Ciencias computacionales

Propedeutico: Teoría de Autómatas y Lenguajes Formales
Autómatas

Contents

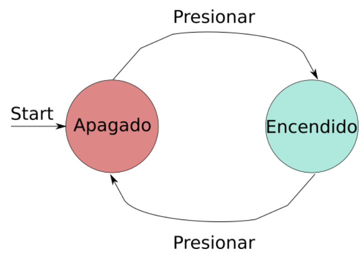

Figure 1: Un autómata finito que modela un switch de encendido y apagado.

1 ¿Por qué estudiar la teoría de autómatas?

La teoría de autómatas es el estudio de dispositivos o máquinas de cómputo abstractas. Turing en los 30's estudió una máquina abstracta con las capacidades de las computadoras actuales (en lo que podían calcular). La meta de Turing era describir la frontera entre lo que una máquina podía hacer y lo que no, su conclusión aplica no solo a las máquinas de Turing, sino a las máquinas actuales.

En los 40's y 50's se estudiaron los Autómatas Finitos para modelar la función cerebral (útiles para otros propósitos). A finales de los 50's, N. Chomsky inicia el estudio formal de las gramáticas. En 1969 S. Cook extiende el trabajo de Turing para estudiar lo que se podía y no calcular (*compute*), y lo que se podía resolver eficientemente o no. La clase de problemas que pueden ser resueltos en un principio pero que en la práctica requerirían mucho tiempo de cómputo son llamados "intratable" o NP-duros.

Todos estos desarrollos tienen repercusiones muy importantes en lo que es la computación actualmente. Algunos conceptos, como los autómatas finitos y cierto tipo de gramáticas formales, son utilizados para el diseño y construcción de tipos importantes de programa. Existen numerosos ejemplos: Analizadores léxicos en compiladores típicos, software para verificación de sistemas de todo tipo, software para diseño y verificación de circuitos digitales entre otros.

Un modelo simple de computación es el autómata finito o máquina de estado finito. Se introduce el concepto de autómata finito en la siguiente subsección.

1.1 Introducción al autómata finito

El autómata finito (AF) es un modelo útil para muchos tipos de hardware y software. El AF realiza una serie de cómputos en forma automática y produce una salida. El componente principal de un AF son estados, los cuales permiten recordar una porción relevante de la historia del sistema. El número de estados de un AF es finito, por lo que require un número limitado de recursos.

Un ejemplo de un AF es el switch de encendido o apagado. El AF recuerda si se encuentra en un estado de encendido o apagado y permite a un usuario presionar un botón cuyo efecto varía dependiendo del estado actual del AF. Es decir, si el estado actual es apagado, al ser presionado el botón, el autómata pasará a el estado de encendido, si el estado es encendido el autómata pasará a el estado de apagado. La Figura 1 muestra la representación gráfica de este simple AF. Los estados son representados por círculos, en el ejemplo se muestran dos estados; Apagado y Encendido. Los arcos entre los estados se describen como entradas que representan influencias externas que actúan sobre el autómata. En el ejemplo los dos arcos se etiquetan como Presionar que representa la acción de presionar el botón. Existe siempre un estado inicial del AF, en este ejemplo el estado Apagado es el estado inicial del autómata.

Los AF pueden tener un estado final o de aceptación. Se considera como buena o aceptable una entrada que permite llegar al estado final de un AF. En la Figura 2 se muestra un analizador léxico de un lenguaje de programación. El trabajo que realiza el autómata es el reconocimiento de una palabra clave *then*. Para lograr esto, se requieren cinco estados. Cada estado representa una posición diferente de la palabra *then* que ha sido alcanzada hasta entonces. La primera posición es la cadena vacía (e. g. nada

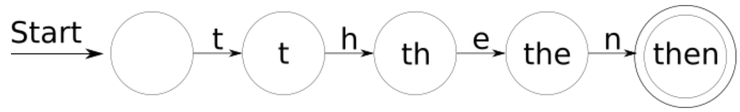

Figure 2: Un autómata finito que modela el reconocimiento de la palabra *then*.

ha sido escrito). Cada nuevo carácter leído permite mover el estado del AF. Si la cadena de caracteres leída por el autómata es *then*, el AF llegará a su estado final.

1.2 Representaciones estructurales

Existen dos nociones importantes para el estudio de autómatas y sus aplicaciones: Las gramáticas y las expresiones regulares. Las gramáticas son modelos útiles cuando se diceña software que procesa datos con una estructura recursiva. El ejemplo más conocido de una gramática es el *parser*, el componente de un compilador que maneja con las características recursivas de un lenguaje de programación e.g. la regla gramatical $E \rightarrow E+E$ indica que una expresión puede ser formada tomando cualesquiera dos expresiones y conectarlas mediante el signo $+$. Las expresiones regulares denotan una estructura de datos, es especial de texto. El estilo de las expresiones regulares varía respecto al estilo de las gramáticas. Un ejemplo de expresión regular es: $[A-Z][a-z]*[\ \ ][A-Z][A-Z]$ la cual representa palabras que inician con mayúscula seguidas de un espacio y dos letras mayúsculas (e.g., Ithaca NY).

1.3 El autómata y su complejidad

Los autómatas son esenciales para el estudio de los límites de la computación. Permiten responder a las preguntas:

- ¿Qué puede hacer una computadora? "Decidability" y los problemas que pueden ser resueltos por una computadora se conocen como "decidable" (decidibles)
- ¿Qué puede una computadora hacer eficientemente? "Intractability" y los problemas que pueden ser resueltos por una computadora en "tiempo polinomial" ("tractable" problems)

2 Introducción a las pruebas formales

La prueba de programas es esencial para saber si dichos programas tienen un comportamiento correcto. Sin embargo, las pruebas que se realizan a los programas mediante multiples ejecuciones no provee demasiada información. En programas complejos e.g. un función recursiva, no siempre es posible probar todas las combinaciones de entrada. Comúnmente para realizar prueba de programas complejos se realiza una hipótesis inductiva y se razona formalmente si dicha hipótesis es consistente con la recursión. Las pruebas formales en los autámatas son generalmente de tipo deductivo e inductivo. Las pruebas deductivas consisten en una secuencia de pasos justificados, mientras que las inductivas consisten en realizar pruebas recursivas de una sentencia parametrizada que "se use a si misma" pero con valores "más bajos" del parámetro.

2.1 Pruebas deductivas

Una prueba deductiva es una secuencia de enunciados cuya verdad nos lleva de un enunciado inicial (hipótesis) a una conclusión. Son del tipo "Si H Entonces C". La hipótesis puede ser verdadera o falsa dependiendo de los valores de sus parámetros. Además la hipótesis puede consistir en varias declaraciones independientes conectadas por un AND lógico (cada declaración se considera una hipótesis o una declaración dada). Por ejemplo la siguiente prueba:

Teorema 1 *Si $x \geq 4$, entonces $2^x \geq x^2$*

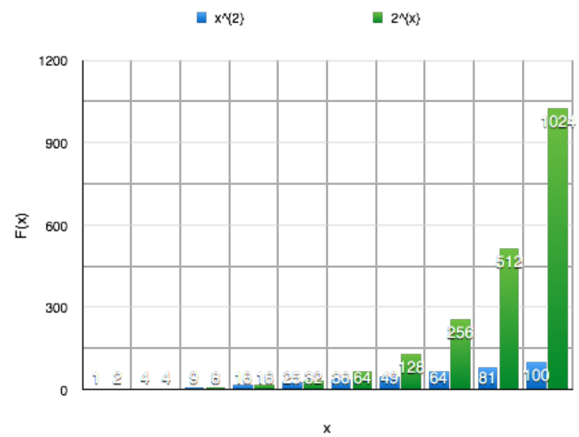

Figure 3: Comparación de las funciones 2^x y x^2. En verde es 2^x y azul es x^2.

	Enunciado	Justificación
1.	$x = a^2 + b^2 + c^2 + d^2$	Dado
2.	$a \geq 1; b \geq 1; c \geq 1; d \geq 1$	Dado
3.	$a^2 \geq 1; b^2 \geq 1; c^2 \geq 1; d^2 \geq 1$	(2) y propiedades de la aritmética
4.	$x \geq 4$	(1), (3) y propiedades de la aritmética
5.	$2^x \geq x^2$	(4) y Ejemplo 1

Table 1: Prueba formal del Teorema 2

La hipótesis del enunciado $x \geq 4$. El parámetro x puede ser cierto o falso. Su valor de verdad depende de x. Por ejemplo, si x es seis, entonces la hipótesis es cierta, por el contrario si x es dos, es falsa. De la misma forma, la conclusión $2^x \geq x^2$ utiliza el parámetro x, que es verdadero para ciertos valores y falso para otros. De manera intuitiva sabemos que la conclusión es verdadera para valores acotados de x. Cuando x es mayor o igual a cuatro, el lado derecho de la conclusión se duplica cada que x se incrementa en uno. Sin embargo el lado derecho crece a una razón de $(\frac{x+1}{x})^2$. En la Figura 3 se muestra gráficamente la comparación entre las funciones 2^x y x^2. Se puede observar que a partir de x es igual a 4, $2^x \geq x^2$.

Teorema 2 *Si x es la suma de los cuadrados de 4 números enteros positivos, entonces $2^x \geq x^2$.*

Intuitivamente, si la hipótesis es verdadera para x, entonces debe ser al menos 4 (ver Ejemplo 1). Si el teorema previo se sostiene, se puede declarar que su conclusión es también verdadera para x. Formalmente, el razonamiento se expresa como una secuencia de pasos. Cada paso puede ser una hipótesis de un teorema probado, parte de una hipótesis, o una declaración que sigue a una o más declaraciones previas. La forma de proceder se le conoce como *modus ponendo ponens*, en forma resumida se dice que si P y Q son dos proposiciones entonces si P implica Q; y P es verdad; entonces Q también es verdad. Se indicar formalmente como: $\frac{P \rightarrow Q, P}{\therefore Q}$. La prueba formal del enunciado principal se muestra en la Tabla 2.1.

2.2 La reducción de las definiciones

En muchos teoremas, incluyendo lo de teoría de autómatas, los términos utilizados en las declaraciones pueden tener implicaciones que son menos obvias. Si no se tiene muy claro cómo empezar una prueba, se pueden cambian los elementos de la hipótesis por sus definiciones. Se muestra un ejemplo cuya prueba resulta fácil una vez que se expresan los enunciados en términos elementales. Utiliza las dos siguientes definiciones:

- Un conjunto S es finito si existe un entero n tal que S tenga exactamente n elementos. El número de elementos de un conjunto se denota $\mid S \mid$. Si S no es finito se le llama infinito. Un conjunto infinito es un conjunto que contiene más que cualquier número entero de elementos.

	Original	Transformado
1.	S es finito	Existe un entero n tal que $\mid S \mid = n$
2.	U es infinito	No existe un entero p tal que$\mid U \mid = p$
3.	T es complemento de S	$S \cup T = U$ y $S \cap T = \emptyset$

Table 2: Transformando los enunciados dados en el Teorema 3

- Si S y T son ambos subconjuntos de algún conjunto U, entonces T es el complemento de S (con respecto a U) if $S \cup T = U$ y $S \cap T = \emptyset$. Cada elemento de U es exactamente uno de S y T, es decir T consiste exactamente de aquellos elemento de U que no están en S.

Teorema 3 *Sea S un subconjunto finito de algún conjunto infinito U. Sea T el complemento de S con respecto a U. Entonces T es infinito.*

La tabla 2.2 muestra la reducción de las definiciones transformadas a enunciados. Para probar el Teorema 3 se requiere utilizar otra técnica formal de prueba llamada "prueba por contradicción". La receta para utilizar este método es la siguiente:

- Se asume que la conclusión es falsa.
- Se utiliza ésta suposición, junto con partes de la hipótesis, para probar lo opuesto para alguna de las declaraciones dadas por la hipótesis.
- Se demuestra que es imposible que todas las partes de la hipótesis sean verdaderas y que la conclusión sea falsa al mismo tiempo.
- La única posibilidad es que la conclusión sea verdadera siempre que la hipótesis sea verdadera, es decir, el teorema es verdadero.

Se procede a demostrar el Teorema 3 usando el método por contradicción: Sabemos que $S \cup T = U$, que $\mid S \mid = n$ para un entero finito y que no existe un entero finito p tal que $\mid U \mid = p$. Suponemos que T es finito (lo contrario a lo queremos probar) e.g. $\mid T \mid = m$ para un entero finito. Entonces $\mid U \mid = \mid S \mid + \mid T \mid = n + m$, lo cual contradice que no existe un entero finito p tal que $\mid U \mid = p$. Por lo tanto, llegamos a una contradicción y se demuestra el Teorema 3 $\blacksquare$.

2.3 Otras formas de teoremas

El tipo de prueba "Si H Entonces C" es la forma más común en matemáticas. Sin embargo, existen otro tipo de pruebas que se pueden probar en forma de teoremas. En esta sección se examinan las formas más comunes de enunciados y que se utiliza usualmente para demostrar los teoremas.

En primer lugar se encuentran los enunciados de teoremas que lucen diferente a la forma "Si H Entonces C", pero que en realidad dicen lo mismo: Si la hipótesis H es verdadera para un cierto valor, entonces la conclusión H es cierta. Algunos ejemplos: H implica C, H únicamente si C, C si H, cuando H se sostiene, C se sigue. El Teorema 1 puede escribirse de las siguientes formas: $x \geq 4$ implica $2^x \geq x^2$, $x \geq 4$ únicamente si $2^x \geq x^2$, $2^x \geq x^2$ si $x \geq 4$.

Muchos teoremas usan los cuantificadores "para todos" $\forall$ y "existe un" $\exists$. El orden de aparición determina cuál va primero. Esto ess n lógica, e.g. sabemos que todos los hombres son mortales y que Socrates en un hombre, entonces Socrates es mortal. Con el uso de cuantificadores se puede escribir el anterior enunciado: $\forall x \text{Hombre}(x) \rightarrow \text{Mortal}(x)$ y Hombre(Socrates) por *modus ponendo ponens* Mortal(Socrates).

Existen también declaraciones del tipo "Si y sólo si H Entonces C". Dichas declaraciones se pueden escribir como H es equivalente a C. De forma más precisa, la declaración es un bicondicional.

Algunas veces es útil romper declaraciones "si y sólo si" en una sucesión de equivalencias:

- A si y sólo si B

- Probar primero "A si y sólo si C"
- Probar "C si y sólo si B"
- Cada declaración "si y sólo si" debe probarse en ambas direcciones

Por ejemplo, sea $\lfloor x \rfloor$, el piso de un número real x, es el mayor entero igual a o menor que x y $\lceil x \rceil$, el techo de un número real x, es al menos un entero igual o mayor que x.

Teorema 4 *Sea x un número real. Entonces $\lfloor x \rfloor = \lceil x \rceil$ si y sólo si x es un entero.*

Primero se asume que $\lfloor x \rfloor = \lceil x \rceil$ y tratamos de probar x es un entero. Por definición $\lfloor x \rfloor \leq x$ y $\lceil x \rceil \geq x$. Pero por suposición, $\lfloor x \rfloor = \lceil x \rceil$. Luego entonces podemos sustituir el piso por el techo en la primera desigualdad $\lceil x \rceil \leq x$. Dado que ambas desigualdades se sostienen, es decir $\lceil x \rceil \leq x, \lceil x \rceil \geq x$ se puede concluir que por propiedades de aritmética de desigualdades $\lceil x \rceil = x$. Ya que $\lceil x \rceil$ es un entero, x debe ser también un entero en este caso $\blacksquare$.

2.4 Teoremas que parecen no ser sentencias If-Then

Algunas veces los teoremas parecen no tener una hipótesis. Un ejemplo es la siguiente expresión trigonométrica.

Teorema 5

$$sen^2(\theta) + cos^2(\theta) = 1$$

En realidad, este enunciado tiene una hipótesis, dicha hipótesis consiste en todos los enunciados que se necesitan conocer para interpretar el enunciado principal. Particularmente para este ejemplo, la hipótesis oculta es que θ es un ángulo, luego entonces las funciones de seno y coseno tienen el significado habitual con ángulos. Con estas definiciones y el Teorema de Pitágoras, se puede probar el teorema 5.

3 Formas adicionales de prueba

Existen formas adicionales de pruebas, particularmente en esta sección se exploran las pruebas de conjuntos, el contra positivo, la prueba por contradicción y los contraejemplos.

3.1 Pruebas de equivalencias en conjuntos

En teoría de autómatas es frecuente probar algún teorema que declare que los conjuntos construidos de dos maneras distintas son los mismos conjuntos e.g. conjuntos de cadenas de caracteres, lenguajes. Si E y F son dos expresiones que representan conjuntos, el enunciado $E = F$ significa que los dos conjuntos represantan lo mismo. De forma más precisa, cada elemento de en el conjunto representado por E se encuentra en el conjunto F, y cada elemento del conjunto representadopor F se encuentra en E.

Por ejemplo, la ley conmutativa de unión: $R \cup S = S \cup R$. En este caso, E es la expresión $R \cup S$ y F es la expresión $S \cup R$. La ley conmutativa de unión indica que $E = F$. La expresión $E = F$ puede ser escrita como un bicondicional.

Teorema 6 *Ley distributiva de la unión sobre la intersección de conjuntos indica que:*

$$R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$$

Las dos expresiones del Teorema 6 a resolver son:

- $E = R \cup (S \cap T)$
- $F = (R \cup S) \cap (R \cup T)$

	Enunciado	Justificación
1.	x está en $R \cup (S \cap T)$	Dado
2.	x está en R o x está en $S \cap T$	(1) y la definición de unión
3.	x está en R o x está en S y T ambas	(2) y la definición de intersección
4.	x está en $R \cup S$	(3) y la definición de unión
5.	x está en $R \cup T$	(3) y la definición de unión
6.	x está en $(R \cup S) \cap (R \cup T)$	(4),(5) y la definición de intersección

Table 3: Transformando los enunciados dados en el Teorema 6

3.2 El contra positivo

Las declaraciones "Si-entonces", tienen una forma equivalente que puede ser útil en algunas circunstancias: "Si no C entonces no H". Una declaración y su contra-positivo son ambos verdaderos o falsos. Por ejemplo el Teorema 1, el contra positivo es "Si no $2^x \geq x^2$ entonces no $x \geq 4$". En términos coloquiales, sabiendo que "no $a \geq b$" es lo mismo que $a < b$, luego entonces el contra positivo del Teorema 1 es "Si $2^x < x^2$ entonces $x < 4$".

3.3 Prueba por contradicción

Una forma de demostrar un enunciado en la forma "Si H entonces C" es probar el siguiente resultado " H y no C implican falso". Para demostrar primero se siguen los siguentes pasos:

- Se quiere demostrar que una afirmación P es verdadera.
- Se asume que P es falsa.
- Se muestran las consecuencias del hecho de que P es falsa.
- Se llega a un absurdo o imposibilidad.
- Como la afirmación P es verdadera o falsa, y ya se demostró que no puede ser falsa ya que esto conlleva a incongruencias matemáticas, se prueba así que P debe ser verdadera.

Para ilustrar la prueba por contradicción, se tiene el siguiente teorema:

Teorema 7 $\sqrt{2}$ *es irracional*

Se comienza suponiendo que $\sqrt{2}$ es racional, es decir puede ser expresado como una fracción:

$$\sqrt{2} = \frac{p}{q} \tag{1}$$

donde p y q no tienen factores comunes. Elevando al cuadrado los dos términos se obtiene:

$$2 = \frac{p^2}{q^2} \tag{2}$$

lo cual implica

$$p^2 = 2q^2 \tag{3}$$

Entonces p^2 es par, esto solo puede ser cierto si p es par. Sin embargo, p^2 es divisible por 4. Luego entonces q^2 y q deben ser pares. De esta forma tanto p y q son pares, lo cual es una contradicción de la hipótesis que p y q no tienen factores comunes. De tal forma que $\sqrt{2}$ no es racional y por lo tanto es irracional $\blacksquare$.

	Cuantificador	Explicacion
1.	$\exists$	Existe al menos un elemento
2.	$\forall$	Para todos los elementos
3.	$\nexists$	No existe un elemento
4.	$\exists!$	Existe exactamente un elemento
5.	$\not\forall$	Para ningún elemento

Table 4: Transformando los enunciados dados en el Teorema 6

3.4 Contraejemplos

Normalmente los teoremas son enunciados acerca de un número infinito de casos, en esos casos suele utilizarse los cuantificadores para expresar casos generales. La tabla muestra los cuantificadores normalmente utilizados en la expresión de teoremas.

Usualmente se agrega | para indicar tal que e.g $\forall x\ | x \geq 2$ eso quiere decir que para todas las x tales que x es mayor igual que dos. Usualmente es más sencillo probar que un enunciado no es un teorema que probar que sí lo es. Por ejemplo el siguiente enunciado.

Teorema 8 *Todos los primos son impares.*

El enunciado 8 se puede refutar de la forma siguiente: El entero 2 es primo, pero es par.

4 Pruebas Inductivas

Las pruebas inductivas son esenciales con objetos definidos recursivamente. Las más conocidas son pruebas inductivas con números enteros. En teoría de autómatas también se necesitan pruebas inductivas sobre conceptos definidos recursivamente, como árboles o expresiones de varios tipos, como expresiones regulares. Estas últimas se conocen como inducciones estructurales.

Dada una declaración $S(n)$, de un entero n a probar:

- La base, mostramos $S(i)$ para un entero en particular i. Normalmente $i = 0$ o $i = 1$ (hay ejemplos donde queremos empezar con algún valor más alto de i, quizás porque S es falso para algunos valores enteros pequeños).
- El paso inductivo, donde asumimos, donde i es el entero base, y mostramos que "Si $S(n)$ entonces $S(n+1)$"

Para ejemplificar el uso de de la inducción, se tiene el siguiente teorema:

Teorema 9 *La suma de los enteros consecutivos es:*

$$\forall n \geq 0; \sum_{i=1}^{n} i = \frac{(n)(n+1)}{2}$$

Para demostrar el Teorema 9 se dividen en dos partes la prueba. El caso base y el general. El caso base es $n = 0$, la sumatoria es cero ya que el limite superior de la sumatoria es menor que el limite inferior, la suma ocurre sobre ningún termino y por lo tanto es cero. El paso inductivo supone que el Teorema 9 se cumple y se quiere probar que

$$\forall n \geq 0; \sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2} \tag{4}$$

Dado que es una sumatoria, la ecuación 4 se puede escribir como:

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (i+1) \tag{5}$$

La suposición es que $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ de modo que la ecuación 5 se escribe como:

$$\sum_{i=1}^{n+1} i = \frac{n(n+1)}{2} + (n+1) \tag{6}$$

Realizando la suma de los dos terminos de la ecuación se tiene

$$\sum_{i=1}^{n+1} i = \frac{n(n+1) + 2(n+1)}{2} \tag{7}$$

Existen factores comunes por lo cual la ecuación 7 se puede escribir como:

$$\frac{(n+1) + (n+2)}{2} \tag{8}$$

La ecuación 8 es la expresión que se quería obtener ■.

Se presenta a continuación otro ejemplo:

Teorema 10 *Si* $x \geq 4$ *entonces* $2^x \geq x^2$

Primero se demuestra el caso base: Si $x = 4$ las dos expresiones son 16 y se cumple. Para el paso de inducción es necesario demostrar la expresión para $x + 1$, esto es $2^{x+1} \geq (x + 1)^2$. Sabemos que $2^{x+1} = 2\text{x}2^x$, para el paso de inducción suponemos que $2^x \geq x^2$ y probamos para el siguiente entero. Por lo que $2^{x+1} \geq 2x^2$ y queremos probar entonces que $2x^2 \geq (x + 1)^2$. Entonces se tiene que $2x^2 \geq (x + 1)^2 \Rightarrow x^2 \geq 2x + 1$, que al dividir entre x nos da: $x \geq 2 + \frac{1}{x}$. Ya que $x \geq 4$ sabemos que $\frac{1}{x} \leq \frac{1}{4}$, lo cual es cierto y se demuestra el Teorema 10 ■.

5 Inducciones estructurales

Existen varios problemas que se pueden representar por estructuras (en lugar de solo números) y que se utilizan en teoría de autómatas (e.g., árboles o gramáticas). Para hacer una inducción sobre estas representaciones se hace básicamente lo mismo: Probar el enunciado $S(X)$ respecto a una familia de objetos X (i.e., enteros, árboles) en dos partes:

1. *Base*: Probar directamente para uno o varios valores pequeños de X.
2. *Paso de Inducción*: Suponer $S(Y)$ para Y "más pequeña que" X; probar $S(X)$ a partir de lo que se supuso

Para ilustrar el uso de la inducción en representaciones de estructura se muestra el siguente ejemplo: Un árbol binario con n hojas tiene $2n - 1$ nodos.

- Formalmente, $S(T)$: Si T es un árbol binario con n hojas, entonces T tiene $2n - 1$ nodos.
- Inducción respecto al tamaño = número de nodos de T.

Base: Si T tiene 1 hoja, se trata de un árbol de un nodo. $1 = 2 \times 1 - 1$, está bien.
Inducción: Suponer $S(U)$ para árboles con menos nodos que T. En particular, suponer para los subárboles de T:

- T debe consistir de una raíz más dos subárboles U y V.
- Si U y V tienen u y v hojas, respectivamente, y T tiene t hojas, entonces $u + v = t$.
- Por la hipótesis de inducción, U y V tienen $2u - 1$ y $2v - 1$ nodos, respectivamente.
- Entonces T tiene $1 + (2u - 1) + (2v - 1)$ nodos.
  - $= 2(u + v) - 1$.
  - $= 2t - 1$, probando con el paso de inducción.

6 Conceptos centrales de la teoría de autómatas

En esta sección se presentan ideas y conceptos centrales para la teoría de autómatas. En primera instancia se presentan los alfabetos, cadenas y gráficos.

6.1 Alfabetos

Un alfabeto es un conjunto finito no vacío de símbolos. Por convención se utiliza el símbolo de sumatoria $\sum$. Por ejemplo el alfabeto binario es representado por $\sum = \{0, 1\}$ y el alfabeto del abecedario en minúsculas $\sum = \{a, b, c, d, \cdots, z\}$.

6.2 Cadenas

Una cadena o palabra es una secuencia finita de símbolos elegidos de un alfabeto. E.g. 01101 es una cadena del alfabeto binario. La cadena vacía una cadena con zero ocurrencias de símbolos. Se denota como ϵ, dicha cadena es una cadena que puede elegirse a partir de cualquier alfabeto. Todas las cadenas de un alfabeto Σ de longitud k se denotan como Σ^k. E.g., $\Sigma^0 = \{\epsilon\}$, si $\Sigma = \{0, 1\}$, entonces, $\Sigma^1 = \{0, 1\}, \Sigma^2 = \{00, 01, 10, 11\}$, etc. El conjunto de todas las cadenas de un alfabeto Σ se denota como Σ^*. $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$. Sin la cadena vacía: Σ^+. Por lo que: $\Sigma^* = \Sigma^0 \cup \Sigma^+$.

Un concepto importante dentro de las cadenas es la concatenación de cadenas. La concatenación de x y $y = xy$. Por lo mismo $\epsilon x = x\epsilon = x$. El lenguaje es un conjunto de cadenas elegidas de algún alfabeto. El lenguaje puede ser infinito, pero existe algún conjunto finito de símbolos de los cuales se componen todas sus cadenas e.g. el conjunto de todas las cadenas binarias que consisten de algún número de 0's seguidos por un número igual de 1's; esto es:
$\{\epsilon, 01, 0011, 000111, \ldots\}$.

Si σ es un alfabeto y L es un lenguaje sobre σ se tiene que decidir si una cadena $w \in \sigma^*$ también $w \in L$. Por ejemplo, si tenemos un autómata que acepta números primos en forma de cadenas binarias, lo podríamos usar para ver si una cadena de un nuevo número es primo o no. Este es el típico problema de los *parsers*, se tiene un autómata que acepta programas escritos en "C" (por ejemplo) y tiene que decidir, si el programa cumple con la sintaxis válida de "C".

6.3 Gráficas

Los autómatas se pueden representar de forma gráfica:

- Un grafo con un número finito de nodos, llamados *estados*.
- Los arcos se etiquetan con uno o más símbolos de algún alfabeto.
- Un estado es designado como el *estado de comienzo* ó *estado inicial*.
- Algunos estados son *estados finales* o *estados de aceptación*.

7 Ejercicios

Ejercicio 1 $2 + 6 + 10 + \ldots + (4n - 2) = 2n^2$

Solución: Se quiere probar que $\sum_{i=1}^{n+1}(4i - 2) = 2(n + 1)^2$

- Base: $\sum_{i=1}^{1}(4i - 2) = 4(1) - 2 = 2 = 2(1)^2$

- Paso inductivo:

$$\sum_{i=1}^{n+1}(4i-2) = \sum_{i=1}^{n}(4i-2) + (4(i+1)-2) \tag{9}$$
$$= 2n^2 + (4(n+1)-2), \text{ por paso inductivo} \tag{10}$$
$$= 2n^2 + 4n + 2, \text{ factorizando se tiene} \tag{11}$$
$$= 2(n+1)^2, \text{ que es lo que se quería demostrar } \blacksquare \tag{12}$$

Ejercicio 2 $1 + 5 + 9 + \ldots + (4n-3) = n(2n-1)$

Ejercicio 3 $1 + 3 + 6 + \ldots + n(n+1)/2 = \frac{n(n+1)(n+2)}{6}$

Solución: Se quiere probar que $\sum_{i=1}^{n+1} i(i+1)/2 = \frac{(n+1)(n+2)(n+3)}{6}$

- Base: $\sum_{i=1}^{1} i(i+1)/2 = 1(1+1)/2 = 1(2)/2 = 1 = 1(2(1)-1) = 1$
- Paso inductivo:

$$\sum_{i=1}^{n+1} i(i+1)/2 = \sum_{i=1}^{n} i(i+1)/2 + [(n+1)((n+1)+1)/2] \tag{13}$$
$$= \frac{n(n+1)(n+2)}{6} + \frac{(n+1)(n+2)}{2}, \text{ por paso inductivo} \tag{14}$$
$$= \frac{(n+1)(n+2)(n+3)}{6}, \text{ juntando factores } \blacksquare \tag{15}$$
$$\tag{16}$$

Ejercicio 4 $5 + 10 + 15 + \ldots + 5n = \frac{5n(n+1)}{2}$

Ejercicio 5 $1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}$

Solución: Se quiere probar que $\sum_{i=1}^{n+1} i^2 = \frac{(n+1)(n+2)(2n+3)}{6}$

- Base: $\sum_{i=1}^{1} i^2 = 1^2 = 1 = (1)^2$
- Paso inductivo:

$$\sum_{i=1}^{n+1} i^2 = \sum_{i=1}^{n} i^2 + (n+1)^2 \tag{17}$$
$$= \frac{n(n+1)(2n+1)}{6} + (n+1)^2, \text{ por paso inductivo} \tag{18}$$
$$= (n+1)[\frac{n(2n+1}{6} + (n+1), \text{ factorizando} \tag{19}$$
$$= \frac{(n+1)(2n^2+7n+6)}{6}, \text{ agrupando} \tag{20}$$
$$= \frac{(n+1)(n+2)(2n+3)}{6}, \text{ factorizando } \blacksquare \tag{21}$$
$$\tag{22}$$

Ejercicio 6 $1^3 + 2^3 + \ldots + n^3 = \frac{n^2(n+1)^2}{4}$

Ejercicio 7 $n^2 > n + 1$ *para* $n > 2$

Ejercicio 8 $2 + 4 + 6 + ... + 2n = n(n + 1)$

Ejercicio 9 $1^2 + 3^2 + ... + (2n - 1)^2 = \frac{n(2n-1)(2n+1)}{3}$

Ejercicio 10 *Probar que* $2^n < n!$, *para* $n \geq 4$