

# Test Modelos de Computación

Daniel Monjas Miguélez

20 de enero de 2021

## Índice

|                             |    |
|-----------------------------|----|
| 1. Test Relación 1          | 3  |
| 2. Test Relación 2          | 6  |
| 3. Test Relación 3          | 10 |
| 4. Test Relación 4          | 18 |
| 5. Test Relación 5          | 21 |
| 6. Test Relación 6          | 26 |
| 7. Ejemplos Pregunta Cortas | 30 |

## 1. Test Relación 1

**1.Si un lenguaje es generado por una gramática dependiente del contexto, entonces dicho lenguaje no es independiente del contexto**

**F.** Si un lenguaje es generado por una gramática dependiente del contexto puede ser independiente del contexto, pues toda gramática independiente del contexto pertenece al conjunto de las gramáticas dependientes del contexto.

**2.Los alfabetos tienen siempre un número finito de elementos, pero los lenguajes, incluso si el alfabeto tiene sólo un símbolo, tienen infinitas palabras**

**F.** Un lenguaje puede tener un número finito de palabras. Por ejemplo, sea el alfabeto  $A = \{0, 1\}$ , defino el lenguaje de las palabras con longitud menor o igual que dos, luego  $L = \{u \in \{0, 1\}^* / |u| \leq 2\} = \{\epsilon, 0, 1, 00, 01, 10, 11\}$ , el cual es finito.

**3.Si  $L$  es un lenguaje no vacío, entonces  $L^*$  es infinito.**

**F.** Supongamos que tenemos un lenguaje  $L = \{\epsilon\}$ , que por contener la palabra vacía es no vacío. Luego se tiene que  $L^* = \cup_{i \geq 0} L^i$ . Como  $L^i = \{\epsilon^i = \epsilon\}$ , entonces  $L^* = \{\epsilon\}$  y por tanto no es infinito.

**4.Todo lenguaje con un número finito de palabras es regular e independiente del contexto.**

**V.** De forma similar a lo visto en la pregunta 1 todo lenguaje regular está contenido en el conjunto de los lenguajes independientes del contexto. Además, un lenguaje con un número finito de palabras es regular, ya que se puede dividir este lenguaje en un número finito de lenguajes, para cada uno se construye un autómata finito que genere dicho lenguaje y finalmente se unen todos los autómatas obteniendo que un autómata finito acepta la unión de todos estos sublenguajes, y por consiguiente existe un autómata que acepta el lenguaje con un número finito de palabras.

**5.Si  $L$  es un lenguaje, entonces siempre  $L^*$  es distinto de  $L^+$**

**F.** Si  $\epsilon \in L \Rightarrow L^* = L^+$ .

**6. $L \cdot \emptyset = L$**

**F.**  $L \cdot \emptyset = \emptyset$ .

**7.Si  $A$  es un alfabeto, la aplicación que transforma cada palabra  $u \in A^*$  en su inversa es un homomorfismo de  $A^*$  en  $A^*$**

**F.** Sea  $A = \{0, 1\}$ , entonces  $h(01) = 10 \neq h(0)h(1) = 01$ .

**8.Si  $\epsilon \in L$ , entonces  $L^+ = L^*$**

V. Lo pone en los apuntes, y además se ha recalcado en la pregunta 5.

**9.La transformación que a cada palabra sobre  $\{0,1\}^*$  le añade 00 al principio y 11 al final es un homomorfismo**

F. Sea 01,  $h(01) = 000111 \neq h(0)h(1) = 0001100111$ .

**10.Se puede construir un programa que tenga como entrada un programa y unos datos que siempre nos diga si el programa leído termina para esos datos**

F. No existe, ya que si existiera (programa  $Stops(P, x)$ ) podríamos construir el algoritmo  $Turing(P)$  con estrada  $P$ , con  $L\ if\ Stops(P, P)\ GOTO\ L$ .

**11.La cabecera del lenguaje L siempre incluye a L**

V. Una palabra pertenece a la cabecera de un lenguaje si para dicha palabra existe un sufijo en  $A^*$  tal que la palabra con dicho sufijo esté en el lenguaje. Si tomo como palabras las palabras del lenguaje y como sufijo la palabra vacía se tiene que toda palabra del lenguaje pertenece a la cabecera.

**12.Un lenguaje no puede ser igual a su inverso.**

F. Sea  $L = \{u|u = u^{-1}\}$ , el lenguaje de los palíndromos sobre cualquier alfabeto. Claramente se tiene que  $L^{-1} = \{u^{-1}|u^{-1} = u\} = L$  por tanto el lenguaje y su inverso son iguales.

**13.La aplicación que transforma cada u sobre el alfabeto  $\{0,1\}^*$  en  $u^3$  es un homomorfismo.**

F. Sea 01 una palabra sobre el alfabeto  $A = \{0,1\}$ . Se tiene que  $h(01) = (01)^3 = 010101 \neq h(0)h(1) = 000111$ .

**14.El lenguaje que contiene sólo la palabra vacía es el elemento neutro para la concatenación de lenguajes**

V. Justamente es la definición de elemento neutro para la concatenación de lenguajes

**15.Si L es un lenguaje, en algunas ocasiones se tiene que  $L^* = L^+$**

V. Es más la única ocasión en la que esto es cierto es aquella en la que  $\epsilon \in L$ .

**16.Hay lenguajes con un número infinito de palabras que no son regulares**

V. Por ejemplo, sobre el alfabeto  $A = \{0,1\}$ , el lenguaje de las palabras cuyo número de 0 es igual al número de 1. Claramente este lenguaje tiene infinitas palabras y no verifica el lema del Bombeo. Pues si bombeas para la palabra  $0^n 1^n$  que pertenece al lenguaje esta no pertenecerá.

**17. Si un lenguaje tiene un conjunto infinito de palabras sabemos que no es regular.**

F. Sea el alfabeto  $A = \{0, 1\}$ , defino el lenguaje de las cadenas de 0 de cualquier longitud. Este lenguaje sería aceptado por un autómata finito determinista y por tanto el lenguaje es regular.

**18. Si  $L$  es un lenguaje finito, entonces su cabecera ( $CAB(L)$ ) también será finita.**

V. A parte de las palabras del lenguaje se añaden por cada palabra tantas palabras como letras tenga, suponiendo que la longitud de una palabra es finita y así llegamos a que por cada palabra se añaden un número finito de palabras como el número de palabras es finito entonces se tiene que a un lenguaje finito se le añaden un número finito de palabras teniendo un lenguaje finito.

**19. El conjunto de palabras sobre un alfabeto dado con la operación de concatenación tiene una estructura de monoide**

V.

**20. La transformación entre el conjunto de palabras del alfabeto  $\{0, 1\}$  que duplica cada símbolo (la palabra 011 se transforma en 001111) es un homomorfismo.**

V. Sea una palabra cualquiera  $a_1 \dots a_n$ , se tiene que  $h(a_1 \dots a_n) = a_1^2 \dots a_n^2 = h(a_1) \dots h(a_n) = a_1^2 \dots a_n^2$ , luego es un homomorfismo.

**21. Si  $f$  es un homomorfismo entre palabras del alfabeto  $A_1$  en palabras del alfabeto  $A_2$ , entonces si conocemos  $f(a)$  para cada  $a \in A_1$  se puede calcular  $f(u)$  para cada palabra  $u \in A_1^*$ .**

V. Si  $u = a_1 \dots a_n \in A_1^*$  se tiene que  $h(u) = h(a_1 \dots a_n) = h(a_1) \dots h(a_n)$ , donde conocemos el valor  $h(a_i)$  para cada  $i \in \{1, \dots, n\}$ .

**22. Si  $A$  es un alfabeto, la aplicación que transforma cada palabra  $u \in A^*$  en su inversa es un homomorfismo de  $A^*$  en  $A^*$ .**

F. Se repite, es la 7.

**23. Si  $\epsilon \in L$ , entonces  $L^+ = L^*$**

V. Se repite es la 8.

**24. Si  $f$  es un homomorfismo, entonces necesariamente se verifica  $f(\epsilon) = \epsilon$**

V. Un homomorfismo debe cumplir siempre la propiedad  $h(uv) = h(u)h(v)$ . Si  $h(\epsilon) \neq \epsilon$  se tiene entonces que (sea  $u = \epsilon$ )  $h(v) = h(\epsilon)h(v)$ , con  $h(\epsilon) \neq \epsilon$  por hipótesis llegando así a una contradicción.

**25. Si  $A$  es un alfabeto, entonces  $A^+$  no incluye la palabra vacía.**  
**V.** Por definición de  $A^+$ .

**26. Es posible diseñar un algoritmo que lea un lenguaje cualquiera sobre el alfabeto  $\{0, 1\}$  y nos diga si es regular o no.**

**F.** No existe algoritmo.

## 2. Test Relación 2

**1. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $(rs)^* = r^*s^*$**

**F.**  $(rs)^*$  puede ser igual por ejemplo a  $rsrsrs$ .

**2. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica  $(r + s)^* = r^* + s^*$**

**F.**  $(r + s)^*$  puede generarse por ejemplo  $rsr$  lo cual no puede ser generado por  $r^* + s^*$ .

**3. Si  $r$  y  $s$  son expresiones regulares, tales que su lenguaje asociado contiene la palabra vacía, entonces  $(r_1r_2)^* = (r_2r_1)^*$ .**

**V.** Se tiene que  $(r_1r_2)^* = (r_2r_1)^*$ , pues el primero genera expresiones de la forma  $r_1r_2r_1r_2 \dots$  y el segundo de la forma  $r_2r_1r_2r_1 \dots$ , los cuales se pueden igualar sustituyendo o bien  $r_1$  o bien  $r_2$  por la palabra vacía.

**4. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $(r + \epsilon)^+ = r^*$**

**V.** Pues  $(r + \epsilon)^+$ , genera obligatoriamente o una palabra vacía o una cadena de  $r$ , que es justo lo que genera  $r^*$ .

**5. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica  $r(r + s)^* = (r + s)^*r$ .**

**F.** Las palabras generadas por  $r.(r + s)^*$  obligatoriamente empiezan por  $r$  y luego pueden terminar o bien por  $s$  o bien por  $r$ . Por otro lado las palabras generadas por  $(r + s)^*r$  obligatoriamente terminan por  $r$  pero pueden empezar por  $r$  o por  $s$ . Por tanto una palabra que empiece por  $s$  puede ser generada por el segundo pero no por el primero y análogo si termina por  $s$ .

**6. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces  $r_1^*r_2^* \subseteq (r_1, r_2)^*$ , en el sentido de que los lenguajes asociados están incluidos.**

**F.** Ya hemos dicho que  $r_1^*r_2^*$  genera producciones de la forma  $r_1r_2 \dots$ , lo cual no significa que las producciones de la forma  $r_1^*r_2^*$  estén incluidas.

**7. Si  $r_1, r_2$  y  $r_3$  son expresiones regulares, entonces  $(r_1 + r_2)^* r_3 = r_1^* r_3 + r_2^* r_3$ .**

**F.**  $(r_1 + r_2)^* r_3$  puede generar por ejemplo  $r_1 r_2 r_3$ , mientras que  $r_1^* r_3 + r_2^* r_3$  no puede.

**8. Si  $r_1$  y  $r_2$  son expresiones regulares entonces:  $(r_1^* r_2^*) = (r_1 + r_2)^*$ .**

**V.** Pues con ambas se puede generar cualquier sucesión de  $r_1$  y  $r_2$ .

**9. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces  $(r_1.r_2)^* = (r_1 + r_2)^*$ .**

**F.** Ya hemos visto que  $(r_1 + r_2)^*$  contiene trivialmente a  $(r_1.r_2)^*$ , pues genera todas las palabras generadas por  $r_1$  y  $r_2$ . Sin embargo,  $(r_1.r_2)^*$  no puede generar expresiones del tipo  $r_1 r_1 \dots$  si  $r_2$  no contiene a  $\epsilon$  y análogo para  $r_1$ , luego no son iguales a menos que ambos contengan la palabra vacía.

**10. Si  $r$  es una expresión regular, entonces  $r^* r^* = r^*$**

**V.** Como  $r^*$  solo puede generar cadenas de la forma  $r^i$ ,  $i \geq 0$  se tiene que es trivial.

**11. Si  $r$  es una expresión regular, entonces  $r\emptyset = r + \emptyset$**

**F.** La primera parte de la expresión es igual a  $\emptyset$ , mientras que la segunda parte puede ser igual a  $r$ .

**12. Si  $r$  es una expresión regular, entonces se verifica que  $r^* \epsilon = r^+ \epsilon$**

**F.** Pues la segunda parte de la igualdad siempre va a ser distinta de la palabra vacía siempre y cuando  $r$  no pueda generar la palabra vacía, mientras que la primera parte si puede ser directamente la palabra vacía.

**13. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces siempre  $r_1(r_2 r_1)^* = (r_1 r_2)^* r_1$ .**

**V.** Una expresión regular apareció en el examen. Todas las cadenas generadas por la primera parte de la expresión tienen la forma  $r_1 r_2 r_1 r_2 r_1 \dots$ , cumpliéndose que debe empezar y terminar por  $r_1$  y que la longitud sea mayor o igual que 1, que es exactamente lo mismo que la segunda parte de la igualdad.

**14. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces siempre se verifica que  $r_1(r_2 r_1)^* = (r_1 r_2)^* r_1$**

**V.** Diría que es lo mismo que la anterior.

**15. Si  $r$  y  $s$  son expresiones regulares, entonces  $(r^* s^*)^* = (r + s)^*$**

**V.** Es la pregunta 8, renombrando a las expresiones regulares.

**16. Si  $r$  es una expresión regular, entonces  $(rr)^* \subseteq r^*$**

V. La expresión  $r^*$  genera sucesiones de cualquier longitud de palabras generadas por  $r$ , incluyendo aquellas sucesiones de longitud para. Si además  $r$  genera  $\epsilon$  se tendría la igualdad.

**17. Si  $r_1$  y  $r_2$  son expresiones regulares, tales que su lenguaje asociado contiene la palabra vacía, entonces  $(r_1 r_2)^* = (r_1 + r_2)^*$ .**

V. Ya se vió en la pregunta 9.

**18. Si  $r_1, r_2$  y  $r_3$  son expresiones regulares, entonces  $r_1(r_2^* + r_3^*) = r_1 r_2^* + r_1 r_3^*$**

V. Se aplica la propiedad distributiva.

**19. La demostración de que la clase de lenguajes aceptados por los autómatas no deterministas es la misma que la aceptada por los autómatas deterministas, se basa en dado un autómata no determinista construir uno determinista que, ante una palabra de entrada, explore todas las posibles opciones que puede seguir el no determinista.**

V. Cuando convertimos un autómata no determinista en uno determinista lo que se hace es partiendo del estado inicial le vamos pasando los posibles símbolos creando así nuevos estados hasta que finalmente de cada estado salen tantas flechas como elementos tiene el alfabeto.

**20. Un autómata finito puede ser determinista y no-determinista a la vez.**

F/V. De hecho las definiciones son excluyentes, o es uno o es otro, si consideramos que un no determinista fuerza a que al menos uno de las transiciones sea no determinista. En los apuntes para demostrar la equivalencia en la aceptación de lenguajes se asume que un determinista es no determinista también.

**21. Para transformar un autómata que acepta el lenguaje  $L$  en uno que acepte  $L^*$ , basta unir los estados finales con el inicial mediante transiciones nulas.**

F. Si hiciésemos sólo esto y el estado inicial no fuese también final, la palabra vacía no estaría incluida con lo que no generaría  $L^*$ .

**22. Para pasar de un autómata que acepte el lenguaje asociado a  $r$  a uno que acepte  $r^*$  basta con unir con transiciones nulas sus estados finales con el estado inicial.**

F. Ocurre lo mismo que en el caso anterior, no solo se unen estados finales con el estado inicial, sino que además se debe poner como final el estado



inicial.

**23.Existe un lenguaje reconocido por un AFD y no generado por una gramática independiente del contexto.**

F. Si un lenguaje es reconocido por un AFD entonces es regular, y si es regular es también independiente del contexto por la jerarquía de Chomsky.

**24.Existen lenguajes aceptados por AFD que no pueden ser aceptados por AF no determinísticos.**

F. Todo lenguaje aceptado por un AFD es también aceptado por un AFND.

**25.La clausura de un lenguaje aceptado por un AFD puede ser representado por una expresión regular.**

V. Sabemos que todo lenguaje aceptado por un AFD es regular, y por tanto, es equivalente a una expresión regular que llamaremos  $r$ . Entonces se hace la clausura de  $r$  y se tiene la expresión regular del lenguaje aceptado por el AFD.

**26.Un lenguaje representado por una expresión regular siempre puede ser reconocido por un AF no determinista.**

V. Si un lenguaje es aceptado por una expresión regular, entonces es reconocido por un AFD. Y si es aceptado por un AFD entonces existe un AFND que acepta ese lenguaje.

**27.Todo lenguaje regular puede ser generado por una gramática libre de contexto**

V. Por la jerarquía de Chomsky.

**28.Un lenguaje con un número finito de palabras siempre puede ser reconocido por un AFND.**

V. Un lenguaje con un número finito de palabras es regular, luego es aceptado por un AFD y por tanto también será aceptado por un AFND.

**29.Todo autómata finito determinista de  $n$  estados, cuyo alfabeto  $A$  contiene  $m$  símbolos debe tener  $m \cdot n$  transiciones.**

V. Un AFD debe tener en cada estado una transición por cada elemento del alfabeto. Como el alfabeto tiene  $m$  elementos y se tienen  $n$  estados esto se plasma en  $n \cdot m$  transiciones.

**30.Para que un autómata con pila sea determinista es necesario que no tenga transiciones nulas.**

**F.** Lo que se requiere es que no tenga transiciones no deterministas, es decir, que si en un estado  $q_0$ , con estado de la pila  $X$  sólo se pueda hacer una transición.

**31. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces se verifica que  $(r_1 + r_2)^* = (r_1^* r_2)^* r_1^*$**

**V.** La primera expresión puede generar cualquier concatenación de palabras de  $r_1$  y  $r_2$ . Por su parte la segunda expresión también puede hacerlo.

**32. Si un lenguaje es infinito no se puede encontrar expresión regular que lo genera.**

**F.** Por ejemplo con el alfabeto  $A = \{0, 1\}$ , se considera el lenguaje de las cadenas de 0 de cualquier longitud, el cual es un lenguaje infinito, y que tiene asociado como expresión regular  $0^*$ .

**33. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces se verifica que  $(r_1 + \epsilon)^+ r_2^+ = r_1^+ (r_2 + \epsilon)^+$**

**F.** La primera expresión podría generar  $r_2$  mientras que la segunda no podría, y por otro lado la segunda expresión puede generar  $r_1$  mientras que la primera expresión no puede.

**34. El conjunto de palabras sobre el alfabeto  $\{0, 1\}$  tales que eliminando los tres últimos símbolos, en la palabra resultante no aparece el patrón 0011 es un lenguaje regular.**

**V.** Ese conjunto de palabras se obtiene al unir las palabras tal que no contienen la subcadena 0011 con las palabras de longitud 3, siendo ambos lenguajes regulares y por consiguiente su unión es regular.

**35. El lenguaje formado por las cadenas sobre  $\{0, 1\}$  que tienen un número impar de 0 y un número par de 1 no es regular**

**F.** Este lenguaje es la intersección del lenguaje de las palabras con un número impar de 0 y el lenguaje de las palabras con un número par de 1, los cuales son ambos regulares, y por consiguiente su intersección es también regular

### 3. Test Relación 3

**1. El lema de bombeo puede usarse para demostrar que un lenguaje es regular**

**F.** Lo normal es utilizarlo para demostrar que un lenguaje no es regular, pues es muy difícil probar una a una todas las constantes y todas las palabras hasta que se encuentra una.

**2. Todo lenguaje con un número finito de palabras es regular**

V. Se divide este lenguaje en tantos sublenguajes como palabras tenga el lenguaje inicial, y se establece un AF para cada uno de estos lenguajes. Se hace la unión de todos ellos y finalmente tenemos un autómata que acepta el lenguaje inicial, luego es regular.

**3. La intersección de lenguajes regulares es siempre regular.**

V.

**4. La demostración del lema del bombeo se basa en que si leemos una palabra de longitud mayor o igual al número de estados del autómata, entonces en el camino que se recorre en el diagrama de transición se produce un ciclo.**

V.

**5. Es más fácil determinar si una palabra pertenece a un lenguaje regular cuando éste viene dado por una expresión regular que cuando viene dado por un autómata finito determinista.**

F. Con el autómata lo único que hay que hacer es ir siguiendo las transiciones al pasarle cada una de las letras o símbolos terminales que conforman la palabra.

**6. En la demostración de que todo autómata finito tiene expresión regular que representa el mismo lenguaje, el conjunto  $R_{ij}^k$  se define como el lenguaje de todas las palabras que llevan al autómata del estado  $q_i$  al estado  $q_j$  pasando por el estado número  $k$ ,  $q_k$**

F. El conjunto  $R_{ij}^k$  contiene las palabras que para ir de  $q_i$  a  $q_j$  no pasan por  $q_k$  y las palabras que para ir de  $q_i$  a  $q_j$  si pasan por  $q_k$ .

**7. El conjunto de todas las expresiones regulares es un lenguaje regular**

V. Cada expresión regular tiene asociado un lenguaje regular, por consiguiente el conjunto de las expresiones regulares es la unión de todas las expresiones regulares, que sería la unión de todos los lenguajes regulares y por tanto sería regular.

**8. A partir de la demostración de que si  $R$  es regular y  $L$  un lenguaje cualquiera, entonces  $R/L$  es regular, se puede obtener un algoritmo para construir el autómata asociado a  $R/L$**

V. Sea  $M = (Q, A, \delta, q_0, F)$  un autómata finito determinístico que acepta el lenguaje  $R$ . Entonces  $R/L$  es aceptado por el autómata  $M' = (Q, A, \delta, q_0, F')$  donde  $F' = \{q \in Q : \exists y \in L \text{ tal que } \delta^*(q, y) \in F\}$ .

**9. En un autómata finito no-determinista, si intercambio entre sí los estados finales y no finales obtengo un autómata que acepta el lenguaje complementario.**

F. Al hacer ese intercambio en un AFD eso es cierto, pero al ser AFND no tiene porque serlo.

**10. Si en un autómata finito no hay estados distinguibles de nivel 2, ya no puede haber estados distinguibles de nivel 4.**

F. Se puede tener que un estado del nivel 2 que sea indistinguible y sin embargo se puede tener un estado final en el nivel 4 que sea distinguible.

**11. Todo lenguaje generado por una gramática lineal por la derecha es también generado por una gramática lineal por la izquierda.**

V.

**12. Un autómata finito determinista sin estados inaccesibles ni indistinguibles es minimal.**

V.

**13. Si  $L$  es un lenguaje sobre el alfabeto  $A$ , entonces  $CAB(L)$  es siempre igual al cociente  $L/A^*$**

V. Por definición son exactamente lo mismo.

**14. El lenguaje de las palabras sobre  $\{0, 1\}$  en las que la diferencia entre el número de ceros y unos es impar es regular ¿?**

**15. En un autómata finito cualquiera, si las transiciones dan lugar a un ciclo, entonces el lenguaje aceptado es infinito.**

V. Ese ciclo se podría repetir indefinidamente, generando una palabra por cada número de repeticiones de ese ciclo.

**16. La expresión recursiva que se emplea para obtener la expresión regular asociada a un autómata finito determinista es:**

$$r_{ij}^k = r_{ij}^{k-1} + r_{i(k-1)}^{k-1} (r_{(k-1)(k-1)}^{k-1})^* r_{(k-1)j}^{k-1}$$

F. La expresión es  $r_{ij}^k = r_{ij}^{k-1} + r_{ik}^{k-1} (r_{kk}^{k-1})^* r_{kj}^{k-1}$ .

**17. Cuando se construye la expresión regular asociada a un autómata finito determinista,  $r_{ii}^0$  no puede ser nunca vacío**

V. Pues esta expresión regular contiene al menos a la palabra vacía.

**18.El conjunto de las palabras  $\{u0011v^{-1} : u, v \in \{0, 1\}^*\}$  es regular**

V. Realmente este es el conjunto de palabras que contienen al menos una vez la subcadena 0011, y es trivial que hay un AF que lo genera.

**19.Si L es un lenguaje finito, entonces su complementario es regular**

V. Si L es finito es regular y por tanto su complementario es regular.

**20.En un AFD la relación de indistinguibilidad es una relación de equivalencia.**

V.

**21. En un AFD siempre debe de existir, al menos, un estado de error.**

F. No es siempre necesario, un autómata puede tener todas sus transiciones aceptables, por tanto no es necesario un estado de error.

**22.El conjunto de los número en binario que son múltiplos de 7 es regular.**

V.

**23.Hay situaciones en las que los estados inaccesibles de un AFD cumplen una función específica.**

F. Un estado inaccesible nunca puede ser accedido pues no hay transiciones que lleguen al mismo luego no tienen función.

**24.Si R es un lenguaje regular y L es un lenguaje independiente del contexto, entonces R/L es regular.**

V. Por definición de lenguaje cociente.

**25.Si en un autómata dos estados son distinguibles de nivel n, entonces serán distinguibles de nivel m para todo  $m \geq n$ .**

V. Las parejas  $\{p, q\}$  que son distinguibles a nivel  $n+1$  son aquellas que lo son a nivel n más aquellas tales que existe un símbolo terminal tal que al añadir una transición lo siguen siendo.

**26.Si h es un homomorfismo y h(L) no es regular, podemos concluir que L no es regular.**

V. La imagen de un lenguaje regular por un homomorfismo es un lenguaje regular.

**27.El lenguaje de todas las palabras en las que los tres primeros símbolos son iguales a los tres últimos es regular.**

F. Se hizo en un examen y se usaba una gramática independiente del contexto

**28. Si un lenguaje verifica la condición que aparece en el lema de bombeo para lenguajes regulares, ya no hay forma de demostrar que no regular.**

F. Un lenguaje puede no ser regular y verificar el lema del bombeo. El lema del bombeo es una condición necesaria de los lenguajes regulares, pero no suficiente.

**29. Si  $f$  es un homomorfismo entre alfabetos  $f : A_1^* \rightarrow A_2^*$  y  $L \subseteq A_1^*$  no es regular, podemos concluir que  $f(L)$  tampoco es regular.**

V. Pues si  $f(L)$  fuese regular entonces se tendría que  $f^{-1}(f(L))$  sería regular.

**30. Todo lenguaje que cumple la condición del lema de bombeo para lenguaje regulares puede ser aceptado por un AFND.**

F. Un lenguaje no regular puede cumplir la condición del lema del bombeo, entonces se tendría que es aceptado por un AFND y por tanto también sería aceptado por un AFD, luego sería un lenguaje regular, lo cual llega a contradicción.

**31. No existe algoritmo para saber si el lenguaje generado por una gramática regular es finito.**

F. Si existe.

**32. Dos autómatas finitos deterministas con diferente número de estados y que aceptan el lenguaje vacío tienen el mismo número de estados finales.**

F. No tiene porque.

**33. Si  $A$  es un alfabeto y  $L$  un lenguaje cualquiera distinto del vacío, entonces se verifica que  $A^*/L = A^*$**

F. Como  $L$  no incluye el vacío significa que el que  $v$  siempre tendrá longitud mayor o igual que 1, luego todas las palabras de  $L$  no estarán incluidas en  $A^*/L$ , y por tanto no puede ser  $A^*$ .

**34. Si  $R_{ij}^k$  son los lenguajes que se usan en la construcción de una expresión regular a partir de un autómata finito, siempre se verifica que  $R_{ij}^{j-1} R_{jk}^{j-1} \subseteq R_{ik}^j$**

V.  $R_{ik}^j = R_{ik}^{j-1} \cup R_{ij}^{j-1} (R_{jj}^{j-1})^* R_{jk}^{j-1}$ , luego se ve que si está contenido.

**35. El lema de bombeo es útil para demostrar que la intersección**

**de dos lenguaje no es regular.**

V. El lema del bombeo siempre es útil para demostrar si un lenguaje no es regular pero no te lo puede asegurar.

**36.Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es infinito.**

V/F.Basta con usar el algoritmo para pasar una gramática a autómeta y luego el algoritmo para comprobar si el lenguaje aceptado por dicho autómeta es finito o infinito

**37.Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es finito o infinito.**

V/F. Basta con usar el algoritmo para pasar una gramática a autómeta y luego el algoritmo para comprobar si el lenguaje aceptado por dicho autómeta es finito o infinito

**38.La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.**

V. La intersección de dos lenguaje regulares da lugar a un lenguaje independiente del contexto y además regular.

**39.Si un lenguaje es infinito no se puede encontrar una expresión regular que lo represente.**

F.

**40.En un AFD sin estados inaccesibles la relación de indistinguibilidad entre los estados es una relación de equivalencia.**

V.

**41.En un AFD, si no hay dos estados que sean indistinguibles entre sí, entonces el autómeta es minimal**

F. Tampoco debe tener estados inaccesibles.

**42.Dada una gramática lineal por la derecha , siempre existe otra gramática lineal por la izquierda que acepte el mismo lenguaje.**

V.

**43.Si  $R$  es un lenguaje regular y  $L$  un lenguaje cualquiera, entonces  $R/L$  es siempre regular.**

V. Es la definición de lenguaje cociente cuando el dividendo es regular.

**44. Si un lenguaje cumple la condición del lema de bombeo para conjunto regulares no nos asegura que sea un lenguaje regular.**

V. Un lenguaje no regular puede cumplir la condición del lema del bombeo, ya que es condición necesaria pero no suficiente.

**45. Existe un algoritmo para determinar si los lenguajes generados por dos gramáticas regulares son iguales.**

V. Basta con hacer  $(L(M_1) \cap \overline{L(M_2)}) \cup (L(M_2) \cap \overline{L(M_1)})$ .

**46. El conjunto de cadenas aceptados por un AFND con transiciones nulas puede ser generado por una gramática independiente del contexto.**

V. Un AFND con transiciones nulas puede ser representado por un AFD y por consiguiente dicho conjunto es aceptado por un AFD y se trata de un lenguaje regular, por consiguiente es independiente del contexto.

**47. El lenguaje resultado de la unión de dos lenguajes regulares con un número infinito de palabras puede ser representado mediante una expresión regular.**

V. El lenguaje resultado de unir dos lenguajes regulares es un lenguaje regular y por tanto tiene una expresión regular asociada.

**48. Una expresión regular siempre representa a un lenguaje que puede ser generado por una gramática independiente del contexto.**

V. Si pues una expresión regular siempre tiene asociada un lenguaje regular que es independiente del contexto.

**49. Existe un algoritmo para comprobar si son iguales los lenguajes aceptados por dos autómatas finitos diferentes.**

V. Igual que el 45.

**50. Si en un AFND intercambio entre sí los estados finales y no finales obtengo un autómata que acepta el lenguaje complementario del aceptado por el autómata original.**

F. Igual que la 9.

**51. Si  $L$  es un lenguaje regular, entonces el lenguaje  $LL^{-1}$  es también regular**

V. Si  $L$  es regular entonces  $L^{-1}$  es regular y por tanto la concatenación de regulares es regular.

**52. El lema de bombeo para lenguajes regulares es útil para demostrar que un lenguaje determinado no es regular**

V. Es útil pero no siempre funciona, pues un lenguaje no regular puede cumplir la condición del lema del bombeo.



**53. Si un lenguaje tiene un conjunto infinito de palabras sabemos que no es regular.**

**F.**

**54. Un AFD sin estados inaccesibles ni indistinguibles es minimal.**

**V.**

**55. El conjunto de palabras  $\{u0011v^{-1} : u, v \in \{0, 1\}^*\}$  es regular.**

**V. Es la concatenación de tres lenguajes regulares.**

**56. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es infinito.**

**V.**

**57. Para cada AFND M existe una gramática independiente del contexto G tal que  $L(M) = L(G)$ .**

**V. Un lenguaje aceptado por un AFND es también aceptado por un AFD y por tanto es regular, luego es también independiente del contexto.**

**58. El lenguaje formado por las cadenas sobre  $\{0, 1\}$  que tienen un número impar de 0 y un número par de 1 no es regular**

**F. Es intersección de dos regulares.**

**59. Si L es un lenguaje regular, entonces la cabecera de L ( $CAB(L)$ ) es siempre regular**

**F. No tiene porque, si bien contiene al lenguaje L, no implica que lo sea.**

**60. En un AFD, si no hay dos estados que sean indistinguibles entre sí, entonces el autómata es minimal**

**F. No deben haber estados inaccesibles.**

**61. La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.**

**V. Es más, la intersección de regulares de un lenguaje regular.**

**62. Si un lenguaje es infinito no se puede encontrar una expresión regular que lo presente.**

**F.**

## 4. Test Relación 4

1. Si un lenguaje de tipo 2 viene generado por una gramática ambigua, siempre puedo encontrar una gramática no ambigua que genere el mismo lenguaje

F. El lenguaje puede ser inherentemente ambiguo.

2. En una gramática de tipo 2 ambigua no puede existir una palabra generada con un único árbol de derivación.

F. Para que una gramática sea ambigua basta con que al menos una palabra se genera por medio de dos árboles de derivación.

3. Dada una gramática independiente del contexto, siempre se puede construir una gramática sin transiciones nulas ni unitarias que genera el mismo lenguaje que la gramática original.

V. Existe un algoritmo para ello.

4. Una gramática independiente del contexto es ambigua si existe una palabra que puede ser generada con dos cadenas de derivación distintas.

V. Es la definición de gramática ambigua.

5. Un lenguaje inherentemente ambiguo puede ser generado por una gramática ambigua.

V. De hecho sólo puede ser generado por gramáticas ambiguas.

6. El lenguaje de las palabras sobre  $\{0, 1\}$  con un número impar de 0 es independiente del contexto.

V. Es más, es regular y por tanto es independiente del contexto.

7. Si en una producción de una gramática independiente del contexto, uno de los símbolos que contiene es útil, entonces la producción es útil

F. Deben ser útiles todos y cada uno de los símbolos.

8. Todo árbol de derivación de una palabra en una gramática independiente del contexto está asociado a una única derivación por la izquierda.

F. No tiene porqué.

9. Para poder aplicar el algoritmo que hemos visto para transformar una gramática a forma normal de Greibach, la gramática tiene que estar en forma normal de Chomsky necesariamente.

**F.** No tiene porqué estar en forma normal de Chomsky, sólo debe cumplir que sus producciones sean de la forma  $A \rightarrow a\alpha$   $a \in T$ ,  $\alpha \in V^*$  y  $A \rightarrow \alpha$ ,  $\alpha \in V^*$ ,  $|\alpha| \geq 2$ , que no es forma normal de Chomsky.

**10.Sólo hay una derivación por la derecha asociada a un árbol de derivación.**

**F.** No tiene porqué.

**11.Si una gramática independiente del contexto no tiene producciones nulas ni unitarias, entonces si u es una palabra de longitud n generada por la gramática, su derivación se obtiene en un número de pasos no superior a 2n-1**

**V.** Si si se requieren de más de 2n-1 pasos entonces podemos decir que la palabra no es generada.

**12.Cada árbol de derivación de una palabra en una gramática de tipo 2, tiene asociada una única derivación por la izquierda de la misma.**

**F.** No tiene porqué, una palabra que tuviese más de un árbol de derivación podría tener más de una derivación por la izquierda.

**13. Existe un lenguaje con un número finito de palabras que no puede ser generado por una gramática libre de contexto**

**F.** Todo lenguaje con un número finito de palabras es regular y por tanto, libre de contexto.

**14.La gramática compuesta por las reglas de producción  $S \rightarrow AA, A \rightarrow aSa|a$  no es ambigua**

**F.** Es ambigua probar con  $a^5$ .

**15.Para poder aplicar el algoritmo que transforma una gramática en forma normal de Greibach es necesario que la gramática esté en forma normal de Chomsky.**

**F.** Ver respuesta del 9.

**16.Un lenguaje libre de contexto es inherentemente ambiguo si existe una gramática ambigua que lo genera.**

**F.** Un lenguaje es inherentemente ambiguo si toda gramática que lo genera es ambigua.

**17.La gramática compeusta por las reglas de producción  $S \rightarrow A, A \rightarrow aSa|a$  es ambigua.**

**F.** Las a's se van añadiendo de dos en dos y finalmente se añade una y no hay otra posibilidad.

**18. Para generar una palabra de longitud  $n$  en una gramática en forma normal de Chomsky hacen falta exactamente  $2n-1$  pasos de derivación.**

V. Diría que sí pero no estoy seguro.

**19. Es imposible que una gramática esté en forma normal de Chomsky y Greibach al mismo tiempo**

F. Si suponemos la gramática cuya única regla de producción es  $S \rightarrow a$  esta está en forma normal de Greibach y de Chomsky.

**20. En una gramática independiente del contexto, si una palabra de longitud  $n$  es generada, entonces el número de pasos de derivación que se emplean debe de ser menor o igual a  $2n-1$**

V.

**21. El algoritmo que pasa una gramática a forma normal de Greibach produce siempre mismo resultado con independencia de cómo se numeren las variables.**

F. Se pueden generar más o menos reglas de producción pero al final serán todas equivalentes.

**22. La gramática compuesta por las siguientes reglas de producción  $\{S \rightarrow A|BA|SS, B \rightarrow a|b, A \rightarrow a\}$  es ambigua**

F. Probar con  $a^3$ .

**23. Si una palabra de longitud  $n$  es generada por una gramática en forma normal de Greibach, entonces lo es con  $n$  pasos de derivación exactamente.**

V. Uno por cada símbolo terminal.

**24. En una gramática independiente del contexto puede existir una palabra que es generada con dos derivaciones por la izquierda distintas que tiene el mismo árbol de derivación**

F. Si tiene dos derivaciones distintas por la izquierda entonces el árbol de derivación ya será distinto.

**25. Una gramática independiente del contexto genera un lenguaje que puede ser representado por una expresión regular.**

F. No siempre, solo aquellas que son regulares.

**26. Para cada AFND  $M$  existe una gramática independiente de contexto  $G$  tal que  $L(M)=L(G)$**

V. Pues si un lenguaje es aceptado por una AFND también es aceptado por un AFD y por consiguiente es regular e independiente del contexto.

**27. Para que un autómata con pila sea determinista es necesario que tenga transiciones nulas.**

F.

**28. El algoritmo que pasa una gramática a una forma normal de Greibach produce siempre el mismo resultado con independencia de cómo se numeren las variables.**

F.

**29. El conjunto de cadenas generado por una gramática independiente del contexto en forma normal de Greibach puede ser reconocido por un AFND con transiciones nulas**

F. Si fuese aceptado por un AFND con transiciones nulas también sería aceptado por un AFND y por un AFD y por tanto sería regular, cuando no todos los lenguajes independientes del contexto son regulares.

**30. La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.**

V. Da lugar a un lenguaje regular y por tanto independiente del contexto.

**31. Si  $L_1$  y  $L_2$  son independientes del contexto, no podemos asegurar que  $L_1 \cap L_2$  también lo sea.**

V.

## 5. Test Relación 5

**1. La clase de los lenguajes aceptados por los autómatas con pila deterministas es igual a la clase de los lenguajes generados por las gramáticas de tipo 2.**

F. La clase de los lenguajes generados por las gramáticas de tipo 2 coinciden con los autómatas con pila, pero no tienen porque ser deterministas.

**2. Una palabra es aceptada por un autómata con pila por el criterio de pila vacía si en algún momento, cuando leemos esta palabra, la pila se queda sin ningún símbolo, con independencia de la cantidad de símbolos que hayamos leído de la palabra de entrada**

F. Es aceptada por el criterio de pila vacía si tras leer la palabra entera se llega a una configuración en la que la pila está vacía.

3. Un autómata con pila siempre acepta el mismo lenguaje por los criterios de pila vacía y de estados finales.

F. Por lo general serán distintos aunque puede ser que coincidan.

4. Todo lenguaje aceptado por un autómata con pila determinista por el criterio de estados finales es también aceptado por un autómata con pila determinista por el criterio de pila vacía.

F. Sólo si el lenguaje cumple la propiedad del prefijo.

5. Para que un autómata con pila sea determinista es suficiente que desde cada configuración se pueda obtener, a lo más, otra configuración en un paso de cálculo.

V.

6. Si un lenguaje de tipo 2 verifica la propiedad del prefijo y es aceptado por un autómata con pila determinista por el criterio de estados finales, entonces también es aceptado por un autómata con pila determinista por el criterio de pila vacía.

V.

7. Para todo autómata con pila existe otro autómata con pila que acepta el mismo lenguaje y tiene un solo estado

F.

8. Si un lenguaje es aceptado por un autómata con pila determinista por el criterio de estados finales, entonces también es aceptado por un autómata con pila determinista por el criterio de pila vacía.

F. Igual que el 4.

9. En un autómata con pila determinista no puede haber transiciones nulas.

F. Puede tenerlas mientras siga siendo determinista.

10. Si  $L$  es independiente del contexto determinista y  $\$ \notin L$  entonces  $L.\{\$\}$  es aceptado por un autómata con pila determinista por el criterio de pila vacía.

F. Se debería verificar que  $\$ \notin A$ , es decir, que no esté en el alfabeto.

11. El conjunto de las palabras  $\{u0011u^{-1} : u \in \{0, 1\}^*\}$  es libre de contexto determinista

F.

**12. En la construcción de una gramática independiente del contexto a partir de un autómata con pila, la variable  $[p, X, q]$  genera todas las palabras que llevan al autómata desde el estado  $p$  al estado  $q$  sustituyendo  $X$  por el símbolo inicial de la pila.**

**F.** Genera las palabras que llevan al autómata desde  $p$  al estado  $q$  quitando una  $X$  de la pila.

**13. En un autómata con pila determinista no puede haber transiciones nulas.**

**F.** Si puede haberlas, siempre y cuando si hay transiciones nulas no haya una transición con misma configuración de partida y dos transiciones de llegadas distintas.

**14. Todo autómata con pila determinista que acepta un lenguaje por pila vacía se puede transformar en otro autómata determinista que acepte el mismo lenguaje por el criterio de estados finales.**

**V.** Las operaciones para transformar un autómata con pila por estados finales en uno por pila vacía son deterministas.

**15. Para que un lenguaje independiente del contexto sea determinista ha de verificar la propiedad del prefijo.**

**F.** Un lenguaje puede ser determinista y no verificar la propiedad del prefijo. La propiedad del prefijo sólo se debe verificar si se quiere pasar de un ACPD por estados finales en un ACPD por pila vacía.

**16. El lenguaje compuesto por las instrucciones completas del lenguaje SQL cumplen la propiedad prefijo.**

**¿?**

**17. En el algoritmo para pasar un autómata con pila gramática que hemos visto, si el autómata tiene 3 estados, entonces la transición  $(p, XYZU) \in \delta(q, \epsilon, H)$  da lugar a  $4^3$  producciones.**

**F.** Son  $3^4$ .

**18. El lenguaje  $\{0^i 1^k 2^i : i, k \geq 0\}$  es independiente del contexto determinista**

**F.**

**19. Si tenemos un lenguaje  $L$  aceptado por un Autómata con Pila por el criterio de estados finales, podemos encontrar otro AP que reconozca  $L$  por el criterio de pila vacía.**

**V.** Lo único que habría que hacer es aplicar el algoritmo.

**20. La propiedad prefijo no tiene ninguna relación con el hecho de que un lenguaje sea aceptado por un autómata con pila determinista por estados finales.**

V. Sólo tiene relación con el hecho de que un lenguaje sea aceptado por un autómata con pila por pila vacía.

**21. Para toda gramática libre de contexto  $G$  siempre se puede encontrar un autómata con pila que acepte el lenguaje generado por  $G$ .**

V. Todo lenguaje generado por una gramática independiente del contexto se tiene que existe un autómata con pila que lo genera.

**22. Si un lenguaje independiente del contexto cumple la propiedad prefijo, entonces puede ser aceptado por un autómata con pila determinista por el criterio de pila vacía.**

F. Debe ser aceptado por un autómata con pila determinista por el criterio de estados finales.

**23. La descripción instantánea de un autómata con pila nos permite saber el estado activo, lo que queda por leer de la cadena de entrada, lo que se ha consumido de la cadena de entrada y lo que nos queda en la pila.**

F. Nos permite saber el estado actual del autómata, lo que queda por leer de la cadena y el contenido de la pila, es decir, el símbolo tope.

**24. Un autómata finito determinista se puede convertir en un autómata con pila que acepta el mismo lenguaje por el criterio de pila vacía.**

V. Un lenguaje aceptado por un AFD es también independiente del contexto luego existe un AP que acepta el lenguaje por ambos criterios.

**25. El conjunto de cadenas generado por una gramática libre de contexto en forma normal de Greibach puede ser reconocido por un autómata con pila determinista por el criterio de pila vacía.**

V. Si está en forma normal de Greibach el lenguaje es regular por tanto es aceptado por un AFD si a este le añadimos una pila y trabajamos con pila vacía se obtiene lo buscado.

**26. Los lenguajes independientes del contexto con la propiedad del prefijo son siempre reconocidos por un autómata con pila determinista por el criterio de pila vacía.**

F. También debe ser aceptado por un AP por el criterio de estados finales.



**27. Puede existir un lenguaje con pila determinista que no sea aceptado por un AP determinista por el criterio de estados finales.**

F. Si es un AP determinista por pila vacía entonces también es aceptado por un AP determinista por estados finales pues solo se añaden operaciones deterministas. Y si es un AP determinista por estados finales es obvio.

**28. Existe un algoritmo para transformar una gramática regular G en un autómata con pila que acepte las cadenas del lenguaje generado por G por el criterio de pila vacía.**

V. Existe un algoritmo para pasar una gramática independiente del contexto a autómata con pila.

**29. Un autómata con pila determinista no puede tener transiciones nulas**

F. Siempre que siga cumpliéndose las reglas de determinismo.

**30. El conjunto de cadenas generadas por una gramática independiente del contexto en forma normal de Chomsky puede ser reconocido por un autómata finito no determinista con transiciones nulas.**

F. Esto se verifica sólo si al pasar de forma normal de Chomsky a forma normal de Greibach nos quedan reglas de producción correspondientes con una gramática regular.

**31. Para que un lenguaje sea aceptado por un autómata con pila determinista por el criterio de pila vacía tiene que verificar la propiedad prefijo**

F. Debe ser también aceptado por un AP determinista por criterio de estados finales.

**32. Un AFD se puede convertir en un AP que acepta el mismo lenguaje por el criterio de pila vacía.**

V.

**33. Un AP determinista no puede tener transiciones nulas**

F.

**34. Todo lenguaje aceptado por un AP determinista por el criterio de estados finales es también aceptado por un AP determinista por el criterio de pila vacía**

F.

**35. Si tenemos un autómata con pila en el que  $(p, \epsilon) \in \delta(q, a, C)$ , entonces para construir una gramática independiente del contexto que genere el mismo lenguaje que acepta el autómata, debemos de**

**añadir la producción  $[p, C, q] \rightarrow a$  (según el procedimiento visto en clase).**

**F.** Se debe añadir la producción  $[q, C, p] \rightarrow a$ , está invertido el orden de los estados.

**36. Para que un AP sea determinista es necesario que no tenga transiciones nulas**

**F.**

**37. El lenguaje  $L = \{u \in \{0, 1\}^* : u = u^{-1}\}$  es independiente del contexto, pero no determinista**

**V.** Mirar las diapositivas.

## **6. Test Relación 6**

**1. La intersección de lenguajes libres de contexto es siempre libre de contexto.**

**F.** La clase de los lenguajes independientes del contexto no es cerrada para la intersección

**2. Existe un algoritmo para determinar si una palabra es generada por una gramática independiente del contexto**

**V.** El algoritmo de Early.

**3. El lenguaje  $\{a^i b^j c^i d^i : i, j \geq 0\}$  es independiente del contexto**

**F.** No cumple el lema del bombeo.

**4. Existe un algoritmo para determinar si una gramática independiente del contexto es ambigua**

**F.** Hay que encontrar una palabra con dos árboles de derivación.

**5. Existe un algoritmo para comprobar cuando dos gramáticas libres de contexto generan el mismo lenguaje**

**F.**

**6. El lenguaje  $L = \{0^i 1^j 2^k : 1 \leq i \leq j \leq k\}$  es independiente del contexto.**

**F.** Probar para  $j=0$

**7. Si el lenguaje  $L$  es independiente del contexto, entonces  $L^{-1}$  es independiente del contexto**

**V.** Se toma el AP con pila que acepta  $L$  y se invierten las transiciones, además de convertir el estado inicial en final y el final en inicial.

**8.Existe un algoritmo que permite determinar si una gramática independiente del contexto genera un lenguaje finito o infinito**  
**V.**

**9.Existe un algoritmo para determinar si una gramática independiente del contexto es ambigua**  
**F.** Igual que el 4.

**11.Existe un algoritmo para comprobar si el lenguaje generado por una gramática libre de contexto es regular**  
**F.** No existe un algoritmo lo más fácil es buscar una gramática regular que lo genera.

**13.El conjunto de palabras  $\{a^n b^n c^i : i \leq n\}$  es independiente del contexto**  
**F**

**14.Si  $L_1$  y  $L_2$  son independientes del contexto, entonces  $L_1 - L_2$  es siempre independiente del contexto**  
**F.**  $L_1 - L_2 = L_1 \cap \overline{L_2}$ , donde sabemos que el complementario de un lenguaje independiente del contexto no tiene porque ser independiente del contexto y lo mismo para la intersección.

**15.Hay lenguajes que no son independientes del contexto y si verifican la condición que aparece en el lema del bombeo para lenguajes independientes del contexto**  
**V.**  $L = \{a^i b^j c^k d^l | (i = 0) \vee (j = k = l)\}$ .

**16.El conjunto de palabras  $\{u011u : u \in \{0,1\}^*\}$  es independiente del contexto**  
**V.** Es fácil encontrar un autómata con pila que genere este lenguaje.

**17.El conjunto de palabras que contiene la subcadena 011 es independiente del contexto**  
**V.** Es más es regular.

**18.En el algoritmo de Cocke-Younger-Kasami calculamos los conjuntos  $V_{ij}$  que son las variables que generan la subcadena de la palabra de entrada que va desde el símbolo en la posición i al símbolo en la posición j.**  
**F.** Los conjuntos  $V_{ij}$  son las variables que generan la subcadena desde el símbolo i con longitud j.

**19. Un lenguaje puede cumplir la negación de la condición que aparece en el lema de bombeo para lenguajes independientes del contexto y ser regular.**

F. Si incumple la condición entonces no es independiente del contexto y por tanto tampoco es regular.

**20. Existe un algoritmo para comprobar si el lenguaje generado con una gramática independiente del contexto es finito o infinito.**

V.

**21. Si  $L_1$  y  $L_2$  son lenguajes independientes de contexto, entonces  $(L_1 L_2 \cup L_1)^*$  es independiente del contexto.**

V. Como son independientes del contexto también lo es la concatenación, la unión y la clausura.

**22. Si  $L_1$  y  $L_2$  son independientes del contexto, entonces  $(L_1 - L_2)$  es independiente del contexto.**

F. Mirar respuesta del 14.

**23. Existe un algoritmo para determinar si una palabra  $u$  tiene más de un árbol de derivación en una gramática independiente del contexto  $G$**

F.

**24. La intersección de dos lenguajes independientes con un número finito de palabras produce siempre un lenguaje regular**

V. Los lenguajes con un número finito de palabras son regulares, por consiguientes la intersección de lenguajes regulares es un lenguaje regular.

**25. El complementario de un lenguaje con un número finito de palabras es siempre libre de contexto.**

V. Pues el complementario de un regular es regular y por tanto independiente del contexto.

**26. Todo lenguaje aceptado por un autómata con pila por el criterio de estados finales cumple la condición que aparece en el lema de bombeo para lenguajes libres de contexto.**

V. Si un lenguaje es aceptado por un AP entonces es independiente del contexto y por consiguiente cumple el lema de bombeo para estos.

**27. No existe algoritmo que para toda gramática libre de contexto  $G$  nos indique si el lenguaje generado por esta gramática  $L(G)$  es finito o infinito**

F. Si existe.

**28. Si  $L_1$  y  $L_2$  son lenguajes independientes de contexto, entonces  $(L_1 L_2 \cup L_1)^*$  puede ser representado por un autómata con pila.**

V. En el 21 se ve que dicha expresión es un lenguaje independiente del contexto y por tanto es aceptado por un AP.

**29. Existe un algoritmo para determinar si un autómata con pila es determinista**

F. No es un algoritmo como tal, basta con comprobar que de una configuración  $C_1$  por un paso de derivación se llega a lo sumo a una configuración  $C_2$ .

**30. La demostración del lema del bombeo para lenguajes independientes del contexto se basa en que si las palabras superan una longitud determinada, entonces en el árbol de derivación debe de aparecer una variable como descendiente de ella misma.**

V.

**31. La unión de dos lenguajes independientes de contexto puede ser siempre aceptada por un autómata con pila**

V. La unión de lenguajes IC es IC y por consiguiente es aceptada por un autómata con pila.

**32. El complementario de un lenguaje libre de contexto con una cantidad finita de palabras no tiene porque producir otro lenguaje libre de contexto**

F. Al tener un número finito de palabras es regular, y por tanto también lo es su complementario.

**33. El lema de bombeo para lenguajes libres de contexto es útil para demostrar que un lenguaje determinado no es libre de contexto**

V.

**34. La intersección de dos lenguajes independientes del contexto da lugar a un lenguaje aceptado por un autómata con pila determinista**

F. La clase de los lenguajes IC no es cerrada para la intersección.

**35. No existe algoritmo que reciba como entrada una gramática independiente del contexto y nos devuelva si el lenguaje generado por esta gramática es finito o infinito.**

F. Si existe.

**36.** En el algoritmo de Cocke-Younger-Kasami si  $A \in V_{1,2}$  y  $B \in V_{3,2}$  y  $C \rightarrow AB$ , entonces podemos deducir que  $C \in V_{1,4}$   
V.

**37.** Si  $L$  es independiente del contexto, entonces  $L^{-1}$  es independiente del contexto

V. Invertir las transiciones del autómata con pila que acepta  $L$ .

**38.** No existe algoritmo que nos diga si son iguales dos lenguajes generados por dos gramáticas independientes del contexto.

V. Si lo hay en caso de que fuesen regulares.

**39.** La intersección de dos lenguajes infinitos da lugar a un lenguaje independiente del contexto

F.

**40.** La unión de dos lenguajes IC puede ser aceptado por un autómata con pila

V

**41.** El lenguaje  $L = \{0^i 1^j 2^k \mid 1 \leq i \leq j \leq k\}$  es independiente del contexto

F.

**42.** Si  $L_1$  y  $L_2$  son IC, no podemos asegurar que  $L_1 \cap L_2$  lo sea  
V.

**43.** Si un lenguaje satisface la condición necesaria del lema de bombeo para lenguajes regulares, entonces también tiene que satisfacer la condición necesaria del lema de bombeo para lenguajes independientes del contexto.

F. Ya hemos dicho que un lenguaje puede verificar la condición del lema de bombeo para lenguajes regulares y no ser regular, por tanto tampoco podemos asegurar que ese lenguaje se IC.

## 7. Ejemplos Pregunta Cortas

Pon un ejemplo de lenguaje independiente del contexto determinista que no pueda ser aceptado por un autómata con pila determinista por el criterio de pila vacía. ¿Cual es la propiedad de este lenguaje que hace que la aceptación por pila vacía por un autómata determinístico sea imposible?

El lenguaje de las palabras con la misma cantidad de 0 que de 1, es aceptado por un autómata con pila determinista por el criterio de estados finales, sin embargo, no puede ser aceptado por un autómata con pila determinista por el criterio de pila vacía pues no cumple la propiedad del prefijo.

**Propiedad del prefijo:** para todo  $x \in L$  no existe prefijo de  $x$  (distinto de  $x$ ) que pertenezca al lenguaje.

**En el autómata producto de dos autómatas finitos  $M_1 = (Q_1, A, \delta_1, q_1, F_1)$   $M_2 = (Q_2, A, \delta_2, q_2, F_2)$  ¿qué estados habría que poner como finales para que el autómata acepte la unión de los lenguajes aceptados por  $M_1$  y  $M_2$ ?**

Si se considera cada estado del autómata producto como un conjunto que contiene estados de  $M_1$  y  $M_2$  a los que se llega por la cadena  $u$  que es la misma para llegar a dicho estado del autómata producto, entonces se tiene que un estado es final si contiene en dicho conjunto o bien un estado final de  $M_1$  o bien uno de  $M_2$  o bien de ambos a la vez.