Av. Universidad 1001, Chamilpa, Cuernavaca, Morelos, México,
C.P. 62209 Tel. (777) 329 7000 y 329 7041 Ext.3148

CURSO PROPEDÉUTICO 2017

LÓGICA MATEMÁTICA

CONTENIDO

Número de horas totales: 14 hrs.

INTRODUCCIÓN

La lógica matemática es una parte de la lógica y las matemáticas, que consiste en el estudio matemático de la lógica y en la aplicación de este estudio a otras áreas de las matemáticas. La lógica matemática tiene estrechas conexiones con las ciencias de la computación y la lógica filosófica. La lógica matemática estudia los sistemas formales en relación con el modo en el que codifican nociones intuitivas de objetos matemáticos como conjuntos, números, demostraciones y computación.

LÓGICA MATEMÁTICA

Es la disciplina que trata de métodos de razonamiento. La lógica proporciona reglas y técnicas para determinar si es o no valido un argumento dado. El razonamiento lógico se emplea en matemáticas para demostrar teoremas; en ciencias de la computación para verificar si son o no correctos los programas.

PROPÓSITO

El propósito de este módulo es construir el andamiaje cognitivo para el desarrollo del pensamiento Lógico-Matemático, el cual está relacionado con la habilidad de trabajar y pensar en términos de números y la capacidad de emplear el razonamiento lógico.

MÓDULO 1. PROPOSICIONES Y TABLAS DE VERDAD

En el desarrollo de cualquier teoría matemática se hacen afirmaciones en forma de frases y que tienen un sentido pleno. Tales afirmaciones, verbales o escritas, las denominaremos enunciados o proposiciones.

1.1 Proposición

> Llamaremos de esta forma a cualquier afirmación que sea verdadera o falsa, pero no ambas cosas a la vez.

Ejemplo 1.1 Las siguientes afirmaciones son proposiciones.
(a) Gabriel García Márquez escribió Cien años de soledad.
(b) 6 es un número primo.
(c) 3+2=6
(d) 1 es un número entero, pero 2 no lo es.

♫ Las proposiciones se notan con letras minúsculas, p, q, r . . . . . .
La notación p:Tres más cuatro es igual a siete se utiliza para definir que p es la proposición “tres más cuatro es igual a siete”.

Este tipo de proposiciones se llaman simples, ya que no pueden descomponerse en otras.

Ejemplo 1.2 Las siguientes no son proposiciones.
(a) $x + y > 5$
(b) ¿Te vas?
(c) Compra cinco azules y cuatro rojas.
(d) $x = 2$

Solución
En efecto, (a) es una afirmación pero no es una proposición ya que será verdadera o falsa dependiendo de los valores de x e y e igual ocurre con la afirmación (d). Los ejemplos (b) y (c) no son afirmaciones, por lo tanto no son proposiciones.
Desde el punto de vista lógico carece de importancia cual sea el contenido material de los enunciados, solamente interesa su valor de verdad.

1.2 Valor de Verdad

Se llama valor verdadero o de verdad de una proposición a su veracidad o falsedad. El valor de verdad de una proposición verdadera es verdad y el de una proposición falsa es falso.

Ejemplo 1.3 Dígase cuáles de las siguientes afirmaciones son proposiciones y determinar el valor de verdad de aquellas que lo sean.
(a) p: Existe Premio Nobel de informática.
(b) q: La tierra es el único planeta del Universo que tiene vida.
(c) r: Teclee Escape para salir de la aplicación.
(d) s: Cinco más siete es grande.

Solución
(a) p es una proposición falsa, es decir su valor de verdad es Falso.
(b) No sabemos si q es una proposición ya que desconocemos si esta afirmación es verdadera o falsa.
(c) r no es una proposición ya que no es verdadera ni es falsa. Es un mandato.
(d) s no es una proposición ya que su enunciado, al carecer de contexto, es ambiguo.

En efecto, cinco niñas más siete niños es un número grande de hijos en una familia, sin embargo cinco monedas de cinco céntimos más siete monedas de un céntimo no constituyen una cantidad de dinero grande.

1.3 Proposición Compuesta

Si las proposiciones simples p1, p2, . . . , pn se combinan para formar la proposición P, diremos que P es una proposición compuesta de p1, p2, . . . , pn.
.

Ejemplo 1.4
"La Matemática Discreta es mi asignatura preferida y Mozart fue un gran compositor", es una proposición compuesta por las proposiciones "La Matemática Discreta es mi asignatura preferida"
y "Mozart fue un gran compositor".

"El es inteligente o estudia todos los días" es una proposición compuesta por dos proposiciones: "El es inteligente" y "El estudia todos los días".

♫ La propiedad fundamental de una proposición compuesta es que su valor de verdad está completamente determinado por los valores de verdad de las proposiciones que la componen junto con la forma en que están conectadas.

1.4 Variables de Enunciado

Es una proposición arbitraria con un valor de verdad no especificado, es decir, puede ser verdadera o falsa.

Sustituiremos por variables de enunciado. Toda variable de enunciado p, puede ser sustituida por cualquier enunciado siendo sus posibles estados, verdadero o falso. El conjunto de los posibles valores de una proposición p, los representaremos en las llamadas tablas de verdad, ideadas por L.Wittgenstein[1].

1.5 Tablas de Verdad

La tabla de verdad de una proposición compuesta *P* enumera todas las posibles combinaciones de los valores de verdad para las proposiciones *p1, p2, . . . , pn*.

Ejemplo 1.5 Por ejemplo, si P es una proposición compuesta por las proposiciones simples p1, p2 y p3, entonces la tabla de verdad de P deberá recoger los siguientes valores de verdad.

[1]Ludwig Wittgenstein (Viena 1889-Cambridge 1951), nacionalizado británico en 1938. Estudió Ingeniería Mecánica en Universidad de Cádiz

p1	**p2**	**p3**
V	V	V
V	V	F
V	F	V
V	F	F
F	V	V
F	V	F
F	F	V
F	F	F

1.6 Conexión entre Proposiciones

En este apartado se estudiaran las distintas formas de conectar proposiciones entre sí. Prestaremos especial atención a las tablas de verdad de las proposiciones compuestas que pueden formarse utilizando las distintas conexiones.

1.6.1 CONJUNCIÓN

Dadas dos proposiciones cualesquiera p y q, se llamará conjunción de ambas a la proposición compuesta "p y q" y su notación $p \wedge q$. Esta proposición será verdadera únicamente en el caso de que ambas proposiciones los sean.

p	q	p ∧ q
F	F	F
F	V	F
V	F	F

p	q	p ∧ q
0	0	0
0	1	0
1	0	0

V	V	V	**1**	1	1

1.6.2 DISYUNCIÓN

Dadas dos proposiciones cualesquiera p y q, se llamará disyunción de ambas a la proposición compuesta "p ó q" y su notación $p \vee q$. Esta proposición será verdadera si al menos una de las dos p ó q lo es.

p	q	p v q	p	q	p v q
F	F	F	**0**	0	0
F	V	V	**0**	1	1
V	F	V	**1**	0	1
V	V	V	**1**	1	1

La palabra "o" se usa en el lenguaje ordinario de dos formas distintas. A veces se utiliza en el sentido de "p ó q, ó ambos", es decir, al menos una de las dos alternativas ocurre y, a veces es usada en el sentido de "p ó q, pero no ambos" es decir, ocurre exactamente una de de las dos alternativas.
Por ejemplo, la proposición "El irá a Madrid o a Bilbao" usa "o" con el ´ultimo sentido. A este tipo de disyunción se le llama disyunción exclusiva.

1.6.3 DISYUNCIÓN EXCLUSIVA

Dadas dos proposiciones cualesquiera p y q, se llamará disyunción exclusiva de ambas a la proposición compuesta "p ´o q pero no ambos" y la notaremos $p \veebar q$. Esta proposición será verdadera si una u otra, pero no ambas son verdaderas.

Según esta definición una disyunción exclusiva de dos proposiciones p y q será verdadera cuando tengan distintos valores de verdad y falsa cuando sus valores de verdad sean iguales. Su tabla de verdad es, por tanto,

p	q	$p \veebar q$	p	q	$p \veebar q$
F	F	F	**0**	0	0
F	V	V	**0**	1	1
V	F	V	**1**	0	1
V	V	F	**1**	1	0

Haciendo el razonamiento contrario si $p \veebar q$ es verdad, únicamente se puede asegurar que una de las dos es verdad y si p Y q es falsa, sólo se puede deducir que ambas tienen el mismo valor de verdad.
♫ Salvo que se especifique lo contrario, "o" será usado en el primero de los sentidos. Esta discusión pone de manifiesto la precisión que se gana con el lenguaje simbólico.

1.6.4 NEGACIÓN

Dadas una proposición cualquiera, p, se llamará "negación de p" a la proposición "*no p* " y la notación es ¬p. Esta proposición será verdadera cuando p sea falsa y falsa cuando sea verdadera.

La tabla de verdad de esta nueva proposición, ¬p, es:

p	¬p	p	¬p
F	V	0	1
V	F	1	0

De esta forma, el valor verdadero de la negación de cualquier proposición es siempre opuesto al valor verdadero de la afirmación original.

Ejemplo 1.6 Estudiar la veracidad o falsedad de las siguientes proposiciones:

p1: El Pentium es un microprocesador.
p2: Es falso que el Pentium sea un microprocesador.
p3: El Pentium no es un microprocesador.
p4: 2 + 2 = 5
p5: Es falso que 2 + 2 = 5
p6: 2 + 2 = 4

Solución

X p2 y p3 son, cada una, la negación de p1.
X p5 y p6 son, cada una, la negación de p4.

Pues bien, de acuerdo con la tabla de verdad para la negación, tendremos:
X p1 es verdad, luego p2 y p3 son falsas.
X p4 es falsa, luego p5 y p6 son verdad.

Ejemplo 1.7 Construir la tabla de verdad de la proposición ¬(p ∧ ¬q).

p	q	¬q	p ∧ ¬q	¬(p ∧ ¬q)
V	V	F	F	V
V	F	V	V	F
F	V	F	F	V
F	F	V	F	V

Solución

Existen proposiciones que son verdaderas (falsas) simplemente por su forma lógica y no por su contenido.

1.7 Tautologías y Contradicciones

Sea P una proposición compuesta de las proposiciones simples p1, p2, . . . , pn
P es una Tautología si es verdadera para todos los valores de verdad que se asignen a p1, p2, . . . , pn.

P es una Contradicción si es falsa para todos los valores de verdad que se asignen a p1, p2, . . . , pn.

En adelante, se notará por "C" a una contradicción y por "T" a una tautología.

Una proposición P que no es tautología ni contradicción se llama, usualmente, Contingencia.

Ejemplo 1.8 Probar que la proposición compuesta p V ¬p es una tautología y la

p ∧ ¬p es una contradicción

Solución

En efecto:

p	¬p	p **V ¬p**	p ∧ ¬p
V	F	V	F
F	V	V	F

1.8 EQUIVALENCIAS LÓGICAS

1.8.1 LEYES DE LA LÓGICA

La siguiente tabla contiene las principales leyes de la lógica en el cálculo proposicional, donde A, B y C son fórmulas bien formadas. Para una mayor claridad con respecto al alcance de una negación en una fórmula bien formada en el cálculo proposicional, se usará en la tabla, la representación de $\overline{A}$ en vez de $\sim A$, donde A es una fórmula bien formada.

Número	Equivalencia Lógica	Nombre Ley
1.	$\overline{\overline{A}} \Leftrightarrow A$	Ley de doble negación
2.	$\overline{A \vee B} \Leftrightarrow \overline{A} \wedge \overline{B}$	Ley de De Morgan
2'.	$\overline{A \wedge B} \Leftrightarrow \overline{A} \vee \overline{B}$	Ley de De Morgan
3.	$A \vee B \Leftrightarrow B \vee A$	Ley conmutativa
3'.	$A \wedge B \Leftrightarrow B \wedge A$	Ley conmutativa
4.	$A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$	Ley asociativa
4'.	$A \wedge (B \wedge C) \Leftrightarrow (A \wedge B) \wedge C$	Ley asociativa
5.	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	Ley distributiva
5'.	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$	Ley distributiva
6.	$A \vee A \Leftrightarrow A$	Ley de idempotencia
6'.	$A \wedge A \Leftrightarrow A$	Ley de idempotencia
7.	$A \vee F \Leftrightarrow A$	Ley de identidad
7'.	$A \wedge V \Leftrightarrow A$	Ley de identidad
8.	$A \vee \overline{A} \Leftrightarrow V$	Ley inversa
8'.	$A \wedge \overline{A} \Leftrightarrow F$	Ley inversa
9.	$A \vee V \Leftrightarrow V$	Ley de dominación
9'.	$A \wedge F \Leftrightarrow F$	Ley de dominación
10.	$A \vee (A \wedge B) \Leftrightarrow A$	Ley de absorción
10'.	$A \wedge (A \vee B) \Leftrightarrow A$	Ley de absorción
11.	$(A \rightarrow B) \Leftrightarrow (\overline{B} \rightarrow \overline{A})$	Ley de transposición

ULO 2. SUCESIONES Y PROGRESIONES

Se llama sucesión a un conjunto de números dispuestos uno a continuación de otro.

$a_1, a_2, a_3, ..., a_n$
3, 6, 9,..., 3n
Los números $a_1, a_2, a_3, ...$; se llaman *términos de la sucesión*.
El subíndice indica el lugar que el término ocupa en la sucesión.

El término general es a_n es una expresión matemática que nos permite determinar cualquier término de la sucesión.

2.1 Determinación de una Sucesión:

- Por el término general

$a_n = 2n-1$
$a_1 = 2 \cdot 1 - 1 = 1$
$a_2 = 2 \cdot 2 - 1 = 3$
$a_3 = 2 \cdot 3 - 1 = 5$
$a_4 = 2 \cdot 4 - 1 = 7$
1, 3, 5, 7,..., 2n-1
No todas las sucesiones tienen término general. Por ejemplo, la sucesión de los números primos:
2, 3, 5, 7, 11, 13, 17, 19, 23,...

- Por una ley de recurrencia

Los términos se obtienen operando con los anteriores.
Escribir una sucesión cuyo primer término es 2, sabiendo que cada término es el cuadrado del anterior.
2, 4, 16, ...

Sucesión de Fibonacci

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...
Los dos primeros términos son unos y los demás se obtienen sumando los dos términos anteriores.

$$a_n = \begin{cases} 1 & \text{si } n = 1 \\ 1 & \text{si } n = 2 \\ a_{n-1} + a_{n-2} & \text{si } n > 2 \end{cases}$$

2.2 Progresiones Aritméticas

Una progresión aritmética es una sucesión de números tales que cada uno de ellos (salvo el primero) es igual al anterior más un número fijo llamado diferencia que se representa por d.

8, 3, -2, -7, -12, ...
3 - 8 = -5
-2 - 3 = -5
-7 - (-2) = -5
-12 - (-7) = -5
d= -5.

Término general de una progresión aritmética

1. Si conocemos el 1er término.

$a_n = a_1 + (n - 1) \cdot d$

8, 3, -2, -7, -12, ..

$a_n = 8 + (n-1)(-5) = 8 - 5n + 5 = = -5n + 13$

2 .Si conocemos el valor que ocupa cualquier otro término de la progresión.

$a_n = a_k + (n - k) \cdot d$

$a_4 = -7$ y $d = -5$

$a_n = -7 + (n - 4) \cdot (-5) = -7 - 5n + 20 = -5n + 13$

Interpolación de términos en una progresión aritmética

Interpolar medios diferenciales o aritméticos entre dos números, es construir una progresión aritmética que tenga por extremos los números ados.

Sean los extremos a y b, y el número de medios a interpolar m.

$$d = \frac{b - a}{m + 1}$$

Interpolar tres medios aritméticos entre 8 y -12.

$$d = \frac{-12 - 8}{3 + 1} = \frac{-20}{4} = -5$$

8, 3, -2, -7 , -12.

Suma de términos equidistantes de una progresión aritmética

Sean a_i y a_j dos términos equidistantes de los extremos, se cumple que la suma de términos equidistantes es igual a la suma de los extremos.

$a_i + a_j = a_1 + a_n$

$a_3 + a_{n-2} = a_2 + a_{n-1} = \ldots = a_1 + a_n$

8, 3, -2, -7, -12, ...

3 + (-7) = (-2) + (-2) = 8 + (-12)

-4 = -4 = -4

Suma de n términos consecutivos de una progresión aritmética

$$S_n = \frac{(a_1 + a_n)n}{2}$$

Calcular la suma de los primeros 5 términos de la progresión : 8, 3, -2, -7, -12, ...

$$S_5 = \frac{(8 - 12)5}{2} = \frac{-20}{2} = -10$$

MÓDULO 3. ALGORITMOS Y MÁQUINAS DE TURING

3.1 ALGORITMOS NUMÉRICOS

> El concepto de algoritmo pertenece a las nociones fundamentales de las matemáticas. Entendemos por algoritmo la prescripción exacta sobre el cumplimiento de cierto sistema de operaciones en un orden determinado para la resolución de problemas de un tipo dado.

Veamos en calidad de ejemplo el algoritmo de Euclides que resuelve todos los problemas del tipo siguiente:

Hallar el máximo común divisor de dos números naturales dados a y b.

1. Examina los dos números a y b. Pasa a la indicación siguiente.
2. Compara los dos números (a== b, o a < b, o a > b); Pasa al a indicación siguiente.
3. Si los números examinados son iguales cada uno de ellos da el resultado que se busca. El proceso de cómputo se para. Si no es así, pasa a la siguiente indicación.
4. Si el primero de los números examinados es menor que el segundo, cámbialos de lugar y continúa su examen. Pasa a la indicación siguiente.
5. Resta el segundo de los números examinados del primero y examina dos números: el sustraendo y el resto. Pasa a la indicación 2.

Después de que las cinco indicaciones se hayan cumplido hay que volver de nuevo a la segunda, pasar a la tercera, a la cuarta, a la quinta y otras vez a la segunda, a la tercera, etc., hasta que se obtengan números iguales, es decir, hasta que se cumpla la condición que se tiene en la tercera indicación; entonces se cesa el proceso.

```
#include <stdio.h>
#include <conio.h>

int MCD (int num1, int num2);

main ()
{
    int num1,num2;
    printf("Tecle dos valores\n");
    scanf("%d %d",&num1, &num2);
        printf("\n El MCD de %d y %d es %d", num1, num2, MCD(num1,num2));
}

int MCD (int num1, int num2)
{
        int rest;
        if((num1%num2)==0)
    {
                rest=num2;
        return (rest);
```

```
    }
else
        rest=MCD(num2,num1%num2);
            return (rest);
    }
    }
```

3.2 Algoritmos para la resolución de problemas lógicos

Estos problemas al igual que los anteriores tratan de lo mismo, de elaborar un algoritmo que permita resolver cualquier problema de la clase examinada de problemas del mismo tipo con un solo método. Pero en estos casos los algoritmos no serán numéricos; más bien atienden a una serie de razonamientos lógicos determinados por las condiciones que se presenten, para de esta manera tomar una decisión. Veamos como ejemplo la búsqueda en un laberinto.

```
#include <stdio.h>
#include <conio.h>

int N, M;
bool fin;
char mapa[1002][1002];
int mov[4][2] =
{
   {0, 1},
   {0, -1},
   {1, 0},
   {-1, 0}
};
/**

# - Pared
. - Camino
I - Inicio
F - Fin
N - Numero de filas
M - Numero de columnas
5 5
#####
#..F#
..##I
.#.#.
.....
**/

void imprime()
{
   for(int i=1; i<=N; i++)
   {
      for(int j=1; j<=M; j++)
      {
         printf("%c", mapa[i][j]);
```

```
            }
            printf("\n");
        }
        printf("\n");
    }
```

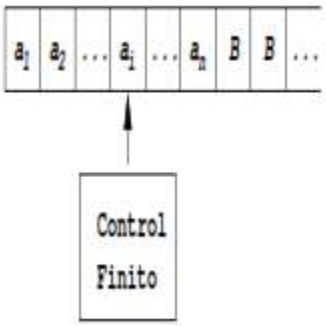
```
void buscar(int x, int y)
{
    if(mapa[x][y] == 'F')
    {
        imprime();
        fin = true;
        return;
    }
    imprime();
    mapa[x][y] = 'V';
    for(int i=0; i<4; i++)
    {
        int newx = x + mov[i][0];
        int newy = y + mov[i][1];
        if(newx * newy == 0 || fin)
            continue;
        if(newx > N || newy > M)
            continue;
        if(mapa[newx][newy] == '#')
            continue;
        if(mapa[newx][newy] == 'V')
            continue;
        buscar(newx, newy);
     }
}

int main()
{
    int x, y;
    scanf("%d %d", &N, &M);
    for(int i=1; i<=N; i++)
    {
        scanf(" %s", &mapa[i][1]);
        for(int j=1; j<=M; j++)
        {
            if(mapa[i][j] == 'I')
            {
                x = i;
                y = j;
            }
        }
    }
    printf("\n");
    buscar(x,y);
    return 0;
}
```

3.3 El Modelo de la máquina de Turing

Un modelo formal de cómputo efectivo debe tener ciertas propiedades. Primero, cada procedimiento debe poder ser descrito de manera finita. Segundo, el procedimiento debe consistir de pasos separados, y cada uno de ellos debe poderse llevar a cabo de manera mecánica. Este modelo fue introducido por Alan Turing en 1936. Aquí presentaremos una variante.

El modelo básico se ilustra en la siguiente figura,



Tiene un control finito, una cinta que está dividida en celdas, y una cabeza de la cinta que explora una celda en la cinta a la vez. La cinta tiene una primera celda ubicada en la posición más a la izquierda de la cinta pero es infinita del lado derecho. Cada celda en la cinta puede contener exactamente un símbolo tomado de un alfabeto finito. Inicialmente, la n celdas más a la izquierda, para algún número finito n ≥ 0, contienen la entrada de la máquina, la cual es una cadena de símbolos que se escogen de un alfabeto llamado símbolos de entrada. Todas las celdas restantes (que forman un conjunto infinito) contienen el símbolo blanco, el cual es un símbolo especial que no forma parte del alfabeto de entrada.

Un movimiento en la Máquina de Turing depende del símbolo explorado por la cabeza de la cinta y del estado de control finito. De manera sintética se hace lo siguiente:

1) dependiendo del símbolo explorado por la cabeza lectora/escritora y del estado en el control finito se cambia de estado,

2) se imprime un símbolo en la celda explorada, reemplazando el símbolo que ésta contenga y

3) se mueve la cabeza de lectura una celda hacia la izquierda o derecha. Note que la diferencia entre la máquina de Turing y un autómata finito de doble vía es que la máquina de Turing tiene la habilidad de cambiar los símbolos en la cinta de entrada.

Formalmente, una máquina de Turing (MT) se denota por:

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

Donde:
Q es un conjunto finito de estados,

Γ es un conjunto finito de símbolos, llamado alfabeto de la cinta,

B es un símbolo en Γ, y el símbolo blanco,

Σ es un subconjunto de Γ que no incluye a B, es el alfabeto de símbolos de entrada,

δ es la función que determina los movimientos de la máquina, es una transformación de $Q \times \Gamma$ a $Q \times \Gamma \times \{L, R\}$ (δ puede estar indefinida para algunos argumentos),
$q_0 \in Q$ es el estado inicial,
$F \subseteq Q$ es el conjunto de estados finales.
Una descripción instantánea (DI) de una máquina de Turing M es una expresión $\alpha 1 q \alpha 2$. Aquí $q \in Q$, es el estado actual de M; α_1 α_2 es una cadena en $\Gamma*$ y es el contenido de la cinta hasta el símbolo no blanco más a la derecha o el símbolo de la izquierda de la cabeza, cualquiera que esté más a la derecha. (Observar que el blanco B puede ocurrir en $\alpha_1\alpha_2$).

Se asume que Q y Γ son disjuntos. Finalmente, se asume que la cabeza de la cinta explora el símbolo más a la izquierda de α_2, o si $\alpha_2 = \in$, la cabeza explora un blanco.

Se define un movimiento en M como: sea $X_1X_2 \cdots X_{i-1}qX_i \cdots X_n$ una DI. Se supone que $\delta(q, X_i) = (p, Y, L)$, si $i - 1 = n$, entonces se considera a X_i como B. Si $i = 1$ entonces no es posible realizar el movimiento a la izquierda así que no hay DI siguiente porque la cabeza de la cinta no tiene permitido desprenderse del fin izquierdo de la cinta. Si $i > 1$, entonces escribimos

1) $X_1X_2 \cdots X_{i-1}qXi \cdots X_n \vdash_M X_1X_2 \cdots X_{i-2p}Xi-_1Y\ X_{i+1} \cdots X_n$

Sin embargo, si cualquier sufijo de $Xi-1Y\ Xi+1 \cdots X_n$ es completamente blanco entonces el sufijo se borra en (1).

Alternativamente si $\delta(q, Xi) = (p, Y, R)$ entonces escribimos:

2) $X_1X_2 \cdots X_{i-1}qX_iX_{i+1} \cdots X_n \vdash_M X_1X_2 \cdots X_{i-1}Y\ pX_{i+1} \cdots Xn$

Notese que en el caso de $i - 1 = n$, la cadena $X_i \cdots X_n$ es vacía, y el lado derecho de (2) es más largo que el lado izquierdo.

Si dos DI están relacionadas por $\vdash_M$, decimos que la segunda resulta de la primera por un movimiento. Si una DI resulta de otra por un número finito de movimientos, incluyendo cero movimientos, ellas están relacionadas por el símbolo $\overset{*}{\vdash}_M$.

El lenguaje aceptado por M, denotado por L(M), es el conjunto de las palabras en Σ^* que causan que M entre a un estado final cuando se colocan al principio de la cinta de M , con M en el estado q0 , y la cabeza de la cinta de M en la celda más a la izquierda. Formalmente, el lenguaje aceptado por $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es:

Dada una MT que reconoce un lenguaje L, asumimos sin perder generalidad que la M T se para, es decir, no tiene más movimientos, cuando la entrada es aceptada. Sin embargo, para algunas palabras no aceptadas, es posible que la MT nunca pare.

Ejemplo:

El diseño de una MT que acepte el lenguaje L = $\{0^n1^n \mid n \geq 1\}$ es: inicialmente, la cinta de entrada de M contendrá $0^n\ 1^n$ seguido de un número infinito de blancos. Repetidamente, M reemplaza el 0 más a la izquierda por X , mueve hacia la derecha la cabeza de la cinta hasta encontrar el 1 que esté más a la izquierda, reemplazándolo por Y , luego se mueve a la izquierda para encontrar la X más a la derecha, entonces mueve la cabeza una celda hacia la derecha y si encuentra un 0 entonces repite el ciclo. Sin embargo, si cuando se busca un 1, M encuentra un blanco en su lugar, entonces M se para sin aceptar su entrada. Si, después de cambiar un 1 por una Y, M no encuentra más 0's, entonces M examina que no haya más 1, si no los hay entonces acepta su entrada.

Sea Q = $\{q_0, q_1, q_2, q_3, q_4\}$, Σ = {0, 1}, Γ = {0, 1, X, Y, B}, y F = $\{q_4\}$. Informalmente, cada estado representa una declaración o un grupo de declaraciones en un programa. Al estado q_0 se ingresa inicialmente y es el estado previo a cada reemplazo del 0 más a la izquierda por una X. El estado q_1 se usa para buscar a la derecha, saltando sobre los 0's y las Y 's, hasta encontrar el 1 más a la izquierda. Si M encuentra 1, lo cambia por Y, y entra al estado q_2 .

El estado q_2 busca a la izquierda una X y entra en el estado q_0 cuando lo encuentra, luego se mueve hacia la derecha hasta el 0 ubicado más a la izquierda. Conforme M busca hacia la derecha en el estado q_1, si B o X aparecen antes de encontrar un 1, entonces la entrada es rechazada; puede que haya demasiados 0's o la entrada no

está en $\mathbf{0*1*}$.
El estado q_0 tiene otro papel. Si, después del estado q_2 se encuentra la X más a la derecha, existe una Y inmediatamente a su derecha, entonces los 0's están agotados. Desde q_0, se explora Y, entra el estado q_3 para explorar sobre las Y 's y examina que no haya 1's restantes. Si las Y 's están seguidas por B, se entra al estado q_4 y la aceptación ocurre; en otro caso, la cadena es rechazada. La función δ es:

Estado	0	1	*X*
q_0	(q_1, X, R)	–	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)
q_3	–	–	–
q_4	–	–	–

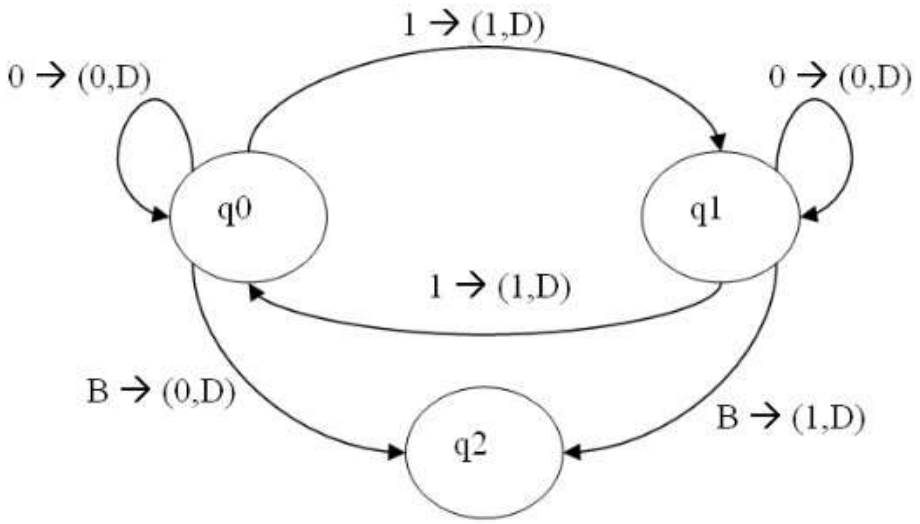
Estado	*Y*	*B*
q_0	(q_3, Y, R)	–
q_1	(q_1, Y, R)	–
q_2	(q_2, Y, L)	–
q_3	(q_3, Y, R)	(q_4, B, R)
q_4	–	–

La siguiente tabla muestra el cálculo de M para la entrada 0011.

$q_0 0011 \vdash X q_1 011$	$\vdash X 0 q_1 11 \vdash X q_2 0 Y 1 \vdash$
$q_2 X 0 Y 1 \vdash X q_0 0 Y 1$	$\vdash X X q_1 Y 1 \vdash X X Y q_1 1 \vdash$
$X X q_2 Y Y \vdash X q_2 X Y Y$	$\vdash X X q_0 Y Y \vdash X X Y q_3 Y \vdash$
$X X Y Y q_3 \vdash X X Y Y B q_4$	

Ejercicio

- Construir una máquina de Turing que agregue un bit de paridad a una secuencia binaria para que la cantidad de "1" sea par. G={0,1,B}.
- El conjunto G es el conjunto de símbolos que pueden encontrarse en la cinta. Este dato es importante porque la máquina se detiene cuando se encuentra en una situación indefinida.



BIBLIOGRAFIA

1. González Gutiérrez, Francisco José. Apuntes de Lógica Matemática 1. Lógica de proposiciones. Universidad de Cádiz. Escuela Superior de Ingeniería, Departamento de Matemáticas. Año 2005.

2. Morales Peña, Hugo Humberto. Material de apoyo para el primer curso de matemáticas computacionales. Universidad Tecnológica de Pereira. Facultad de Ciencias Básicas. Departamento de Matemáticas. Pereira, Risaralda. 2010. Recuperado de http://www.slideshare.net/JuanPabloRomeroL/libro-matematicas-computacionaleshugohumbertomoralesversionjulio28de2010

3. B A trajtenbrot. Los algoritmos y la resolución automática de problemas. Editorial MIR, Moscú 1977.

4. Feliú Sagols Troncoso. Curso básico de computación. Matemáticas-Cinvestav. Recuperado de: http://acme.math.cinvestav.mx/~basico/apache/siete.pdf