Autómatas y Lenguajes formales 2019-2

Ejercicio Semanal 2

Sandra del Mar Soto Corderi
Edgar Quiroz Castañeda

Fecha de entrega: 7 de febrero del 2019

1. Una cadena es palíndorma si es de la forma ww^R.

   a) Define el conjunto P de cadenas palíndromas de forma recursiva.
   - ϵ es cadena palíndroma.
   - Si $a \in \Sigma$ y $p \in P$, entonces $apa \in P$
   - Estos son todos las cadenas palíndromas

   b) Enuncia el principio de Inducción generado por la definición del inciso anterior.
   Sea Pr alguna propiedad. Para demostrar que Pr se cumple para todas las cadenas paíndromas, hay que hacer lo siguiente
   - Demostrar directamente que $Pr(\epsilon)$ es verdad.
   - Asumir como hipótesis que si alguna cadena palíndroma es de la forma, apa, entonces $Pr(p)$ es verdad.
   - Demostrar que $P(apa)$ es verdad usando la hipótesis del inciso anterior.

   c) Demuestra mediante inducción estrucutral que todas las cadenas palíndromas **de ese tipo** tienen un número par de símbolos.
   - Caso base, con $w = \epsilon$.
     Entonces $|w| = 0 = 2 \cdot 0$.
     Por lo que $w = \epsilon$ tiene una cantidad par de símbolos.
   - La hipótesis es que si awa es cadena palíndorma, entonces w tiene una cantidad par de símbolos, esto es que $|w| = 2k$ para alguna $k \in \mathbb{N}$
   - Notemos que $a \in \Sigma$ es un símbolo, para también puede considerarse una cadena de un sólo símbolo, por lo que las propiedades válidas para cadenas también son válidas para a.
     En particular, tenemos que $|vw| = |v| + |w|$ (demostrado en la tarea anterior).
     Por lo que
     $$|awa| = |a| + |aw| = |a| + |w| + |a| = 1 + |w| + 1 = |w| + 2|$$
     Luego, por la hipótesis de inducción, $|w| = 2k$.
     Por lo que $|awa| = 2k + 2 = 2(k + 1)$, que es un número par.
     Por lo que awa tiene una cantidad par de símbolos.

2. Dada la siguiente definición de árbol binario de elementos del conjunto A
   - Un árbol vacío *void* es un árbol binario.
   - Si T_1 y T_2 son árboles binarios, y c es un elemento de A, entonces $(tree\ T_1, c, T_2)$ es un árbol binario.
   - Estos son todos los árboles binarios.

   Responde a lo siguiente

   a) Define recursivamente la función *aplana* que toma un árbol binario y devuelve una lista de sus elementos con un recorrido *in order*.

   Usando la definición de lista de objetos que está en las notas, podemos definir *aplana* como:

   aplana : Árbol binario c $\to \mathcal{L}$(c)
   $aplana\ (void) = nil$

$aplana\ (tree\ T_1, c, T_2) = cons(cons(aplana(T_1), c), aplana(T_2))$

En haskell sería algo así:
aplana :: Arbol a → [a]
aplana void = []
aplana (tree a t1 t2) = aplana t1 ++ [a] ++ aplana t2

b) Demuestre que para cualquier árbol binario T se cumple que el número de nodos de T es igual a la longitud de la lista $aplana(T)$.

Demostramos realizando inducción sobre T y usando la definición de lista de objetos que está en las notas:
Sea n el número de nodos de T

- Caso base: $T = void$
  $n = 0$ por la definición de árbol binario
  $aplana(T) = nil$ por la definición de *aplana*
  $len(nil) = 0$ por la definición de longitud en listas
  Entonces como los nodos y la longitud de la lista son 0, se cumple el caso base.
- Hipótesis de inducción: Suponemos verdadero que $n = len(aplana(T))$ para los árboles T_1 y T_2
  Sea n_1 el número de nodos de T_1 y n_2 el número de nodos de T_2:
- Paso inductivo, tenemos que demostrar que $len(aplana(tree\ T_1, c, T_2)) = n_1 + n_2 + 1$
  $aplana\ (tree\ T_1, c, T_2) = cons(cons(aplana(T_1), c), aplana(T_2))$ por definición de *aplana*
  $len(cons(cons(aplana(T_1), c), aplana(T_2))) = len(cons(aplana(T_1), c)) + len(aplana(T_2))$ por definición de longitud
  $len(cons(aplana(T_1), c)) = len(aplana(T_1)) + 1$ por definición de longitud
  De ahí, $len(aplana(tree\ T_1, c, T_2)) = len(aplana(T_1)) + len(aplana(T_2)) + 1$
  Usando la hipótesis de inducción, $len(aplana(tree\ T_1, c, T_2)) = n_1 + n_2 + 1$

Por lo tanto, se cumple que el número de nodos de T es igual a la longitud de la lista $aplana(T)$.

3. Define recursivamente la función $sub?(u, v)$ que nos dice si u es subcadena de v.
Tenemos que $sub? : \Sigma^{*2} \to \{0, 1\}$ dada por

- $sub?(\epsilon, v) = 1$
- $sub?(u, \epsilon) = 0$
- $sub?(\alpha u, \beta v) = ((\alpha == \beta) \wedge sub?(u, v)) \vee (sub?(\alpha u, v))$

4. Considera los lenguajes $L = \{0, 1, 10, 11, 100, 101, 111\}$ y $M = \{00, 01, 001, 11, 0100, 101, 0111\}$.
Da los lenguajes resultantes de cada inciso

a) $L \cup M = \{0, 1, 10, 11, 100, 101, 111, 00, 01, 001, 0100, 0111\}$

b) $M \cap L^R = \{01, 001, 11, 101\}$

c) $L - (M \cup L^R) = \{10, 100\}$

d) $(ML)^R = \{000, 100, 0100, 1100, 00100, 10100, 11100, 010, 110, 0110, 1110, 00110, 10110, 11110, 01100, 001100, 101100, 111100, 011, 111, 0111, 1111, 00111, 10111, 11111, 00010, 10010, 010010, 110010, 0010010, 1010010, 1110010, 0101, 1101, 01101, 11101, 001101, 101101, 111101, 01110, 011110, 111110, 0011110, 1011110, 1111110\}$

e) $L^R M^R = \{000, 100, 0100, 1100, 00100, 10100, 11100, 010, 110, 0110, 1110, 00110, 10110, 11110, 01100, 001100, 101100, 111100, 011, 111, 0111, 1111, 00111, 10111, 11111, 00010, 10010, 010010, 110010, 0010010, 1010010, 1110010, 0101, 1101, 01101, 11101, 001101, 101101, 111101, 01110, 011110, 111110, 0011110, 1011110, 1111110\}$