) = \lim_{n\to\infty} a_n - \lim_{n\to\infty} b_n$
- $\lim_{n\to\infty}(a_n \times b_n) = \lim_{n\to\infty} a_n \times \lim_{n\to\infty} b_n$
- $\lim_{n\to\infty} \frac{a_n}{b_n} = \frac{\lim_{n\to\infty} a_n}{\lim_{n\to\infty} b_n} \quad \text{donde} \quad \lim_{n\to\infty} b_n \neq 0 \ \text{ y } \ b_n \neq 0 \ \text{ para todo } \ n$
- $\lim_{n\to\infty} c = c$

2.11. Series

- $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ (serie geométrica)
- $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ (caso especial de la precedente)
- $\sum_{i=1}^{n} i \times (i+1) \times (i+2) \times \ldots \times (i+k) = n \times (n+1) \times (n+2) \times \ldots \times (n+k+1)/(k+2)$ para todo k natural
- $\sum_{i=1}^{n} i = \frac{n\times(n+1)}{2} \approx \frac{n^2}{2}$ (serie aritmética, caso especial de la anterior)
- $\sum_{i=1}^{n} i^2 = \frac{n\times(n+1)\times(2n+1)}{6} \approx \frac{n^3}{3}$
- $\sum_{i=1}^{n} i^k = n^{k+1}/(r+1) + p_r(n) \approx \frac{n^{k+1}}{|k+1|}$ para todo k natural, siendo $p_r(n)$ un polinomio en n de grado a lo sumo r.
- $\sum_{i=1}^{n} \frac{1}{i} \approx \log_e n$ (suma H_n de los n primeros números armónicos)
- $\sum_{i=1}^{n} f(n) = n \times f(n)$
- $\sum_{i=m}^{n} f(i) = \sum_{i=1}^{n} f(i) - \sum_{i=1}^{m-1} f(i)$

2.12. Función logarítmica iterada

Se usa la notación $\lg^* n$ para denotar el logaritmo iterado, que se define a continuación. Sea $\lg^{(i)} n$ definida recursivamente para i natural

$$\lg^{(i)} n = \begin{cases} n & \text{si } i = 0 \\ \lg(\lg^{(i-1)} n) & \text{si } i > 0 \text{ y } \lg^{(i-1)} n > 0 \\ \text{indefinido} & \text{en otro caso} \end{cases}$$

No hay que confundir $\lg^{(i)} n$ (la función logaritmo aplicada i veces en sucesión) con $(\lg n)^i$ (el logaritmo elevado a la i-ésima potencia, a veces notado con $\lg^i n$). Entonces la función logaritmo iterada está definida

$$\lg^* n = \min \left\{ i \geq 0 : \lg^{(i)} n \leq 1 \right\}$$

Es una función de un crecimiento *muy* lento:

$$\begin{aligned} \lg^* 2 &= 1 \\ \lg^* 4 &= 2 \\ \lg^* 16 &= 3 \\ \lg^* 65536 &= 4 \\ \lg^*(2^{65536}) &= 5 \end{aligned}$$

Dado que el número de átomos en el universo observable está estimado en aproximadamente 10^{80}, que es mucho menor que 2^{65536}, es raro encontrar un valor de n tal que $\lg^* n > 5$.