Conceptos matemáticos preliminares

Pedro O. Pérez M., PhD.

Implementación de métodos computacionales
Tecnológico de Monterrey

pperezm@tec.mx

02-2023

Contenido I

Probar los programas es fundamental. Sin embargo, la realización de pruebas sólo llega hasta cierto punto, ya que no es posible probar los programas para todas las posibles entrada. Aún más importante, si el programa es complejo, por ejemplo contiene recursiones o iteraciones, entonces si no se comprende qué es lo que ocurre al ejecutar un ciclo o una llamada a una función en forma recursiva, es poco probable que podamos escribir el código correctamente. Si al probar el código resulta ser incorrecto, será necesario corregirlo.

Para conseguir iteraciones o recursiones correctas, es necesario establecer hipótesis inductivas, y resulta útil razonar, formal o informalmente, que la hipótesis es coherente con la iteración o recursión. Este proceso sirve para compreder que el trabajo que realiza un programa correcto es esencialmente el mismo que el proceso de demostrar teoremas por inducción.

Como mencionamos anteriormente, una demostración deductiva consta de una secuencia de proposiciones cuya veracidad se comprueba partiendo de una proposición inicial, conocida como hipótesis o de una serie de proposiciones dadas, hasta llegar a una conclusión. Cada uno de los pasos de la demostración, hay que deducirlos mediante algún principio lógico aceptado, bien a partir de los postulados o de algunas de las proposiciones anteriores de la demostración deductiva o de una combinación de éstas.

Existen varias formas en que podemos construir demostraciones:

1. Empleando conjuntos.
2. Por reducción al absurdo.
3. Mediante contra-ejemplo.

Demostración de equivalencias entre conjuntos

En la teoría de autómatas, frecuentemente es necesario demostrar un teorema que establece que los conjuntos construidos de dos formas diferentes son el mismo conjunto. A menudo, se trata de conjuntos de cadenas de caracteres y se denominan "lenguajes".

Veamos un ejemplo. Si E y F son dos expresiones que representan conjuntos, la proposición $E = F$ quiere decir que los dos conjuntos representados son iguales. De forma má precisa, cada uno de los elementos del conjunto representado por E está en el conjunto representado por F, y cada uno de los elementos del conjunto representado por F está en el conjunto representado por E.
Demostrar, Teorema 6. $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$.

Figura: Pasos correspondientes a la parse "si" del Teorema 6

	Proposición	Justificación
1.	x pertenece a $R \cup (S \cap T)$	Postulado
2.	x pertenece a R o x pertenece a $S \cap T$	(1) y la definición de unión
3.	x pertenece a R o x pertenece a S y T	(2) y la definición de intersección
4.	x pertenece a $R \cup S$	(3) y la definición de unión
5.	x pertenece a $R \cup T$	(3) y la definición de unión
6.	x pertenece a $(R \cup S) \cap (R \cup T)$	(4), (5), y la definición de intersección

Figura: Pasos correspondientes a la parse "sólo si" del Teorema 6

	Proposición	Justificación
1.	x pertenece a $(R \cup S) \cap (R \cup T)$	Postulado
2.	x pertenece a $R \cup S$	(1) y la definición de intersección
3.	x pertenece a $R \cup T$	(1) y la definición de intersección
4.	x pertenece a R o x pertenece a S y T	(2), (3), y el razonamiento sobre la unión
5.	x pertenece a R o x pertenece a $S \cap T$	(4) y la definición de intersección
6.	x pertenece a $R \cup (S \cap T)$	(5) y la definición de unión

Demostración por reducción al absurdo

Otra forma de demostrar una proposición de la forma "si H entonces C" consiste en demostrar la proposición: "H y no C implica falsedad". Es decir, comenzamos suponiendo que tanto la hipótesis H como la negación de la conclusión C son verdaderas. La demostración se completa probando que algo que se sabe que es falso se deduce lógicamente a partir de H y C. Esta forma de demostración se conoce como demostración por reducción al absurdo.

Recuerda la forma en que demostramos el Teorema 3: Sea S un subconjunto finitos de un determinaod conjunto infinito U. Sea T el conjunto complementario de S con respecto a U. Entonces T es infinito.

Demostrar, Teorema 3: Sea S un subconjunto finitos de un determinado conjunto infinito U. Sea T el conjunto complementario de S con respecto a U. Entonces T es infinito.

1. Un conjunto S es finito si existe un entero n tal que S tiene exactamente n elementos. Escribimos $||S||$ se utiliza para designar el número de elementos de un conjunto S. Si el conjunto S no es finito, decimos que S es infinito. Intuitivamente, un conjunto infinito es un conjunto que contiene más que cualquier número entero de elementos.
2. Si S y T son subconjuntos de algún conjunto U, entonces T es el complementario de S (con respecto a U) si $S \cup T = U$ y $S \cap T = \emptyset$. Es decir, cada elemento de U es exactamente uno de S y otro de T; dicho de otra manera, T consta exactamente de aquellos elementos de U que no pertenecen a S.
3. Demostración por reducción a lo absurdo.

Una de las pruebas más importantes y reconocidas que utiliza la reducción al absurdo es "The Halting Problem - Prueba de que las computadoras no pueden hacer todo (El Problema de la Parada)"

En grupos de tres personas, discute las siguientes preguntas:

- ¿Porqué crees que es importante esta demostración?
- ¿Qué implicaciones tiene?

Revisemos el siguiente vídeo. En él, encontraremos respuestas a las preguntas antes planteadas: “The Halting Problem - An Impossible Problem to Solve”

En la práctica, no se habla de demostrar un teorema, sino que tenemos que enfrentarnos a algo que parece que es cierto, por ejemplo, una estrategia para implementar un programa y tenemos que decidir si el "teorema" es o no verdadero. Para resolver este problema, podemos intentar demostrar el teorema, y si no es posible, intentar demostrar que la proposición es falsa.

Generalmente, los teoremas son proposiciones que incluyen un número infinito de casos, quizás todos los valores de sus parámetros.

Suele ser más fácil demostrar que una proposición no es un teorema que demostrar que sí lo es.

Existe una forma especial de demostración, denominada "inductiva", que es esencial a la hora de tratar con objetos definidos de forma recursiva. Muchas de las demostraciones inductivas más habituales trabajan con enteros, pero en la teoría de autómatas, también necesitamos demostraciones inductivas, por ejemplo, para conceptos definidos recursivamente como pueden ser árboles y expresiones de diversas clases, como expresiones regulares.

Supón que tenemos que demostrar una proposición $S(n)$ acerca de un número entero n. Un enfoque que se emplea habitualmente consiste en demostrar dos cosas:

1. El *caso base*, donde demostramos $S(i)$ para un determinado entero i. Normalmente, $i = 0$ o $i = 1$, pero habrá ejemplos en los que desearemos comenzar en cualquier valor mayor de i, quizá porque la proposición S sea falsa para los enteros más pequeños.
2. El *paso de inducción*, donde suponemos que $n \geq i$, siendo i el entero empleado en el caso base, y demostramos que "si $S(n)$ entonces $S(n+1)$".

Intuitivamente, estas dos partes deberían convencernos de que $S(n)$ es verdadera para todo entero n que sea igual o mayor que el entero de partida i.

Para algunos de los ejemplos que vamos a desarrollar, nos basaremos en alguno de los siguientes teoremas:

- Si $A \leq B$, entonces $A + C \leq B + C$.
- Si $A \leq B$ y $B \leq C$, entonces $A \leq C$.

Demostrar, para todo $n \geq 3$:

$$2n + 1 \leq 2^n$$

Demostrar, para todo $n \geq 4$:

$$n^2 \leq 2^n$$

Demostrar, para todo $n \geq 1$:

$$\sum_1^n i = \frac{n(n+1)}{2}$$

Supón una sucesión de números $a_1, a_2, \ldots$ que cumplen las siguientes reglas:

1. $a_1 = 1$
2. $a_{n+1} = 2a_n + 1$ para toda $n \geq 1$.

Demostrar, para todo $n \geq 1$:

$$a_n = 2^n - 1$$