Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2213 — Lógica para Ciencias de la Computación — 1' 2022

Ayudantía 1

Pregunta 1: Calentamiento

Construya máquinas de Turing que decidan los siguientes lenguajes (como subconjuntos de $\{a, b\}^*$):

1. $\{w \mid w$ contiene al menos una $a\}$
2. $\varnothing$ (el conjunto vacío)
3. $\{\epsilon\}$ (la palabra vacía)

---

1. Definimos nuestro estados como q_0: No he visto una "a" y q_1: Ya vi una "a".

   $Q = \{q_0, q_1\}$
   $A = \{a, b\}$
   $F = \{q_1\}$

   <u>Función Delta:</u>
   $\delta(q_0, a) = (q_1, a, \rightarrow)$
   $\delta(q_0, b) = (q_0, b, \rightarrow)$

2. Aquí queremos que nuestra máquina **no acepte ningún input**, por lo que se puede solucionar definiendo nuestro conjunto de estados finales como vacío:
   $F = \{\}$

3. En este caso queremos que solo acepte la palabra vacía como input, asi que definimos nuestros estamos como q_0: no he visto ningún caracter, q_1: Vi algun caracter.

   $Q = \{q_0, q_1\}$
   $A = \{a, b\}$
   $F = \{q_0\}$

   <u>Función Delta:</u>
   $\delta(q_0, a) = (q_1, a, \rightarrow)$
   $\delta(q_0, b) = (q_1, b, \rightarrow)$

Construya máquinas de Turing que computen las siguientes funciones:

1. $f(w) = \begin{cases} 0 & \text{si } \|w\| \text{ es par} \\ \\ 1 & \text{si } \|w\| \text{ es impar} \end{cases}$

2. $f(n) = n + 1$, con n un número natural codificado en binario

---

1. La idea es revisar el input e ir borrando los caracteres leídos mientras se mantiene la cuenta de si he leido hasta ahora una cantidad par o impar de caracteres. Al final, dependiendo del estado en el que me encuetro, escribo un 0 o un 1. Definimos nuestro estados como q_p: He leido una cantidad par de caracteres, q_i: He leido una cantidad impar de caracteres, q_f: Ya revisé todo el input.

   $Q = \{q_p, q_i, q_f\}$
   $A = \{a, b, 0, 1\}$
   $q_0 = q_p$
   $F = \{q_f\}$

   <u>Función Delta:</u>
   $\delta(q_p, *) = (q_i, B, \rightarrow)\ \forall * \in A$
   $\delta(q_i, *) = (q_p, B, \rightarrow)\ \forall * \in A$

   $\delta(q_p, B) = (q_f, 0, \rightarrow)$
   $\delta(q_i, B) = (q_f, 1, \rightarrow)$

   DEMOSTRACIÓN

   Dada una máquina de Turing M creada con las reglas anteriores, cualquier ejecución con input w de largo mayor a 1 será de la forma:
   $(q_p, 0, t_0) \xrightarrow{*} (q_i, 1, t_1) \xrightarrow{*} ... \xrightarrow{*} (q_p, j, t_j) \xrightarrow{*} (q_i, j+1, t_{j+1}) \xrightarrow{*} ... \xrightarrow{*} (q_x, n, t_n) \xrightarrow{\text{B}} (q_f, n+1, t_{n+1})$

   Donde

   $t_{j+1}(x) = t_j(x) \qquad \forall x \neq j+1,\ j \in [1, n-1]$
   $t_{j+1}(j+1) = B \qquad \forall j \in [1, n-1]$

   , lo que quiere decir que en cada paso de la ejecución, se reemplaza la última casilla revisada por un blanco.Y además:

   $t_{n+1}(x) = t_n(x) \quad \forall x \neq n+1$
   $t_{n+1}(n+1) = \begin{cases} 0 & \text{si } x = p \\ \\ 1 & \text{si } x = i \end{cases}$

   , que significa que cuando llega al último paso, se reemplaza la útima casilla por un 0 o un 1 según corresponda al estado actual.

Luego, por inducción es posible demostrar que para un input de largo n el estado actual de la máquina de las primeras n configuraciones representa exactamente si la posición del cabezal es par o impar. Además, como el cabezal pasa por todo el input en los estados q_p o. q_i, se anota un blanco en cada espacio y solo queda el último dígito anotado, el cual corresponde justamente a 0 si nos encontrábamos en q_p o 1 si estábamos en q_i.

2. Lo que busca hacer la máquina que haremos es avanzar hasta el final del input (para lo que utilizaremos el estado q_0) y desde ahí ir sumando a la izquierda (en el estado q_c) hasta que lea un 0 o un B y por lo tanto no deba seguir almacenando un "carry" para la suma, pasando a q_f.

   $Q = \{q_0, q_c, q_f\}$
   $A = \{0, 1\}$
   $F = \{q_f\}$

   Función Delta:
   $\delta(q_0, *) = (q_0, *, \rightarrow)\ \forall * \in A$
   $\delta(q_B, *) = (q_c, B, \leftarrow)$

   $\delta(q_c, 0) = (q_f, 1, \rightarrow)$
   $\delta(q_c, 1) = (q_c, 0, \rightarrow)$

Pregunta 2: Propuesto

Sean n, m dos números en codificación binaria con $\|n\| = \|m\|$. Construya una máquina de Turing que reciba un input de la forma $n\$m$ y retorne en codificación binaria el número $n + m$.

Pregunta 3: Máquina de Turing con cinta para un lado

Definición: Una máquina de Turing con cinta para un solo lado es una tupla $M = (Q, \mathbf{A}, B, \vdash, q_0, F, \delta)$, donde

- Q es un conjunto de estados
- $\mathbf{A}$ es un alfabeto que no contiene ni a B ni a $\vdash$
- $q_0 \in Q$ es el estado inicial
- $F \subseteq Q$ son los estados finales
- δ es la función de transición, una función parcial definida como

  $$\delta : Q \times (\mathbf{A} \cup \{B, \vdash\}) \to Q \times (\mathbf{A} \cup \{B\}) \times \{\leftarrow, \to\}$$

  en donde además $\delta(q, \vdash)$ está siempre en $Q \times (\vdash) \times \{\to\}$, lo que quiere decir que cada vez que la máquina lee $\vdash$, tiene que volver a escribir $\vdash$ y además moverse a la derecha.

El lenguaje aceptado por una máquina de Turing con cinta para un solo lado $M = (Q, \mathbf{A}, B, \vdash, q_0, F, \delta)$, escrito como $L(M)$, corresponde a todas las palabras $w \in \mathbf{A}^*$ tal que M acepta a la palabra $w = a_1 ... a_n$ cuando comienza con la palabra $\vdash a_1 ... a_n$ escrita y el resto de las celdas vacías y con la cinta lectora apuntando al símbolo a_1.

Demuestre que para cada máquina de Turing M existe una máquina de Turing con cinta para un solo lado M' tal que $L(M) = L(M')$, y que para cada máquina de Turing con cinta para un solo lado M existe una máquina de Turing M' tal que $L(M) = L(M')$.

Proximamente...
