Ejercicio 1. Sea $\Sigma = \{0, 1\}$ un alfabeto y sea G la siguiente gramática regular:

$$
\begin{aligned}
G : S &\to 1A|0 \\
A &\to 0A|1A|\epsilon
\end{aligned}
$$

a) Hallar α_1, $\alpha_2 \in \Sigma^*$ tal que $\alpha_1, \alpha_2 \in L(G)$ y dar su derivación correspondiente en G.

b) Hallar α_3, $\alpha_4 \in \Sigma^*$ tal que $\alpha_3, \alpha_4 \notin L(G)$.

c) Definir $L(G)$ dando una expresión regular que lo denote. Que conjunto es coloquialmente hablando?

a) Deriv:

Para $\alpha_1 = 1$

$$S \underset{G}{\Rightarrow} 1A \Rightarrow 1\varepsilon = 1$$

Para $\alpha_2 = 10$

$$S \Rightarrow 1A \Rightarrow 10A \Rightarrow 10\varepsilon = 10$$

b)

Gramática:

1. $S \to 1A \mid 0$: Las cadenas pueden comenzar con 1 y continuar con cualquier derivación de A, o simplemente ser 0.
2. $A \to 0A \mid 1A \mid \varepsilon$: Esto permite generar cualquier combinación arbitraria de 0 y 1 después de un 1.

Restricción:

Las únicas cadenas que no pueden generarse son aquellas que:

- Comienzan con 0 pero tienen símbolos adicionales (porque 0 no puede ser seguido por nada más).
- No siguen la estructura de comenzar con 1 seguido por 0 o 1 (o la cadena vacía).

c)

Parte (c): Definir $L(G)$ mediante una expresión regular

Analizando las producciones:

1. La producción $S \to 0$ genera la cadena $\{0\}$.
2. La producción $S \to 1A$ genera cadenas que comienzan con 1, seguidas de cualquier combinación de 0 y 1, porque A puede derivar ε, $0A$, o $1A$.

Por lo tanto:

$$L(G) = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}.$$

Expresión regular equivalente:

$$L(G) = 0 \mid 1(0 \mid 1)^*.$$

Descripción coloquial: El lenguaje $L(G)$ contiene todas las cadenas binarias que:

- Son 0, o
- Comienzan con 1 y pueden ser seguidas por cualquier combinación de 0 y 1 (incluyendo la cadena vacía después del 1).

Ejercicio 2. Sea $\Sigma = \{0, 1, 2, 3, \ldots, 9\}$ un alfabeto, probar que los siguientes lenguajes son regulares dando una gramática regular que los genere:

a) $L_{\mathbb{N}} = \{0, 1, 2, 3, \ldots\}$ (el lenguaje de los números naturales)

b) $L_{\mathbb{Z}} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ (el lenguaje de los números enteros)

c) L_{float} el lenguaje de los números flotantes.

a)

Sea $G = (N, \Sigma, P, S)$, donde:

- $N = \{S\}$: Un único no terminal.
- $\Sigma = \{0, 1, 2, \ldots, 9\}$: Alfabeto de dígitos.
- P: Producciones, definidas como:

$$S \to \text{dígitos} \mid \text{dígitos } S$$

Específicamente:

$$S \to 0 \mid 1 \mid 2 \mid \ldots \mid 9 \mid 0S \mid 1S \mid \ldots \mid 9S$$

- S: Símbolo inicial.

Explicación: Esta gramática genera cualquier número natural permitiendo producir una cadena de uno o más dígitos.

b)

Gramática Regular:

Sea $G = (N, \Sigma, P, S)$, donde:

- $N = \{S, A\}$: Con S como símbolo inicial y A como un auxiliar.
- $\Sigma = \{0, 1, 2, \ldots, 9, -\}$: Alfabeto extendido con el símbolo de signo negativo ($-$).
- P: Producciones, definidas como:

$$S \to 0 \mid A \mid -A$$

$$A \to 1 \mid 2 \mid \ldots \mid 9 \mid 1A \mid 2A \mid \ldots \mid 9A$$

Explicación:

- S permite generar 0, cualquier número positivo (A), o cualquier número negativo ($-A$).
- A genera cualquier número natural diferente de 0.

c)

Sea $G = (N, \Sigma, P, S)$, donde:

- $N = \{S, A, B\}$: Con S como símbolo inicial y A, B auxiliares.
- $\Sigma = \{0, 1, 2, \ldots, 9, -, .\}$: Alfabeto extendido con el signo negativo ($-$) y el punto decimal (.).
- P: Producciones, definidas como:

$$S \to A.B \mid -A.B$$

$$A \to 0 \mid 1 \mid 2 \mid \ldots \mid 9 \mid 0A \mid 1A \mid \ldots \mid 9A$$

$$B \to 0 \mid 1 \mid 2 \mid \ldots \mid 9 \mid 0B \mid 1B \mid \ldots \mid 9B$$

Explicación:

- S genera números flotantes positivos ($A.B$) o negativos ($-A.B$).
- A genera la parte entera del número (una secuencia de dígitos).
- B genera la parte fraccionaria del número (una secuencia de dígitos).

Ejercicio 3. Sea $\Sigma = \{a, b\}$ un alfabeto, probar que $L = \{\alpha : |\alpha| \; es \; impar\} \in LR^{\Sigma}$ dando una gramática regular G tal que $L(G) = L$, y luego, probar que efectivamente se cumple dicha igualdad mediante inducción.

Paso 1: Construcción de la gramática regular

La gramática regular $G = (N, \Sigma, P, S)$ es la siguiente:

- $N = \{S, A\}$: Con S y A como no terminales.
- $\Sigma = \{a, b\}$: Alfabeto.
- Producciones:

$$S \to aA \mid bA$$

$$A \to aS \mid bS \mid \varepsilon$$

- S es el símbolo inicial.

Explicación de la gramática:

- S genera cadenas con un número impar de símbolos. Esto se logra al alternar entre S (longitud impar) y A (longitud par).
- A genera cadenas de longitud par (siendo un estado intermedio que puede volver a S con un símbolo más).

Paso 2: Validación de $L(G) = L$

Inducción sobre la longitud de las cadenas (n)

Probaremos que:

1. S genera cadenas de longitud impar.
2. A genera cadenas de longitud par.

Base del caso ($n = 1$):

- S genera aA o bA, y luego A puede generar ε:

$$S \to aA \to a, \quad S \to bA \to b.$$

Longitud = 1 (impar).

Paso inductivo:

Supongamos que:

- S genera cadenas de longitud impar ($2k + 1$).
- A genera cadenas de longitud par ($2k$).

Demostrar:

1. S genera cadenas de longitud impar para $2k + 3$.
2. A genera cadenas de longitud par para $2k + 2$.

1. S**:** Si $S \to aA$, y A genera una cadena de longitud $2k$, entonces:

$$\text{Longitud total} = 1 (\text{de } a) + 2k = 2k + 1 \text{ (impar)}.$$

2. A**:** Si $A \to aS$, y S genera una cadena de longitud $2k + 1$, entonces:

$$\text{Longitud total} = 1 (\text{de } a) + 2k + 1 = 2k + 2 \text{ (par)}.$$

Por lo tanto, la gramática genera alternadamente cadenas de longitud impar y par, cumpliendo la definición de L.

Ejercicio 9. Para la siguiente gramática regular (de paso único), obtener su AF equivalente:

$$S \to aS|bA$$
$$A \to aA|bB$$
$$B \to aB|\epsilon$$

Paso 1: Identificar los elementos de la gramática

La gramática tiene:

- No terminales: $\{S, A, B\}$.
- Terminales: $\{a, b\}$.
- Producciones que definen cómo se generan cadenas.

Paso 2: Construcción del AF

Para convertir la gramática a un AF, seguimos estos pasos:

1. **Estados del AF:**
    - Cada no terminal de la gramática corresponde a un estado del autómata.
    - Añadimos un estado adicional llamado q_f (estado de aceptación) para manejar las derivaciones que terminan en ε.

    Estados: $Q = \{S, A, B, q_f\}$.

2. **Transiciones:**
    - Las producciones de la gramática determinan las transiciones entre los estados:
        - Si $X \to aY$, entonces hay una transición (X, a, Y) en el AF.
        - Si $X \to \varepsilon$, entonces hay una transición (X, ε, q_f) hacia el estado de aceptación.

    Transiciones:

    - $S \xrightarrow{a} S$ (de $S \to aS$).
    - $S \xrightarrow{b} A$ (de $S \to bA$).
    - $A \xrightarrow{a} A$ (de $A \to aA$).
    - $A \xrightarrow{b} B$ (de $A \to bB$).
    - $B \xrightarrow{a} B$ (de $B \to aB$).
    - $B \xrightarrow{\varepsilon} q_f$ (de $B \to \varepsilon$).

3. **Estado inicial:**
    - El estado inicial del AF es S, ya que corresponde al símbolo inicial de la gramática.

4. **Estado(s) de aceptación:**
    - q_f es el único estado de aceptación, ya que maneja las producciones que terminan en ε.

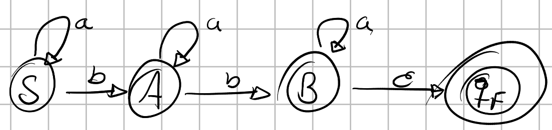
Paso 3: Representación del AF

El autómata finito equivalente es:

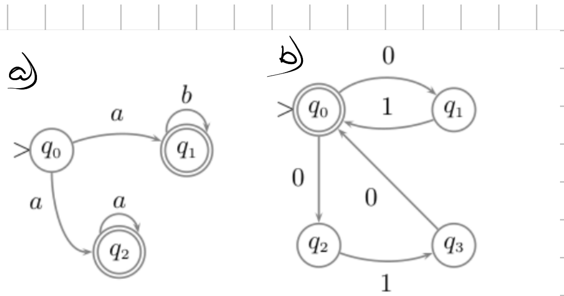
- **Estados**: $Q = \{S, A, B, q_f\}$
- **Alfabeto**: $\Sigma = \{a, b\}$
- **Transiciones**:

$$\Delta = \{(S, a, S), (S, b, A), (A, a, A), (A, b, B), (B, a, B), (B, \varepsilon, q_f)\}$$

- **Estado inicial**: S
- **Estado de aceptación**: q_f



Ejercicio 8. Para cada uno de los siguientes AF, obtener su gramática regular (de paso único) equivalente:



b)

La gramática regular es:

$$S \to 0A \mid 0B \mid \varepsilon$$
$$A \to 1S$$
$$B \to 1C$$
$$C \to 0S$$

a)

La gramática regular es:

$$S \to aA \mid aB$$
$$A \to bA \mid \varepsilon$$
$$B \to aB \mid \varepsilon$$