Tarea 4
Teoría de la Computación

Benjamín Ruvera

7 de octubre de 2020

Ejercicio 3.4

Los **Enumeradores (E)** son maquinas con doble cadena. Una de ellas es la impresora, donde únicamente se imprimiran elementos; la otra es la cadena de trabajo, que empieza vacia. Además los E ignoran cualquier entrada que reciban. Estos se encargan de generar lenguajes, lo que implica que imprime TODAS las cadenas que pertenecen a cierto lenguaje. Para definir el autómata debemos dar la *7-tupla* de una **Máquina de Turing (MT)**.

$$(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r) \tag{1}$$

Para empezar, sabemos que los **E** son autómatas con doble pila, por lo que debemos considerar la siguiente tupla

$$(Q, \Sigma, \Gamma_1, \Gamma_2, \delta, q_0, q_a, q_r) \tag{2}$$

donde estos elementos quedan definidos como

- Q**:** El conjunto (finito) de estados del autómata.
- Σ :] El alfabeto de entrad. Este podría ser vacio dado que la entrada es ignorada por el autómata.
- Γ_1**:** El alfabeto de la cinta de traajo que debe incluir el algabero del lenguaje a enumerar.
- Γ_2**:** El alfabeto de la impresora. ESte únciamente contiene el alfabeto del elnguaje a enumerar. De manera que $\Gamma_1 \subset \Gamma_2$.
- δ**:** La función de transición tal que

$$\delta : Q \times \Gamma_1 \mapsto Q \times \Gamma_1 \times \Gamma_2 \times L, R$$

- q_0**:** El estado inicial.
- q_a**:** El conjunto de los estados aceptores.
- q_r**:** El conjunto de los estados rechazados.

Lenguaje enumerado

De manera que un lenguaje enumerado es aquel que, para algún enumerador E (si este se deja corriendo indefinidamente) eventualmente imprimira todo el lenguaje. O dicho de otra forma; es aquel cuyos elementos son numerables[1]

[1] Es numerable si y solo si existe una biyección con los naturales $\mathbb{N}$

Ejercicio 3.6

Entonces, reproduciendo la demostración inicial, tenemos que si la Maquina de Turing M reconoce al lenguaje A, podemos contruir el enumerador E para A. Siendo $s_1, s_2, \ldots$ una lista de strings en Σ^*.

```
E := ignorar el input
   1. Repetir lo siguiente para i=1,2,3,...
   2.    Ejecutar M para s_i
   3.    Si es aceptado, imprimir s_1
```

Podemos ver que, por definicion en 2, el enumerador únicamente imprimira elementos que sean aceptados por M. Por lo que todo elemento que sea imprimido por E es aceptado por M

Ejercicio 3.8

A continuación esta el pseudocódigo del ejercicio y más adelante esta la implementación explicita en forma gráfica

```
Si marcamos con x esta pendiente    [x]
Si marcamos con y esta verificado   [y]


Leemos el elemento en la cabeza [Inicio]

        Si encotnramos el fin de la cadena
                Aceptamos la cadena
                [Fin]
        Si leemos 0
                Marcamos con x
                Hacemos una de la siguientes
                        Buscamos 0 y 1 y marcamos [y]
        Si leemos 1
                Buscamos dos ceros y marcamos [y]

        Regresamos el lector hasta x y marcamos [y]
        [Inicio]

[Fin]
```

Ejercicio 3.12

Vemos que la diferencia entre la **Maquina de Turing con Reset (MTR)** y la **Maquina de Turing (MT)** es que el primero sustituye el movimiento izquierda L con el reset $RESET$. Este queda definido por

$$\begin{aligned} \{q, x \ldots xqa\} &\rightarrow \{q_r, q_r x \ldots xb\} \\ \delta(q, a) &= (r, b, RESET) \end{aligned}$$

Para poder verificar la equivalencia entre la MTR y la MT debemos encontrar una forma de emular el movimiento `L` en función de los movimientos de MTR `RESET, R`

```
Tenemos que RESET
    {xx...xqa -> qxx...xb}

Sea U un caracter unico que no este en el abecedario original.
Sea x' el valor alterno a x para identificar

Empezamos con
  {xxx...xxqa}
    Usamos a -> U, RESET
      {xxx...xxqa -> qxxx...xxU}

    Si lees U estas al inicio [Fin]
    Si lees x avanza x->R
        Si lees x ve a [prod]
        Si lees U, estabas casi al inicio
               RESET y ve a [Fin]

    [prod]
    Si lees x' avanzas hasta encontrar x.  x'->R
      {qx'x'x...xxU -> x'qx'x...xxU -> x'x'qx...xxU}
    Si lees x, marcas y avanzas dos (x->x',R)->R
      {qxxx...xxU -> x'xqx...xxU}
        Si lees x vamos RESET y a [prod] x->RESET
        Si lees U ponemos b y vamos a [casi] U->b->RESEET
    Si lees

    [casi]
    Avanza quitando primas  x'->x,R
      {qx'x'...x'xb -> xqx'...x'xb -> xx...qx'xb -> xx...xqxb}
    Cuando ya no encuentres primas estas en L [Fin]

[Fin]
```

Ejercicio 3.13

Sea la **Maquina de Turing Degenerada (MTD)** como se define en el ejercicio, esta se diferencia de la **Maquina de Turing (MT)** porque no puede mover el cabezal a la izquierda.

La principal diferencia entra los autómatas con pila (PL) y las MT es la maner en que acceden a su memoria. Mientras que los primeros se basan en `FIFO`, la segunda tiene libertad para leer su memoria. La MTD no puede leer su mempria libremente, únicamente lo puede hacer asia adelante, asi que puede ser que sea equivalente a los AP, pero definitivamente no es equivalente a la MT.

Por otro lado, la diferencia principal entre los AP y los AF es que los primero no tiene memoria. Aunque parece que los MTD si tienen memoria, únicamente pueden escribir sobre el caracter en que esten viendo, por lo que cualquier informacion que tengan guardad en este se perdera en cuanto avancen. Además, como no pueden retroceder, cualquier cosa que traten de guardar en espacios anteriores queda inaccesible por el MTD en cuanto avance. Por lo que el MTD tampoco es similar a los AP.

Una *definición rapida* de los AF, es que son maquinas que leen caracteres, sin rettroceder, y cambian de estado. Como esto es lo únic que hace nuestro MTD, entonces el MTD es equivalente a un AF.