Modelos Avanzados de Computación

Entrega 1

María del Mar Ruiz Martín
Doble Grado de Ingeniería Informática y Matemáticas
Universidad de Granada - UGR
18001 Granada, Spain

Curso 2016/2017

Ejercicio 6

Describir de manera informal MTs con varias cintas que enumeren (produzcan como salida una lista que contenga todas sus palabras) los siguientes lenguajes (se supone que los números se escriben en binario):
(a) El conjunto de los cuadrados perfectos.
(b) El conjunto de todos los naturales primos.
(c) El conjunto de todos los números naturales n tales que la MT cuya descripción es la palabra w_n acepta la palabra w_n como entrada (w_n es la palabra sobre 0, 1 cuyo número asociado es n).

(a) El conjunto de los cuadrados perfectos.

Vamos a considerar como alfabeto de entrada A = {0,1}, y además B = {0,1,/,#}. La máquina de Turing contará con 3 cintas: en la primera habrá un contador que recorrerá todos los naturales, la segunda será una cinta auxiliar para hacer la multiplicación del número que está en la primera cinta consigo mismo, y en la tercera se van introduciendo los cuadrados perfectos, separando dichos números por el símbolo $/ \in B$.

La función δ la podríamos dividir en las siguientes subrutinas:

- Incremento: se encargará de incrementar en una unidad el número existente en la primera cinta.

- Cálculo del cuadrado: haciendo uso de la igualdad $(n-1)^2 + n + (n-1) = n^2$ se calcula el cuadrado del nuevo número de la siguiente forma: puesto que en la segunda cinta entontramos el cuadrado de n-1 le sumamos dicho número, a continuación se pasa a la subrutina de incremento, y se suma el número que entonces hay en la primera cinta. De esta forma en la segunda cinta aparece el cuadrado de n, esto es, un cuadrado perfecto.

- Copia del cuadrado: finalmente se cuenta con una subrutina que escribe al final de la última cinta el símbolo / y pasa a copiar el número de la segunda cinta en la última.

Es fácil comprobar que dicha máquina de Turing efectivamente enumera todos los cuadrados perfectos, puesto que recorre los naturales escribiendo su cuadrado en la última cinta. Además, no producirá en la salida ningún número que no lo sea.

(b) El conjunto de todos los naturales primos.

Para realizar esta máquina nos basaremos en la criba de Eratóstenes. Para ello de nuevo consideraremos A = {0,1} B ={0,1,/,#}, y una máquina con 3 cintas:

- En la primera cinta se recorrerán todos los números naturales mayores o iguales que 2.

- La segunda cinta será una cinta auxiliar para realizar divisiones.

- Finalmente en la última aparecerán todos los números primos separados por el símbolo /.

La función de transición vendrá dada por las siguientes rutinas:

- Incremento (igual que en el apartado anterior), da paso a la subrutina de división.
- División: se divide el número de la primera cinta entre el número sobre el que se encuentre la máquina de Turing de la última cinta, y se almacena el resto en la segunda cinta y se pasa a comprobar la primalidad del número de la primera cinta
- Comprobación de primalidad: si el resto almacenado en la segunda cinta es igual a 0 podemos concluir que el número en cuestión no es primo y pasa al estudio del siguiente y en la primera cinta el cabezal debe estar sobre el primer elemento de la lista. En caso contario, en la última cinta se pasa al siguiente número de la lista y se pasa a la división. Si durante este proceso se llega al símbolo /, esto es, se ha terminado de recorrer los números primos menores que n, se concluye que n es primo y se pasa a la rutina de copia.
- Copia: al final de la última cinta se escribe el símbolo / seguido del número de la primera cinta, se desplaza hasta el principio de la lista de primos y finalmente se pasa a la función incremento.

Inicialmente en la primera cinta aparecerá el número 2, puesto que si no se incluiría en primer lugar al 1 como primo, y después a todos los naturales.

(c) El conjunto de todos los números naturales n tales que la MT cuya descripción es la palabra w_n acepta la palabra w_n como entrada (w_n es la palabra sobre 0, 1 cuyo número asociado es n).

En este caso contaremos con la siguiente serie de cintas:

- Una cinta en la que se almacenará un natural, que denotaremos i.
- Cinta en la que se almacenará otro natural, j.
- Cinta para almacenar un tercer natural, k.
- Cinta para almacenar la suma i+j.
- Conjunto de cintas necesarias para ejecutar j pasos de la máquina de Turing que acepta el lenguaje universal, para comprobar si, en esos j pasos, la máquina de Turing con codificación i acepta a i.
- Cinta que almacena los naturales buscados.

El funcionamiento de la máquina será el siguiente: para cada k, se recorren las parejas (i,j) con i+j = k. i denotará la máquina de Turing con codificación i, y j será el número de pasos. Para cada pareja (i,j) que verifique lo anterior se comprueba que i no esté aún en la lista de palabras aceptadas. En caso de que no esté, se ejecutarán j pasos de la máquina de Turing que acepta el lenguaje universal para la máquina i junto con la cadena i. En caso de que se acepte la palabra, añadimos i a la lista. En caso contrario, pasamos a la siguiente pareja (i,j). Cuando se han recorrido todas las parejas (i,j), se incrementa k y se repite el proceso.

Las principales subrutinas con las que contaremos son las siguientes:

- Incremento de k.
- Recorrido de (i,j) verificando las condiciones explicadas.
- Suma de i + j
- Ejecución de j pasos de la MT del lenguaje universal para R(i,i).
- Búsqueda de la palabra i en la lista de palabras aceptadas.
- Copia de i en la última cinta si procede.

Ejercicio 7

Sean L_1 , . . . , L_k (k $\geq$ 2) un conjunto de lenguajes sobre el alfabeto A tales que:
(a) Para cada $i \neq j$, tenenos que $L_i \cap L_j = \emptyset$.
(b) $\cup_{i=1}^{k} L_i = A^*$.
(c) $\forall i \in \{1, ..., k\}$, el lenguaje L_i es r.e.
Demostrar que $\forall$i $\in$ {1, . . . , k}, el lenguaje L_i es recursivo.

Para probar este resultado en primer lugar usaremos el siguiente resultado de teoría:

Teorema: Si L y $\overline{L}$ son ambos recursivamente enumerables, entonces L es recursivo.

Queremos además probar otro resultado:

Resultado: Sean $L_1, L_2, ..., L_k$ lenguajes recursivamente enumerables. Entonces $\cup_{i=1}^{k} L_i$ es recursivamente enumerable.

Es fácil comprobar este hecho, puesto que si L_i es recursivamente enumerable $\forall i \in \{1, 2, ..., k\}$, entonces existen máquinas de Turing que aceptan cada uno de estos lenguajes. Sea M_i la máquina que acepta el lenguaje L_i, entonces contruímos una máquina de Turing no determinista, que presenta un estado inicial con k transiciones posibles, donde la transición i-ésima conduce a las transiciones correspondientes a la máquina de Turing M_i. Finalmente, puesto que existe una equivalencia entre máquinas de Turing y máquinas de Turing no deterministas, podemos encontrar una máquina de Turing que acepte el lenguaje $\cup_{i=1}^{k} L_i$, y la existencia de esta máquina es la que nos afirma que el lenguaje es recursivamente enumerable.

Finalmente queda hacer la siguiente apreciación: $\overline{L_i} = \cup_{j=1,\ j\neq i}^{k} L_j$. Esto es inmediato sin más que aplicar las condiciones (a) y (b) dadas como hipótesis en el enunciado. Aplicamos entonces el siguiente razonamiento:

Sea $i \in \{1, 2, ..., k\}$. L_i es recursivamente enumerable por hipótesis, y también lo es $\overline{L_i}$ en virtud de la apreciación anterior junto con el resultado demostrado. Aplicando el teorema concluímos que L_i es recursivo. Puesto que i era fijo pero arbitrario, concluímos que L_i es recursivo $\forall i \in \{1, 2, ..., k\}$

Ejercicio 8

Sea L r.e., pero no recursivo. Considérese el lenguaje

$$L_0 = \{0w|w \in L\} \cup \{1w|w \notin L\}$$

¿Puede asegurarse que L_0 o su complementario son recursivos, r.e. o no r.e.?

En primer lugar hacemos la siguiente apreciación: puesto que L es recursivamente enumerable, pero no recursivo, $\overline{L}$ no puede ser recursivamente enumerable, puesto que si no se llegaría una contradicción con el teorema del ejercicio anterior. Podemos reescribir la expresión de L_0 como:

$$L_0 = \{0w/\ w \in L\} \cup \{1w/\ w \in \overline{L}\}$$

De la expresión anterior es fácil ver cómo se construiría la máquina de Turing que aceptase este lenguaje:

- Si desde el estado inicial se lee un 0 se pasa a las transiciones correspondientes a la máquina asociada al lenguaje L.
- Si se lee un 1 se pasa a las transiciones correspondientes a las palabras de $\overline{L}$.

Sin embargo, para el segundo caso no es posible encontrar tales transiciones, puesto que si existiesen, tendríamos una serie de transiciones que leen $\overline{L}$, y tomando estas transiciones, y como estado inicial aquel estado al que llegamos tras leer un 1 en la máquina de L_0 tendríamos una máquina de Turing que aceptaría $\overline{L}$, lo cual es una contradicción.

Por tanto, como no existe una máquina de Turing que acepte el lenguaje L_0 podemos afirmar que no es recursivamente enumerable.

Análogamente, podemos ver $\overline{L_0}$ como:

$$\overline{L_0} = \{0w/\ w \in \overline{L}\} \cup \{1w/\ w \in L\}$$

Aplicando el mismo razonamiento que el usado para L_0, llegamos a la conclusión de que ninguno de los dos son recursivamente enumerables.