

1) Dada la siguiente metaER encuentre una ER básica equivalente

$[ac]\{2,4\}^?|(tm)^+|rs^*$
 $(a+c)(a+c)(a+c+\epsilon)(a+c+\epsilon)^+\epsilon+tm(tm)^*+rs^*$

2) Dada la siguiente ER básica encuentre una metaER equivalente

$(x+y)(x+y)(x+y)(x+y+\epsilon)(x+y+\epsilon)am(am)^*+fm$
 $[xy]\{3,5\}(am)^+|fm$

3) Dado el siguiente fragmento de programa en ANSI C, arme una tabla para los errores de compilación con las columnas: Nro de línea, descripción del error.

```
1 void *p; int v[10], i = 10;
2 *p = 5;
3 for (i--) {
4     v[i] = 5++; }
```

Nro de Línea	Error
2	No se puede des referenciar un puntero a void sin casting
3	Faltan los ; en el for
4	No se puede incrementar 5 porque no es un valorL modificable

4) Diga cuál es la semántica del siguiente fragmento de programa en ANSI C

```
{int a = 8; while (--a) if (a%2) printf("%d\n", a+1); }
```

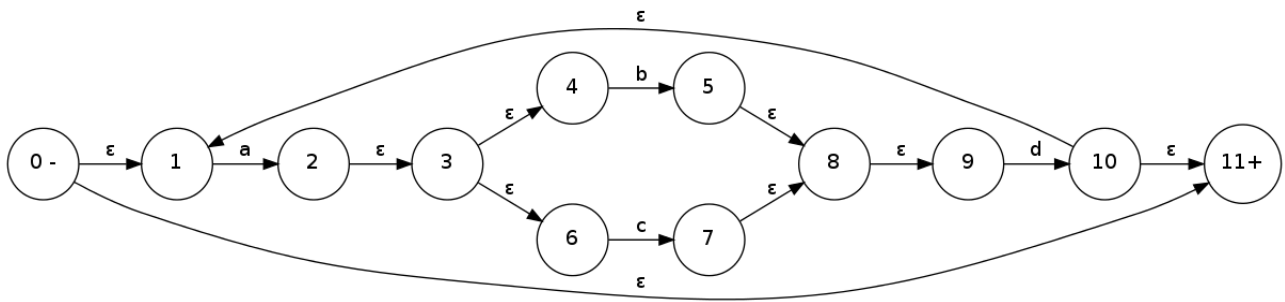
Muestra por pantalla los números pares entre el 8 y el 2, uno por línea, quedando al final de la ejecución el cursor en una nueva línea.

5) Marque con una cruz si los siguientes constructos ANSI C tiene errores semánticos, errores sintácticos, o no tiene error de compilación. Asuma que las funciones estándar están disponibles.

	Errores Semánticos	Errores Sintácticos	Sin Error
{int a=2,b=3,c; c = a > b ? a ■ ;}		x	
{int *p; p = &5 ; *p++;}	x		
{int i=7; do printf("%d\n", i--); while (i);}			x
{int i=0, v[10]; while (i<10 ■ v[i++]-- ;}		x	

6) Dada la siguiente ER $(a(b+c)d)^*$ encuentre el diagrama de transición correspondiente a aplicar el algoritmo de Thompson

Solución



7) Convierta el AFN- ϵ dado por la siguiente tabla de transición a un AFD

TT	a	b	ϵ
0-	{2}	{0,2}	{4}
1	-	{2}	{3}
2	{3}	-	-
3	{1,3}	{4}	-
4+	-	-	{2}

Solución

Clausura- ϵ ({0}) = {0,2,4}
 Hacia({0,2,4},a) = {2,3}
 Clausura- ϵ ({2,3}) = {2,3}
 Hacia({0,2,4},b) = {0,2}
 Clausura- ϵ ({0,2}) = {0,2,4}
 Hacia({2,3},a) = {1,3}
 Clausura- ϵ ({1,3}) = {1,3}
 Hacia({2,3},b) = {4}
 Clausura- ϵ ({4}) = {2,4}
 Hacia({1,3},a) = {1,3}
 Hacia({1,3},b) = {2,4}
 Clausura- ϵ ({2,4}) = {2,4}
 Hacia({2,4},a) = {3}
 Clausura- ϵ ({3}) = {3}
 Hacia({2,4},b) = -
 Hacia({3},a) = {1,3}
 Hacia({3},b) = {4}

TT	a	b
{0,2,4}±	{2,3}	{0,2,4}
{2,3}	{1,3}	{2,4}
{1,3}	{1,3}	{2,4}
{2,4}+	{3}	-
{3}	{1,3}	{2,4}

TT	a	b
0±	1	0
1	2	3
2	2	3
3+	4	-
4	2	3

No se pedía pero se pueden simplificar los estados equivalentes 1, 2 y 4 quedando

TT	a	b
0±	1	0
1	1	3
3+	1	-

8) Obtenga la tabla de transición del AFD mínimo equivalente al AFD de la siguiente tabla

Solución

TT	a	b
0-	1	3
1	4	2
2+	2	3
3	1	0
4	3	5
5+	5	0
6	3	4

Tabla Inicial. Estado 6 inalcanzable.

TT	a	b	
0-	1	3	C0
1	4	2	
3	1	0	
4	3	5	
2+	2	3	C1
5+	5	0	

Separamos por finales y no finales para armar las clases

TT	a	b	
0-	C0	C0	C0
1	C0	C1	
3	C0	C0	
4	C0	C1	
2+	C1	C0	C1
5+	C1	C0	

Los estados 1 y 4 forman una nueva clase

TT	a	b	
0-	C2	C0	C0
3	C2	C0	
1	C2	C1	C2
4	C0	C1	
2+	C1	C0	C1
5+	C1	C0	

Los estados 1 y 4 se separan

TT	a	b	
0-	C2	C0	C0
3	C2	C0	
1	C3	C1	C2
4	C0	C1	C3
2+	C1	C0	C1
5+	C1	C0	

No se puede dividir más

TT	a	b
0-	1	0
1	4	2
2+	2	0
4	0	2

Dejamos solo un estado por clase

9) Obtenga el complemento de la intersección de los siguientes AFD

TT1	a	b	c
0-	-	1	0
1+	1	1	-

TT2	a	b	c
2-	3	2	2
3+	3	2	-

Solución

TT	a	b	c
(0,2)-	-	(1,2)	(0,2)
(1,2)	(1,3)	(1,2)	-
(1,3)+	(1,3)	(1,2)	-

Aplicando algoritmo de intersección

TT	a	b	c
0-	-	1	0
1	2	1	-
2+	2	1	-

Renombrando

TT	a	b	c
0-	3	1	0
1	2	1	3
2+	2	1	3
3	3	3	3

Completando estados

TT	a	b	c
0±	3	1	0
1+	2	1	3
2	2	1	3
3+	3	3	3

Complementado

10) Obtenga la ER correspondiente a la siguiente tabla de transición

TT	a	b
0-	2	1
1	3	2
2	-	1
3+	-	-

Solución

Las ecuaciones:

$$0 = a2 + b1$$

$$1 = a3 + b2$$

$$2 = b1$$

$$3 = \epsilon$$

Por tanto:

$$1 = a3 + b2 = a\epsilon + bb1 = bb1 + a$$

$$1 = (bb)^*a$$

$$2 = b(bb)^*a$$

$$0 = a2 + b1 = ab(bb)^*a + b(bb)^*a = (ab + b)(bb)^*a$$

11) Dado el siguiente fragmento de código ANSI C arme una tabla los lexemas que reconoce el escáner e indique a que categoría pertenecen

```
int i=0, v[10];
for (; i < 10; i++) /* loop */
    v[i] = 15 - i;
```

lexema	token	lexema	token	lexema	token
int	palab. Reservada	for	palab. Reservada	v	ident
i	identificador	(carác. puntuación	[operador
=	carác. puntuación	;	carác. puntuación	i	identificador
0	constante	i	identificador]	operador
,	carác. puntuación	<	operador	=	operador
v	identificador	10	constante	15	constante
[carác. puntuación	;	carác. puntuación	-	operador

10	constante	i	identificador	i	identificador
]	carác. puntuación	++	operador	;	carác. puntuación
;	carác. puntuación)	carác. puntuación		