# TEORÍA DE LA COMPUTACIÓN

**Grado en Ingeniería Informática**

**Soluciones del Boletín de Ejercicios nº 4**

## Máquinas de Turing

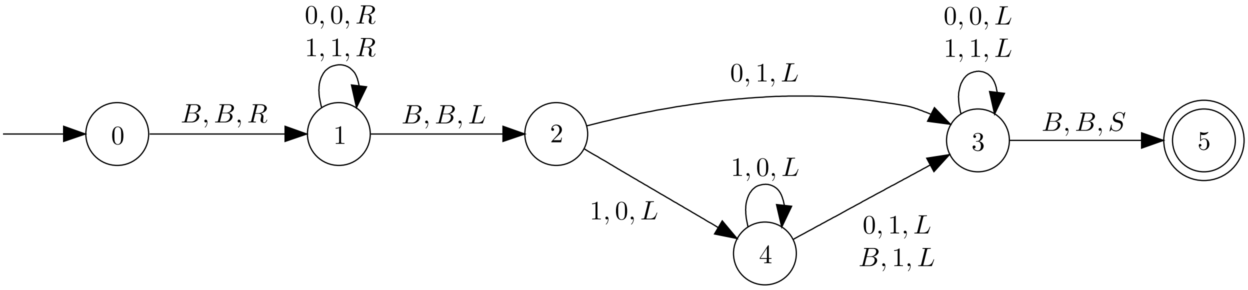
61. Construya una máquina de Turing $M$ que dado un número en binario, el cual se representa en la cinta mediante una cadena de 0,s y 1,s acotada por blancos, lo incremente en una unidad. Ejemplos de ejecución:

    – La entrada $\underline{B}\,1\,0\,1\,0\,1\,B$ se transforma en $\underline{B}\,1\,0\,1\,1\,0\,B$.

    – La entrada $\underline{B}\,1\,1\,1\,1\,1\,B$ se transforma en $\underline{B}\,1\,0\,0\,0\,0\,0\,B$.

    A la hora de especificar la máquina $M$ obtenida, hágalo dibujando su correspondiente grafo.

    <u>Solución</u>:




62. Mediante composición de máquinas básicas (tales como $R$, $L$, $R_B$, $L_B$, $S_R$, $S_L$, $C$, etc.), construya una máquina de Turing que, dada una cadena de $n$ $a$es acotada por blancos, genere una nueva cadena de $2^n$ $a$es también acotada por blancos. La nueva cadena puede generarse en cualquier zona de la cinta. La cadena de entrada puede quedar destruida o no. Ejemplos de ejecución:

    – La entrada $\underline{B}B$ debe producir la salida $\underline{B}aB$, donde $B$ es el símbolo blanco.

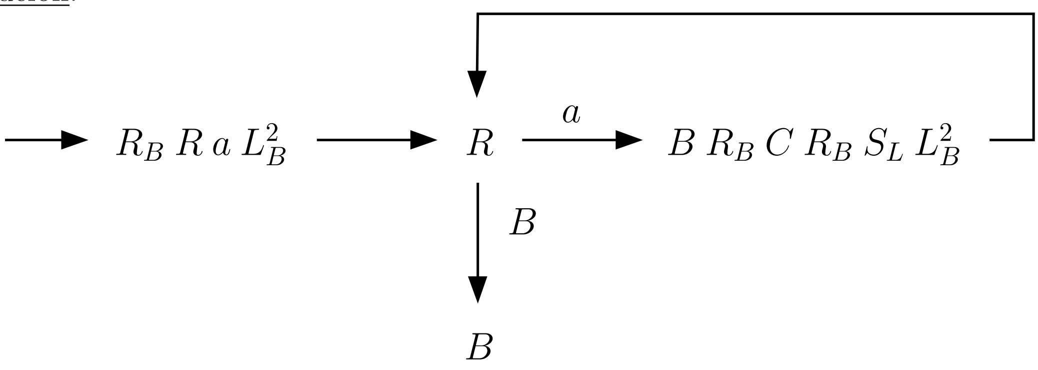
    – La entrada $\underline{B}aB$ debe producir la salida $\underline{B}aaB$.

    – La entrada $\underline{B}aaB$ debe producir la salida $\underline{B}aaaaB$.

    – La entrada $\underline{B}aaaB$ debe producir la salida $\underline{B}aaaaaaaaB$.

    – Etc...

    <u>Solución</u>:



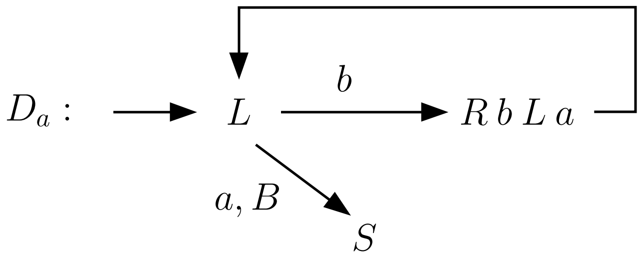
63. Mediante composición de máquinas básicas, construya una máquina de Turing $M$ que transforme cadenas de $(a \cup b)^*$ delimitadas por blancos en cadenas de $a^*b^*$, pero respetando el número de $a$es y el número de $b$es de la cadena original. Es decir, dada una cadena de entrada, $M$ agrupa todas las $a$es al principio y todas las $b$es al final.

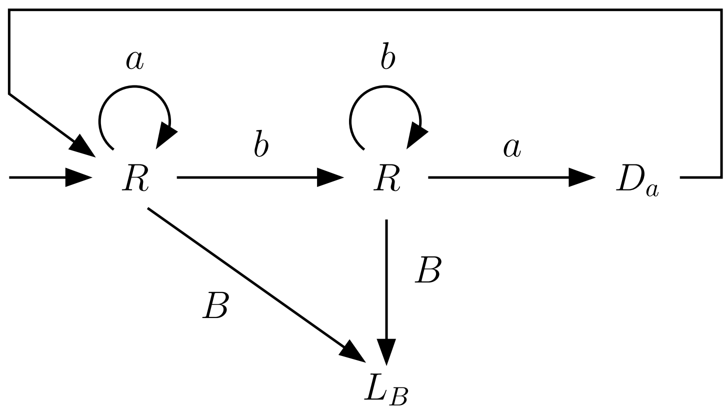
    Ejemplo: la cadena $\underline{B}bbabaB$ se transforma en $\underline{B}aabbbB$, donde $B$ es el símbolo blanco.

    Nota: La cabeza de lectura escritura está posicionada inicialmente en el símbolo blanco que precede a la cadena de entrada, y debe quedar posicionada al final en el símbolo blanco que precede a la cadena resultado. La cadena resultado debe quedar ocupando las mismas celdas que la cadena de entrada original. Y, además, durante la generación del resultado, $M$ sólo puede acceder a dichas celdas y a los blancos delimitadores.

    Sugerencia: Si durante el proceso de construcción de $M$ identifica alguna operación auxiliar que se repita con frecuencia, puede definir previamente dicha operación como una nueva máquina básica y utilizarla en la definición de $M$.

    <u>Solución</u>:




64. Mediante composición de máquinas básicas, construya las siguientes máquinas de Turing:

    – *Split*, la cual, dada una cadena de entrada de longitud par y acotada por blancos, devuelve dos cadenas de la misma longitud, la primera formada por los símbolos que ocupan posición impar en la cadena original, y la segunda formada por los símbolos que ocupan posición par, y en ambos casos con el mismo orden relativo. Ante una cadena vacía, *Split* debe generar dos cadenas vacías consecutivas. Ante una cadena de longitud impar, *Split* debe finalizar su computación en un estado de error.

      Ejemplos de ejecución:

      * La entrada $\underline{B}B$ debe producir la salida $\underline{B}BB$, donde $B$ es el símbolo blanco.
      * La entrada $\underline{B}abcdefB$ debe producir la salida $\underline{B}aceBbdfB$.
      * La entrada $\underline{B}abcB$ debe producir una situación de error.

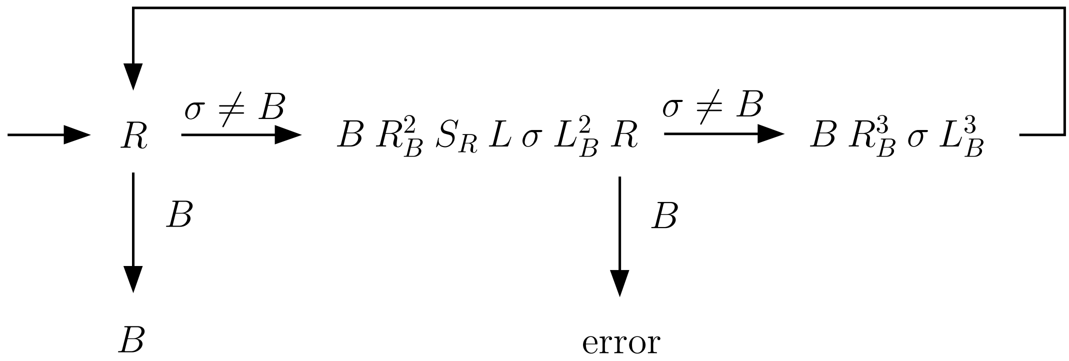
    – *Combine*, la cual, dadas dos cadenas de entrada $u$ y $v$ acotadas por blancos, genera una nueva cadena $w$, de forma que los símbolos de $u$ aparecen en las posiciones impares de $w$, los símbolos de $v$ aparecen en las posiciones pares de $w$ y, en ambos casos, con el mismo orden relativo. Si las cadenas $u$ y $v$ son ambas vacías, la máquina *Combine* debe generar la cadena vacía. Si las cadenas $u$ y $v$ tienen distinta longitud, la máquina *Combine* debe finalizar su computación en un estado de error.

Ejemplos de ejecución:

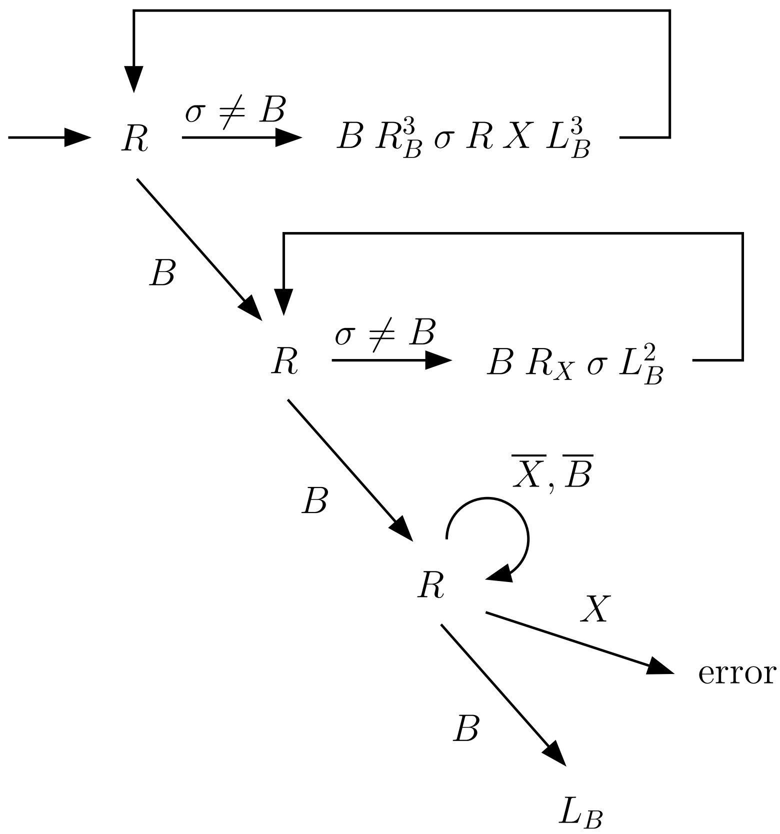
- ∗ La entrada $\underline{B}BB$ debe producir la salida $\underline{B}B$.
- ∗ La entrada $\underline{B}aceBbdfB$ debe producir la salida $\underline{B}abcdefB$.
- ∗ La entrada $\underline{B}aceBbdB$ debe producir una situación de error.

Solución:

*Split*:




*Combine*:



65. Diseñe una máquina de Turing $M$ que, dada una cadena de entrada de $n$ $a$es, devuelva la cadena de ceros y unos que corresponde a la codificación binaria del número $n$. Por ejemplo, ante la entrada $\underline{B}aaaaaaaaaaaaaB$, $M$ debe generar como salida la cadena $\underline{B}1101B$. Explique cómo podría construirse dicha máquina $M$, sin especificar de manera completa su función de transición, pero sí indicando el máximo detalle posible sobre su arquitectura y sobre la misión de cada uno de sus componentes internos. Es decir, realice un diseño esquemático, no completamente detallado. Es suficiente con indicar, por ejemplo, cuántas cintas debe tener la máquina, qué almacena cada cinta, y cómo la máquina hace uso de esos contenidos durante su funcionamiento. Eso sí, no olvide especificar todas las posibles situaciones en las que la máquina se detiene, indicando dónde queda almacenada la cadena de salida.

    Solución:

    La máquina de Turing $M$ tendrá dos cintas:

    - La primera cinta almacenará la cadena de entrada, es decir, la cadena de $a$es.
    - La segunda cinta almacenará el número binario. Su configuración inicial será $\underline{B}0B$.

    $M$ irá marcando con blancos las $a$es de la primera cinta. Por cada $a$ que marque, incrementará en uno el número binario de la segunda cinta (por ejemplo, usando la máquina del ejercicio 61). El proceso termina cuando ya no quedan más $a$es por marcar en la primera cinta. El resultado será el contenido de la segunda cinta.

## Lenguajes recursivos y recursivamente enumerables

66. Supongamos que $L$ es recursivamente enumerable, pero no recursivo. Demuestre que, para toda máquina de Turing $M$ que acepte $L$, existe un número infinito de cadenas que pueden hacer que $M$ no pare.

    Solución:

    Supongamos que $M$ es la máquina de Turing que acepta $L$. Supongamos que el número de cadenas que hacen que $M$ no pare es finito. Si es finito, serían aceptadas por un autómata finito.

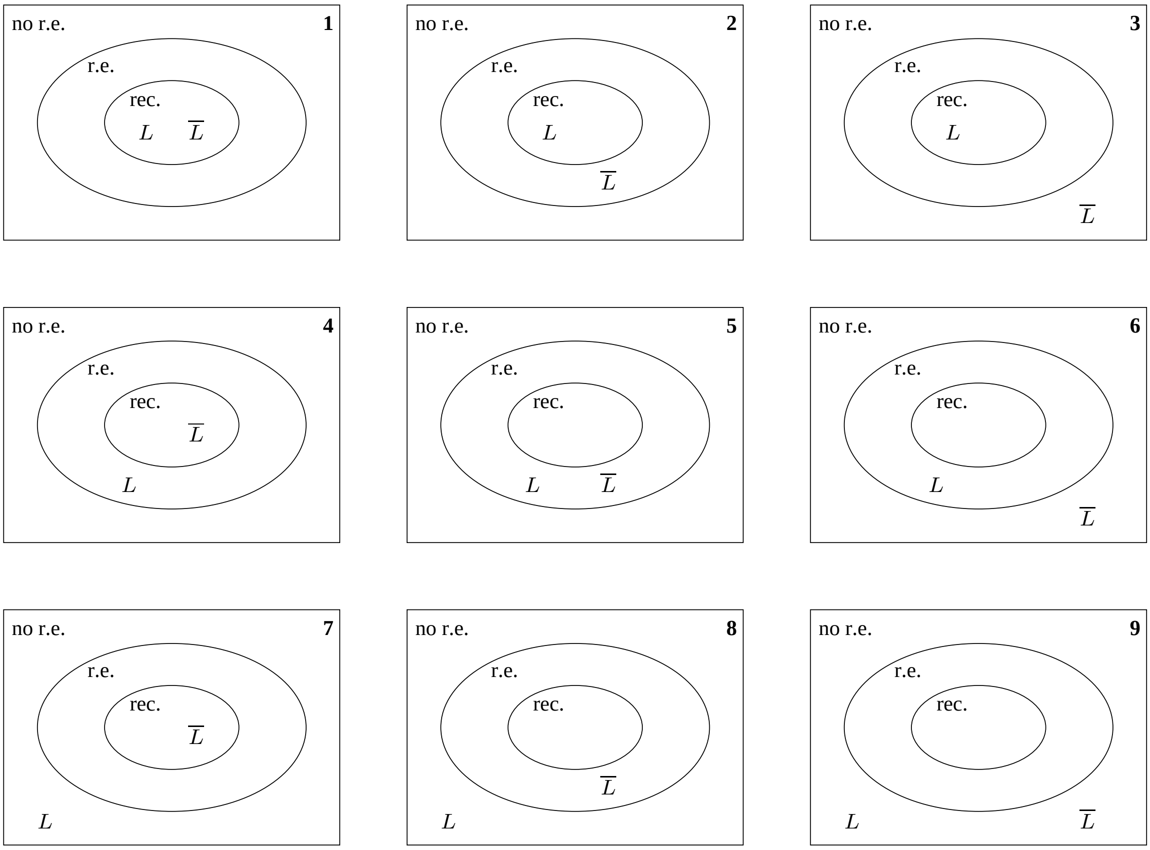
    Entonces construimos $M'$ mediante la combinación de $M$ y de este autómata finito, de tal forma que, cuando llega una cadena de entrada $w$, si el autómata finito la acepta, $M'$ para y rechaza; y si el autómata finito la rechaza, $M'$ simula a $M$ sobre $w$.

    Por tanto, $M'$ acepta $L$ y $M'$ siempre para ante cualquier cadena de entrada, lo cual es una contradicción, porque $L$ es recursivamente enumerable, no recursivo.

67. Dado un lenguaje $L$, denotamos su complementario mediante $\overline{L}$. Así mismo, denotamos también mediante “rec.”, “r.e.” y “no r.e.”, los lenguajes “recursivos”, “recursivamente enumerables (pero no recursivos)” y “no recursivamente enumerables”, respectivamente.

    Indique entonces cuáles de las nueve situaciones de la siguiente figura son posibles y cuáles no lo son.

    Indique también, de forma clara y ordenada, los resultados utilizados en el razonamiento, sin probarlos.




Solución:

El teorema que dice "*si un lenguaje es recursivo, su complementario también lo es*" hace válida la situación 1 y hace no válidas las situaciones 2, 3, 4 y 7.

El teorema que dice "*si un lenguaje es recursivamente enumerable y su complementario también, entonces ambos son recursivos*" hace no válida la situación 5.

El teorema que dice "*hay al menos un lenguaje recursivamente enumerable cuyo complementario no es recursivamente enumerable*" hace válidas las situaciones 6 y 8.

No hemos visto ningún resultado que haga no válida la situación 9.

# El problema de la parada

68. Mediante las reducciones que sean apropiadas, demuestre que el siguiente problema es irresoluble:

> Para una máquina de Turing arbitraria $M$, con alfabeto de cinta $\Gamma$, y dado un símbolo $a \in \Gamma$, si $M$ comienza con la cinta en blanco, ¿escribirá $M$ el símbolo $a$ en la cinta alguna vez?

Solución:

Sea $M$ una instancia del problema de la cinta en blanco.

A partir de $M$, creamos una máquina $M'$ como sigue: para cada par $(q, \sigma)$ para el cual $\delta(q, \sigma)$ no está definido, añadimos la transición $\delta(q, \sigma) = (p, a, R)$, donde $p$ es un nuevo estado.

Por tanto, si $M$ para con la cinta en blanco, $M'$ escribe el símbolo $a$ y también para.

Como no se puede "decidir" si $M$ parará, tampoco se puede "decidir" si $M'$ escribirá el símbolo $a$ en algún momento.

Recordemos que el principio de reducibilidad aplicado aquí ha sido el siguiente: si $P$ es no decidible y $P$ es reducible a $P'$, entonces $P'$ es también no decidible (si $P'$ fuera decidible, $P$ también lo sería a través de la composición del algortimo de reducción y del algoritmo que resuelve $P'$).

En este caso, $P$ es el problema de la cinta en blanco y $P'$ es el problema de la escritura de un símbolo.

69. Dadas dos máquinas de Turing $M_1$ y $M_2$ sobre el mismo alfabeto $\Sigma$, considere el problema de determinar si existe alguna cadena $w \in \Sigma^*$ tal que tanto $M_1$ como $M_2$ paran al procesar $w$. Indique si este problema es resoluble o no y por qué.

    Solución:

    Se podría construir una máquina $M$ tal que $L(M) = L(M_1) \cap L(M_2)$. La máquina $M$ tendría dos cintas: una para simular a $M_1$ sobre ella, y la otra para simular a $M_2$ sobre ella. La entrada $w$ se copia en ambas cintas. De esta forma, si $M_1$ o $M_2$ no paran con $w$, $M$ tampoco lo hará. Por tanto, $w \in L(M)$ sólo cuando $M_1$ y $M_2$ paren ambas con $w$ en un estado de aceptación.

    Pues bien, saber si existe una cadena $w$ capaz de hacer que $M$ pare supondría ir probando $M$ sobre todas las posibles cadenas $w$. Pero, para cada una de las cadenas $w$ individuales, el problema de la parada es irresoluble.

# El problema de correspondencia de Post

70. Demuestre que el problema de correspondencia de Post sobre alfabetos de un solo símbolo (por ejemplo, $\Sigma = \{a\}$) es resoluble.

    Solución:

    En este caso, existen fichas de tres tipos:

    - Positivas. Por ejemplo: $\frac{aaaaa}{aa} \equiv 3$.
    - Negativas. Por ejemplo: $\frac{a}{aaa} \equiv -2$.
    - Cero. Por ejemplo: $\frac{aa}{aa} \equiv 0$.

    Y por tanto el algoritmo sería como sigue:

    - Si existe alguna ficha "cero", esa ficha ya constituye una solución.
    - Si todas las fichas son "positivas", entonces no hay solución.
    - Si todas las fichas son "negativas", entonces tampoco hay solución.
    - Si hay al menos una ficha positiva (por ejemplo, $\frac{aaaaa}{aa} \equiv 3$) y una negativa (por ejemplo, $\frac{a}{aaa} \equiv -2$), entonces hay soluciones, las cuales vienen dadas por la ecuación:

    $$3x - 2y = 0$$

Es decir, cogiendo por ejemplo $x = 2$ veces la pieza 3 e $y = 3$ veces la pieza 2, obtendremos el mismo número de $a$es arriba y abajo:

$$\begin{array}{ccccccccccccc} \frac{aaaaa}{aa} & + & \frac{aaaaa}{aa} & + & \frac{a}{aaa} & + & \frac{a}{aaa} & + & \frac{a}{aaa} & \equiv & \frac{a^{13}}{a^{13}} \\ 3 & + & 3 & + & (-2) & + & (-2) & + & (-2) & = & 0 \end{array}$$