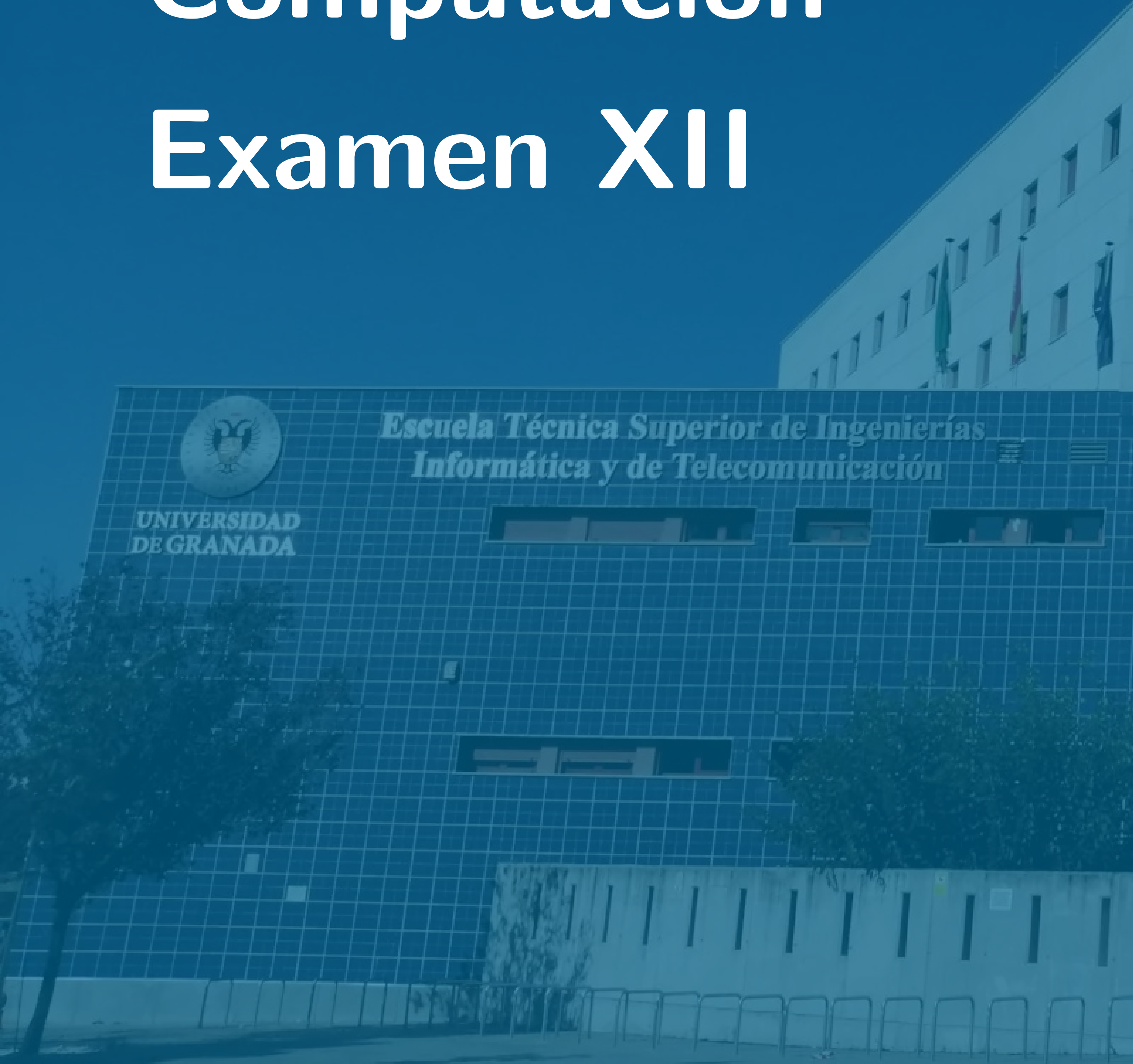
# Modelos de Computación Examen XII




**Los Del DGIIM**, `losdeldgiim.github.io`

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada

# Modelos de Computación Examen XII

Los Del DGIIM, `losdeldgiim.github.io`

Arturo Olivares Martos

Granada, 2024-2025

**Asignatura** Modelos de Computación

**Curso Académico** 2020-21.

**Grado** Doble Grado en Ingeniería Informática y Matemáticas o ADE.

**Grupo** Único.

**Descripción** Convocatoria Ordinaria.

**Fecha** 20 de enero de 2021.

**Duración** 2,5 horas.

**Ejercicio 1** (2.5 puntos)**.** Determinar cuales de los siguientes lenguajes sobre el alfabeto $\{0, 1\}$ son regulares y/o independientes del contexto. Justificar las respuestas.

1. $L_1 = \{u0^n u^{-1} \mid u \in \{0,1\}^*, n \geqslant 0\}$
2. $L_2 = \{0^i 1^j \mid 1 \leqslant i \leqslant j \leqslant 2i\}$
3. $L_3 = LL'$ donde $L$ es el conjunto de las palabras que contienen la subcadena "01" y $L'$ el conjunto de palabras que contienen la subcadena "00".
4. $L_4 = \{0^i 1^j 0^i 1^j \mid i, j \geqslant 1\}$

**Ejercicio 2** (2.5 puntos)**.** Construir un autómata con pila determinista que acepte el lenguaje $L$ sobre el alfabeto $\{0, 1\}$ dado por las palabras de longitud mayor o igual que 1 que:

- Si empiezan por 0, tienen más 0's que 1's.
- Si empiezan por 1, tienen más 1's que 0's.

A partir del autómata anterior construir un autómata con pila que acepte las palabras de $L$ que, además tengan un número par de 0's. Se valorará que se haga con el procedimiento visto en clase para la intersección de un lenguaje regular y un lenguaje independiente del contexto.

**Ejercicio 3** (1.25 puntos)**.** Comenta si existe alguna dificultad en realizar un algoritmo que lea como entrada un lenguaje cualquiera sobre el alfabeto $\{0, 1\}$ y determine si el lenguaje es finito.

**Ejercicio 4** (1.25 puntos)**.** Cuando se construye una gramática lineal por la derecha a partir de un autómata finito determinista, ¿puede ser ambigua dicha gramática? Justifica la respuesta. ¿Puede ser un lenguaje regular inherentemente ambiguo?

**Ejercicio 5** (1.25 puntos)**.** Sea $G$ una gramática independiente del contexto cualquiera, si construimos un grafo en el que los nodos son las variables y existe un enlace de la variable $A$ a la variable $B$ si existe una producción $A \to \alpha B \beta$, donde $\alpha, \beta \in (V \cup T)^*$ y dicho grafo tiene un ciclo, ¿qué conclusión importante podemos extraer sobre el lenguaje generado por la gramática?

**Ejercicio 6** (1.25 puntos)**.** ¿Puede tener un autómata finito determinista transiciones nulas? ¿puede tener un autómata con pila determinista transiciones nulas? En ambos casos, si la respuesta fuese afirmativa di bajo qué condiciones pueden existir esas transiciones nulas.

# Soluciones

**Ejercicio 1** (2.5 puntos)**.** Determinar cuales de los siguientes lenguajes sobre el alfabeto $\{0,1\}$ son regulares y/o independientes del contexto. Justificar las respuestas.

1. $L_1 = \{u0^n u^{-1} \mid u \in \{0,1\}^*, n \geqslant 0\}$

   Veamos en primer lugar que no es regular usando el recíproco del lema de bombeo. Para cada $n \in \mathbb{N}$, sea $z = 1^n01^n \in L_1$, con $|z| = 2n+1 \geqslant n$. Cualquier descomposición $z = uvw$ con $|uv| \leqslant n$ y $|v| \geqslant 1$ debe cumplir:

   $$u = 1^k \quad v = 1^l \quad w = 1^{n-k-l}01^n \quad \text{con } 0 \leqslant k+l \leqslant n \quad \text{y } l \geqslant 1$$

   Bombeando con $i = 2$ obtenemos:

   $$uv^2w = 1^k1^{2l}1^{n-k-l}01^n = 1^{n+l}01^n \notin L_1$$

   Por tanto, por el recíproco del lema de bombeo, $L_1$ no es regular. No obstante, sí es independiente del contexto, puesto que es generado por la gramática $G = (\{S, X\}, \{0,1\}, P, S)$ con $P$ dado por:

   $$\begin{aligned} S &\to 0S0 \mid 1S1 \mid X \\ X &\to 0X \mid \varepsilon \end{aligned}$$

2. $L_2 = \{0^i1^j \mid 1 \leqslant i \leqslant j \leqslant 2i\}$

   Veamos en primer lugar que no es regular usando el recíproco del lema de bombeo. Para cada $n \in \mathbb{N}$, sea $z = 0^n1^n \in L_2$, con $|z| = 2n \geqslant n$. Cualquier descomposición $z = uvw$ con $|uv| \leqslant n$ y $|v| \geqslant 1$ debe cumplir:

   $$u = 0^k \quad v = 0^l \quad w = 0^{n-k-l}1^n \quad \text{con } 0 \leqslant k+l \leqslant n \quad \text{y } l \geqslant 1$$

   Bombeando con $i = 2$ obtenemos:

   $$uv^2w = 0^k0^{2l}0^{n-k-l}1^n = 0^{n+l}1^n \notin L_2$$

   ya que $n + l > n$. Por tanto, por el recíproco del lema de bombeo, $L_2$ no es regular. No obstante, sí es independiente del contexto, puesto que es generado por la gramática $G = (\{S\}, \{0,1\}, P, S)$ con $P$ dado por:

   $$S \to 0S1 \mid 0S11 \mid 01 \mid 011$$

   En vez de dar una gramática, se podría pensar en dar un APND que lo aceptase. Este se encuentra disponible en el Ejercicio 1.5.11 de las relaciones, por lo que no se ha incluido aquí (además de por ser una solución bastante más compleja).

3. $L_3 = LL'$ donde $L$ es el conjunto de las palabras que contienen la subcadena "01" y $L'$ el conjunto de palabras que contienen la subcadena "00".

Sabemos que $L, L'$ son regulares con expresiones regulares asociadas:

$$L \equiv (0+1)^*01(0+1)^*$$
$$L' \equiv (0+1)^*00(0+1)^*$$

Por tanto, $L_3$ es regular, ya que la concatenación de dos lenguajes regulares es regular. De hecho, su expresión regular asociada es:

$$L_3 \equiv (0+1)^*01(0+1)^*00(0+1)^*$$

4. $L_4 = \{0^i1^j0^i1^j \mid i, j \geqslant 1\}$

   Demostremos que no es independiente del contexto (y por tanto tampoco regular) usando el recíproco del lema de bombeo para lenguajes independientes del contexto. Para cada $n \in \mathbb{N}$, sea $z = 0^n1^n0^n1^n \in L_4$, con $|z| = 4n \geqslant n$. Consideramos ahora cualquier descomposición $z = uvwxy$ con $|vwx| \leqslant n$ y $|vx| \geqslant 1$. Considerando cada una de las 4 partes de $z$, como $|vwx| \leqslant n$, $uvw$ estará entera contenida en una o dos de las partes, pero no pertenecerá a más de dos. Por tanto, al bombear con $i > 1$, como $|vx| \geqslant 1$, se bombeará algún caracter que provoque un desequilibrio, teniendo entonces que:

   $$uv^iwx^iy \notin L_4$$

   Por tanto, por el recíproco del lema de bombeo, $L_4$ no es independiente del contexto, y por tanto tampoco regular.

**Ejercicio 2** (2.5 puntos)**.** Construir un autómata con pila determinista que acepte el lenguaje $L$ sobre el alfabeto $\{0, 1\}$ dado por las palabras de longitud mayor o igual que 1 que:

- Si empiezan por 0, tienen más 0's que 1's.
- Si empiezan por 1, tienen más 1's que 0's.

A partir del autómata anterior construir un autómata con pila que acepte las palabras de $L$ que, además tengan un número par de 0's. Se valorará que se haga con el procedimiento visto en clase para la intersección de un lenguaje regular y un lenguaje independiente del contexto.

Para el primer apartado, consideramos los estados:

- $q$: Estado inicial. Desde él, se decidirá si la palabra empieza por 0 o por 1, y por tanto qué camino tomar.
- $q_0$: La palabra ha empezado por 0, pero $N_0(u) \leqslant N_1(u)$, por lo que no es un estado final.
- $q_1$: La palabra ha empezado por 1, pero $N_1(u) \leqslant N_0(u)$, por lo que no es un estado final.
- $q_0^f$: La palabra ha empezado por 0, y $N_0(u) > N_1(u)$, por lo que es un estado final.

- $q_1^f$: La palabra ha empezado por 1, y $N_1(u) > N_0(u)$, por lo que es un estado final.

Además, la variable 0 de la pila se usará para contar el número de 0's que llevamos, y la variable 1 para contar el número de 1's. Las respectivas variables con el subíndice $i$ se usarán para notar que son el último que provoca la diferencia, y que al leer el símbolo contrario se provocará la igualdad, pasando por tanto a un estado no final.
Por tanto, el autómata con pila que acepta $L$ por el criterio de estados finales es:

$$M = (\{q, q_0, q_1, q_0^f, q_1^f\}, \{0, 1\}, \{Z_0, 0, 1, 0_i, 1_i\}, \delta, q, Z_0, \{q_0^f, q_1^f\})$$

donde la función de transición $\delta$ viene dada por:

$$\begin{aligned}
\delta(q, 0, Z_0) &= \left\{\left(q_0^f, 0_iZ_0\right)\right\} \\
\delta(q, 1, Z_0) &= \left\{\left(q_1^f, 1_iZ_0\right)\right\} \\
\\
\delta(q_0^f, 0, 0) &= \left\{\left(q_0^f, 00\right)\right\} \\
\delta(q_0^f, 1, 0) &= \left\{\left(q_0^f, \varepsilon\right)\right\} \\
\delta(q_0^f, 0, 0_i) &= \left\{\left(q_0^f, 00_i\right)\right\} \\
\delta(q_0^f, 1, 0_i) &= \{(q_0, \varepsilon)\} \\
\delta(q_0, 0, 1) &= \{(q_0, \varepsilon)\} \\
\delta(q_0, 1, 1) &= \{(q_0, 11)\} \\
\delta(q_0, 0, Z_0) &= \left\{\left(q_0^f, 0_iZ_0\right)\right\} \\
\delta(q_0, 1, Z_0) &= \{(q_0, 1Z_0)\} \\
\\
\delta(q_1^f, 0, 1) &= \left\{\left(q_1^f, \varepsilon\right)\right\} \\
\delta(q_1^f, 1, 1) &= \left\{\left(q_1^f, 11\right)\right\} \\
\delta(q_1^f, 0, 1_i) &= \{(q_1, \varepsilon)\} \\
\delta(q_1^f, 1, 1_i) &= \left\{\left(q_1^f, 11_i\right)\right\} \\
\delta(q_1, 1, 0) &= \{(q_1, \varepsilon)\} \\
\delta(q_1, 0, 0) &= \{(q_1, 00)\} \\
\delta(q_1, 1, Z_0) &= \left\{\left(q_1^f, 1_iZ_0\right)\right\} \\
\delta(q_1, 0, Z_0) &= \{(q_1, 0Z_0)\}
\end{aligned}$$

Para el segundo apartado, consideramos en primer lugar el AFD de la Figura 1 que acepta el lenguaje $L_0$ de las palabras con un número par de 0's.

Sea $M' = (\{p_0, p_1\}, \{0, 1\}, \delta', p_0, \{p_1\})$ el AFD de la Figura 1. Para construir el autómata con pila que acepta la intersección de $L$ y $L_0$ por el criterio de estados

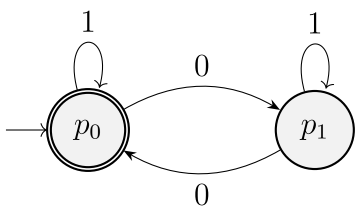



Figura 1: Autómata finito determinista que acepta el lenguaje $L_0$ de las palabras con un número par de 0's.

finales, consideramos el autómata con pila siguiente:

$$\begin{aligned}
M'' &= \left(Q \times P, \{0,1\}, B, \delta'', (q_0, p_0), Z_0, \left\{\left(q_0^f, p_0\right), \left(q_1^f, p_1\right)\right\}\right)\\
B &= \{Z_0, 0, 1, 0_i, 1_i\}\\
P &= \{p_0, p_1\}\\
Q &= \{q, q_0, q_1, q_0^f, q_1^f\}
\end{aligned}$$

donde la función de transición $\delta''$ viene dada por:

$$\delta''((q,p),a,X) = \{((q', \delta'(p,a)), \alpha) \mid \delta(q,a,X) = \{(q', \alpha)\}\} \qquad \forall (q,p) \in Q \times P, a \in \{0,1\}, X \in B$$

**Ejercicio 3** (1.25 puntos)**.** Comenta si existe alguna dificultad en realizar un algoritmo que lea como entrada un lenguaje cualquiera sobre el alfabeto $\{0,1\}$ y determine si el lenguaje es finito.

Sí, hay diversos problemas.

1. En primer lugar, el hecho de "leer" un lenguaje no es sencillo, y hay distintas alternativas. El lenguaje se puede dar mediante una expresión matemática ($0^n 1^n$, por ejemplo), se puede dar mediante una gramática, se puede dar mediante una expresión semántica ("las palabras que empiezan por 0 y acaban por 1"), etc. Además, en función del tipo de lenguaje, se podría dar de otras formas, como autómatas o una expresión regular.

2. Por otro lado, en la asignatura hemos visto algoritmos para resolver este hecho si el lenguaje es regular o independiente del contexto. No obstante, no hay un algoritmo para determinar de qué tipo es un lenguaje, por lo que no sabríamos qué opción tomar.

3. En el caso de que el lenguaje sea regular y, además, tuviésemos un reconocedor suyo, sí se podría. Distinguimos en función del reconocedor:

   - AFD: Se eliminarían los estados inalcanzables y se comprobaría si hay un ciclo. Si hay ciclos, el lenguaje es infinito; en caso contrario, es finito.
   - Expresión regular, gramática regular o AFND: Se convertiría a AFD y se aplicaría el mismo procedimiento.

   No obstante, aunque el lenguaje fuese regular, si no tuviésemos un reconocedor, no podríamos determinar si es finito o no; ya que no hay un algoritmo que, dado un lenguaje regular, te proporcione un reconocedor del mismo.

4. En el caso de que el lenguaje fuese independiente del contexto y, además, tuviésemos un reconocedor suyo, sí se podría. Distinguimos en función del reconocedor:

   - Gramática independiente del contexto: Tras eliminar producciones y símbolos inútiles y producciones unitarias y nulas, se construye un grafo en el que los nodos son las variables y existe un arco de la variable $A$ a la variable $B$ si existe una producción $A \to \alpha B \beta$, donde $\alpha, \beta \in (V \cup T)^*$. Si el grafo tiene un ciclo, el lenguaje es infinito; en caso contrario, es finito.
   - APND: Se obtiene la gramática independiente del contexto asociada y se aplica el procedimiento anterior.

   No obstante, aunque el lenguaje fuese independiente del contexto, si no tuviésemos un reconocedor, no podríamos determinar si es finito o no; ya que no hay un algoritmo que, dado un lenguaje independiente del contexto, te proporcione un reconocedor del mismo.

5. En el caso de que el lenguaje no fuese independiente del contexto, no conocemos ningún algoritmo que determine si es finito o no.

**Ejercicio 4** (1.25 puntos)**.** Cuando se construye una gramática lineal por la derecha a partir de un autómata finito determinista, ¿puede ser ambigua dicha gramática? Justifica la respuesta. ¿Puede ser un lenguaje regular inherentemente ambiguo?

No, debido al determinismo del AFD. Dado un AFD $M = (Q, A, \delta, q_0, F)$, la gramática lineal por la derecha $G = (V, T, P, S)$ asociada a $M$ es:

$$\begin{aligned} V &= Q \\ T &= A \\ S &= q_0 \\ P &= \left\{ \begin{array}{ll} q \to aq' & \text{si } \delta(q,a) = q' \\ q \to \varepsilon & \text{si } q \in F \end{array} \right. \end{aligned}$$

De esta forma, $\mathcal{L}(G) = \mathcal{L}(M)$. Sea ahora $z \in \mathcal{L}(G)$, y veamos que tan solo hay una derivación para $z$.

- Si $z$ es la palabra vacía, y como desde $q_0$ tan solo podemos, o introducir un símbolo de $A$ o aplicar $q_0 \to \varepsilon$, la única derivación es $q_0 \Rightarrow \varepsilon$.

- Si $|z| \geqslant 1$, y $z = a_1 a_2 \dots a_n$, con $a_i \in A$, entonces para generar $a_1$ desde $q_0$ podemos aplicar cualquier regla de las de la forma $q_0 \to a_1 q'$ tal que $\delta(q_0, a_1) = q'$, y por ser el autómata determinista esta transición es única, luego tan solo hay una única derivación para $a_1$. De forma análoga, para generar $a_2$ desde $q'$, tan solo hay una única derivación, y así sucesivamente hasta $a_n$. La derivación única derivación para $z$, por tanto, es:

  $$q_0 \Rightarrow a_1\delta^*(q_0, a_1) \Rightarrow a_1a_2\delta^*(q_0, a_1a_2) \Rightarrow \dots \Rightarrow a_1a_2 \dots a_n\delta^*(q_0, a_1a_2 \dots a_n) \Rightarrow a_1a_2 \dots a_n = z$$

  donde, al terminar, hacemos uso de que $\delta^*(q_0, z) = \delta^*(q_0, a_1a_2 \dots a_n) \in F$, luego podemos aplicar $\delta^*(q_0, a_1a_2 \dots a_n) \to \varepsilon$.

**Ejercicio 5** (1.25 puntos)**.** Sea $G$ una gramática independiente del contexto cualquiera, si construimos un grafo en el que los nodos son las variables y existe un enlace de la variable $A$ a la variable $B$ si existe una producción $A \to \alpha B \beta$, donde $\alpha, \beta \in (V \cup T)^*$ y dicho grafo tiene un ciclo, ¿qué conclusión importante podemos extraer sobre el lenguaje generado por la gramática?

Si la gramática no tiene símbolos ni producciones inútiles, ni producciones unitarias, ni producciones nulas, entonces el lenguaje generado por la gramática es infinito. Supongamos que el ciclo se encuentra entre las variables $A$ y $B$. Esto implica que desde $A$ se puede derivar $B$, y desde $B$ se puede derivar $A$, y al no haber producciones unitarias siempre estaremos añadiendo símbolos o variables, y al no haber producciones nulas estas variables no se pueden eliminar sin incluir un símbolo terminal. Por tanto, siempre se podrá seguir añadiendo símbolos y variables, y por tanto el lenguaje es infinito.

Notemos que es importante que la gramática no tenga símbolos ni producciones inútiles, ya que el ciclo podría estar entre variables inútiles, y por tanto no afectar al lenguaje generado por la gramática. De igual forma, es importante que no haya producciones unitarias, ya que si $A \to B$ y $B \to A$ habría un ciclo entre ellas pero no se estaría añadiendo símbolos o variables, y por tanto el lenguaje podría ser finito. Por último, es importante que no haya producciones nulas, ya que aunque haya ciclos entre variables, si se puede eliminar una variable sin añadir un símbolo terminal, el lenguaje podría ser finito.

**Ejercicio 6** (1.25 puntos)**.** ¿Puede tener un autómata finito determinista transiciones nulas? ¿puede tener un autómata con pila determinista transiciones nulas? En ambos casos, si la respuesta fuese afirmativa di bajo qué condiciones pueden existir esas transiciones nulas.

Por definición, dado un AFD $M = (Q, A, \delta, q_0, F)$, la función de transición $\delta$ es tal que $\delta : Q \times A \to Q$. Por tanto, no puede haber transiciones nulas, ya que $\varepsilon \notin A$.

Por otro lado, dado un APND $M = (Q, A, B, \delta, q_0, Z_0, F)$, la función de transición $\delta$ es una función del tipo:

$$\delta : Q \times (A \cup \{\varepsilon\}) \times B \to \mathcal{P}(Q \times B^*)$$

Este es el caso general por lo que sí puede haber transiciones nulas en un APND general. En los APD, se exige que en cada configuración, leyendo cierto símbolo de $A \cup \{\varepsilon\}$ tan solo se pueda llegar a una configuración. Es decir, se exige que:

$$|\delta(q, a, X)| \leqslant 1 \qquad \forall q \in Q, a \in A \cup \{\varepsilon\}, X \in B$$

Además, no puede haber transiciones nulas desde configuraciones que ya tienen una transición para cierto símbolo del alfabeto. Es decir, para cada $q \in Q, X \in B$ si $\delta(q, a, X) \neq \emptyset$ para algún $a \in A$, entonces $\delta(q, \varepsilon, X) = \emptyset$. Bajo estas condiciones, pueden existir transiciones nulas en un APD.