Teoría de la Computación
Licenciatura en Informática con Orientación en Desarrollo de Software
Universidad Nacional de Quilmes

Práctica 2
Lenguajes decidibles y semi-decidibles

Ejercicio 1. Considerar el siguiente lenguaje sobre el alfabeto $\Sigma = \{a, 0, 1\}$:

$$\mathsf{Bin} = \{w\, a^n \mid w \in \{0,1\}^* \text{ es la codificación en binario de } n \in \mathbb{N}\}$$

Demostrar que Bin es decidible.

Ejercicio 2. Sea $L \subseteq \Sigma^*$ un lenguaje finito. Demostrar que L es decidible.

Ejercicio 3. Sea $L \subseteq \Sigma^*$ un lenguaje decidible. Demostrar que L^r es decidible. Recordemos que L^r es el reverso de L, es decir, $w \in L^r$ si y sólo si existe una palabra $v \in L$ tal que $w = v^r$.

Ejercicio 4. Demostrar las siguientes afirmaciones:

1. Todos los lenguajes regulares son decidibles.
2. Existen lenguajes decidibles que no son regulares.

Ejercicio 5. Decidir si son verdaderas o falsas y demostrar (o dar un contraejemplo):

1. Si $L_1 \subseteq \Sigma^*$ y $L_2 \subseteq \Sigma^*$ son decidibles, entonces $L_1 \setminus L_2$ es decidible.
2. Si $L_1 \subseteq \Sigma^*$ y $L_2 \subseteq \Sigma^*$ son semi-decidibles, entonces $L_1 \setminus L_2$ es semi-decidible.

Ejercicio 6. Usando el método de diagonalización, demostrar que el siguiente lenguaje es indecidible:

$$\text{REV} = \{\langle M, w\rangle \mid M(w) \text{ termina dejando la palabra } w^r \text{ escrita en la cinta}\}$$

Ejercicio 7. Recordemos que dada una máquina de Turing M y una palabra $w \in \Sigma^*$, decimos que $M(w)$ *se cuelga* si la ejecución de M sobre la palabra w nunca llega a una configuración de aceptación ni de rechazo (es decir, no termina). Considerar el lenguaje:

$$\mathsf{HANG} = \{\langle M, w\rangle \mid M(w) \text{ se cuelga}\}$$

1. Demostrar que $\Sigma^* \setminus \mathsf{HANG}$ es semi-decidible.
2. Demostrar que HANG es indecidible (es decir, no es decidible).
3. Usando los dos ítems anteriores, demostrar que HANG no es semi-decidible.

Dos variantes de la noción de máquina de Turing

Recordemos que una máquina de Turing es una 7-upla $(\Sigma, \Gamma, Q, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ donde la función de transición es de tipo $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$.

Ejercicio 8. Consideremos la siguiente noción alternativa. Una **máquina de Turing morosa** es una 7-upla: $(\Sigma, \Gamma, Q, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ igual que antes, pero donde la función de transición es $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$, donde S representa la posibilidad de dejar el cabezal quieto en el lugar, sin moverlo a la izquierda ni a la derecha[1]. Las nociones de aceptación, rechazo, lenguaje decidible, semi-decidible, etc. se adaptan a esta nueva noción de la manera esperable.

Dado un lenguaje $L \subseteq \Sigma^*$, demostrar que existe una *máquina de Turing* que decide L si y sólo si existe una *máquina de Turing morosa* que decide L.

Ejercicio 9. Consideremos otra noción alternativa. Una **máquina de Turing derechista** es una 7-upla: $(\Sigma, \Gamma, Q, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ igual que antes, pero donde la función de transición es $\delta : Q \times \Gamma \to Q \times \Gamma \times \{S, R\}$. Igual que en el ejercicio anterior, S representa la posibilidad de quedarse quieto. Observar que en una máquina de Turing derechista el cabezal no puede moverse hacia la izquierda.

Demostrar que si existe una *máquina de Turing derechista* que decide L, entonces L es regular.

[1] La letra S proviene del inglés *stay put*, es decir, quedarse quieto.