Universitatea din Bucureşti
FMI
Calculabilitate şi Complexitate
Examen, 3 Februarie, Nivelul I,
Subiecte: $\boxed{\text{A}}$

Puncte:25. Timp: 50 min
Data: 3-02-2025

Instrucțiuni I. Completați (pe foaia auxiliarã) rãspunsul/rãspunsurile corecte. Toate intrebãrile conteazã in mod egal. Puteți folosi marginile hârtiei drept ciornã, **dar nicun alt material**. O intrebare poate avea mai multe rãspunsuri corecte. Toate conteazã in mod egal. **Pe de altã parte, dacã alegeți un rãspuns greșit, punctajul vostru la intrebare este zero.**

1. Funcţia lui Ackermann **nu** este
   - (a) primitiv recursivã.
   - (b) recursivã.
   - (c) parţial recursivã.
   - (d) calculabilã de un program LOOP.
2. Care operaţie, aplicatã unei(unor) funcţii primitiv recursive, garanteazã cã rezultatul este o funcţie primitiv recursivã?
   - (a) compunere
   - (b) recursie primitivã
   - (c) minimizare
   - (d) suma a douã funcţii.
3. Consider am funcţia $f(n) = 0$ dacã conjectura lui Collatz (vezi cursul 1) este adevãratã, $f(n) = 1$ altfel. f este o funcţie
   - (a) recursivã.
   - (b) primitiv recursivã.
   - (c) care poate fi calculatã de o maşinã Turing.
   - (d) care creşte mai rapid decât orice funcţie primitiv recursivã.
4. Cum putem crea o funcţie care **nu** e primitiv recursivã ?
   - (a) enumerãm toate funcţiile primitiv recursive. Creãm o funcţie care pe inputul i returneazã valoarea $f_i(i) + 1$.
   - (b) folosim operaţia de minimizare, dacã rezultatul nu este definitã pentru toate inputurile i.
   - (c) prin compunere din douã funcţii primitiv recursive.
   - (d) Cu un automat finit.

5. Dacã A, B sunt probleme de decizie iar $A \oplus B = \{x0|x \in A\} \cup \{y1|y \in B\}$ atunci
   (a) $B \leq_m A \oplus B$.
   (b) $A \oplus B \leq_m A$.
   (c) Ambele reduceri sunt adevãrate.
   (d) Nicio reducere nu e adevãratã
6. Dacã A este o mulţime nevidã şi recursivã iar K este problema opririi, atunci
   (a) $A \leq_m A \oplus K$.
   (b) $K \leq_m A \oplus K$.
   (c) Ambele reduceri sunt adevãrate.
   (d) Nicio reducere nu e adevãratã.
7. Care din urmãtoarele modele de calcul recunosc o clasã de limbaje inchisã la operaţia de complementare?
   (a) automat finit
   (b) maşinã Turing cu o bandã.
   (c) maşinã Turing cu douã benzi.
   (d) Toate modelele menţionate.
8. Care din problemele urmãtoare sunt reductibile la problema determinãrii dacã un numãr este prim sau nu? **NU** cerem ca reducerea sã fie calculabilã în timp polinomial.
   (a) problema opririi K.
   (b) problema de a decide dacã un graf este 2-colorabil.
   (c) problema Quantified Boolean Formula (QBF).
   (d) Niciuna din problemele listate.
9. Care din clasele urmãtoare de probleme de decizie conţin clasa limbajelor regulate?
   (a) clasa problemelor de decizie recursive
   (b) clasa problemelor de decizie primitiv recursive.
   (c) P, clasa problemelor care au algoritmi poliniomiali.
   (d) niciuna din clase.
10. Care din problemele urmãtoare **nu** sunt recursive ?
   (a) $K_1 = \{< x, y >: M_x(y)$ se opreşte intr-un pas$\}$
   (b) $K = \{< x, y >: M_x(y)$ se opreşte$\}$.
   (c) $\overline{K} = \{< x, y >: M_x(y)$ nu se opreşte$\}$
   (d) toate problemele sunt recursive.

11. Pentru care din următoarele probleme puteţi da algoritmi care să le rezolve?
    (a) Fiind date o mulţime de tipuri de pavaje, decideţi dacă putem pava planul cu pavajele Wang date.
    (b) Fiind date o mulţime de tipuri de pavaje, decideţi dacă putem pava un pătrat 3x3 cu pavajele Wang date
    (c) Fiind dat un polinom cu coeficienţi intregi $p(x_1, x_2, \ldots, x_n)$ există soluţii intregi pentru ecuaţia $p(x_1, x_2, \ldots, x_n) = 0$?
    (d) Fiind dat un polinom cu coeficienţi intregi $p(x_1, x_2, \ldots, x_n)$ există soluţii pentru ecuaţia $p(x_1, x_2, \ldots, x_n) = 0$ cu $|x_1|, \ldots, |x_n| \geq 1000$?
12. Fiind dată o maşină Turing deterministă, care din afirmaţiile următoare sunt adevărate?
    (a) Dacă pe o intrare x maşina rulează in $f(|x|)$ paşi, atunci pe orice intrare y spaţiul folosit de maşină este $O(f(|y|))$.
    (b) Dacă pe o intrare x maşina rulează in spaţiu $f(|x|)$, atunci pe orice intrare y maşina rulează in $O(f(|y|))$ paşi.
    (c) Dacă pe o intrare x maşina nu se opreşte atunci spaţiul folosit de $M(x)$ este infinit.
    (d) niciuna din afirmaţiile celelalte nu este adevărată.
13. Fiind dată formula următoare: $(x \vee y \vee z) \wedge (y \vee \overline{t}) \wedge (\overline{z} \vee \overline{t}) \wedge \overline{x}$, care literali sunt puri ?
    (a) x
    (b) y
    (c) z
    (d) t
14. Care din următoarele afirmaţii sunt cunoscute drept adevărate ?
    (a) Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate polinomială.
    (b) Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate $O(2^{n^{O(1)}})$.
    (c) Dându-se o formulă booleană in care toate variabilele sunt cuantificate, există un algoritm polinomial pentru a decide dacă fomulele sunt adevărate sau false.
    (d) Pentru o parte a afirmaţiilor de mai sus nu se cunoaşte statutul lor de adevăr.

15. Care din următoarele probleme **nu** sunt cunoscute ca fiind NP-complete ?
    (a) HORN-SAT
    (b) 4-SAT
    (c) XOR-SAT.
    (d) problema 2-colorării unui graf.

16. Dacă problema A este NP-completă atunci
    (a) orice problem a de decizie nevidă $B \in P$ se reduce la A
    (b) orice problem a de decizie nevidă $B \in NP$ se reduce la A
    (c) A este NP-hard.
    (d) niciunul din răspunsuri nu este adevărat.

17. Care din următoarele afirmaţii sunt adevărate ?
    (a) Dacă $A \leq_m^P B$ şi $B \in P$ atunci $A \in P$.
    (b) Dacă $A \leq_m^P B$ şi $B \in NP$ atunci $A \in NP$.
    (c) Dacă $A \leq_m^P B$ şi B este NP-completă atunci A este NP-completă.
    (d) Dacă $A \leq_m^P B$ şi B este NP-hard atunci A este NP-hard.

18. Care din următoarele afirmaţii este adevărată ?
    (a) O formulă logică CNF cu cel mult trei literali in fiecare clauză are demonstraţii prin rezoluţie de lungime polinomială in n.
    (b) O formulă logică CNF cu cel mult un literal pozitiv in fiecare clauză are demonstraţii prin rezoluţie de lungime polinomială in n.
    (c) Există un algoritm de tip Davis-Putnam care pe instanţe nesatisfiabile pentru 3-SAT rulează in timp polinomial in n.
    (d) Există un algoritm de tip DPLL care pe instanţe care codifică principiul cutiei (Principiul lui Dirichlet) rulează in timp polinomial in n.

19. Care din următoarele probleme sunt cunoscute că au algoritmi polinomiali ?
    (a) Fiind dată o mulţime de numere, putem alege o submulţime a cărei sumă ia o valoare dată ?
    (b) Fiind dat un graf orientat G şi două vârfuri s, t, putem ajunge in cel mult cinci paşi de la s la t?
    (c) Putem colora un graf cu 3 culori astfel incât orice două vârfuri adiacente să aibă culori diferite ?
    (d) Fiind dată o formulă propoziţională in forma normală conjunctivă in care in fiecare clauză apare cel mult doi literal pozitivi, este formula satisfiabilă ?

20. Dacã $P = NP$ atunci ...
    (a) Putem colora un graf cu numãrul minim de culori in timp polinomial.
    (b) putem rezolva orice problem a cu un algoritm polinomial.
    (c) putem gãsi o soluţie pentru problema Vertex Cover cu un algoritm cu complexitate polinomialã.
    (d) Orice problemã rezolvabilã in timp polinomial folosind SAT ca subrutinã are un algoritm polinomial.
21. Care din urmãtoarele clase de complexitate are probleme complete ?
    (a) co-NP.
    (b) toate clasele Σ_k^P din ierarhia polinomialã.
    (c) PSPACE.
    (d) Niciuna din clase.
22. Care din urmãtoarele afirmaţii descriu funcţionarea algoritmilor moderni de tip CDCL pentru problema satisfiabilitãţii ?
    (a) Algoritmii folosesc propagare unitarã.
    (b) Algoritmii adaugã la formule constrângeri pe care le "invaţã" ca urmare a eşecurilor anterioare.
    (c) Algoritmii exploateazã tranziţia de fazã in problema satisfiabilitãţii.
    (d) Niciunul din celelalte rãspunsuri nu este corect.
23. Dacã $P = NP$ atunci ...
    (a) $P = co - NP$.
    (b) Putem testa izomorfismul a douã grafuri in timp polinomial.
    (c) Orice problemã NP-hard este in P
    (d) Niciuna din celelalte variante nu este adevãratã.
24. Fie R un predicat recursiv cu 5 argumente. Definim problema de decizie A prin $A = \{x \in \Sigma^* : (\forall y)(\forall z)(\exists t)(\exists u)R(x, y, z, t, u)\}$. Care din afirmaţiile de mai jos sunt adevãrate ?
    (a) A este recursiv enumerabilã.
    (b) $A \in NP$.
    (c) $A \in \Pi_4^P$.
    (d) $A \in \Pi_2$.

25. Care din următoarele probleme **nu** are o tranziţie de fază de tip SAT/UNSAT ?

(a) 1-k-SAT, $k \geq 3$.

(b) 3-SAT.

(c) Horn-SAT.

(d) Toate au.

Answer Key for Exam $\boxed{\text{A}}$

Instrucțiuni I. Completați (pe foaia auxiliarã) rãspunsul/rãspunsurile corecte. Toate intrebãrile conteazã in mod egal. Puteți folosi marginile hârtiei drept ciornã, **dar nicun alt material**. O intrebare poate avea mai multe rãspunsuri corecte. Toate conteazã in mod egal. **Pe de altã parte, dacã alegeți un rãspuns greșit, punctajul vostru la intrebare este zero.**

1. Funcţia lui Ackermann **nu** este
   - $\boxed{\text{(a)}}$ primitiv recursivã.
   - (b) recursivã.
   - (c) parţial recursivã.
   - $\boxed{\text{(d)}}$ calculabilã de un program LOOP.
2. Care operaţie, aplicatã unei(unor) funcţii primitiv recursive, garanteazã cã rezultatul este o funcţie primitiv recursivã?
   - $\boxed{\text{(a)}}$ compunere
   - $\boxed{\text{(b)}}$ recursie primitivã
   - (c) minimizare
   - $\boxed{\text{(d)}}$ suma a douã funcţii.
3. Consider am funcţia $f(n) = 0$ dacã conjectura lui Collatz (vezi cursul 1) este adevãratã, $f(n) = 1$ altfel. f este o funcţie
   - $\boxed{\text{(a)}}$ recursivã.
   - $\boxed{\text{(b)}}$ primitiv recursivã.
   - $\boxed{\text{(c)}}$ care poate fi calculatã de o maşinã Turing.
   - (d) care creşte mai rapid decât orice funcţie primitiv recursivã.
4. Cum putem crea o funcţie care **nu** e primitiv recursivã ?
   - $\boxed{\text{(a)}}$ enumerãm toate funcţiile primitiv recursive. Creãm o funcţie care pe inputul i returneazã valoarea $f_i(i) + 1$.
   - $\boxed{\text{(b)}}$ folosim operaţia de minimizare, dacã rezultatul nu este definitã pentru toate inputurile i.
   - (c) prin compunere din douã funcţii primitiv recursive.
   - (d) Cu un automat finit.
5. Dacã A, B sunt probleme de decizie iar $A \oplus B = \{x0 | x \in A\} \cup \{y1 | y \in B\}$ atunci
   - $\boxed{\text{(a)}}$ $B \leq_m A \oplus B$.
   - (b) $A \oplus B \leq_m A$.
   - (c) Ambele reduceri sunt adevãrate.
   - (d) Nicio reducere nu e adevãratã

6. Dacã A este o mulţime nevidã şi recursivã iar K este problema opririi, atunci

   $\boxed{\text{(a)}}$ $A \leq_m A \oplus K$.

   $\boxed{\text{(b)}}$ $K \leq_m A \oplus K$.

   $\boxed{\text{(c)}}$ Ambele reduceri sunt adevãrate.

   (d) Nicio reducere nu e adevãratã.

7. Care din urmãtoarele modele de calcul recunosc o clasã de limbaje inchisã la operaţia de complementare?

   $\boxed{\text{(a)}}$ automat finit

   (b) maşinã Turing cu o bandã.

   (c) maşinã Turing cu douã benzi.

   (d) Toate modelele menţionate.

8. Care din problemele urmãtoare sunt reductibile la problema determinãrii dacã un numãr este prim sau nu? **NU** cerem ca reducerea sã fie calculabilã în timp polinomial.

   (a) problema opririi K.

   $\boxed{\text{(b)}}$ problema de a decide dacã un graf este 2-colorabil.

   $\boxed{\text{(c)}}$ problema Quantified Boolean Formula (QBF).

   (d) Niciuna din problemele listate.

9. Care din clasele urmãtoare de probleme de decizie conţin clasa limbajelor regulate?

   $\boxed{\text{(a)}}$ clasa problemelor de decizie recursive

   $\boxed{\text{(b)}}$ clasa problemelor de decizie primitiv recursive.

   $\boxed{\text{(c)}}$ P, clasa problemelor care au algoritmi poliniomiali.

   (d) niciuna din clase.

10. Care din problemele urmãtoare **nu** sunt recursive ?

   (a) $K_1 = \{< x, y >: M_x(y) \text{ se opreşte intr-un pas}\}$

   $\boxed{\text{(b)}}$ $K = \{< x, y >: M_x(y) \text{ se opreşte}\}$.

   $\boxed{\text{(c)}}$ $\overline{K} = \{< x, y >: M_x(y) \text{ nu se opreşte}\}$

   (d) toate problemele sunt recursive.

11. Pentru care din următoarele probleme puteţi da algoritmi care să le rezolve?

    (a) Fiind date o mulţime de tipuri de pavaje, decideţi dacă putem pava planul cu pavajele Wang date.

    $\boxed{\text{(b)}}$ Fiind date o mulţime de tipuri de pavaje, decideţi dacă putem pava un pătrat 3x3 cu pavajele Wang date

    (c) Fiind dat un polinom cu coeficienţi intregi $p(x_1, x_2, \ldots, x_n)$ există soluţii intregi pentru ecuaţia $p(x_1, x_2, \ldots, x_n) = 0$?

    (d) Fiind dat un polinom cu coeficienţi intregi $p(x_1, x_2, \ldots, x_n)$ există soluţii pentru ecuaţia $p(x_1, x_2, \ldots, x_n) = 0$ cu $|x_1|, \ldots, |x_n| \geq 1000$?

12. Fiind dată o maşină Turing deterministă, care din afirmaţiile următoare sunt adevărate?

    $\boxed{\text{(a)}}$ Dacă pe o intrare x maşina rulează in $f(|x|)$ paşi, atunci pe orice intrare y spaţiul folosit de maşină este $O(f(|y|))$.

    (b) Dacă pe o intrare x maşina rulează in spaţiu $f(|x|)$, atunci pe orice intrare y maşina rulează in $O(f(|y|))$ paşi.

    (c) Dacă pe o intrare x maşina nu se opreşte atunci spaţiul folosit de $M(x)$ este infinit.

    (d) niciuna din afirmaţiile celelalte nu este adevărată.

13. Fiind dată formula următoare: $(x \vee y \vee z) \wedge (y \vee \bar{t}) \wedge (\bar{z} \vee \bar{t}) \wedge \bar{x}$, care literali sunt puri ?

    (a) x

    $\boxed{\text{(b)}}$ y

    (c) z

    $\boxed{\text{(d)}}$ t

14. Care din următoarele afirmaţii sunt cunoscute drept adevărate ?

    (a) Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate polinomială.

    $\boxed{\text{(b)}}$ Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate $O(2^{n^{O(1)}})$.

    (c) Dându-se o formulă booleană in care toate variabilele sunt cuantificate, există un algoritm polinomial pentru a decide dacă fomulele sunt adevărate sau false.

    $\boxed{\text{(d)}}$ Pentru o parte a afirmaţiilor de mai sus nu se cunoaşte statutul lor de adevăr.

15. Care din urmãtoarele probleme **nu** sunt cunoscute ca fiind NP-complete ?

$\boxed{\text{(a)}}$ HORN-SAT

(b) 4-SAT

$\boxed{\text{(c)}}$ XOR-SAT.

$\boxed{\text{(d)}}$ problema 2-colorãrii unui graf.

16. Dacã problema A este NP-completã atunci

$\boxed{\text{(a)}}$ orice problem a de decizie nevidã $B \in P$ se reduce la A

$\boxed{\text{(b)}}$ orice problem a de decizie nevidã $B \in NP$ se reduce la A

$\boxed{\text{(c)}}$ A este NP-hard.

(d) niciunul din rãspunsuri nu este adevãrat.

17. Care din urmãtoarele afirmaţii sunt adevãrate ?

$\boxed{\text{(a)}}$ Dacã $A \leq_m^P B$ şi $B \in P$ atunci $A \in P$.

$\boxed{\text{(b)}}$ Dacã $A \leq_m^P B$ şi $B \in NP$ atunci $A \in NP$.

(c) Dacã $A \leq_m^P B$ şi B este NP-completã atunci A este NP-completã.

(d) Dacã $A \leq_m^P B$ şi B este NP-hard atunci A este NP-hard.

18. Care din urmãtoarele afirmaţii este adevãratã ?

(a) O formulã logicã CNF cu cel mult trei literali in fiecare clauzã are demonstraţii prin rezoluţie de lungime polinomialã in n.

$\boxed{\text{(b)}}$ O formulã logicã CNF cu cel mult un literal pozitiv in fiecare clauzã are demonstraţii prin rezoluţie de lungime polinomialã in n.

(c) Existã un algoritm de tip Davis-Putnam care pe instanţe nesatisfiabile pentru 3-SAT ruleazã in timp polinomial in n.

(d) Existã un algoritm de tip DPLL care pe instanţe care codificã principiul cutiei (Principiul lui Dirichlet) ruleazã in timp polinomial in n.

19. Care din urmãtoarele probleme sunt cunoscute cã au algoritmi polinomiali ?

(a) Fiind datã o mulţime de numere, putem alege o submulţime a cãrei sumã ia o valoare datã ?

$\boxed{\text{(b)}}$ Fiind dat un graf orientat G şi douã vârfuri s, t, putem ajunge in cel mult cinci paşi de la s la t?

(c) Putem colora un graf cu 3 culori astfel incât orice douã vârfuri adiacente sã aibã culori diferite ?

(d) Fiind datã o formulã propoziţionalã in forma normalã conjunctivã in care in fiecare clauzã apare cel mult doi literal pozitivi, este formula satisfiabilã ?

20. Dacã $P = NP$ atunci ...
   - $\boxed{\text{(a)}}$ Putem colora un graf cu numãrul minim de culori in timp polinomial.
   - (b) putem rezolva orice problem a cu un algoritm polinomial.
   - $\boxed{\text{(c)}}$ putem gãsi o soluţie pentru problema Vertex Cover cu un algoritm cu complexitate polinomialã.
   - $\boxed{\text{(d)}}$ Orice problemã rezolvabilã in timp polinomial folosind SAT ca subrutinã are un algoritm polinomial.

21. Care din urmãtoarele clase de complexitate are probleme complete ?
   - $\boxed{\text{(a)}}$ co-NP.
   - $\boxed{\text{(b)}}$ toate clasele Σ_k^P din ierarhia polinomialã.
   - $\boxed{\text{(c)}}$ PSPACE.
   - (d) Niciuna din clase.

22. Care din urmãtoarele afirmaţii descriu funcţionarea algoritmilor moderni de tip CDCL pentru problema satisfiabilitãţii ?
   - $\boxed{\text{(a)}}$ Algoritmii folosesc propagare unitarã.
   - $\boxed{\text{(b)}}$ Algoritmii adaugã la formule constrângeri pe care le ”invaţã” ca urmare a eşecurilor anterioare.
   - (c) Algoritmii exploateazã tranziţia de fazã in problema satisfiabilitãţii.
   - (d) Niciunul din celelalte rãspunsuri nu este corect.

23. Dacã $P = NP$ atunci ...
   - $\boxed{\text{(a)}}$ $P = co - NP$.
   - $\boxed{\text{(b)}}$ Putem testa izomorfismul a douã grafuri in timp polinomial.
   - (c) Orice problemã NP-hard este in P
   - (d) Niciuna din celelalte variante nu este adevãratã.

24. Fie R un predicat recursiv cu 5 argumente. Definim problema de decizie A prin $A = \{x \in \Sigma^* : (\forall y)(\forall z)(\exists t)(\exists u)R(x, y, z, t, u)\}$. Care din afirmaţiile de mai jos sunt adevãrate ?
   - (a) A este recursiv enumerabilã.
   - (b) $A \in NP$.
   - (c) $A \in \Pi_4^P$.
   - $\boxed{\text{(d)}}$ $A \in \Pi_2$.

25. Care din urmãtoarele probleme **nu** are o tranziţie de fazã de tip SAT/UNSAT ?
   - (a) 1-k-SAT, $k \geq 3$.
   - (b) 3-SAT.
   - $\boxed{\text{(c)}}$ Horn-SAT.
   - (d) Toate au.