Práctico 9

Teoría de Lenguajes

Los objetivos de este práctico son que el/la estudiante

- comprenda y pueda clasificar un lenguaje en la **Jerarquía de Chomsky**.
- pueda construir **gramáticas irrestrictas** que generen lenguajes recursivamente enumerables;
- pueda construir **máquinas de Turing** que **reconozcan** lenguajes recursivamente enumerables;
- pueda construir **máquinas de Turing** que **computen** funciones.

Ejercicios fundamentales

Ejercicio 1

Parte A

Defina la **Jerarquía de Chomsky** para lenguajes formales.

Parte B

Utilizando la **Jerarquía de Chomsky** junto con ejemplos[1] y propiedades de los diferentes tipos de lenguajes presentes en ella, determine si las siguientes afirmaciones son **verdaderas** o **falsas**.

Si L_1 es un lenguaje regular, L_2 es libre de contexto, L_3 es finito y no vacío, L_4 es recursivamente enumerable y L_5 es libre de contexto y no regular, entonces:

1. $L_3 = L_2 \cap L_a \implies L_a$ es libre de contexto.
2. $L_b = L_3 \cap L_4 \implies L_b$ es libre de contexto.
3. $L_c = L_2 - L_1 \implies L_c$ es libre de contexto.
4. $L_d = L_4 \cap L_5 \implies L_d$ es libre de contexto.

Ejercicio 2

Cree **gramáticas irrestrictas** que generen los lenguajes:

1. $L_1 = \{a^n b^{2n} c^n : n \geq 1\}$
2. $L_2 = \{a^n b^n c^j d^n : j \geq 1 \wedge n \geq 1\}$
3. $L_3 = \{w\#w : w \in L((0|1)(0|1)^*)\}$
4. $L_4 = \{a^n b^k : k = 3^n \wedge n \geq 0\}$
5. $L_5 = \{a^i b^j a^k : j = \max(i, k) \wedge j \geq 1\}$
6. $L_6 = \{a^n b^n c^i : i \neq n \wedge (i \geq 1 \vee n \geq 1)\}$
7. $L_7 = \{a^n b^k : n \geq 1 \wedge k = n^2\}$

---

[1] Siempre asuma demostrado que $\{0^k : k \geq 0\}$ es regular, $\{0^k 1^k : k \geq 0\}$ es libre de contexto pero no regular, y que $\{0^k 1^k 2^k : k \geq 0\}$ es recursivamente enumerable pero no libre de contexto.

Ejercicio 3

Cree **máquinas de Turing** que reconozcan los lenguajes del ejercicio anterior.

Ejercicio 4

En este ejercicio practicará el uso de **máquinas de Turing como computadoras de funciones**. Tenga en cuenta que, al finalizar el cómputo de la función, la máquina se detiene en su único estado final [2] y la cinta debe tener el resultado esperado rodeado de infinitos *blancos* (ƀ) a cada lado.

Parte A

Construya **máquinas de Turing** que computen las siguientes funciones sobre $\Sigma = \{0, 1\}$:

1. **Entrada**: $w \in \Sigma^* : |w| \geq 1$
   Salida: w^r
2. **Entrada**: $w \in \Sigma^* : |w| \geq 1$
   Salida: $NOT\ w$

   Se considera la operación NOT bit a bit. El resultado no debe tener ceros no significativos.

Parte B

Considere una cinta de la forma $v\#w$ donde v y w son naturales en *unario* (e.g. el número 3 se representa como 111 y el 5 como 11111). Construya **máquinas de Turing** que computen las siguientes funciones, expresando la salida también en *unario*:

1. La suma de v y w.
2. La resta de v y w.
3. El producto de v y w.
4. El módulo entre v y w.

Ejercicios complementarios

Ejercicio 5

Cree una gramática irrestricta que genere el lenguaje $L = \{b_i\#b_{i+1} : i \geq 1 \wedge b_i \text{ es el entero } i \text{ en binario}\}$. Construya también un autómata que lo acepte.

Ejercicio 6

Considere la parte B del ejercicio 4, pero con la modificación de que ahora v y w están representadas en *binario*. Construya las **máquinas de Turing** que computen cada una de las cuatro funciones anteriores.

[2] Recuerde que las Máquinas de Turing finalizan su ejecución al entrar en el estado final, por lo que de él **no pueden salir transiciones**. Esto no era así en otros formalismos de autómatas vistos en el curso, donde los estados finales podían tener transiciones salientes.

Ejercicio 7

Considere un juego compuesto por una cinta infinita donde cada celda contiene la letra a o b ($\Sigma = \{a, b\}$). **Se asegura** que en la cinta infinita existe alguna subtira $w \in \Sigma^*$ que comienza y termina con una letra b y que está rodeada de tiras infinitas de símbolos a. Por ejemplo, para $w = baabbbaab$, la cinta infinita tendría la forma[3]:

$$...aaa...aaa...aabaabbbaabaa...aaa...aaa...$$

Teniendo en cuenta las siguientes reglas del juego

- Los movimientos posibles son
  - convertir bba en aab
  - convertir abb en baa
- Una **configuración inicial** es la ya mencionada subtira w, que comienza y termina con letras b.

  Una configuración inicial es **ganadora** si existe una secuencia finita de movimientos que lleve a una **configuración final** ganadora.
- Una **configuración final** es cualquier configuración en la cual no sea aplicable ninguna de las dos reglas de movimiento.

  Una configuración final es **ganadora** si la cinta presenta una única b rodeada de letras a. Es decir, si la cinta infinita tiene la forma $...aaa...aaa...abaa...aaa...$.

1. Construya una **gramática** que genere las configuraciones **iniciales** ganadoras.
2. Construya una **máquina de Turing** que, a partir de una configuración inicial, realice una de las secuencias de jugadas posibles hasta llegar a una configuración **final**. La máquina debe parar siempre, diferenciando los estados finales correspondientes a **juego ganado** y **juego perdido**. Considere que la cinta viene con el formato $...C\ Configuracion\ inicial\ F...$.
3. Demuestre por inducción completa que la expresión regular $b(ba)^*(ab)^*b$ genera un conjunto de configuraciones **iniciales** ganadoras.

[3]Note que, en la cinta infinita, la tira w comienza con la letra b de más a la izquierda y la b de más a la derecha. Esto implica que la tira w pueda contener, a su vez, subtiras que comiencen y terminen con letras b.