



Auxiliar 1: Inducción

Miércoles 14 de Septiembre, 2016

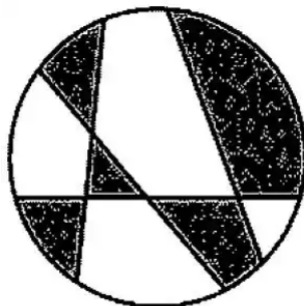
P1. Demuestre que para todo natural n se cumple que $7^n 2^n$ es un múltiplo de 5.

Solución:

Caso base: $k = 1 \rightarrow 7^1 - 2^1 = 7 - 2 = 5$ el cual es claramente múltiplo de 5. Hipótesis inductiva : $7^n - 2^n = 5k$, $k \in \mathbb{N}$

$$7^{n+1} - 2^{n+1} = 7(5k - 2^n) - 2^{n+1} = 7 * 5k - 2^n(7 - 2) = 7 * 5k - 5 * 2^n = 5(7k - 2^n) = 5q, q \in \mathbb{N}$$

P2. Sea $n \in \mathbb{N}$. Demuestre que al trazar n rectas en una circunferencia tal que no pasan más de dos rectas por el mismo punto y además no existen rectas paralelas entre sí, entonces cada región formada por la intersección de rectas puede colorearse con dos colores de forma que las regiones adyacentes son de distinto color. Se da el ejemplo para $n = 4$.



P3. Recuerde que el dominio constructible de las expresiones aritméticas $\mathcal{E}_{\mathbb{N}}$ se define por:

- Si k es natural, entonces k es una expresión.
- Si E_1 y E_2 son expresiones, entonces $E_1 + E_2$ es una expresión.
- Si E_1 y E_2 son expresiones, entonces $E_1 * E_2$ es una expresión.
- Si E es una expresión, entonces (E) es una expresión.

a) Defina inductivamente en la construcción de $\mathcal{E}_{\mathbb{N}}$ el operador $\#_{op} : \mathcal{E}_{\mathbb{N}} \rightarrow \mathbb{N}$ que cuenta la cantidad de operadores aritméticos (+ y *) de una expresión.

Solución:

Sea k natural, entonces k es una expresión, luego la cantidad de operadores de k es $\#_{op}(k) = 0$.

Hipótesis inductiva: E_1 tiene $\#_{op}(E_1)$ operaciones y E_2 con $\#_{op}(E_2)$ operaciones.

Veamos que $E_1 + E_2$ es una expresión y $\#_{op}(E_1 + E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1$ que es la cantidad de operadores por hipótesis y el nuevo operador entre las expresiones.

Análogamente $E_1 * E_2$ es una expresión y $\#_{op}(E_1 * E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1$ que es la cantidad de operadores por hipótesis y el nuevo operador entre las expresiones.

Finalmente $\#_{op}((E_1)) = \#_{op}(E_1)$ por lo que se suma a los operadores que contiene E_1 .

- b) Defina inductivamente en la construcción de $\mathcal{E}_{\mathbb{N}}$ el operador $\#_{num} : \mathcal{E}_{\mathbb{N}} \rightarrow \mathbb{N}$ que cuenta la cantidad de naturales que aparecen en una expresión.

Solución:

Sea k natural, entonces k es una expresión, luego la cantidad de naturales que aparecen en la expresión es uno, k . Luego $\#_{num}(k) = 1$.

Definamos por medio de inducción fuerte a E . Sea E formado por dos expresiones y un operador $E_1 + E_2$ por hipótesis inductiva esto es $\#_{num}(E_1) + \#_{num}(E_2)$ que es la cantidad de naturales que aparecen en cada expresión, y como el operador no cambia esta cantidad entonces la suma de naturales de ambas expresiones entrega el valor total de naturales $\#_{num}(E)$. Análogamente para $E_1 * E_2$ por hipótesis $\#_{num}(E_1) + \#_{num}(E_2)$ es el valor total de naturales en E . Por último, sea E expresión con $\#_{num}(E)$ naturales, entonces (E) posee la misma cantidad de naturales.

- c) Demuestre por inducción en la construcción de $\mathcal{E}_{\mathbb{N}}$ que para toda expresión $E \in \mathcal{E}_{\mathbb{N}}$ se cumple:

$$\#_{op}(E) = \#_{num}(E) - 1$$

Solución:

Caso base: Sea k natural, entonces k es expresión, sigue que $\#_{op}(k) = 0$ y $\#_{num}(k) = 1$ por la definición anterior, luego se tiene que $\#_{op}(k) = \#_{num}(k) - 1$. Caso inductivo: Sea E_1 y E_2 expresiones tales que $\#_{op}(E_1) = \#_{num}(E_1) - 1$ y $\#_{op}(E_2) = \#_{num}(E_2) - 1$ luego:

- $\#_{op}(E_1 + E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2) - 1 + 1 = \#_{num}(E_1) + \#_{num}(E_2) - 1 = \#_{num}(E_1 + E_2) - 1$
- $\#_{op}(E_1 * E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2) - 1 + 1 = \#_{num}(E_1) + \#_{num}(E_2) - 1 = \#_{num}(E_1 * E_2) - 1$

$$\#_{op}((E_1)) = \#_{op}(E_1) = \#_{num}(E_1) - 1 = \#_{num}((E_1)) - 1$$

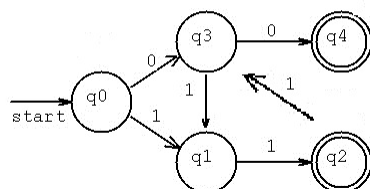
- P4.** Un *automata finito determinista* (afd) es una tupla $A = (Q, \Sigma, \delta, s, F)$ donde Q es un conjunto de estados, Σ es un alfabeto finito de *símbolos* (e.g a, b, c), $\delta : Q \times \Sigma \rightarrow Q$ es una *función de transición*, $s \in Q$ se denomina *estado inicial* y finalmente $F \subseteq Q$ es el conjunto de *estados finales*. Dado un afd A y un entero $n \in \mathbb{N}$, se define $L_n(A)$ como el conjunto de palabras $w = a_1 \dots a_n$, formada por símbolos $a_1, \dots, a_n \in \Sigma$, *aceptadas* por el autómata, esto es, tales que existe una secuencia de estados $q_0, \dots, q_n \in Q$ que cumplen que (1) $q_0 = s$, (2) $\delta(q_i, a_{i+1}) = q_{i+1}$ para cada $i = 0, \dots, n-1$, y (3) $q_n \in F$. En este problema, dados un número $n \in \mathbb{N}$ y un afd A , queremos calcular la cantidad $|L_n(A)|$ de palabras de tamaño n que acepta A . Para esto construiremos la siguiente construcción. Para cada estado $q \in Q$ y $k \leq n$, sea $N_{k,q}$ el número de palabras $w = a_1 \dots a_k$ de tamaño k , tales que existe una secuencia de estados $p_0, \dots, p_k \in Q$, con (1) $p_0 = s$, (2) $\delta(p_i, a_{i+1}) = p_{i+1}$ para cada $i = 0, \dots, k-1$, y (3) $p_k = q$ (es decir, la cantidad de palabras de tamaño k que llevan del estado s al estado q en el autómata). Suponga que $N_{0,s} = 1$ y $N_{0,q} = 0$ para cualquier $q \in Q$, $q \neq s$.

1. Muestre usando inducción que para $0 \leq k \leq n-1$ y $q \in Q$, se tiene que

$$N_{k+1,q} = \sum_{(p,\sigma) \in Q \times \Sigma : \delta(p,\sigma)=q} N_{k,p}$$

Solución:

Primero un pequeño ejemplo de autómatas: En el ejemplo anterior se tienen 5 estados: q_0, q_1, q_2, q_3, q_4 . Donde q_2 y q_4 son estados finales. La función de transición se muestra explícita con las flechas, por ejemplo, si estoy en el estado inicial q_0 y recibo el carácter 1 entonces voy al estado q_1 , por otro lado, si estoy en q_0 y recibo un string de caracteres 01 entonces primero removeré 0 de la lista e iré a q_3 , luego removeré 1 de



la lista e iré de q_3 a q_1 . Para que una cadena de palabras sea aceptadas por el autómata esta cadena tiene que terminar en uno de los estados finales, por ejemplo 00 es aceptada por el autómata pues al final de la cadena termino en el estado q_4 , pero 01 y 111 no son aceptados, pues en el primero termino en q_1 mientras que para el segundo, pese a pasar por el estado final q_2 , al recibir el último 1 paso a q_3 que no es un estado final.

Los autómatas son muy usados en computación pues permiten definir gran parte de los lenguajes posibles, por otro lado tienen aplicaciones en procesamiento de texto, compiladores, etc.

Ahora demostraremos que para $0 \leq k \leq n-1$ y $q \in Q$, se tiene que $N_{k+1,q} = \sum_{(p,\sigma) \in Qx\Sigma: \delta(p,\sigma)=q} N_{k,p}$. Por inducción en k , el caso base, $k=1$ se tiene del enunciado, pues sabemos que $N_{0,q} = 0 \forall q \neq s$, por lo que el unico caso que suma un término distinto de cero en la sumatoria del lado derecho es el caso $N_{0,s}$, luego, a partir de s deberían sumarse una unidad por cada palabra de largo 1 que permite llegar a q , esto es, por cada q se suma 1 si existe un símbolo que lleve de s a q que es equivalente a la cantidad de transiciones de $\delta(s, \sigma) = q$.

Para el paso inductivo, veremos cuantas palabras de tamaño $n-1$ nos permiten llevar a un estado para el cual al agregarle un símbolo podemos llegar finalmente hasta q , veremos que la cantidad de combinaciones de formas de llegar por medio de este método equivale a la fórmula.

Si la propiedad se cumple para $n-1$ entonces, como $N_{n,q}$ corresponde al número de palabras de tamaño n que llevan desde el estado s al estado q y $N_{n-1,p}$ corresponde a la cantidad de palabras de tamaño $n-1$ que llevan desde s a p , entonces para cada símbolo σ tal que $\delta(p, \sigma) = q$ debe existir una palabra de tamaño 1 que lleva desde p a q por medio del carácter σ . Luego, si tomamos la cantidad de formas que permiten llegar a p por medio de una palabra de tamaño $n-1$, esto corresponde a la hipótesis inductiva $N_{n-1,p}$, luego debemos notar que para cada combinación $(p, \sigma) \in Qx\Sigma$ tal que $\delta(p, \sigma) = q$ se suma una palabra de largo 1 distinta, y por cada palabra de largo 1 debemos tomar las combinaciones de palabras que permitieron llegar a p , es decir $N_{n-1,p}$, esto último es equivalente a las sumas de combinaciones de palabras posibles entregadas por la ecuación $\sum_{(p,\sigma) \in Qx\Sigma: \delta(p,\sigma)=q} N_{n-1,p}$ y notando que esto es equivalente (por la descripción anterior) a la cantidad de palabras de tamaño n que permiten llegar desde s a q se tiene demostrado lo pedido.

2. Concluya, utilizando la recurrencia de la parte anterior, que la cantidad $|L(A)_n|$ es igual a $\sum_{f \in F} N_{n,f}$.

Solución:

Claramente $\sum_{f \in F} N_{n,f}$ representa la suma de de todas las palabras de tamaño n que van desde s a cada f , vale decir, la cantidad total de palabras que permiten ir desde s a un estado final, esto último significa que la palabra es aceptada por el autómata (pues el último carácter debe llegar exactamente a f para que el autómata acepte la palabra), esto es equivalente a la cantidad de palabras de tamaño n que acepta el autómata, es decir $|L_n(A)|$ con lo que se muestra la equivalencia.

P5. Defina inductivamente el conjunto $\mathcal{P}_{\mathbb{N}}$ de todas las expresiones aritméticas en notación prefija. Las expresiones contienen números naturales y los símbolos $+$ y $*$.

Solución:

Sea k natural $\rightarrow k$ esta en $\mathcal{P}_{\mathbb{N}}$.

Sean E_1 y E_2 expresiones aritméticas en notación prefija, es decir $E_1, E_2 \in \mathcal{P}_{\mathbb{N}}$

- $+E_1 E_2 \in \mathcal{P}_{\mathbb{N}}$
- $*E_1 E_2 \in \mathcal{P}_{\mathbb{N}}$

No es necesario definir el caso para el paréntesis, pues de hecho las expresiones aritméticas en notación prefija o polaca no requieren de paréntesis para definir la precedencia de operaciones lo que lo hace "más fácil" de procesar para la máquina pues no deben buscarse paréntesis en la expresión u otros ordenes de precedencia antes de comenzar a operar.

- Explique paso a paso cómo usando su definición anterior se crea la expresión

$$* * + 4 5 * 3 + 7 1 9$$

De la misma manera explique por qué la siguiente no es una expresión bien formada (o sea trate de construirla paso a paso y deténgase cuando la construcción falle)

$$* + + 5 + + 6 7 8 * 9 1$$

Solución:

La expresión $* * + 4 5 * 3 + 7 1 9$ puede formarse comenzando desde el par de valores más cercanos al último operador, en este caso $+7 1$ es una expresión en $\mathcal{P}_{\mathbb{N}}$ pues 7 y 1 son naturales y por tanto (por caso base) están en $\mathcal{P}_{\mathbb{N}}$. Luego se pasa al siguiente operador, en este caso $*3+7 1$, como 3 es natural y $+71$ esta en notación prefija, entonces $*3 +7 1$ está en $\mathcal{P}_{\mathbb{N}}$ como fue definido anteriormente. El siguiente operador en prioridad usa los dos números consecutivos al operador, en este caso $+4 5$ (y no el resultado de la operación anterior), como ambos números son naturales entonces la expresión está en $\mathcal{P}_{\mathbb{N}}$. Los valores consecutivos al siguiente operador son $+4 5$ y $*3+7 1$ pues ambas son expresiones en notación prefija y son los valores consecutivos al operador, luego $* +4 5 *3+7 1$. Finalmente el resultado de esta última expresión se multiplica con el 9 al final de la expresión original, pues $* +4 5 *3+7$ es expresión y 9 es natural y por tanto expresión en notación prefija por caso base. Finalmente concluimos que puede construirse esta expresión por medio de la definición inductiva anterior.

Por otro lado, veamos si $* + + 5 + + 6 7 8 * 9 1$ puede construirse de la misma forma. Partimos desde $*9 1$ lo cual es expresión, y luego operamos el $+6 7$ y el siguiente más debería juntar los resultados del $+6 7$ y el 8, luego el siguiente $+$ debería sumar los valores de 5 y $+ + 6 7 8$, luego el siguiente $+$ debería sumar $+ 5 + + 6 7 8$ y el resultado obtenido al inicio de $*9 1$ todo esto sigue siendo expresión en notación prefija, pero luego se tiene el último operador $*$ que solo tiene una expresión $+ + 5 + + 6 7 8 * 9 1$, luego como el $*$ no esta definido como una expresión, entonces esta expresión no está en $\mathcal{P}_{\mathbb{N}}$.

- De manera similar al P3) defina los operadores $\#_{op}$ y $\#_{num}$ para $\mathcal{P}_{\mathbb{N}}$ y demuestre que se cumple $\#_{op}(E) = \#_{num}(E) - 1$. Use este hecho para justificar por qué la última expresión del ejercicio anterior no estaba bien formada.

Solución:

- Sea k natural, entonces k es expresión en $\mathcal{P}_{\mathbb{N}}$, $\#_{op}(k) = 0$.
- Sean $E_1, E_2 \in \mathcal{P}_{\mathbb{N}}$ luego $\#_{op}(+E_1 E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1$
- Sean $E_1, E_2 \in \mathcal{P}_{\mathbb{N}}$ luego $\#_{op}(*E_1 E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1$
- Sea k natural, entonces k es expresión en $\mathcal{P}_{\mathbb{N}}$, $\#_{num}(k) = 1$.

- Sean $E_1, E_2 \in \mathcal{P}_N$ luego $\#_{num}(+E_1E_2) = \#_{num}(E_1) + \#_{num}(E_2)$
- Sean $E_1, E_2 \in \mathcal{P}_N$ luego $\#_{num}(*E_1E_2) = \#_{num}(E_1) + \#_{num}(E_2)$

Veamos que $\#_{op}(E) = \#_{num}(E) - 1$, sea k natural, entonces k es expresión y luego $\#_{op}(k) = 0 = \#_{num}(k) - 1 = 1 - 1 = 0$. Caso inductivo: Sean E_1, E_2 expresiones en notación prefija tal que $\#_{op}(E_1) = \#_{num}(E_1) - 1$ y $\#_{op}(E_2) = \#_{num}(E_2) - 1$.

- $\#_{op}(+E_1E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2) - 1 + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2)$ que es equivalente a $\#_{num}(+E_1E_2) - 1 = \#_{num}(E_1) + \#_{num}(E_2) - 1$ con lo que se tiene lo buscado.
- $\#_{op}(*E_1E_2) = \#_{op}(E_1) + \#_{op}(E_2) + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2) - 1 + 1 = \#_{num}(E_1) - 1 + \#_{num}(E_2)$ que es equivalente a $\#_{num}(*E_1E_2) - 1 = \#_{num}(E_1) + \#_{num}(E_2) - 1$ con lo que se tiene lo buscado.

Ahora que se ha demostrado esta propiedad podemos ver que $\#_{op}(* + + 5 + + 6 7 8 * 9 1) = 6$ y que $\#_{num}(* + + 5 + + 6 7 8 * 9 1) = 6$, en caso de que hubieramos aceptado esta expresión como válida en la construcción realizada no se cumpliría la propiedad anteriormente demostrada.

- Una de las virtudes de las expresiones en notación prefija es que pueden ser evaluadas sin la necesidad de utilizar paréntesis para establecer precedencia de operadores. Defina inductivamente la función $eval : \mathcal{P}_N \rightarrow N$ que evalúa una expresión aritmética prefija cualquiera. Por ejemplo:

$$eval(* + 4 2 + 3 2) = 30$$

Discuta la factibilidad de hacer una definición similar para evaluar expresiones en notación infija como las del P3).

Solución:

Caso base: Sea k natural, luego k es expresión en notación prefija y $eval(k) = k$. Caso inductivo: Sea E_1 y E_2 expresiones en notación prefija tales que $eval(E_1)$ es el resultado de la evaluación de todas las operaciones de E_1 y $eval(E_2)$ es el resultado de la evaluación de todas las operaciones de E_2 .

- $eval(*E_1E_2) = eval(E_1)eval(E_2)$ (multiplicación común, * es simplemente un carácter al cual le estamos dando un significado por medio de la definición inductiva)
- $eval(+E_1E_2) = eval(E_1) + eval(E_2)$ (este último signo + es la suma que conocemos, en $+E_1E_2$ lo estamos usando como un carácter cuyo significado definimos)

Como ya se mencionó anteriormente, la evaluación en la notación infija requiere conocer la precedencia u orden en que se debe operar buscando en primer lugar los paréntesis y luego operando según las reglas que definimos para los operadores, esto requiere mucho mayor esfuerzo computacional y muchos más casos por definir que los que se dieron para definir la evaluación en notación prefija.

Dudas sobre la pauta a andruso11@gmail.com (Andrés Olivares)