

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid (LExp LExp)$

$Var \rightarrow x \mid y$

Care stringuri apartin limbajului gramaticii ?

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid (LExp LExp)$

$Var \rightarrow x \mid y$

Care stringuri apartin limbajului gramaticii?

Fiecare varianta corecta se puncteaza, fiecare varianta incorecta se depuncteaza

Select one or more:

☒ a.  $(\lambda x \cdot (\lambda x \cdot x))$



☒ b.  $(\lambda x \cdot y)$



☐ c.  $((\lambda x \cdot x) (\lambda x))$

Your answer is correct.

The correct answers are:  $(\lambda x \cdot y)$

,  $(\lambda x \cdot (\lambda x \cdot x))$

Se da gramatica cu productiile

$A \rightarrow ( Es) \quad Es \rightarrow e, Es \mid e$

Si simbolurile terminale  $e, (, )$  si ,

Se aplica LL(s) string . Alegeti afirmatiile adevarate .

Se da gramatica cu productiile

$A \rightarrow ( Es)$

$Es \rightarrow e, Es \mid e$

si simbolurile terminale  $e, (, )$  si ,

Se aplica LL(2) strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

Select one or more:

☒ a. Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$



☐ b. in analiza lui  $q_0$  se identifica o noua stare  $q_1 = [A \rightarrow \cdot (Es)]$

☐ c. in analiza lui  $q_0$  se identifica o noua stare  $q_1 = [Es \rightarrow \cdot e, Es]$

☒ d. Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{\quad} q_1$



☐ e. Din starea  $q_0$  se adauga productia  $q_0 e \rightarrow q_1$

Your answer is correct.

The correct answers are: Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$

, Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{\quad} q_1$

Care dintre expresiile regulate exprima limbajul  $\{w \mid w \text{ contine pe orice pozitie para } 0\}$

Care dintre expresiile regulate exprima limbajul  $\{w \mid w \text{ contine pe orice pozitie para } 0\}$

Alegerile gresite se penalizeaza

Select one or more:

- ☐ a.  $(0+1)^*0(1+0)^*$
- ☐ b.  $(101)^*$
- ☒ c.  $((0+1)0)^* (0+1+\epsilon)$  ✓
- ☒ d.  $((0+1)0)^* 1$  ✗

Your answer is partially correct.

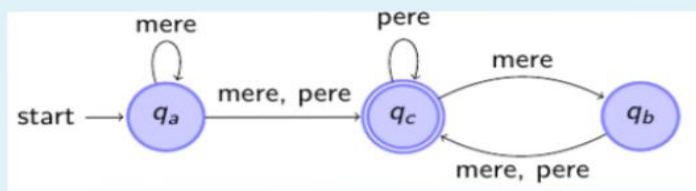
You have selected too many options.

The correct answer is:  $((0+1)0)^* (0+1+\epsilon)$

Ce fel de automat este?

Ce fel de automat este?

$$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$



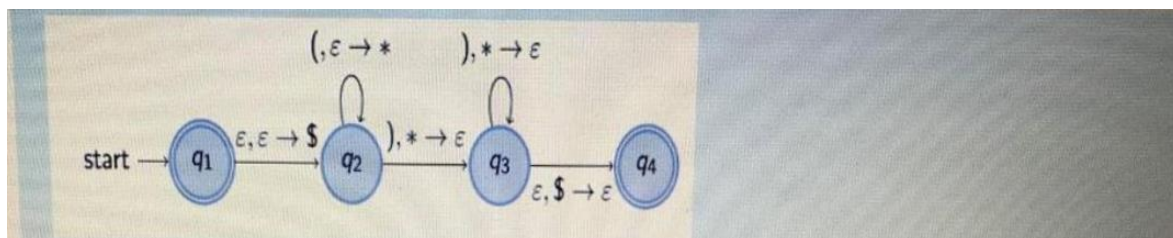
Select one or more:

- ☒ a. automat finit nedeterminist ✓
- ☐ b. automat finit determinist
- ☐ c. automat stiva

Your answer is correct.

The correct answer is: automat finit nedeterminist

Fie automatul



Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

Alegeți răspunsul corect

Select one or more:

- ☐ a. cand se citește un simbol (, se scoate de pe stiva \* și automatul trece în q3
- ☐ b. cand se citește un simbol (, se scoate de pe stiva \* și automatul ramane în starea q2
- ☒ c. cand se citește un simbol (, se introduce pe stiva \* și automatul ramane în starea q2 ✓

Your answer is correct.

The correct answer is: cand se citește un simbol (, se introduce pe stiva \* și automatul ramane în starea q2

Ce fel de automat este ?

Ce fel de automat este?

$$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$

$\delta$	mere	pere
$\rightarrow$	$q_a$	$q_c$
*	$q_c$	$q_c$
	$q_b$	$q_c$

Select one or more:

- ☐ a. automat finit determinist
- ☐ b. automat stiva
- ☒ c. automat finit nedeterminist ✓

Your answer is correct.

The correct answer is: automat finit nedeterminist

Se da gramatica cu productiile , food modifier quality stiva top-down fara lookahead

Se da gramatica G cu productiile

S → Food 'is' Quality  
 Food → Modifier Food | 'pizza' | 'avocado'  
 Modifier → 'large'|'big'|'small'|'this'|'that'  
 Quality → 'fresh'|'healthy'|'expensive'|'not' Quality

si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se construiesc automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

S q →  
 Quality  
 'is' este o productie din acest automat prin care se scoate un simbol de pe stiva si se inlocuieste cu partea dreapta a unei productii din gramatica  
 Food ✗  
 q  
 'large'  
 q nu este productie in acest automat  
 'large' ✗  
 → q

Your answer is incorrect.

The correct answer is: S q → Quality 'is' Food q → este o productie din acest automat prin care se inlocuieste un simbol de pe stiva cu partea dreapta a unei productii din gramatica in ordinea inversata, 'large' q 'large' → q → este o productie din acest automat de tip consum de la intrare pe baza stivei

Activa  
Go to S

Care dintre expresiile regulate exprima limbajul  $\{w \mid w \text{ contine pe orice pozitie para } 0\}$

Care dintre expresiile regulate exprima limbajul  $\{w \mid w \text{ contine pe orice pozitie para } 0\}$

Alegerile gresite se penalizeaza

Select one or more:

☐ a.  $(101)^*$

☒ b.  $((0+1)0)^* (0+1+\epsilon)$  ✓

☐ c.  $(0+1)^* 0 (1+0)^*$

☒ d.  $((0+1)0)^* 1$  ✗

Your answer is partially correct.

You have selected too many options.

The correct answer is:  $((0+1)0)^* (0+1+\epsilon)$

Se dau gramaticile  $G1 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii1})$ ,  $G2 = (\{e, ', '\}, \{P, P2\}, Ps, \text{Productii2})$ ,

Se dau gramaticile  $G1 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii1})$ ,  $G2 = (\{e, ', '\}, \{P, P2\}, Ps, \text{Productii2})$ ,

$\text{Productii1} = \{Ps \rightarrow P \ Ps \mid \epsilon\}$

$P \rightarrow e \ ', '\}$

$\text{Productii2} = \{Ps \rightarrow P \ Ps \mid P\}$

$P \rightarrow e \ ', '\}$

Care afirmatii sunt adevarate?

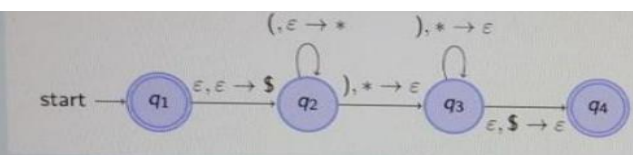
e,e,e apartine ☐ nici  $L(G1)$  nici  $L(G2)$  ☒ ☐

e,e,e, apartine ☐ atat  $L(G1)$ , cat si  $L(G2)$  ☒ ☐

Your answer is correct.

The correct answer is: e,e,e apartine  $\rightarrow$  nici  $L(G1)$  nici  $L(G2)$ , e,e,e, apartine  $\rightarrow$  atat  $L(G1)$ , cat si  $L(G2)$

Fie automatul



Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

Alegeti raspunsul corect

Select one or more:

- ☐ a. cand se citește un simbol (, se scoate de pe stiva \* si automatul trece in q3
- ☐ b. cand se citește un simbol (, se scoate de pe stiva \* si automatul ramane in starea q2
- ☒ c. cand se citește un simbol (, se introduce pe stiva \* si automatul ramane in starea q2 ☒

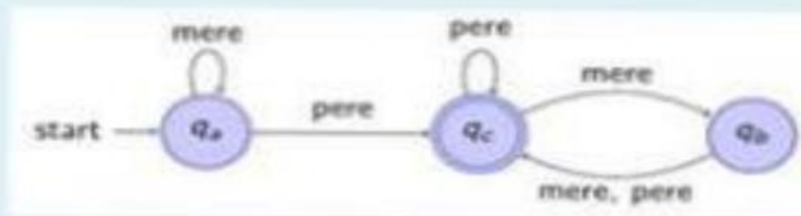
Your answer is correct.

The correct answer is: cand se citește un simbol (, se introduce pe stiva \* si automatul ramane in starea q2

Ce fel de automat este ?

Ce fel de automat este?

$$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$



Select one or more:

- ☒ a. automat finit determinist ✓
- ☐ b. automat finit nedeterminist
- ☐ c. automat stiva

Your answer is correct.

The correct answer is: automat finit determinist



Se da gramatica G cu productiile

Se da gramatica G cu productiile

S-> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

Si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se da gramatica G cu productiile

S -> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se construiesc automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

Daca varful stivei e 'large' se va folosi productia 'large' q 'large' -> q pt a consuma de la intrare un 'large' ✓

Daca varful stivei e Modifier si in automat avem productiile Modifier q -> 'large' q | 'big' q | 'small' q | 'this' q | 'that' q se vor incerca toate productiile de forma Modifier q -> anything pana cand fie se accepta stringul, fie se respinge; nu exista criteriu de alegere in acest automat. ✓

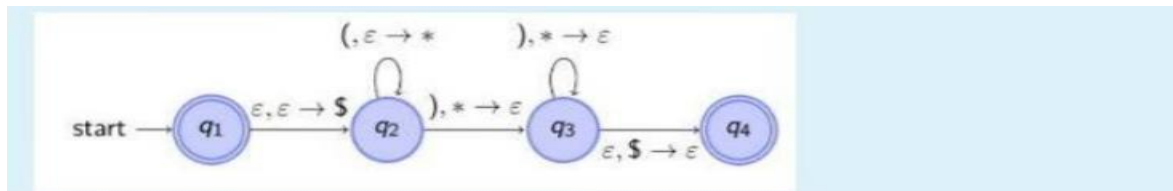
Your answer is correct.

The correct answer is: Daca varful stivei e 'large' -> se va folosi productia 'large' q 'large' -> q pt a consuma de la intrare un 'large'; Daca varful stivei e Modifier si in automat avem productiile Modifier q -> 'large' q | 'big' q | 'small' q | 'this' q | 'that' q -> se vor incerca toate productiile de forma Modifier q -> anything pana cand fie se accepta stringul, fie se respinge; nu exista criteriu de alegere in acest automat.

Activate Windows  
Go to Settings to activate Windows.

Activat  
Go to Settings to activate Windows.

Fie automatul





Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

Alegeti raspunsul corect

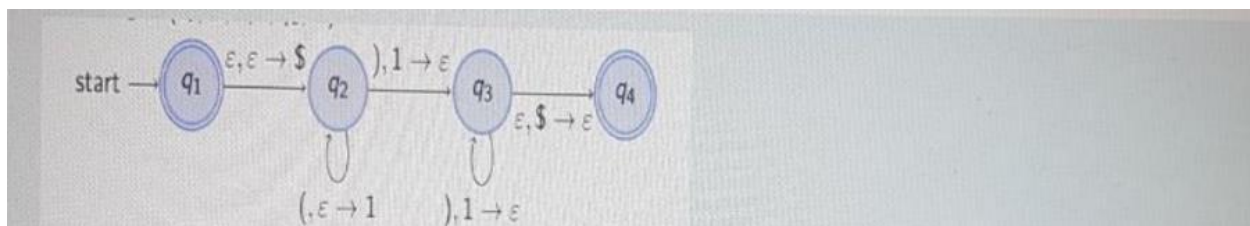
Select one or more:

- ☒ a. este un automat finit determinist 
- ☐ b. este un automat finit nedeterminist
- ☒ c. este un automat stiva 

Your answer is incorrect.

The correct answer is: este un automat stiva

Fie automatul



Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

Alegeti raspunsul corect

Select one or more:

- ☐ a. in starea q2 citeste ) doar daca poate scoate de pe stiva (
- ☒ b. in starea q2 citeste ) doar daca poate scoate de pe stiva un 1, altfel stringul e respins ✓
- ☐ c. in starea q2 citeste ) doar daca poate scoate de pe stiva un 1, altfel revine in q1

Your answer is correct.

The correct answer is: in starea q2 citeste ) doar daca poate scoate de pe stiva un 1, altfel stringul e respins



Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{LExp, Var\}, \{LExp, P\})$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp) .$

$Var \rightarrow x \mid y$

Completati (prin drag & drop ) derivarea cea mai din stanga pentru stringul  $(\lambda x.y y)$

$x$   $y$   $)$   $LExp$   $\lambda$   $($

Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{LExp, Var\}, \{LExp, P\})$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din stanga pentru stringul  $(\lambda x.y y)$

$LExp \Rightarrow ($   $LExp$   $\checkmark$   $LExp$   $\checkmark$   $) \Rightarrow (\lambda Var.$   $LExp$   $\checkmark$   $LExp$   $\checkmark$   $) \Rightarrow^* (\lambda x.y y)$

Your answer is correct.

Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{LExp, Var\}, \{LExp, P\})$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp) .$

$Var \rightarrow x \mid y$

Completati (prin drag & drop ) derivarea cea mai din stanga pentru stringul  $(\lambda x.x x)$

$y$   $x$   $LExp$   $)$   $($   $Var$

Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{LExp, Var\}, \{LExp, P\})$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din stanga pentru stringul  $(\lambda x.x x)$

$LExp \Rightarrow ($   $LExp$   $LExp$   $\checkmark$   $) \Rightarrow (\lambda$   $Var$   $\checkmark$   $.$   $LExp$   $\checkmark$   $LExp$   $\checkmark$   $) \Rightarrow^* (\lambda x.x x)$

Your answer is correct.

Se da gramatica G cu productiile

forst  $\rightarrow$  FOR vardecl '=' E 'to' stepE sts

stepE  $\rightarrow$  STEP E | epsilon

sts  $\rightarrow$  'S' sts | epsilon

STEP  $\rightarrow$  'step'

FOR  $\rightarrow$  'for'

Vardecl  $\rightarrow$  'a1' | 'a2'

E  $\rightarrow$  'e1' | 'e2' | 'e3'

Terminalele sunt toate stringurile marcate intre "

Se construiește automatul stiva in abordarea simpla top-down

Se da gramatica G cu productiile

forst  $\rightarrow$  FOR vardecl '=' E 'to' E stepE sts

stepE  $\rightarrow$  STEP E | epsilon

sts  $\rightarrow$  'S' sts | epsilon

STEP  $\rightarrow$  'step'

FOR  $\rightarrow$  'for'

vardecl  $\rightarrow$  'a1' | 'a2'

E  $\rightarrow$  'e1' | 'e2' | 'e3'

terminalele sunt toate stringurile marcate intre "

Se construiește automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

'e1' q

'e1' - nu este productie in acest automat

> q

stepE

q ->

E

STEP

q

Your answer is partially correct.

You have correctly selected 1.

The correct answer is: 'e1' q 'e1'  $\rightarrow$  q  $\rightarrow$  este o productie din acest automat de tip consum de la intrare pe baza stivei, stepE q  $\rightarrow$  E STEP q  $\rightarrow$  este o productie din acest automat prin care se inlocuieste un simbol de pe stiva cu partea dreapta a unei productii din gramatica in ordinea inversata

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

Si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(2) strong. Alegeti afirmatiile adevarate.

Presupunem ca s-a ajuns cu analiza la starea  $q_3 = [Es \rightarrow .e, Es]$ .

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(2) strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

Presupunem ca s-a ajuns cu analiza la starea  $q_3 = [Es \rightarrow .e, Es]$ .

Select one or more:

- ☒ a. Noua starea  $q_4$  este  $[Es \rightarrow e., Es]$  ✓
- ☒ b. se adauga productia  $q_3e \rightarrow q_4$  ✓
- ☐ c. se adauga productia  $q_3e \rightarrow q_4e$
- ☒ d. starea  $q_3$  corespunde unei stari in care se poate consuma  $e$  fara nicio verificare suplimentara ✓
- ☐ e. starea  $q_3$  corespunde unei stari in care se foloseste un lookahead pentru a verifica tranzitia in  $q_4$

Your answer is correct.

The correct answers are: Noua starea  $q_4$  este  $[Es \rightarrow e., Es]$

, se adauga productia  $q_3e \rightarrow q_4$

, starea  $q_3$  corespunde unei stari in care se poate consuma  $e$  fara nicio verificare suplimentara

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde  $P$  contine

$Song = do \ SongDo \mid mi \ SongMi \mid sol \ SongSol$

$SongDo = mi \ SongMi \mid sol \ SongSol$

$SongMi = mi \ SongMi \mid sol \ SongSol \mid do \ SongDo$

$SongSol = sol \ SongSol \mid mi \ SongMi \mid \epsilon$

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde  $P$  contine

$Song = do \ SongDo \mid mi \ SongMi \mid sol \ SongSol$

$SongDo = mi \ SongMi \mid sol \ SongSol$

$SongMi = mi \ SongMi \mid sol \ SongSol \mid do \ songDo$

$SongSol = sol \ SongSol \mid mi \ SongMi \mid \epsilon$

Notatia folosita pt expresiile regulate este cea conceptuala, nu cea din Lex

Alegeti afirmatiile corecte.

Alegerile gresite se penalizeaza.

Select one or more:

- ☐ a. Limbajul gramaticii este acelasi cu al expresiei regulate  $(do + mi + sol)^*$
- ☒ b. Limbajul gramaticii nu este acelasi cu al expresiei regulate  $(do + mi + sol)(do + mi + sol)^*$  ✓
- ☐ c. Limbajul gramaticii este acelasi cu al expresiei regulate  $do^*mi^*sol^*$
- ☐ d. Limbajul gramaticii este acelasi cu al expresiei regulate  $do + mi + sol$

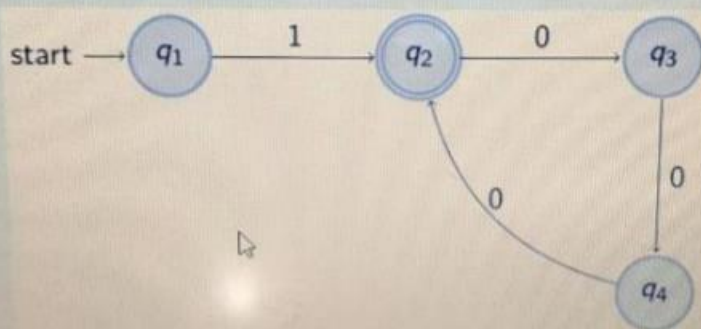
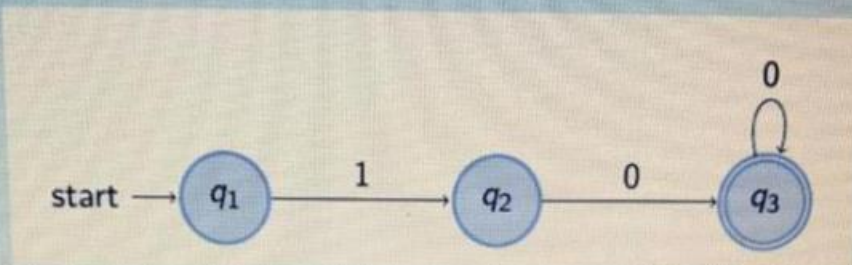
Your answer is correct.

The correct answer is: Limbajul gramaticii nu este acelasi cu al expresiei regulate  $(do + mi + sol)(do + mi + sol)^*$



Se dau urmatoarele doua automate

Se dau urmatoarele doua automate



Alegeti afirmatiile corecte. Alegeri incorecte se penalizeaza

Select one or more:

- ☒ a. sunt diferite, primul este echivalent cu  $100^*$ , in timp ce al doilea accepta si 1 ✓
- ☐ b. orimul are acelasi limbaj cu  $10^*$
- ☐ c. au acelasi limbaj

Your answer is correct.

When several answers could be correct, the first is equivalent to  $100^*$  in timp ce al doilea accepta si 1.



Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{L, V\}, L, P)$ , unde  $P$  contine

$L \rightarrow V \mid \lambda V . L \mid ( L L )$

$Var \rightarrow x \mid y$

Se da gramatica  $G = (\{\lambda, x, y, ., (, )\}, \{L, V\}, L, P)$ , unde  $P$  contine

$L \rightarrow V \mid \lambda V . L \mid ( L L )$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din dreapta pentru stringul  $(x y)$

$L \Rightarrow ( L L ) \Rightarrow ( L \oplus V ) \Rightarrow ( L \oplus y ) \Rightarrow (x y)$

Your answer is correct.

Ce fel de automat este?

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Ce fel de automat este?

$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$

```

graph LR
    start((start)) --> qa((qa))
    qa -- mere --> qa
    qa -- pere --> qc((qc))
    qc -- pere --> qc
    qc -- mere --> qb((qb))
    qb -- "mere, pere" --> qc
  
```

Select one or more:

- ☒ a. automat finit determinist ✓
- ☐ b. automat stiva
- ☐ c. automat finit nedeterminist

Your answer is correct.

The correct answer is: automat finit determinist

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e , Es \mid e$

Si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(2) strong.

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e , Es \mid e$

si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(2) strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

Select one or more:

- ☒ a. Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$   
✓
- ☐ b. in analiza lui  $q_0$  se identifica o noua stare  $q_1 = [A \rightarrow \cdot (Es)]$
- ☐ c. in analiza lui  $q_0$  se identifica o noua stare  $q_1 = [Es \rightarrow \cdot e, Es]$
- ☒ d. Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{\cdot} q_1$   
✓
- ☐ e. Din starea  $q_0$  se adauga productia  $q_0 e \rightarrow q_1$

Your answer is correct.

The correct answers are: Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$

, Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{\cdot} q_1$

Se da A1

Se da A1

$$A_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$

$\delta$	mere	pere
$\rightarrow$	$q_a$	$q_c$
*	$q_c$	$q_c$
	$q_b$	$q_c$

si A2:

Alegeti raspunsul corect

Select one or more:

- a.  $L(A_1)$  intersectat cu  $L(A_2)$  contine cateva elemente, dar  $L(A_1)$  nu e egal cu  $L(A_2)$
- b.  $L(A_1)$  intersectat cu  $L(A_2) =$  multimea vida
- c.  $L(A_1) = L(A_2)$  ✓

Your answer is correct.  
The correct answer is:  $L(A_1) = L(A_2)$

Stiind ca gramatica de mai jos are doua terminale, \ si / , simbolul de start este X , iar  $L(G) = \{ \text{string cu numar egal de } \vee \text{ si } \wedge \}$ , completati spatiile libere , in fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal .

Z epsilon / Y \

Stiind ca gramatica de mai jos are doua terminale, \ si / , simbolul de start este X , iar  $L(G) = \{ \text{string cu numar egal de } \vee \text{ si } \wedge \}$ , completati spatiile libere. In fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal.

$X \rightarrow$  \ ☒ N Y ☒ / Z

$Y \rightarrow$  \ Y ☒ / Y \

$Z \rightarrow$  \ Z ☒ / Z ☒ \

Your answer is partially correct.

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

$x \mid LExp \mid \lambda \mid y$

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var . LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din stanga pentru stringul  $(\lambda x.y \cdot y)$

$LExp \Rightarrow ($   ☒  $LExp$  ☒  $) \Rightarrow (\lambda Var.$

☒  ☒  $) \Rightarrow^* (\lambda x.y \cdot y)$

Your answer is correct.

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde  $P$  contine

$Song = do \ SongDo \mid mi \ SongMi \mid sol \ SongSol$

$SongDo = mi \ SongMi \mid sol \ SongSol$

$SongMi = mi \ SongMi \mid sol \ SongSol \mid do \ SongDo$

$SongSol = sol \ SongSol \mid mi \ SongMi \mid \epsilon$

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde  $P$  contine

$Song = do \ SongDo \mid mi \ SongMi \mid sol \ SongSol$

$SongDo = mi \ SongMi \mid sol \ SongSol$

$SongMi = mi \ SongMi \mid sol \ SongSol \mid do \ songDo$

$SongSol = sol \ SongSol \mid mi \ SongMi \mid \epsilon$

Care afirmatii sunt adevarate?

Fiecare varianta corecta se puncteaza, fiecare varianta incorecta se depuncteaza

For fun after the test testati cantecelele ce apartin limbajului gramaticii date <https://www.musicca.com/piano>, tinand cont ca do= C, mi=E, sol=G.

Select one or more:

- ☒ a. do mi sol mi sol - apartine limbajului gramaticii ✓
- ☒ b. mi mi sol sol - apartine limbajului gramaticii ✓
- ☒ c. do mi sol - apartine limbajului gramaticii ✓
- ☒ d. unele stringuri care apartin limbajului gramaticii contin aceeasi nota repetata de mai multe ori ✓
- ☐ e. niciunul dintre stringurile care apartin limbajului gramaticii nu contine aceeasi nota repetata de mai multe ori
- ☐ f. toate stringurile care apartin limbajului trebuie sa inceapa cu do

Your answer is correct.

The correct answers are: do mi sol - apartine limbajului gramaticii, do mi sol mi sol - apartine limbajului gramaticii, mi mi sol sol - apartine limbajului gramaticii, unele stringuri care apartin limbajului gramaticii contin aceeasi nota repetata de mai multe ori



Se dau urmatoarele doua automate

Se dau urmatoarele doua automate

Alegeti afirmatiile corecte. Alegeri incorecte se penalizeaza

Select one or more:

- ☐ a. oricul are acelasi limbaj cu  $10^*$
- ☐ b. au acelasi limbaj
- ☒ c. sunt diferite, primul este echivalent cu  $100^*$ , in timp ce al doilea accepta si 1 ✓

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{L, V\}, L, P)$ , unde P contine

$L \rightarrow V \mid \lambda V \cdot L \mid ( L L )$

$Var \rightarrow x \mid y$

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{L, V\}, L, P)$ , unde P contine

$L \rightarrow V \mid \lambda V \cdot L \mid ( L L )$

$Var \rightarrow x \mid y$

Completati (prin drag &drop) derivarea cea mai din dreapta pentru stringul (x y)

$L \Rightarrow ( L L ) \Rightarrow ( L \cdot V \cdot ) \Rightarrow ( L \cdot y \cdot ) \Rightarrow^*(x y)$

Your answer is correct.

Ce fel de automat este?

Ce fel de automat este?

$$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$

$\delta$	mere	pere
$\rightarrow q_a$	$\{q_a, q_b\}$	$q_c$
$* q_c$	$q_b$	$q_c$
$q_b$	$q_c$	$q_c$

Select one or more:

☐ a. automat finit determinist

☒ b. automat finit nedeterminist ✓

☐ c. automat stiva

Your answer is correct.

The correct answer is: automat finit nedeterminist

Se dau gramaticile  $G_1 = (\{e, ', '\}, \{P, P_2\}, P_s, \text{Productii1})$ ,  $G_2 = (\{e, ', '\}, \{P, P_s\}, P_s, \text{Productii2})$ ,

Se dau gramaticile  $G_1 = (\{e, ', '\}, \{P, P_s\}, P_s, \text{Productii1})$ ,  $G_2 = (\{e, ', '\}, \{P, P_s\}, P_s, \text{Productii2})$ ,

Productii1 =  $\{P_s \rightarrow P P_s \mid \epsilon$

$P \rightarrow e ', '\}$

Productii2 =  $\{P_s \rightarrow P P_s \mid P$

$P \rightarrow e ', '\}$

Care afirmatii sunt adevarate?

e,e,e apartine nici  $L(G_1)$  nici  $L(G_2)$  ⇌ ✓

e,e,e, apartine atat  $L(G_1)$ , cat si  $L(G_2)$  ⇌ ✓

Your answer is correct.

The correct answer is: e,e,e apartine → nici  $L(G_1)$  nici  $L(G_2)$ , e,e,e, apartine → atat  $L(G_1)$ , cat si  $L(G_2)$

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

Si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(k) strong. Alegeti afirmatiile adevarate.

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

si simbolurile terminale  $e, (, )$  si  $,$

Se aplica LL(k) strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

Select one or more:

- ☐ a. Pentru  $k=0$ , automatul rezultat este nedeterminist
- ☒ b. Pentru  $k=1$ , automatul rezultat este nedeterminist ✓
- ☐ c. Pentru  $k=1$ , automatul rezultat este determinist
- ☐ d. un lookahead de 1 singur terminal este suficient
- ☒ e. este necesar cel putin un lookahead de 2 simboluri terminale ✓

Your answer is correct.

The correct answers are: Pentru  $k=1$ , automatul rezultat este nedeterminist, este necesar cel putin un lookahead de 2 simboluri terminale

Se da gramatica G cu productiile

forst -> FOR vardecl '=' E 'to' stepE sts

stepE -> STEP E | epsilon

sts -> 'S' sts | epsilon

STEP -> 'step'

FOR -> 'for'

Vardecl -> 'a1' | 'a2'

E -> 'e1' | 'e2' | 'e3'

Tot ce e marcat intre ' este terminal

Se construiesc automatul stiva in abordarea simpla top-down

Se da gramatica G cu productiile

forst -> FOR vardecl '=' E 'to' E stepE sts

stepE -> STEP E | epsilon

sts -> 'S' sts | epsilon

STEP -> 'step'

FOR -> 'for'

vardecl -> 'a1' | 'a2'

E -> 'e1' | 'e2' | 'e3'

tot ce e marcat intre ' este terminal

Se construiesc automatul stiva in abordarea simpla top-down  
(descendentă) (algoritmul fara lookahead).

'for'

q

este o productie prin care se mentioneaza ca se poate consuma ori

'for'

-> q

sts

q -

>

este o productie din automat; in momentul folosirii, se prezice ca se

sts

'S'

q

Your answer is partially correct.

You have correctly selected 1.

The correct answer is: 'for' q 'for' -> q -- este o productie din acest  
automat de tip consum de la intrare pe baza stivei, sts q -> sts 'S'  
q -- este o productie din automat; in momentul folosirii, se prezice  
ca se va identifica in stringul de la intrare un string derivat din 'S'  
sts

Stiind ca gramatica de mai jos are doua terminale, \ si /, simbolul de start este X, iar  $L(G) = \{\text{string cu numar egal de } \backslash \text{ si } /\}$ , completati spatiile libere.

Y / \ epsilon Z

Stiind ca gramatica de mai jos are doua terminale, \ si /, simbolul de start este X, iar  $L(G) = \{\text{string cu numar egal de } \backslash \text{ si } /\}$ , completati spatiile libere. In fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal.

$X \rightarrow$  Z ✗ N Y | Y ✗ / Z

$Y \rightarrow$  \ Y | / Y | \

$Z \rightarrow$  \ Z | / Z | / ✓

Your answer is partially correct.

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde P contine

$Song = do SongDo \mid mi SongMi \mid sol SongSol$

$SongDo = mi SongMi \mid sol SongSol$

$SongMi = mi SongMi \mid sol SongSol \mid do SongDo$

$SongSol = sol SongSol \mid mi SongMi \mid \epsilon$

Se da gramatica  $G = (\{do, mi, sol\}, \{Song, SongDo, SongMi, SongSol\}, Song, P)$ , unde P contine

$Song = do SongDo \mid mi SongMi \mid sol SongSol$

$SongDo = mi SongMi \mid sol SongSol$

$SongMi = mi SongMi \mid sol SongSol \mid do SongDo$

$SongSol = sol SongSol \mid mi SongMi \mid \epsilon$

Notatia folosita pt expresiile regulate este cea conceptuala, nu cea din Lex

Alegeti afirmatiile corecte.

Alegerile gresite se penalizeaza.

Select one or more:

☐ a. Limbajul gramaticii este acelasi cu al expresiei regulate  $(do + mi + sol)^*$

☒ b. Limbajul gramaticii nu este acelasi cu al expresiei regulate  $(do + mi + sol)(do + mi + sol)^*$  ✓

☐ c. Limbajul gramaticii este acelasi cu al expresiei regulate  $do^*mi^*sol^*$

☐ d. Limbajul gramaticii este acelasi cu al expresiei regulate  $do + mi + sol$

Your answer is correct.

The correct answer is: Limbajul gramaticii nu este acelasi cu al expresiei regulate  $(do + mi + sol)(do + mi + sol)^*$



Stiind ca gramatica de mai jos este o gramatica regulata al carei limbaj contine toate stringurile formate din 0 si 1 mai putin cele care incep cu 0 sau care se termina cu 1 .

C
1
epsilon
0
B
A

Stiind ca gramatica de mai jos este o gramatica regulata al carei limbaj contine toate stringurile formate din 0 si 1 mai putin cele care incep cu 0 sau care se termina cu 1. Completati spatiile libere. In fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal.

A-> 1 ✓ B

B-> 0 ✓ B ✓ | 1 B

B ✓ -> 0

Your answer is correct.

Se dau gramaticile  $G_1 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii1})$ ,  $G_2 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii2})$ ,

Se dau gramaticile  $G_1 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii1})$ ,  $G_2 = (\{e, ', '\}, \{P, Ps\}, Ps, \text{Productii2})$ .

Productii1 = {  $Ps \rightarrow P \ Ps$  |  $\epsilon$   
 $P \rightarrow e \ ', '\}$

Productii2 = {  $Ps \rightarrow P \ Ps$  |  $P$   
 $P \rightarrow e \ ', '\}$

Care afirmatii sunt adevarate?

e,e,e apartine L(G2), dar nu si L(G1) ✖

e,e,e, apartine L(G1), dar nu si L(G2) ✖

Your answer is incorrect.

The correct answer is: e,e,e apartine → nici L(G1) nici L(G2), e,e,e, apartine → atat L(G1), cat si L(G2)

Se da gramatica cu productiile  
 $A \rightarrow ( Es )$   
 $Es \rightarrow e, Es \mid e$   
 si simbolurile terminale  $e, (, )$  si  $,$   
 Se aplica LL(2) strong

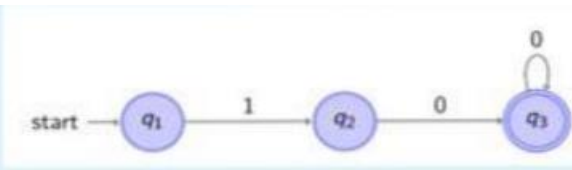
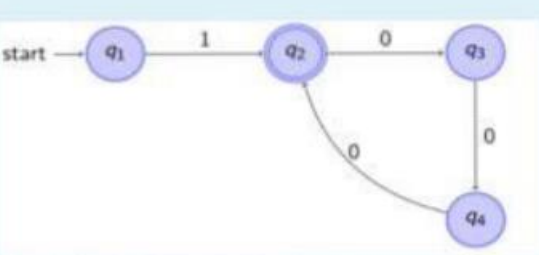
Se da gramatica cu productiile  
 $A \rightarrow ( Es )$   
 $Es \rightarrow e, Es \mid e$   
 si simbolurile terminale  $e, (, )$  si  $,$   
 Se aplica LL(2) strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

Select one or more:

- ☒ a. Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$  ✓
- ☐ b. In analiza lui  $q_0$  se identifica o noua stare  $q_1 = [A \rightarrow \cdot (Es)]$
- ☐ c. In analiza lui  $q_0$  se identifica o noua stare  $q_1 = [Es \rightarrow \cdot e, Es]$
- ☒ d. Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{0} q_1$  ✓
- ☐ e. Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{e} q_1$

Your answer is partially correct.  
 You have correctly selected 2.  
 The correct answers are: Se porneste de la situatia  $q_0 = [A \rightarrow \cdot (Es)]$   
 , in analiza lui  $q_0$  se identifica o noua stare  $q_1 = [A \rightarrow \cdot (Es)]$   
 , Din starea  $q_0$  se adauga productia  $q_0 \xrightarrow{0} q_1$

Se dau urmatoarele 2 automate

Alegeti afirmatiile corecte. Alegeri incorecte se penalizeaza

Select one or more:

- ☐ a. oricul are acelasi limbaj cu  $10^*$
- ☐ b. au acelasi limbaj
- ☒ c. sunt diferite, primul este echivalent cu  $100^*$ , in timp ce al doilea accepta si 1 ✓

Se da gramatica G cu productiile

S-> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

Si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se da gramatica G cu productiile

S -> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se construiesc automatul stiva in abordarea simpla top-down (descendentă) (algoritmul fara lookahead).

Daca varful stivei e  
'large'

Choose...

Daca varful stivei e  
Modifier si in  
automat avem

productiile Modifier

Choose...

q -> 'large' q | 'big'

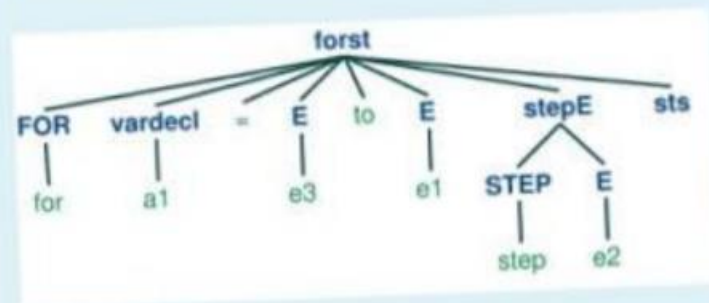
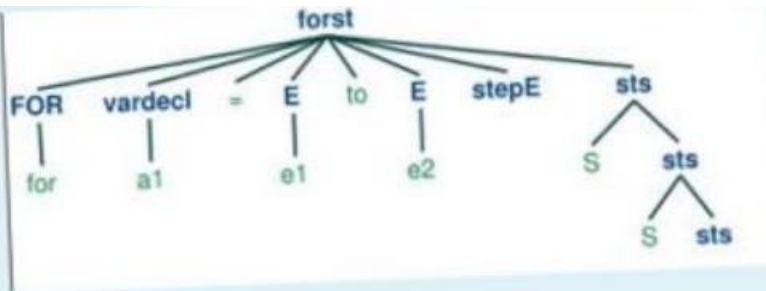
q | 'small' q | 'this' q

| 'that' q

Your answer is incorrect.

The correct answer is: Daca varful stivei e 'large' → se va folosi productia 'large' q 'large' -> q pt a consuma de la intrare un 'large', Daca varful stivei e Modifier si in automat avem productiile Modifier q -> 'large' q | 'big' q | 'small' q | 'this' q | 'that' q → se vor incerca toate productiile de forma Modifier q -> anything pana cand fie se accepta stringul, fie se respinge; nu exista criteriu de alegere in acest automat

Completati gramatica



forst -> FOR vardecl '=' E 'to' E  ☒ stepE  ☒ sts

stepE -> STEP E| epsilon ☒

sts -> 'S' sts| epsilon ☒

STEP -> 'step'

FOR -> 'for'

forst -> 'for' 'a1' '=' 'e3' 'to' 'e1' 'step' 'step' 'e2' 'sts'

Ce fel de automat este?

Ce fel de automat este?

$$D_1 = (\{mere, pere\}, \{q_a, q_c, q_b\}, \delta, q_a, \{q_c\})$$

$\delta$	mere	pere
$\rightarrow$	$q_a$	$q_c$
$*$	$q_c$	$q_c$
	$q_b$	$q_c$

Select one or more:

- ☐ a. automat finit determinist
- ☐ b. automat stiva
- ☒ c. automat finit nedeterminist ✓

Your answer is correct.

The correct answer is: automat finit nedeterminist

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde P contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

x } y Var LExp {

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde P contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din stanga pentru stringul  $(\lambda x.x \cdot x)$

$LExp \Rightarrow ( LExp LExp ) \Rightarrow ( \lambda Var \cdot LExp LExp ) \Rightarrow * (\lambda x.x \cdot x)$

Your answer is correct.



Fie automatul

Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

Alegeti raspunsul corect

Select one or more:

- ☒ a. este un automat stiva ✓
- ☐ b. este un automat finit nedeterminist
- ☐ c. este un automat finit determinist

Your answer is correct.

The correct answer is: este un automat stiva

Se da gramatica G cu productiile

forst  $\rightarrow$  FOR vardecl 'a' E 'to' E stepE sts

stepE  $\rightarrow$  STEP E | epsilon

sts  $\rightarrow$  'S' sts | epsilon

STEP  $\rightarrow$  'step'

FOR  $\rightarrow$  'for'

Vardecl  $\rightarrow$  'a1' | 'a2'

E  $\rightarrow$  'e1' | 'e2' | 'e3'

Tot ce e marcat intre ' este terminal

Se construiesc automatul stiva in abordarea simpla top-down

Se da gramatica G cu productiile

forst  $\rightarrow$  FOR vardecl 'a' E 'to' E stepE sts

stepE  $\rightarrow$  STEP E | epsilon

sts  $\rightarrow$  'S' sts | epsilon

STEP  $\rightarrow$  'step'

FOR  $\rightarrow$  'for'

vardecl  $\rightarrow$  'a1' | 'a2'

E  $\rightarrow$  'e1' | 'e2' | 'e3'

terminalele sunt toate stringurile marcate intre "

Se construiesc automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

stepE q $\rightarrow$ E STEP q	este o productie din acest automat prin care se inlocuieste un simbol de pe stiva cu partea dreapta a unei productii din gramatica in ordinea inversata	\$ ✓
'e1' q 'e1' $\rightarrow$ q	este o productie din acest automat de tip consum de la intrare pe baza stivei	\$ ✓

Your answer is correct.

The correct answer is: stepE q  $\rightarrow$  E STEP q  $\rightarrow$  este o productie din acest automat prin care se inlocuieste un simbol de pe stiva cu partea dreapta a unei productii din gramatica in ordinea inversata, 'e1' q 'e1'  $\rightarrow$  q  $\rightarrow$  este o productie din acest automat de tip consum de la intrare pe baza stivei

Activate Windows

Stiind ca gramatica de mai jos este o gramatica regulata al carei limbaj contine toate stringurile formate din 0 si 1 mai putin cele care incep cu 0 sau care se termina cu 1.

epsilon 1 B 0 A C

Stiind ca gramatica de mai jos este o gramatica regulata al carei limbaj contine toate stringurile formate din 0 si 1 mai putin cele care incep cu 0 sau care se termina cu 1. Completati spatiile libere. In fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal.

A → 1 B

B → epsilon 0 C B | 1 B

C → B 0

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde P contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

LExp y x ) λ (

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde P contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Completati (prin drag & drop) derivarea cea mai din stanga pentru stringul  $(\lambda x y \cdot y)$

$LExp \Rightarrow ( LExp LExp ) \Rightarrow (\lambda Var. LExp LExp ) \Rightarrow (\lambda x y y)$

Your answer is correct.

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Se da gramatica  $G = (\{\lambda, x, y, \cdot, (, )\}, \{LExp, Var\}, LExp, P)$ , unde  $P$  contine

$LExp \rightarrow Var \mid \lambda Var \cdot LExp \mid ( LExp LExp)$

$Var \rightarrow x \mid y$

Care stringuri apartin limbajului gramaticii?

Fiecare varianta corecta se puncteaza, fiecare varianta incorecta se depuncteaza

Select one or more:

- ☒ a.  $(x y)$  ✓
- ☐ b.  $x y$
- ☐ c.  $(x \cdot y)$
- ☒ d.  $(\lambda y \cdot y)$  ✓

Your answer is correct.

The correct answers are:  $(x y)$ ,  $(\lambda y \cdot y)$

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

si simbolurile terminale  $e, (, )$  si ,

Se aplica  $LL(k)$  strong

Se da gramatica cu productiile

$A \rightarrow ( Es )$

$Es \rightarrow e, Es \mid e$

si simbolurile terminale  $e, (, )$  si ,

Se aplica  $LL(k)$  strong. Alegeti afirmatiile adevarate. Alegerile incorecte se penalizeaza

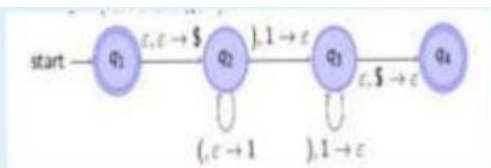
Select one or more:

- ☒ a. Pentru  $k=0$ , automatul rezultat este nedeterminist ✓
- ☐ b. Pentru  $k=1$ , automatul rezultat este nedeterminist
- ☒ c. Pentru  $k=1$ , automatul rezultat este determinist ✗
- ☒ d. un lookahead de 1 singur terminal este suficient ✗
- ☐ e. este necesar cel putin un lookahead de 2 simboluri terminale

Your answer is incorrect.

The correct answers are: Pentru  $k=0$ , automatul rezultat este nedeterminist, Pentru  $k=1$ , automatul rezultat este nedeterminist, este necesar cel putin un lookahead de 2 simboluri terminale

Fie automatul



Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$

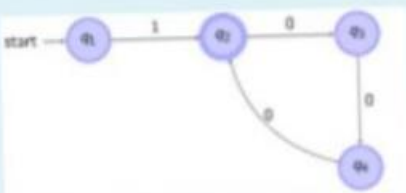
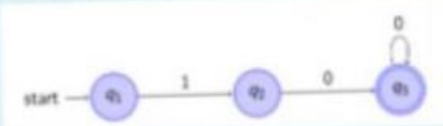
Alegeti raspunsul corect

Select one or more:

- ☐ a. in starea q2 citeste ) doar daca poate scoate de pe stiva un 1, altfel revine in q1
- ☒ b. in starea q2 citeste ) doar daca poate scoate de pe stiva un 1, altfel stringul e respins ✓
- ☐ c. in starea q2 citeste ) doar daca poate scoate de pe stiva (

Se dau urmatoarele doua automate

Se dau urmatoarele doua automate



Alegeti afirmatiile corecte. Alegeri incorecte se penalizeaza

Select one or more:

- ☐ a. primul are acelasi limbaj cu 10\*
- ☐ b. au acelasi limbaj
- ☒ c. sunt diferite, primul este echivalent cu 100\*, in timp ce al doilea accepta si 1 ✓

Your answer is correct.

The correct answer is: sunt diferite, primul este echivalent cu 100\*, in timp ce al doilea accepta si 1

Se da A1

Se da A1

$$A_1 = \{(mere, pere), \{q_1, q_2, q_3\}, \delta, q_1, \{q_1\}\}$$

$\delta$	mere	pere
$\rightarrow$	$q_1$	$q_1$
$\rightarrow$	$q_2$	$q_2$
$\rightarrow$	$q_3$	$q_3$

si A2:

Alagati raspunsul corect

Select one or more:

- ☒ a. L(A1) intersectat cu L(A2) contine cateva elemente, dar L(A1) nu e egal cu L(A2) ✓
- ☐ b. L(A1) = L(A2)
- ☐ c. L(A1) intersectat cu L(A2) = multimea vida

Completati gramatica stiind ca arborii de parsare de mai jos sunt corecti

S healthy Quality pizza not avocado Quality Modifier Food that

Completati gramatica stiind ca arborii de parsare de mai jos sunt corecti.

S -> Food 'is' Quality ✓

Food -> Modifier ✓ Food ✓ | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality ✓

Your answer is correct.

Se da gramatica  $G = (\{ '*', '+', '(', ')' \}, \{ A, S, E \}, A, P)$ , unde  $P$  este

$A \rightarrow \{ ' S ' \}$

$S \rightarrow E S \mid \epsilon$

$E \rightarrow '+' \mid '*'$

Se da gramatica  $G = (\{ '*', '+', '(', ')' \}, \{ A, S, E \}, A, P)$ , unde  $P$  este

$A \rightarrow \{ ' S ' \}$

$S \rightarrow E S \mid \epsilon$

$E \rightarrow '+' \mid '*'$

Alegeti afirmatiile adevarate.

Alegerile incorecte aduc penalizari.

Select one or more:

- ☒ a. pentru a include in limbaj si stringuri de forma  $\{++\{+\}\}$ , cu oricate niveluri de imbricare, trebuie modificata a doua productie ✖
- ☐ b. pentru a include in limbaj si stringuri de forma  $\{++\{+\}\}$ , cu oricate niveluri de imbricare, nu trebuie modificat nimic, pentru ca limbajul deja le include
- ☒ c. pentru a include in limbaj si stringuri de forma  $\{++\{+\}\}$ , cu oricate niveluri de imbricare, trebuie modificata a treia productie ✔
- ☐ d. pentru a include in limbaj si stringuri de forma  $\{++\{+\}\}$ , cu oricate niveluri de imbricare, trebuie modificata prima productie
- ☒ e. orice string format din oricat de multe  $+$  si  $*$  cuprins intre doua acolade (deci fara (...) imbricate) face parte din limbajul gramaticii ✔

Your answer is correct.

The correct answers are: orice string format din oricat de multe  $+$  si  $*$  cuprins intre doua acolade (deci fara (...) imbricate) face parte din limbajul gramaticii, pentru a include in limbaj si stringuri de forma  $\{++\{+\}\}$ , cu oricate niveluri de imbricare, trebuie modificata a treia productie



Se da gramatica G cu productiile

S -> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

Si is , large, big, small, this, that , fresh , healthy, expensive , not drept terminale .

Se da gramatica G cu productiile

S -> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se construiesc automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

S q ->

Quality

'is'

Food

q

'large'

q

'large'

-> q

este o productie din acest automat prin care se scoate un simbol de pe stiva si se inlocuieste cu partea dreapta a unei productii din gramati

x

nu este productie in acest automat

x

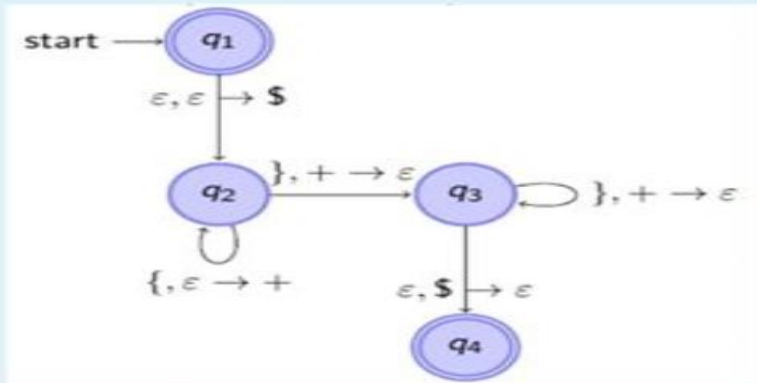
Your answer is incorrect.

The correct answer is: S q -> Quality 'is' Food q -> este o productie din acest automat prin care se scoate un simbol de pe stiva cu partea dreapta a unei productii din gramati in ordinea inversata, 'large' q 'large' -> q -> este o productie din acest automat de tip consum de la intrare pe baza stivei

Fie automatul

Fie automatul

$$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$$



Alegeți raspunsul corect

Select one or more:

- ☒ a. automatul accepta doar stringuri care contin n acolade deschise urmate de n acolade inchise ✓
- ☐ b. automatul accepta stringul { } { }
- ☐ c. automatul accepta doar stringuri care contin n acolade deschise urmate de m acolade inchise, unde m si n pot fi diferite

Se dau gramaticile  $G_1 = (\{e, '\}, \{P, Ps\}, P_2, \text{Productii1})$ ,  $G_2 = (\{e, '\}, \{P, P_2\}, Ps, \text{Productii2})$ ,

Se dau gramaticile  $G_1 = (\{e, '\}, \{P, Ps\}, Ps, \text{Productii1})$ ,  $G_2 = (\{e, '\}, \{P, Ps\}, Ps, \text{Productii2})$ ,

Productii1 = {  $Ps \rightarrow P'$ ,  $Ps \mid \epsilon$   
 $P \rightarrow e$  }

Productii2 = {  $Ps \rightarrow P'$ ,  $Ps \mid P$   
 $P \rightarrow e$  }

Care afirmatii sunt adevarate?

- e,e,e apartine  ✓
- e,e,e, apartine  ✓

Your answer is correct.

The correct answer is: e,e,e apartine → L(G2), dar nu si L(G1),  
 e,e,e, apartine → L(G1), dar nu si L(G2)

Stiind ca gramatica de mai jos are doua terminale, \ si / , simbolul de start este X, iar  $L(G) = \{ \text{string cu numar egal de } V \text{ si } \wedge \}$ , completati spatiile libere.

epsilon Z / \ Y

Stiind ca gramatica de mai jos are doua terminale, \ si / , simbolul de start este X, iar  $L(G) = \{ \text{string cu numar egal de } V \text{ si } \wedge \}$ , completati spatiile libere. In fiecare spatiu liber puteti scrie fie doar un nonterminal, fie doar un terminal.

X -> \ X Y / Z

Y -> \ Y / Y \

Z -> \ Z / Z \

Your answer is partially correct.

Se da gramatica  $G = (\{ \text{cat, dog, rug, chased, sat, in, on} \}, \{ S, NP, VP, PP, N, V, P \}, S, \text{PROD})$ , unde PROD este setul de productii:

Se da gramatica  $G = (\{ \text{cat, dog, rug, chased, sat, in, on} \}, \{ S, NP, VP, PP, N, V, P \}, S, \text{PROD})$ , unde PROD este setul de productii:

S -> NP VP  
 PP -> P NP  
 NP -> 'the' N | N PP | 'the' N PP  
 VP -> V NP | V PP | V NP PP  
 N -> 'cat'  
 N -> 'dog'  
 N -> 'rug'  
 V -> 'chased'  
 V -> 'sat'  
 P -> 'in'  
 P -> 'on'

Pentru automatul stiva construit in abordarea descendenta (top-down) simpla, alegeți afirmatiile adevarate. Alegeri gresite sunt depunctate. Nu se poate obtine punctaj negativ

Select one or more:

☐ a. N q -> 'rug' q - nu apartine automatului ✗

☐ b. automatul rezultat este determinist ✗

☐ c. automatul rezultat nu este determinist si pentru ca sunt mai multe productii cu partea stanga VP q ✓

☐ d. productia  $Sq \rightarrow NP VP q$  - apartine automatului ✗

☐ e. productia  $Sq \rightarrow VP NP q$  apartine automatului ✓

☐ f. N q -> 'sat' q apartine automatului ✓

☐ g. productia  $Sq \rightarrow VP NP q$  apartine automatului ✓

Your answer is partially correct.  
 You have selected too many options.

Se da gramatica G cu productiile

S-> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large' | 'big' | 'small' | 'this' | 'that'

Quality -> 'fresh' | 'healthy' | 'expensive' | 'not' Quality

Si is , large, big, small, this, that , fresh , healthy, expensive , not drept terminale .

Se da gramatica G cu productiile

S -> Food 'is' Quality

Food -> Modifier Food | 'pizza' | 'avocado'

Modifier -> 'large'|'big'|'small'|'this'|'that'

Quality -> 'fresh'| 'healthy'|'expensive'|'not' Quality

si is, large, big, small, this, that, fresh, healthy, expensive, not drept terminale.

Se construiesc automatul stiva in abordarea simpla top-down (descendenta) (algoritmul fara lookahead).

S q -> Quality 'is' Food q este o productie din acest automat de tip consum de la intrare pe baza stivei

'not' q 'not' -> q este o productie prin care se mentioneaza ca se poate consuma oricand un terminal de la intrare

Stiind ca gramatica de mai jos are terminalele  $\{ \}$  1 si 2, simbolul de start este A, iar arborele de mai jos este arborele de parsare pentru  $\{ 1 \{ 2 \} 2 \}$ , completati gramatica

