CC3102 - 2018/01
Soluciones Control 3

Responda cada pregunta en una hoja separada
Ponga su nombre en cada hoja de respuesta
Dispone de una hora y 45 minutos para resolver el control

1. Construya en detalle una máquina de Turing M (listando todas las transiciones) que acepte el lenguaje $L = \{0^{2^i} \mid i \geq 0\}$. Es decir, M debe aceptar a todos los strings de 0's cuyo largo es una potencia de 2. Explique claramente la idea de su construcción.

   Idea de Solución: La idea de esta solución era muy similar a la de la tarea. Dado un string de la forma $w = \mathtt{0}^N$, lo que hacemos es ir chequeando que N es divisible por 2, luego que es divisible por $4 = 2^2$ luego que es divisible por $8 = 2^3$ etc. Para esto, marcamos los símbolos de w uno por medio usando un nuevo símbolo `X`. Partimos marcando el primer `0` no marcado, y avanzando hacia la derecha, nos "saltamos" el siguiente `0` y marcamos el subsiguiente. Repetimos hasta que llegamos al final del string. Luego de la primera ejecución de esta rutina, la mitad de los `0`'s estarán marcados. Por ejemplo, para el string $\mathtt{0}^8 =$ `00000000`, después de ejecutar este procedimiento obtenemos el string

   `X0X0X0X0`

   Luego volvemos al inicio del string y repetimos. Con el mismo ejemplo anterior, después de la segunda repetición, tendremos el string:

   `XXX0XXX0`

   Si repetimos dos veces más el mismo procedimiento obtendremos primero `XXXXXXX0` y luego `XXXXXXXX`. Finalmente, si en cualquier momento de la ejecución, vemos un símbolo blanco justo después de haber marcado una `X`, terminamos el marcado y solo nos faltaría verificar que todos los símbolos `0`'s fueron efectivamente marcados. En el caso anterior, aceptaríamos porque efectivamente todos los `0` se marcaron. Note que si ejecutamos el proceso en un string de la forma $\mathtt{0}^N$ donde N no es una potencia de 2, entonces en algún momento marcaremos el último 0 del string (justo antes de un símbolo blanco) pero otros `0`'s quedarán sin marcar. Por ejemplo, suponga que ejecutamos este procedimiento sobre el string $\mathtt{0}^{10} =$ `0000000000`. En este caso la ejecución se verá así:

   `0000000000`

   `X0X0X0X0X0`

   `XXX0XXX0XX`

   Y por lo tanto rechazaremos el string. La construcción de la tabla de transiciones sigue la idea de la solución de arriba.

2. Definimos a las *máquinas de Turing sin arrepentimiento* (MTSA), como la clase de máquinas de Turing que sólo pueden mover su cabezal a la derecha (nunca a la izquierda). Encuentre un lenguaje recursivo que **no pueda** ser aceptado por una MTSA. Justifique claramente su respuesta.

   Solución: Dado que la MTSA sólo puede mover su cabezal a la derecha, la función de transición se ve de la forma $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\to\}$. Más aún, dado que siempre se debe mover, lo que sea que escriba en la cinta será invisible para la máquina, luego podemos pensar que la función de transiciónn es simplemente de la forma $\delta' : Q \times \Gamma \to Q$ y por lo tanto la MTSA funciona exactamente igual a un autómata finito, y luego están solo capacitadas para aceptar lenguajes regulares. Con esta discusión es fácil encontrar un lenguaje recursivo que no sea aceptado por una MTSA: por ejemplo el lenguaje $L = \{a^i b^i \mid i \geq 0\}$ es un lenguaje no regular, luego no puede ser aceptado por una MTSA, pero es libre de contexto y por lo tanto es recursivo.

3. (a) Considere una máquina de Turing generadora[1] M y suponga que $G(M)$ (el lenguaje generado por M) es infinito. Demuestre que siempre puede construir una nueva máquina generadora M' que genera un subconjunto de los strings de M, es decir que $G(M') \subseteq G(M)$, tal que $G(M')$ es también infinito y para cada par de strings $u, v \in G(M')$, si u es generado por M' antes que v, entonces el largo de u es menor o igual al largo de v.

   (b) Use la anterior propiedad para demostrar lo siguiente. Sea L un lenguaje recursivamente enumerable cualquiera y suponga que L es infinito. Demuestre que L siempre tiene un subconjunto $S \subseteq L$ tal que S es infinito y recursivo.

Solución:

(a) Considere la máquina M descrita en el enunciado. Podemos crear una nueva máquina M' que funciona de la siguiente forma. Inicialmente M' simula a M para generar el primer string que M genera, digamos w_1, y escribirlo en su cinta de output. Ahora M' simulará a M pero en una nueva cinta (no la de output), y generará todos los strings $w_2, w_3, w_4, \ldots$ hasta generar uno que tenga largo mayor a w_1. Note que, dado que M genera un lenguaje infinito, estamos completamente seguros que el algún momento M generará un string de largo mayor a $|w_1|$. Llamemos w_i a ese string. Una vez encontrado w_i, M lo escribirá en su cinta de output, y repetirá el proceso, generando en la otra cinta los strings $w_{i+1}, w_{i+2}, w_{i+3}, \ldots$ hasta generar un string, digamos w_j tal que $|w_j| > |w_i|$, y entonces escribirá w_j en su cinta de output, y continuará con el mismo proceso. Note que este proceso continuará para siempre y hará que M' genere strings en su cinta de output tales que cada string es también generado por M pero todos los strings cumplirán con estar ordenados por tamaño. Luego $G(M')$ satisface con lo pedido en el enunciado.

(b) Esta parte es una consecuencia directa de la anterior. Sea L un lenguaje recursivamente enumerable infinito. Sabemos entonces que existe una máquina de Turing generadora M tal que $L = G(M)$. Por la parte (a) sabemos que existe una M' tal que $G(M') \subseteq G(M)$ tal que $G(M')$ es infinito y los strings son generados en orden de largo por M'. Luego $G(M')$ es un lenguaje recursivo dado que es generado en orden por una máquina de Turing. Finalmente, si llamamos $S = G(M')$ tenemos que $S \subseteq G(M) = L$, y S es infinito y recursivo, lo que demuestra lo pedido.

[1] Recuerde que una máquina generadora tiene una cinta especial en donde escribe strings separados por #.