

## MODELOS DE COMPUTACION

### Preguntas Tipo Test - Tema 1

1. Si un lenguaje es generado por una gramática dependiente del contexto, entonces dicho lenguaje no es independiente 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.
3. Si  $L$  es un lenguaje no vacío, entonces  $L^*$  es infinito.
4. Todo lenguaje con un número finito de palabras es regular e independiente del contexto.
5. Si  $L$  es un lenguaje, entonces siempre  $L^*$  es distinto de  $L^+$ .
6.  $L.\emptyset = L$
7. Si  $A$  es un alfabeto, la aplicación que transforma cada palabra  $u \in A^*$  en su inversa es un homomorfismo de  $A^*$  en  $A^*$ .
8. Si  $\epsilon \in L$ , entonces  $L^+ = L^*$ .
9. La transformación que a cada palabra sobre  $\{0, 1\}^*$  le añade 00 al principio y 11 al final es un homomorfismo.
10. Se puede construir un programa que tenga como entrada un programa y unos datos y que siempre nos diga si el programa leído termina para esos datos.
11. La cabecera del lenguaje  $L$  siempre incluye a  $L$ .
12. Un lenguaje nunca puede ser igual a su inverso.
13. La aplicación que transforma cada palabra  $u$  sobre el alfabeto  $\{0, 1\}$  en  $u^3$  es un homomorfismo.
14. El lenguaje que contiene sólo la palabra vacía es el elemento neutro para la concatenación de lenguajes.
15. Si  $L$  es un lenguaje, en algunas ocasiones se tiene que  $L^* = L^+$ .
16. Hay lenguajes con un número infinito de palabras que no son regulares.
17. Si un lenguaje tiene un conjunto infinito de palabras sabemos que no es regular.
18. Si  $L$  es un lenguaje finito, entonces su cabecera ( $CAB(L)$ ) también será finita.
19. El conjunto de palabras sobre un alfabeto dado con la operación de concatenación tiene una estructura de monoide.
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.
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  $u \in A_1^*$ .

## MODELOS DE COMPUTACION

### Preguntas Tipo Test - Tema 2

1. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $(rs)^* = r^*s^*$
2. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $(r + s)^* = r^* + s^*$
3. Si  $r_1$  y  $r_2$  son expresiones regulares, tales que su lenguaje asociado contiene la palabra vacía, entonces  $(r_1r_2)^* = (r_2r_1)^*$ .
4. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $(r + \epsilon)^+ = r^*$
5. Si  $r$  y  $s$  son expresiones regulares, tenemos que siempre se verifica que  $r(r + s)^* = (r + s)^*r$
6. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces  $r_1^*r_2^* \subseteq (r_1r_2)^*$ , en el sentido de que los lenguajes asociados están incluidos.
7. Si  $r_1, r_2$  y  $r_3$  son expresiones regulares, entonces  $(r_1 + r_2)^*r_3 = r_1^*r_3 + r_2^*r_3$ .
8. Si  $\mathbf{r}_1$  y  $\mathbf{r}_2$  son expresiones regulares entonces:  $(\mathbf{r}_1^*\mathbf{r}_2^*)^* = (\mathbf{r}_1 + \mathbf{r}_2)^*$
9. Si  $\mathbf{r}_1$  y  $\mathbf{r}_2$  son expresiones regulares, entonces  $(\mathbf{r}_1.\mathbf{r}_2)^* = (\mathbf{r}_1 + \mathbf{r}_2)^*$ .
10. Si  $\mathbf{r}$  es una expresión regular, entonces  $\mathbf{r}^*\mathbf{r}^* = \mathbf{r}^*$ .
11. Si  $\mathbf{r}$  es una expresión regular, entonces  $\mathbf{r}\emptyset = \mathbf{r} + \emptyset$ .
12. Si  $\mathbf{r}$  es una expresión regular, entonces se verifica que  $\mathbf{r}^*\epsilon = \mathbf{r}^+\epsilon$
13. Si  $\mathbf{r}_1$  y  $\mathbf{r}_2$  son expresiones regulares, entonces siempre  $\mathbf{r}_1(\mathbf{r}_2\mathbf{r}_1)^* = (\mathbf{r}_1\mathbf{r}_2)^*\mathbf{r}_1$
14. Si  $r_1$  y  $r_2$  son expresiones regulares, entonces siempre se verifica que  $r_1(r_2r_1)^* = (r_1r_2)^*r_1$ .
15. Si  $\mathbf{r}$  y  $\mathbf{s}$  son expresiones regulares, entonces  $(\mathbf{r}^*\mathbf{s}^*)^* = (\mathbf{r} + \mathbf{s})^*$ .
16. Si  $\mathbf{r}$  es una expresión regular, entonces  $(\mathbf{r}\mathbf{r})^* \subseteq \mathbf{r}^*$ .
17. Si  $r_1$  y  $r_2$  son expresiones regulares, tales que su lenguaje asociado contiene la palabra vacía, entonces  $(r_1r_2)^* = (r_1 + r_2)^*$ .
18. Si  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$  son expresiones regulares, entonces  $\mathbf{r}_1(\mathbf{r}_2^* + \mathbf{r}_3^*) = \mathbf{r}_1\mathbf{r}_2^* + \mathbf{r}_1\mathbf{r}_3^*$ .
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.
20. Un autómata finito puede ser determinista y no-determinista a la vez.
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.
22. Para pasar de un autómata que acepte el lenguaje asociado a  $\mathbf{r}$  a uno que acepte  $\mathbf{r}^*$  basta con unir con transiciones nulas sus estados finales con el estado inicial.
23. Existe un lenguaje reconocido por un AFD y no generado por una gramática independiente del contexto.
24. Existen lenguajes aceptados por AFD que no pueden ser aceptados por AF no determinísticos.
25. La clausura de un lenguaje aceptado por un AFD puede ser representado con una expresión regular.

## MODELOS DE COMPUTACION

### Preguntas Tipo Test - Tema 3

1. El lema de bombeo puede usarse para demostrar que un lenguaje determinado es regular.
2. Todo lenguaje con un número finito de palabras es regular.
3. La intersección de lenguajes regulares es siempre regular.
4. La demostración del lema de 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.
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.
6. En la demostración de que todo autómata finito tiene una 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  $k$ ,  $q_k$ .
7. El conjunto de todas las expresiones regulares es un lenguaje 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  $R/L$ .
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.
10. Si en un autómata finito no hay estados distinguibles de nivel 2, ya no puede haber estados distinguibles de nivel 4.
11. Todo lenguaje generado por una gramática lineal por la derecha es también generado por una gramática lineal por la izquierda.
12. Un autómata finito determinista sin estados inaccesibles ni indistinguibles es minimal.
13. Si  $L$  es un lenguaje sobre el alfabeto  $A$ , entonces  $CAB(L)$  es siempre igual al cociente  $L/A^*$ .
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.
16. La expresión recursiva que se emplea para obtener la expresión regular asociada a un autómata finito determinista es:  $r_{ij}^k = r_{ij}^{k-1} + r_{i(k-1)}^{k-1} (r_{(k-1)(k-1)}^{k-1})^* r_{(k-1)j}^{k-1}$
17. Cuando se construye la expresión regular asociada a un autómata finito determinista,  $r_{ii}^0$  no puede ser nunca vacío.
18. El conjunto de las palabras  $\{u0011v^{-1} : u, v \in \{0,1\}^*\}$  es regular.
19. Si  $L$  es un lenguaje finito, entonces su complementario es siempre regular.
20. En un autómata finito determinista la relación de indistinguibilidad es una relación de equivalencia.
21. En un autómata finito determinista siempre debe de existir, al menos, un estado de error.
22. El conjunto de los números en binario que son múltiplos de 7 es regular.

23. Hay situaciones en las que los estados inaccesibles de un AFD cumplen una función específica.
24. Si  $R$  es un lenguaje regular y  $L$  un lenguaje independiente del contexto, entonces  $R/L$  es regular.
25. Si en un autómata dos estados son distinguibles de nivel  $n$ , entonces serán distinguibles de nivel  $m$  para todo  $m \geq n$ .
26. Si  $h$  es un homomorfismo y  $h(L)$  no es regular, podemos concluir que  $L$  no es regular.
27. El lenguaje de todas las palabras en las que los tres primeros símbolos son iguales a los tres últimos es regular.
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 es regular.
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.
30. Todo lenguaje que cumple la condición del lema de bombeo para lenguajes regulares puede ser aceptado por un autómata finito no determinista.
31. No existe algoritmo para saber si el lenguaje generado por una gramática regular es finito.
32. Dos autómatas finitos deterministas con diferente número de estados y que aceptan el lenguaje vacío tienen el mismo número de estados finales.
33. Si  $A$  es un alfabeto y  $L$  un lenguaje cualquiera distinto del vacío, entonces se verifica que  $A^*/L = A^*$ .
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  $R_{ij}^{j-1} R_{jk}^{j-1} \subseteq R_{ik}^j$ .
35. El lema de bombeo es útil para demostrar que la intersección de dos lenguajes regulares no es regular.
36. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es infinito.
37. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es finito o infinito.
38. La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.
39. Si un lenguaje es infinito no se puede encontrar una expresión regular que lo represente.
40. En un autómata finito determinista sin estados inaccesibles la relación de indistinguibilidad entre los estados es una relación de equivalencia.
41. En un autómata finito determinista, si no hay dos estados que sean indistinguibles entre sí, entonces el autómata es minimal.
42. Dada una gramática lineal por la derecha, siempre existe otra gramática lineal por la izquierda que acepte el mismo lenguaje.
43. Si  $R$  es un lenguaje regular y  $L$  un lenguaje cualquiera, entonces  $R/L$  es siempre un lenguaje regular.
44. Si un lenguaje cumple la condición del lema de bombeo para conjuntos regulares no nos asegura que sea un lenguaje regular.
45. Existe un algoritmo para determinar si los lenguajes generados por dos gramáticas regulares son iguales o no.

## MODELOS DE COMPUTACION

### Preguntas Tipo Test - Tema 4

Indicar si son verdaderas o falsas las siguientes afirmaciones:

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.
2. En una gramática de tipo 2 ambigua no puede existir una palabra generada con un único árbol de derivación.
3. Dada una gramática independiente del contexto, siempre se puede construir una gramática sin transiciones nulas ni unitarias que genere exactamente el mismo lenguaje que la gramática original.
4. Una gramática independiente del contexto es ambigua si existe una palabra que puede ser generada con dos cadenas de derivación distintas.
5. Un lenguaje inherentemente ambiguo puede ser generado por una gramática ambigua.
6. El lenguaje de las palabras sobre  $\{0, 1\}$  con un número impar de ceros 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.
8. Todo árbol de derivación de una palabra en una gramática independiente del contexto está asociado a una única derivación por la izquierda.
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.
10. Sólo hay una derivación por la derecha asociada a un árbol de derivación.
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  $2n - 1$ .
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.
13. Existe un lenguaje con un número finito de palabras que no puede ser generado por una gramática libre de contexto.
14. La gramática compuesta por las reglas de producción  $S \rightarrow AA, A \rightarrow aSa, A \rightarrow a$  no es ambigua.
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.
16. Un lenguaje libre de contexto es inherentemente ambiguo si existe una gramática ambigua que lo genera.
17. La gramática compuesta por las reglas de producción  $S \rightarrow A, A \rightarrow aSa, A \rightarrow a$  es ambigua.
18. Para generar una palabra de longitud  $n$  en una gramática en forma normal de Chomsky hacen falta exactamente  $2n - 1$  pasos de derivación.
19. Es imposible que una gramática esté en forma normal de Chomsky y Greibach al mismo tiempo.
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  $2n - 1$ .

21. El algoritmo que pasa una gramática a forma normal de Greibach produce siempre el mismo resultado con independencia de cómo se numeren las variables.
22. La gramática compuesta por la siguientes reglas de producción  $\{S \rightarrow A|BA|SS, B \rightarrow a|b, A \rightarrow a\}$  es ambigua.
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.
24. En una gramática independiente del contexto puede existir una palabra que es generada con dos derivaciones por la izquierda distintas que tienen el mismo árbol de derivación.
25. Una gramática independiente del contexto genera un lenguaje que puede ser representado por una expresión regular.

## MODELOS DE COMPUTACION

### Preguntas Tipo Test - Tema 5

Indicar si son verdaderas o falsas las siguientes afirmaciones:

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.
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.
3. Un autómata con pila siempre acepta el mismo lenguaje por los criterios de pila vacía y de estados finales.
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.
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.
6. Si un lenguaje de tipo 2 verifica la propiedad 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.
7. Para todo autómata con pila existe otro autómata con pila que acepta el mismo lenguaje y tiene un solo estado.
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.
9. En un autómata con pila determinista no puede haber transiciones nulas.
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.
11. El conjunto de las palabras  $\{u0011u^{-1} : u \in \{0,1\}^*\}$  es libre del contexto determinista.
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.
13. En un autómata con pila determinista no puede haber transiciones nulas.
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.
15. Para que un lenguaje independiente del contexto sea determinista ha de verificar la propiedad prefijo.
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 a 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.
18. El lenguaje  $\{0^i 1^k 2^i : i, k \geq 0\}$  es independiente del contexto determinista.
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.

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.
21. Para toda gramática libre de contexto  $G$  siempre se puede encontrar un autómata con pila que acepte el lenguaje generado por  $G$ .
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.
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.
24. Un autómata finito determinista se puede convertir en un autómata con pila que acepta el mismo lenguaje por el criterio de pila vacía.
25. El conjunto de cadenas generado por una gramática libre de contexto en forma normal de Greibach puede ser reconocido por un autómata finito no determinista con transiciones nulas.
26. Los lenguajes independientes del contexto con la propiedad prefijo son siempre reconocidos por un autómata con pila determinista por el criterio de pila vacía.
27. Puede existir un lenguaje con pila determinista que no sea aceptado por un autómata con pila determinista por el criterio de estados finales.
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.
29. Un autómata con pila determinista no puede tener transiciones nulas.
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.



**MODELOS DE COMPUTACION**  
**Preguntas Tipo Test - Tema 6**

Indicar si son verdaderas o falsas las siguientes afirmaciones:

1. La intersección de lenguajes libres de contexto es siempre libre de contexto.
2. Existe un algoritmo para determinar si una palabra es generada por una gramática independiente del contexto.
3. El lenguaje  $\{a^i b^j c^i d^i : i, j \geq 0\}$  es independiente del contexto.
4. Existe un algoritmo para determinar si una gramática independiente del contexto es ambigua.
5. Existe un algoritmo para comprobar cuando dos gramáticas libres de contexto generan el mismo lenguaje.
6. El lenguaje  $L = \{0^i 1^j 2^k : 1 \leq i \leq j \leq k\}$  es independiente del contexto.
7. Si el lenguaje  $L$  es independiente del contexto, entonces  $L^{-1}$  es independiente del contexto.
8. Existe un algoritmo que permite determinar si una gramática independiente del contexto genera un lenguaje finito o infinito.
9. Existe un algoritmo para determinar si una gramática independiente del contexto es ambigua.
10. En el algoritmo de Earley, la presencia del registro  $(2, 5, A, CD, adS)$  implica que a partir de  $CD$  se puede generar la subcadena de la palabra de entrada que va del carácter 3 al 5.
11. Existe un algoritmo para comprobar si el lenguaje generado por una gramática libre de contexto es regular.
12. El algoritmo de Earley se puede aplicar a cualquier gramática independiente del contexto (sin producciones nulas ni unitarias).
13. El conjunto de palabras  $\{a^n b^n c^i : i \leq n\}$  es independiente del contexto.
14. Si  $L_1$  y  $L_2$  son independientes del contexto, entonces  $L_1 - L_2$  es siempre independiente del contexto.
15. Hay lenguajes que no son independientes del contexto y si verifican la condición que aparece en el lema de bombeo para lenguajes independientes del contexto.
16. El conjunto de palabras  $\{u011u : u \in \{0, 1\}^*\}$  es independiente del contexto.
17. El conjunto de palabras que contienen la subcadena 011 es independiente del contexto.
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  $j$ .
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.
20. Existe un algoritmo para comprobar si el lenguaje generado con una gramática independiente del contexto es finito o infinito.
21. Si  $L_1$  y  $L_2$  son lenguajes independientes de contexto, entonces  $(L_1 L_2 \cup L_1)^*$  es independiente de contexto.
22. Si  $L_1$  y  $L_2$  son lenguajes independientes de contexto, entonces  $(L_1 - L_2)$  es independiente del contexto.

23. Existe un algoritmo para determinar si una palabra  $u$  tiene más de un árbol de derivación en una gramática independiente del contexto  $G$ .
24. La intersección de dos lenguajes independientes de contexto con un número finito de palabras produce siempre un lenguaje regular.
25. El complementario de un lenguaje con un número finitos de palabras es siempre libre de 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.
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.
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.
29. Existe un algoritmo para determinar si un autómata con pila es determinista.
30. La demostración del lema de 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.
31. La unión de dos lenguajes independientes contexto puede ser siempre 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.
33. El lema de bombeo para lenguajes libres de contexto es útil para demostrar que un lenguaje determinado no es libre de contexto.
34. La intersección de dos lenguajes independientes del contexto da lugar a un lenguaje aceptado por un autómata con pila determinista.
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.
36. En el algoritmo de Cocke-Younger-Kasami si  $A \in V_{1,2}$  y  $B \in V_{3,2}$  y  $C \rightarrow AB$ , podemos deducir que  $C \in V_{1,4}$ .
37. Si  $L$  es independiente del contexto, entonces  $L^{-1}$  es independiente del contexto.
38. No existe un algoritmo que nos diga si son iguales los lenguajes generados por dos gramáticas independientes del contexto  $G_1$  y  $G_2$ .
39. La intersección de dos lenguajes infinitos da lugar a un lenguaje independiente del contexto.