FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN. PARTE 1. COMPUTABILIDAD.  
Trabajo Práctico Nro 1 Año 2020. Licenciatura en Sistemas.  
  
Máquinas de Turing

Comentario: Tratar de hacer mínimamente los ejercicios del 1 al 5.Ejercicio 1. Responder breve y claramente los siguientes incisos:  
1. ¿En qué se diferencia un problema de búsqueda de un problema de decisión?

* En los problemas de búsqueda se debe encontrar y dar la solución, por ejemplo dado un grafo encontrar un camino del nodo A al nodo N.
* En los problemas de decisión dado un input se acepta o rechaza.

2. ¿Por qué en el caso de los problemas de decisión, podemos referirnos indistintamente a problemas y lenguajes?

* Un lenguaje es un conjunto de strings y un problema (en este contexto) es un conjunto de strings, así que son lo mismo

3. En la clase 1 se presentó el problema de satisfactibilidad de las fórmulas booleanas, en su forma de decisión: “Dada una fórmula φ, ¿existe una asignación A de valores de verdad que la hacen verdadera?” Enunciar el problema de búsqueda asociado.

Dada una fórmula booleana φ, por ejemplo: (x1 https://lh5.googleusercontent.com/hb8S6yYf0yNXt_o9HXf0XB4ZRCzIrI9HdP9ltu-GtXrLh4qL3dc221wV-j5aSTMXc-prwocwBZ73HmsdFDBTMk8WD7r6xCngnS2qovmJFfG_FOcuxC-zi1agYHRpFZr2egDjcz0Z x2 ) https://lh6.googleusercontent.com/ZQb_VZaHi4lZ1dmbrctq3l2mhUeGahvAPZ_jppiugVteDRqeWDgg2VaAMmULinXAvR6ZBmHba-ln2cSgI1SFlu8LuDXaq3WkH6BiWRBivXqiNVX_eUGuyqRTazFd-Y1_1vClphsP (x3 https://lh6.googleusercontent.com/ZQb_VZaHi4lZ1dmbrctq3l2mhUeGahvAPZ_jppiugVteDRqeWDgg2VaAMmULinXAvR6ZBmHba-ln2cSgI1SFlu8LuDXaq3WkH6BiWRBivXqiNVX_eUGuyqRTazFd-Y1_1vClphsP x4 ), ¿existe una asignación A de valores de verdad, que la satisface, es decir que la hace verdadera? Por ejemplo, A1 = (V, F, V, V) satisface φ, y A2 = (F, F, V, V) no.

4. Además de la visión de Máquina de Turing (o MT) que reconoce lenguajes (visión  
reconocedora), está la visión de MT que los genera (visión generadora). En el caso del  
problema del inciso anterior, ¿qué lenguaje generaría la MT de visión generadora que  
resuelve el problema?

5. ¿Qué postula la Tesis de Church-Turing?

La tesis de Church-Turing formula hipotéticamente la equivalencia entre los conceptos de función computable y máquina de Turing, que expresado en lenguaje corriente vendría a ser "todo algoritmo es equivalente a una máquina de Turing"

6. ¿Cuándo dos MT son equivalentes? ¿Y cuándo dos modelos de MT son equivalentes?

* Dos MT son equivalentes (computacionalmente) si reconocen el mismo lenguaje.
* Dos modelos de MT son equivalentes cuando para toda MT M1 de un modelo existe una MT M2 equivalente del otro, es decir que L(M1) = L(M2).

Ejercicio 2. Dado el alfabeto Ʃ = {0, 1}:  
1. Obtener el conjunto Ʃ\* y el lenguaje incluido en Ʃ\* con cadenas de a lo sumo 2 símbolos.  
2. Dado el lenguaje L = {0n1n | n ≥ 0}, calcular Ʃ\* ⋂ L, Ʃ\* ⋃ L y LC con respecto a Ʃ\*.

**1. Obtener el conjunto Ʃ\* y el lenguaje incluido en Ʃ\* con cadenas de a lo sumo 2 símbolos.**

* CONJUNTO SIGMA ESTRELLA son todos los strings que puedo formar con 0 y/o con 1
* CONJUNTO SIGMA ESTRELLA CON CADENAS DE A LO SUMO 2 CARACTERES {0,1,00,11,01,10}

**2. Dado el lenguaje L = {0**n **1**n **| n ≥ 0}, calcular Ʃ\* ⋂ L, Ʃ\* ⋃ L y L^C con respecto a Ʃ\*.**

* Ʃ\* ⋂ L = L
* Ʃ\* ⋃ L = Ʃ\*
* Complemento de L con respecto a Ʃ\*, seria Ʃ\* -  L

Ejercicio 3. En la clase 1 se construyó una MT no determinística (o MTN) para aceptar las cadenas de la forma han o hbn, con n ≥ 0. Construir una MT determinística (o MTD) equivalente.

Es importante diferenciar entre una máquina de Turing determinística, de una no determinística:

*En una MT no determinística, en lugar de una función de transición δ, M tiene una relación de transición Δ, es decir que para un mismo par (q, a), la máquina puede responder de más de una manera, por ejemplo: Δ(q, a) = {(q1, a1, L), (q2, a2, R), (q3, a3, S)}*

• *La relación de transición se define así: Δ: Q x Γ → P((Q ∪ {qA, qR}) x Γ x {L, R, S})*

• *Una MTN acepta si y sólo si al menos una computación acepta*

Resolución del ejercicio:

* Idea general: Tenemos que ver que el input comience con 'h'. Después, tiene que seguir con 'a' o 'b'. En caso de seguir con 'a', todos los caracteres deben ser 'a', hasta llegar a un blanco. Lo mismo en caso de que siga con una 'b' después de la 'h' inicial.
* Construcción de la MT M = (Q, Ʃ, Γ, δ, q0, qA, qR):
* Q = { qh, qa, qb, qab } qh es el estado inicial, en donde se busca una 'h'. qa es el estado en el que se encuentra una 'a', y se espera seguir encontrando 'a' o blanco (B). qb es el estado en el que se encuentra una 'b', y se espera seguir encontrando 'b' o blanco (B). qab es el estado en el que se encontró una 'h', y ahora se espera encontrar una 'a' o 'b'.
* Ʃ = {a, b, h}
* Γ = {a, b, h, B}
* q0 = qh

|  | **h** | **a** | **b** | **B** |
| --- | --- | --- | --- | --- |
| qh | qab, x, R |  |  |  |
| qab |  | qa, a, R | qb, b, R | qA, B, S |
| qa |  | qa, a, R |  | qA, B, S |
| qb |  |  | qb, b, R | qA, B, S |

* Función de transición:

MTM(Q,E,r,FT,qo,qA,qR)

Q={q1,q2,q3} (q0 NO poner)

E={h,a,b}

r={h,a,b,B}

q0 estado inicial , q1=determina con cual sigue   q2= busca una a q3= busca una b

h RECHAZO

ha ACEPTACIÓN

hb ACEPTACIÓN

hab RECHAZO

hba RECHAZO

hh RECHAZO

h                    a b         B

q0 (q1,h,R)     (qR,a,S)     (qR,b,S)   (qR,B,S)

q1       (qR,h,S)     (q2,a,R)         (q3,b,R)            (qR,B,S)

q2    (qR,h,S)     (q2,a,R)       (qR,b,S) (qA,B,S)

q3       (qR,h,S)     (qR,a,S) (q3,b,R)             (qA,B,S)

Ejercicio 4. Describir una MT de K cintas (plantear la idea general, opcionalmente construirla formalmente), que acepte de la manera más eficiente posible el lenguaje L = {anbncn | n ≥ 0}.

* Idea general: La máquina va a comenzar apuntando con el cabezal al extremo izquierdo de la cinta. El primer elemento debe ser una "a", el cual debe reemplazar con "x". Irá hacia la derecha en busca de un caracter "b". Al encontrarlo, lo reemplazará con "y". Luego irá nuevamente hacia la derecha, en busca de un caracter "c", al cual reemplazará con "z". En este momento,
* deberá volver hacia la izquierda, buscando una "x". Luego, irá hacia la derecha, buscando una 'a'. Si la encuentra, debe repetir el proceso. Si no encuentra una "a", entonces tampoco debe encontrar caracteres "b", ni caracteres "c", ya que tiene que haber la misma cantidad de "a", "b", y "c".
* Construcción de la máquina de Turing: M = (Q, Ʃ, Γ, δ2, q0, qA, qR), con:
* Q = { qa, qb, qH } qa: es el estado en el que se busca una 'a'. qb: es el estado en el que se busca una 'b'. qc: es el estado en el que se busca una 'c'. qH: es el estado en el que no hay más 'a', y por ende no debería haber mas letras. qL: es el estado en el que vuelvo a la izquiera después de haber encontrado una 'c', y busco una 'x'.
* Ʃ = {a, b, c}
* Γ = {a, b, c, x, y, z, B}
* q0 = qa

|  | **A** | **b** | **c** | **x** | **y** | **z** | **B** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| qa | qb, x, R |  |  |  |  |  | qA, B, S |
| qb | qb, a, R | qc, y, R |  |  | qH, y, R |  |  |
| qc |  | qc, y, R | qL, z, L |  |  |  |  |
| qH |  |  |  |  | qH, y, R | qH, z, R | qA, B, S |
| qL | qL, a, L | qL, b, L |  | qa, x, R | qL, y, L | qL, z, L |  |

* La función de transición δ es:

Las celdas en blanco representan los casos de rechazo (estado qR)

Ejercicio 5. Explicar (informal pero claramente) cómo simular una MT por otra que no tenga el movimiento S (es decir el no movimiento).

Bajo esta situación, entra la necesidad de hacer movimientos innecesarios del cabezal. De nuevo serán necesarios estados adicionales. Como no existe el movimiento S siempre se hará un movimiento adicional para la derecha o para la izquierda. Estos nuevos estados deberán realizar un movimiento en sentido contrario, para volver a la posición necesaria.

Ejercicio 6. En la clase 1 se construyó una MT con dos cintas para aceptar el lenguaje L = {w | w ∈ {a, b}\* y w es un palíndromo o “capicúa”}. Construir una MT con una cinta para aceptar el mismo lenguaje (se puede considerar la idea de solución propuesta en la clase).

* Sea L = {w | w ∈ {a, b}\* y w es un palíndromo o “capicúa”}
* w es un palíndromo o “capicúa” sii w = wR, siendo wR la cadena inversa de w
* Queremos construir una MT M que acepte L
* Idea general: Lo ideal sería que la máquina vaya recorriendo de derecha a izquierda. Comparamos el primer caracter con el último, y a medida que realizo la verificación, voy marcándolos con un B (blanco). Si el resultado final es una cadena en blanco, entonces la entrada era un palídromo.
* Q = { qab, qa, qb, qaB, qbB, qLB }
  + qab es el estado en el que se busca una 'a' o una 'b'.
  + qa es el estado en el que se busca que una 'a' sea el último caracter de la cadena. (La idea es que este estado esté presente en solo un movimiento, del blanco, a su inmediato anterior). Solo se mueve una vez a la izquierda en este estado.
  + qb es el estado en el que se busca que una 'b' sea el último caracter de la cadena. (La idea es que este estado esté presente en solo un movimiento, del blanco, a su inmediato anterior). Solo se mueve una vez a la izquierda en este estado.
  + qaB es el estado en el que se busca un blanco, luego de encontrar una 'a'.
  + qbB es el estado en el que se busca un blanco, luego de encontrar una 'b'.
  + qLB es el estado en el que se vuelve a la izquierda, hasta encontrar un blanco, para luego encontrar el próximo primer caracter
* Ʃ = {a, b }
* Γ = {a, b, B}
* q0 = qa
* La función de transición es:

|  | **a** | **b** | **B** |
| --- | --- | --- | --- |
| Qab | qaB, B, R | qbB, B, R | qA, B, S |
| Qa | qL, B, L |  | qA, B, S |
| Qb |  | qL, B, L | qA, B, S |
| qaB | qaB, a, R | qbB, b, R | qa, B, L |
| qbB | qaB, a, R | qbB, b, R | qb, B, L |
| qL | qL, a, L | qL, b, L | qab, B, R |

Ejercicio 7. Construir una MT que calcule la resta de dos números (se puede considerar la idea de solución propuesta en la clase 1).

La resolución de este ejercicio la tomé del capítulo uno del libro.

Se va a construir una MT M que calcula la resta m – n, tal que m y n son dos números naturales representados en notación unaria. Cuando m ≤ n, M devuelve la cadena vacía λ. En la entrada, m y n aparecen separados por el dígito 0.

Idea general: Dado w = 1^m 0 1^n, con m ≥ 0 y n ≥ 0, la MT M itera de la siguiente manera. En cada paso elimina el primer 1 del minuendo, y correspondientemente reemplaza el primer 1 del sustraendo por un 0. Al final elimina todos los 0 (caso m > n), o bien elimina todos los dígitos (caso m ≤ n).

Construcción de la MT M. La MT M = (Q, Ʃ, Γ, δ, q0, F) es tal que: Q = {q0, q1, q2, q3, q4, q5, q6}. El estado q0 es el estado de inicio de una iteración. El estado q1 es el estado en que M busca el primer 0 yendo a la derecha. El estado q2 es el estado en que M encuentra el primer 0 yendo a la derecha. El estado q3 es el estado en que M encuentra un 1 después de un 0 yendo a la derecha. El estado q4 es el estado en que M yendo a la derecha buscando un 1 después de un 0 encuentra en cambio un blanco. El estado q5 es el estado en que M, iniciando una iteración, no encuentra como primer dígito un 1. El estado q6 es el estado final.

Ʃ = {1, 0} Γ = {1, 0, B} F = {q6} La función de transición δ se define de la siguiente manera:

1. δ(q0, 1) = (q1, B, R)
2. δ(q1, 1) = (q1, 1, R)
3. δ(q1, 0) = (q2, 0, R)
4. δ(q2, 1) = (q3, 0, L)
5. δ(q2, 0) = (q2, 0, R)
6. δ(q3, 0) = (q3, 0, L)
7. δ(q3, 1) = (q3, 1, L)
8. δ(q3, B) = (q0, B, R)
9. δ(q2, B) = (q4, B, L)
10. δ(q4, 0) = (q4, B, L)
11. δ(q4, 1) = (q4, 1, L)
12. δ(q4, B) = (q6, 1, S)
13. δ(q0, 0) = (q5, B, R)
14. δ(q5, 0) = (q5, B, R)
15. δ(q5, 1) = (q5, B, R)
16. δ(q5, B) = (q6, B, S)

Ejercicio 8. Construir una MT que genere todas las cadenas de la forma anbn, con n ≥ 1 (se puede considerar la idea de solución propuesta en la clase 1)

Las MT que generan lenguajes tienen una cinta de salida de sólo escritura, en la que el cabezal se mueve únicamente hacia la derecha, y las cadenas se separan por el símbolo. Se asume que la entrada es la cadena vacía λ.

El lenguaje generado por una MT es el conjunto de cadenas que escribe en su cinta de salida. Las cadenas se pueden repetir, y no aparecen en ningún orden determinado.

La expresión G(M) denota el lenguaje generado por la MT M.

* Idea general: Podemos plantear una MT M con dos cintas. Una cinta sirve para contar, y la otra para escribir los caracteres 'a' o 'b' según corresponda. Usamos notación unaria en la cinta que usamos para contar. Cuando leo B (blanco), pongo un valor unario "i" que represente la cantidad de apariciones de 'a' y 'b' en la cinta de escritura de caracteres. En la cinta de caracteres, a partir de B, escribo i veces 'a', e i veces 'b'. Luego imprimo un separador, e incremento el valor de la cinta contadora. Es infinito.

Pasado en limpio, los pasos serían:

1. i := 1
2. imprimir i veces a, imprimir i veces b, e imprimir separador
3. i := i + 1 y volver a (2)