INFORME DE LA PRÁCTICA 8

Nº 1.- Dadas las siguiente gramáticas, simplificar y pasar a forma normal de Chomsky

a) $S \rightarrow aBAAb|bABBa|aCa$
$A \rightarrow aBa|B|a|\varepsilon$
$B \rightarrow bAb|A|b|\varepsilon$
$C \rightarrow aCa|bDb$
$D \rightarrow aCa|bDb$

En el primer paso, la eliminación de producciones vacías, vemos que tenemos dos, en *A* y en *B*. Procedemos a eliminarlas, y donde aparezcan, contemplaremos la opción de que ese símbolo no terminal aparezca o no. Nos quedaría una gramática de la siguiente manera:

$S \rightarrow aBAAb|aBAb|aBb|aAAb|aAb|ab|bABBa|bABa|bAa|bBBa|bBa|ba|aCa$
$A \rightarrow aBa|aa|B|a$
$B \rightarrow bAb|bb|A|b$
$C \rightarrow aCa|bDb$
$D \rightarrow aCa|bDb$

En el segundo paso, nos encargamos de eliminar las producciones unitarias, por lo que B y A, colocadas en A y B respectivamente, tienen que desaparecer, sustituyendo la B en la A y la A en la B:

$S \rightarrow aBAAb|aBAb|aBb|aAAb|aAb|ab|bABBa|bABa|bAa|bBBa|bBa|ba|aCa$
$A \rightarrow aBa|aa|bAb|bb|b|a$
$B \rightarrow bAb|bb|aBa|aa|a|b$
$C \rightarrow aCa|bDb$
$D \rightarrow aCa|bDb$

En el tercer paso, eliminación de símbolos y producciones inútiles, buscaremos aquellos símbolos no terminales que son derivables, y eliminaremos los que no lo son. Seguidamente, con esos símbolos, veremos cuáles no son alcanzables.

Para comprobar que símbolos no terminales son derivables, tendremos en cuenta los que tienen símbolos terminales:

$V' = \{S, A\}$

Nos faltan los símbolos *C* y *D*, pero como podemos comprobar, se puede hacer desde S, pero después, aparte de no tener símbolos terminales, todos sus símbolos no terminales no se encuentran en *V'*, es decir, nuestra cadena nunca acabaría, por lo que nos encontramos ante un bucle y por ello, *C* y *D* tienen que ser eliminados. Esta sería nuestra gramática, a falta del último paso:

$S \rightarrow aBAAb|aBAb|aBb|aAAb|aAb|ab|bABBa|bABa|bAa|bBBa|bBa|ba$
$A \rightarrow aBa|aa|bAb|bb|b|a$
$B \rightarrow bAb|bb|aBa|aa|a|b$

Ahora, dentro del paso de eliminación de símbolos y producciones inútiles, tenemos que analizar que símbolos no terminales y símbolos del alfabeto no son alcanzables desde S, ya que tendríamos que eliminarlos.

$V' = \{S, A, B\}$
$T' = \{a, b\}$

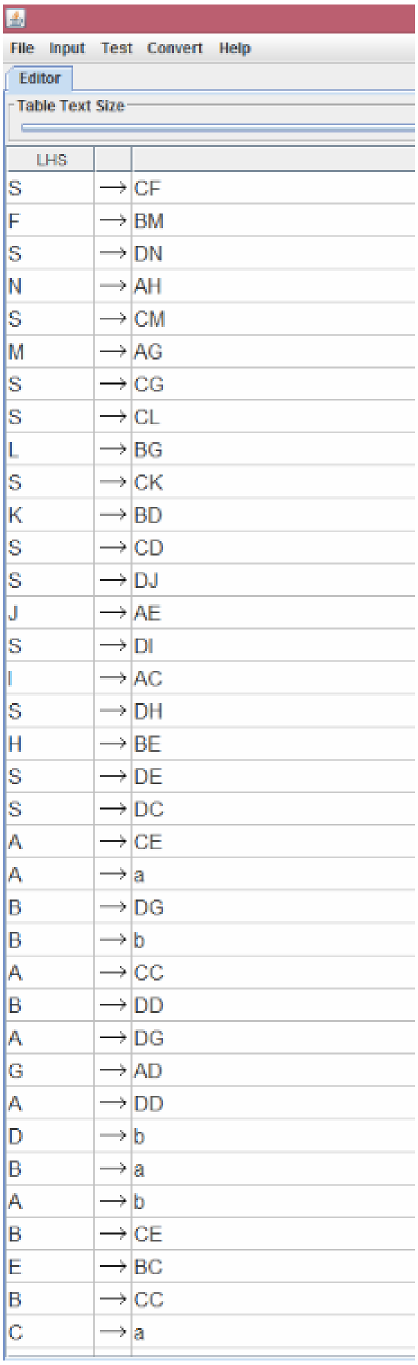
Como hemos comprobado, nuestra gramática esta simplificada, puesto que hemos alcanzado todos los símbolos no terminales y todos los símbolos del alfabeto. Además, nos hemos dado cuenta de que las producciones de *A* y *B* son exactamente las mismas, lo que significa que podemos eliminar uno de los dos símbolos y que toda *B* llegue hasta *A*. Tendríamos esta gramática:

$S \rightarrow aAAAb|aAAb|aAb|ab|bAAAa|bAAa|bAa|ba$
$A \rightarrow aAa|aa|bAb|bb|b|a$

Para la forma normal de Chomsky, si nos damos cuenta, las producciones del símbolo S con más de dos símbolos siempre acaban de manera *Aa* o *Ab* y empiezan por *aA* o *bA*. Esto simplificará mucho el paso a la forma normal.

$S \rightarrow BH|FE|BE|BC|CJ|GI|CI|CA$
$A \rightarrow BI|BB|CE|CC|b|a$
$B \rightarrow a$
$C \rightarrow b$
$D \rightarrow AA = AA$
$E \rightarrow AC = Ab$
$F \rightarrow BA = aA$
$G \rightarrow CA = bA$
$H \rightarrow CE = AAAb$
$I \rightarrow AB = Aa$
$J \rightarrow DI = AAAa$

Al comprobarlo en JFLAP, seleccionamos la opción “Grammar”, introducimos nuestra gramática original. A continuación pulsamos en “Transform Grammar” en el menú “Convert”. Nos hará los pasos de eliminación de producciones vacías, le damos a “Do All” y luego a “Proceed” y ahora JFLAP proseguirá con la eliminación de producciones unitarias. Nuevamente le damos a “Do All” y “Proceed” y ahora eliminaremos las producciones y símbolos inútiles, y nuevamente “Do All” y “Export”. Ya tendríamos nuestra gramática en Forma Normal de Chomsky.



LHS		
S	→	CF
F	→	BM
S	→	DN
N	→	AH
S	→	CM
M	→	AG
S	→	CG
S	→	CL
L	→	BG
S	→	CK
K	→	BD
S	→	CD
S	→	DJ
J	→	AE
S	→	DI
I	→	AC
S	→	DH
H	→	BE
S	→	DE
S	→	DC
A	→	CE
A	→	a
B	→	DG
B	→	b
A	→	CC
B	→	DD
A	→	DG
G	→	AD
A	→	DD
D	→	b
B	→	a
A	→	b
B	→	CE
E	→	BC
B	→	CC
C	→	a

Como se puede apreciar, en el JFLAP obtenemos una cantidad superior de símbolos no terminales. Esto se debe a que el programa no eliminó el símbolo *B* que era igual al *A*, ya que eso no se implementa en los algoritmos. Por ello, nuestras gramáticas son equivalentes, pero una mayor que la otra.

b) $S \rightarrow aY|Ybb|Y$
$X \rightarrow a|\varepsilon$
$Y \rightarrow aXY|bb|XXa$

En el primer paso, la eliminación de producciones vacías, vemos que tenemos una en *X*. Procedemos a eliminarla, y donde aparezca, contemplaremos la opción de que ese símbolo no terminal aparezca o no. Nos quedaría una gramática de la siguiente manera:

$S \rightarrow aY|Ybb|Y$
$X \rightarrow a$
$Y \rightarrow aXY|aY|bb|XXa|Xa|a$

En el segundo paso, nos encargamos de eliminar las producciones unitarias, por lo que la *Y* de S, tiene que desaparecer:

$S \rightarrow aY|Ybb|aXY|bb|XXa|Xa|a$
$X \rightarrow a$
$Y \rightarrow aXY|aY|bb|XXa|Xa|a$

En el tercer paso, eliminación de símbolos y producciones inútiles, buscaremos aquellos símbolos no terminales que son derivables, y eliminaremos los que no lo son. Seguidamente, con esos símbolos, veremos cuáles no son alcanzables.

Para comprobar que símbolos no terminales son derivables, tendremos en cuenta los que tienen símbolos terminales:

$V' = \{S, X, Y\}$

Como todos son símbolos derivables, no quitamos ninguno de ellos. Ahora comprobamos que todos son alcanzables, así como los símbolos del alfabeto:

$V' = \{S, X, Y\}$
$T' = \{a, b\}$

Podemos llegar a todos los símbolos no terminales y todos los símbolos del alfabeto desde la primera iteración, lo que quiere indicarnos que la gramática ya está minimizada:

$S \rightarrow aY|Ybb|aXY|bb|XXa|Xa|a$
$X \rightarrow a$
$Y \rightarrow aXY|aY|bb|XXa|Xa|a$

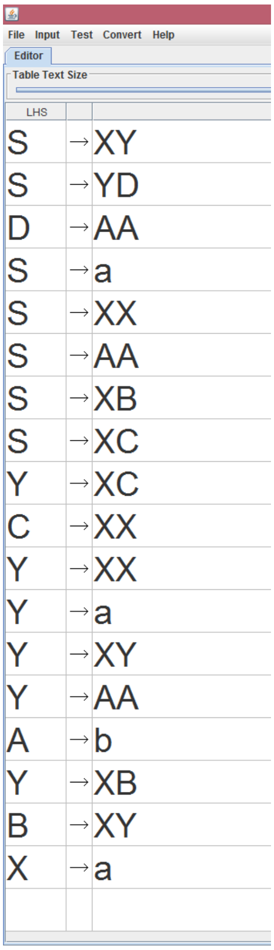
Para hacer la forma normal de Chomsky, he preferido llamar *X* como *A* e *Y* como *B* por comodidad, dejándonos la gramática de la siguiente forma:

$S \rightarrow aB|Bbb|aAB|bb|AAa|Aa|a$
$A \rightarrow a$
$B \rightarrow aAB|aB|bb|AAa|Aa|a$

Procedemos a la forma normal de Chomsky:

$S \rightarrow AB|BD|EB|CC|EA|AA|a$
$A \rightarrow a$
$B \rightarrow EB|AB|CC|EA|EA|a$
$C \rightarrow b$
$D \rightarrow CC = bb$
$E \rightarrow AA = aA$

Al comprobarlo en JFLAP, seleccionamos la opción “Grammar”, introducimos nuestra gramática original. A continuación pulsamos en “Transform Grammar” en el menú “Convert”. Nos hará los pasos de eliminación de producciones vacías, le damos a “Do All” y luego a “Proceed” y ahora JFLAP proseguirá con la eliminación de producciones unitarias. Nuevamente le damos a “Do All” y “Proceed” y ahora eliminaremos las producciones y símbolos inútiles, y nuevamente “Do All” y “Export”. Ya tendríamos nuestra gramática en Forma Normal de Chomsky.



Podemos comprobar que las gramáticas en Forma Normal de Chomsky son las mismas.

c) $S \rightarrow A|B$
$A \rightarrow C|D$
$B \rightarrow D|E$
$C \rightarrow S|a|\varepsilon$
$D \rightarrow S|b$
$E \rightarrow S|c|\varepsilon$

El primer paso consiste en la eliminación de producciones vacías. En este caso, tenemos dos, en *C* y en *E*, por lo tanto, tenemos que quitarlas, y donde aparezca *C* o *E*, contemplaremos la posibilidad de que dicho símbolo salga o no:

$S \rightarrow A|B$
$A \rightarrow C|D$
$B \rightarrow D|E$
$C \rightarrow S|a$
$D \rightarrow S|b$
$E \rightarrow S|c$

Desde S podemos llegar a *C* y a *D* sin añadir ningún símbolo terminal. Por lo tanto, la cadena vacía es una posibilidad, por lo que tenemos que añadir ε a S:

$S \rightarrow A|B|\varepsilon$
$A \rightarrow C|D$
$B \rightarrow D|E$
$C \rightarrow S|a$
$D \rightarrow S|b$
$E \rightarrow S|c$

En el segundo paso, eliminamos las producciones vacías, encontrándonos con una en cada símbolo no terminal:

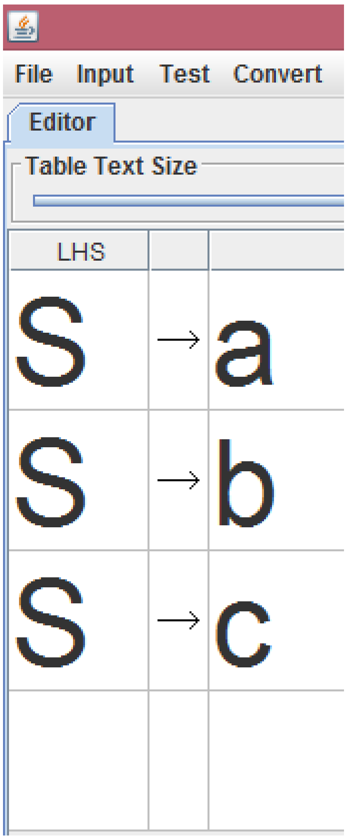
$S \rightarrow a|b|c|\varepsilon$
$A \rightarrow S|a|b$
$B \rightarrow S|b|c$
$C \rightarrow S|a$
$D \rightarrow S|b$
$E \rightarrow S|c$

Como podemos ver, si los símbolos no terminales tienen S como producción unitaria, nos damos cuenta de que va a salir el mismo resultado para todas ellas, por lo que nos quedamos con una gramática con un solo símbolo no terminal, el S. Como podemos ver, desde S se deriva a S y se alcanzan todos los símbolos del alfabeto, por lo que ya está simplificada:

$S \rightarrow a|b|c|\varepsilon$

Esta gramática ya se encuentra en forma normal de Chomsky, por lo que no tenemos que modificar nada.

Al comprobarlo en JFLAP, seleccionamos la opción "Grammar", introducimos nuestra gramática original. A continuación pulsamos en "Transform Grammar" en el menú "Convert". Nos hará los pasos de eliminación de producciones vacías, le damos a "Do All" y luego a "Proceed" y ahora JFLAP proseguirá con la eliminación de producciones unitarias. Nuevamente le damos a "Do All" y "Proceed" y ahora eliminaremos las producciones y símbolos inútiles, y nuevamente "Do All" y "Export". Ya tendríamos nuestra gramática en Forma Normal de Chomsky.



Las gramáticas en Forma Normal de Chomsky son exactamente las mismas, por lo que hemos aplicado correctamente todos los procedimientos.

d) $S \to A|B$
$A \to aB|bS|b$
$B \to AB|Ba|CC$
$C \to AS|b|\varepsilon$

En el primer paso de la simplificación, tenemos que eliminar las producciones vacías. En este caso, solo aparece en *C*, por lo que eliminamos su producción vacía, y donde aparezca *C* contemplamos la posibilidad de que salga o no:

$S \to A|B$
$A \to aB|bS|b$
$B \to AB|Ba|CC|C$
$C \to AS|b$

Al comprobar si podemos llegar desde S hasta *C* si añadir ningún símbolo terminal, vemos que sí es posible ($S \to B \to C$) por lo que añadimos la posibilidad de la cadena vacía a nuestra gramática:

$S \to A|B|\varepsilon$
$A \to aB|bS|b$
$B \to AB|Ba|CC|C$
$C \to AS|b$

En el segundo paso tenemos que eliminar las producciones unitarias y donde aparezcan, tendremos que sustituirlas por las producciones completas de dicho símbolo. En este caso encontramos 3, *A* y *B* en S, y *C* en *B*:

$S \to aB|bS|b|AB|Ba|CC|AS|\varepsilon$
$A \to aB|bS|b$
$B \to AB|Ba|CC|AS|b$
$C \to AS|b$

En el tercer paso, el de eliminación de producciones y símbolos inútiles, miramos que símbolos no terminales son derivables, y desechamos aquellos que no aparezcan en nuestra lista:

$V' = \{S, A, B, C\}$

Como vemos, todos los símbolos no terminales son derivables en nuestra primera vuelta, por lo que todos incluyen al menos un símbolo terminal. Ahora comprobamos si alguno de los símbolos, terminales o no, no se pueden alcanzar:

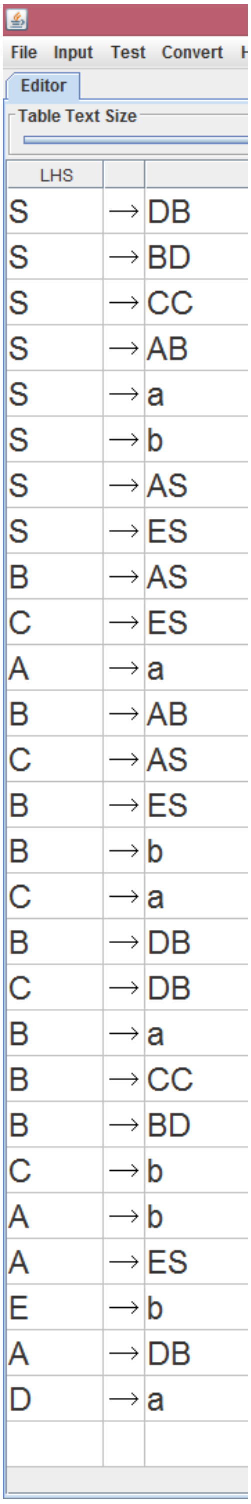
$V' = \{S, B, A, C\}$
$T = \{a, b\}$

Podemos alcanzar todos los símbolos, por lo que nuestra gramática ya está simplificada y podemos hacerla en la forma normal de Chomsky.

Forma normal de Chomsky:

$S \rightarrow DB|ES|b|AB|BD|CC|AS|\varepsilon$
$A \rightarrow DB|ES|b$
$B \rightarrow AB|BD|CC|AS|b$
$C \rightarrow AS|b$
$D \rightarrow a$
$E \rightarrow b$

Al comprobarlo en JFLAP, seleccionamos la opción "Grammar", introducimos nuestra gramática original. A continuación pulsamos en "Transform Grammar" en el menú "Convert". Nos hará los pasos de eliminación de producciones vacías, le damos a "Do All" y luego a "Proceed" y ahora JFLAP proseguirá con la eliminación de producciones unitarias. Nuevamente le damos a "Do All" y "Proceed" y ahora eliminaremos las producciones y símbolos inútiles, y nuevamente "Do All" y "Export". Ya tendríamos nuestra gramática en Forma Normal de Chomsky.



LHS		
S	→	DB
S	→	BD
S	→	CC
S	→	AB
S	→	a
S	→	b
S	→	AS
S	→	ES
B	→	AS
C	→	ES
A	→	a
B	→	AB
C	→	AS
B	→	ES
B	→	b
C	→	a
B	→	DB
C	→	DB
B	→	a
B	→	CC
B	→	BD
C	→	b
A	→	b
A	→	ES
E	→	b
A	→	DB
D	→	a

Podemos determinar que ambas gramáticas en Formal Normal de Chomsky son las mismas.

Nº 2.- Analizar si las siguientes cadenas forman parte del lenguaje de la gramática a).

a) $w_1 = abababb$

Utilizando la gramática simplificada, vemos que si podemos obtener la cadena de una manera sencilla:

$S \rightarrow aAAAb|aAAb|aAb|ab|bAAAa|bAAa|bAa|ba$
$A \rightarrow aAa|aa|bAb|bb|b|a$

$aAAAb \rightarrow abAAb \rightarrow abaAb \rightarrow ababAbb \rightarrow abababb$

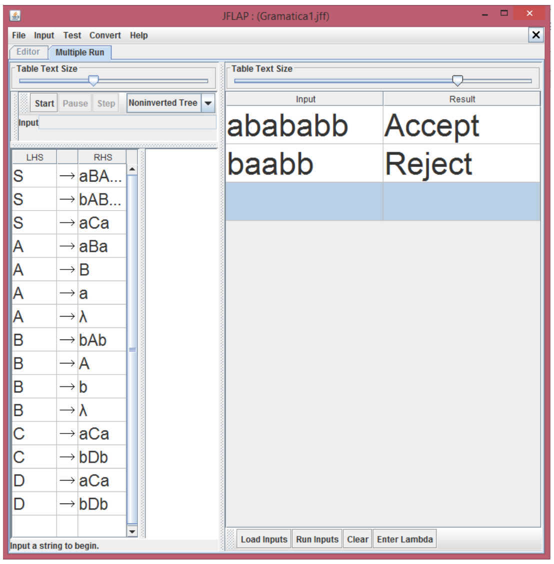
b) $w_2 = baabb$

Analizamos esta cadena también utilizando el algoritmo de CYK.

	b	a	a	b	b
1	A, C	A, B	A, B	A, C	A, C
2	D, I, S, G	D, I, F, A	D, E, F, S	D, E, S, G, A	
3	D	S, D, E	D, S, F		
4	∅	S, D			
5	∅				

Como se puede comprobar, S no aparece en la última iteración, por lo que la cadena $w_2 = baabb$ no pertenece al lenguaje de la gramática.

Con el JFLAP, una vez introducida la gramática, tenemos una opción "Multiple Brute Force Parse" en el menú "Input" que nos permite introducir cadenas, y el programa detectar si se aceptan o no usando la fuerza bruta (probar todas las combinaciones posibles).



MODIFICACIÓN

$S \rightarrow bA|aBa|C$
$A \rightarrow Aa|\varepsilon$
$B \rightarrow bAB|AB|b$
$C \rightarrow A|cc$

Iniciamos el proceso de simplificación de gramáticas analizando las producciones vacías. En este caso solo tenemos una en *A*, por lo que la quitamos, y en cuanto aparezca *A* en nuestra gramática contemplamos la opción de que aparezca o no:

$S \rightarrow bA|b|aBa|C$
$A \rightarrow Aa|a$
$B \rightarrow bAB|bB|AB|b$
$C \rightarrow A|cc$

Ahora analizamos si podemos llegar desde S hasta *A* sin añadir ningún símbolo terminal, es decir, ver si la cadena vacía es aceptada por nuestra gramática. Podemos si realizamos el siguiente recorrido, $S \rightarrow C \rightarrow A$, por lo que añadimos ε a S:

$S \rightarrow bA|b|aBa|C|\varepsilon$
$A \rightarrow Aa|a$
$B \rightarrow bAB|bB|AB|b$
$C \rightarrow A|cc$

El segundo paso consiste en la eliminación de producciones unitarias, es decir, en nuestro caso la *B* en S y la *A* en C:

$S \rightarrow bA|b|aBa|Aa|a|cc|\varepsilon$
$A \rightarrow Aa|a$
$B \rightarrow bAB|bB|AB|b$
$C \rightarrow Aa|a|cc$

En el último paso de eliminación de producciones y símbolos inútiles se divide en dos etapas, primero se mira los símbolos que tienen una producción terminal, en este caso:

$V' = \{S, A, B, C\}$

Como podemos ver, todos los símbolos no terminales son derivables, por lo que no podemos eliminar ninguno. Vamos a la segunda fase, donde comprobamos que todos los estados son alcanzables, así como todos los símbolos del alfabeto:

$V' = \{S, A, B\}$
$T = \{a, b, c\}$

Por lo tanto, el estado *C* no se puede alcanzar y se elimina. En el caso de los símbolos terminales, todos son útiles, por lo que ninguno se puede quitar, dejándonos la gramática simplificada de la siguiente manera:

$S \rightarrow bA|b|aBa|Aa|a|cc|\varepsilon$
$A \rightarrow Aa|a$
$B \rightarrow bAB|bB|AB|b$

Pasamos a la forma normal de Chomsky:

$S \rightarrow DA|b|CF|AC|a|EE|\varepsilon$
$A \rightarrow AC|a$
$B \rightarrow DG|DB|AB|b$
$C \rightarrow a$
$D \rightarrow b$
$E \rightarrow c$
$F \rightarrow BC = Ba$
$G \rightarrow AB = AB$

- **COMPROBAR LA SIGUIENTE CADENA:** $w_1 = aaab$

Esta cadena no pertenece al lenguaje de la gramática, ya que todas las cadenas mayores de longitud dos, no pueden acabar con el símbolo b.

Aplicamos el algoritmo de CYK para comprobarlo:

	a	a	a	b
1	S, A, C	S, A, C	S, A, C	S, B, D
2	S, A	S, A	B, G	
3	S, A	B, G		
4	B, G			

Al hacer el algoritmo de CYK comprobamos que no aparece S en la última iteración, por lo que la cadena $w_1 = aaab$ no es aceptada.