

Subiecte la proba scrisă a examenului de licență iunie 2015

Cunoștințe fundamentale – teorie

T1. Demonstrați că familia limbajelor independente de context este închisă la intersecția cu limbaje regulate pornind de la o gramatică independentă de context și un automat.

T2. Tratați următoarele cerințe:

- (a) Descrieți formal operatorul DIVISION al algebrei relaționale. Dați un exemplu de cerere în limbaj natural a cărei rezolvare necesită acest operator. Rezolvați cererea în SQL.
- (b) Precizați trei reguli de optimizare a cererilor enunțând, pentru fiecare dintre acestea, proprietatea operatorilor algebrei relaționale care se aplică.
- (c) Descrieți și exemplificați forma normală Boyce-Codd.

T3. Arbori binari de căutare echilibrați AVL. Definiție. Inserarea unui nod cu reechilibrare. Limite pentru înălțime (enunț teorema AVL).

T4. Tratați următoarele cerințe:

- (a) Definiți noțiunea de algebră Boole și explicați pe scurt legătura dintre algebrele Boole și calculul propozițional clasic.
- (b) Demonstrați teorema de reprezentare a lui Stone: Pentru orice algebră Boole $\mathbf{A} = (\mathbf{A}, \vee, \wedge, \neg, 0, 1)$ există o mulțime X astfel încât \mathbf{A} și $\mathcal{P}(X) = (\mathcal{P}(X), \cup, \cap, \neg, \emptyset, X)$ sunt algebre Boole izomorfe.

Rezultatele auxiliare folosite trebuie enunțate, dar nu demonstrate.

Cunoștințe fundamentale – probleme

P1. Tratați următoarele cerințe:

- (a) Spuneți dacă limbajul următor este limbaj independent de context sau nu; dacă da, construiți o gramatică independentă de context care să îl genereze, dacă nu, demonstrați că nu este (folosind, spre exemplu, lema de pompare).

$$L_1 = \{a^l b^n c^m \mid l+n \neq m\}$$

- (b) Spuneți dacă limbajul următor este limbaj independent de context sau nu; dacă da, construiți o gramatică independentă de context care să îl genereze, dacă nu, demonstrați că nu este (folosind, spre exemplu, lema de pompare).

$$L_2 = \{a^i b^j a^{2i} b^{3j+3} \mid i \geq 0, j \geq 0\}$$

P2. Se dă un număr natural S . Tratați următoarele cerințe (algoritm, implementare într-un limbaj de programare, complexitate).

- (a) Se poate scrie S ca sumă de numere prime (nu neaparat distincte)? Dacă da, afișați o astfel de posibilitate.
- (b) Se poate scrie S ca sumă de numere prime distincte? Dacă da, afișați o astfel de posibilitate.
- (c) În câte moduri pot scrie S ca sumă de numere prime distincte ?

Prin sumă de numere prime înțelegem suma a cel puțin un număr prim. 1 nu este număr prim.

P3. Tratați următoarele cerințe:

- (a) Să se proiecteze diagrama conceptuală a modelului de date corespunzător următorului scenariu:
 - O firmă de distribuție are angajați de tip funcționar și curier. Firma deține autovehicule având diferite capacități de transport. Funcționarii firmei preiau comenzile de livrare ale clienților. Clienții comandă transportul de la o adresă de ridicare până la o adresă de recepție. Funcționarul care a preluat comanda unui client delegă curierilor efectuarea transportului corespunzător. În funcție de volumul livrat, transportul acestuia este delegat unuia sau mai multor curieri, ce vor folosi unul sau mai multe autovehicule. În cazul transportului cu autovehicule, sunt desemnați curierii care le vor conduce.
- (b) Să se creeze structura unui tabel asociativ din diagrama conceptuală de la punctul anterior. Să se adauge constrângerile de integritate asupra acestui tabel.
- (c) Pentru clienții care au lansat cele mai multe comenzi în anul 2015, să se obțină numele acestora și numărul de comenzi livrate la care a participat un singur autovehicul.
- (d) Să se obțină numele funcționarilor care au preluat comenzi de livrare la care au participat cele mai multe autovehicule.

P4. Se consideră un vector format din n numere întregi ($1 \leq n \leq 100000$). Să se descrie un algoritm eficient pentru rearanjarea elementelor tabloului astfel încât toate numerele pare să fie situate înaintea numerelor impare. Se va descrie informal algoritmul, se va implementa într-un limbaj de programare și se va preciza complexitatea obținută.

P5. Aplicați noțiunea de semafor la una dintre problemele clasice de programare, la alegere (producator/consumator, filozofi, cititori/scriitori, frizerul somnoros). Se va scrie soluția în pseudocod.

P6. Se dă o matrice $n \times m$ ($n, m \leq 1000$), cu $P \leq 100$ puncte marcate cu 1 (restul valorilor din matrice vor fi 0). Distanța dintre 2 puncte ale matricii se măsoară în locații străbătute mergând pe orizontală și pe verticală între cele 2 puncte (distanța Manhattan).

Se dă o mulțime de Q puncte din matrice ($Q \leq 1000000$). Să se calculeze cât mai eficient pentru fiecare dintre cele Q puncte date, care este cea mai apropiată locație marcată cu 1 din matrice. Se va descrie informal algoritmul, se va implementa într-un limbaj de programare și se va preciza complexitatea obținută.

P7. Fie funcția booleană: $f : B_2^3 \rightarrow B_2$, $f(x, y, z) = 0$ dacă și numai dacă x, y, z este o secvență palindromică (i.e. simetrică).

- (a) Construiți tabelul de valori al lui f și scrieți f în FND și FNC.
- (b) Implementați f printr-un circuit format doar din porți NOT, AND, OR, conținând un număr cât mai mic de porți (justificați construcția).
- (c) Implementați f printr-un PROM și respectiv printr-un multiplexor.
- (d) Construiți un circuit secvențial care primește la intrare, succesiv, o secvență de biți și de fiecare dată furnizează la ieșire valoarea $f(x, y, z)$, unde x, y, z sunt ultimii 3 biți citiți (în această ordine).

P8. Fie *Form* mulțimea formulelor calculului propozițional clasic. Pentru ψ și ϕ formule în calculul propozițional clasic, definim $\psi \sim \phi$ dacă și numai dacă $\vdash \psi \leftrightarrow \phi$

- (a) Demonstrați că \sim este relație de echivalență pe *Form*.
- (b) Arătați că mulțimea Form/\sim poate fi înzestrată cu o structură de algebră Boole.

Cunoștințe de specialitate

S1. Șabloane (template) în limbajul de programare C++.

- (a) Funcții șablon: caracteristici generale, mod de definire (sintaxa), mod de implementare, mod de folosire, restricții, supraîncărcare.
- (b) Clase șablon: caracteristici generale, mod de definire (sintaxa, se va trata inclusiv cazul când metodele sunt implementate în afara clasei), mod de implementare, mod de folosire, restricții.

S2. Linii hamiltoniene (definiții, exemple). Condiții suficiente de existență - Teorema lui Dirac (enunț și demonstrație).

S3. Clasa de complexitate TIME.

S4. Comparatie între criptografia simetrică și cu cheie publică. Definiții, exemple și scurtă descriere pentru sisteme de criptare de ambele tipuri. Limitările criptografiei simetrice. Avantaje și dezavantaje pentru criptografia cu cheie publică.

S5. Tratați următoarele cerințe:

- (a) Definiți relațiile de congruență sintactică și congruență semantică în logica ecuațională multisortată.
- (b) Demonstrați completitudinea logicii ecuaționale multisortate: relația de congruență semantică este inclusă în relația de congruență sintactică.

S6. Nivelul transport (adresarea, stabilirea, eliberarea conexiunii, controlul fluxului, multiplexarea, protocoalele specifice, porturi).

S7. Algoritmul alpha-beta pentru jocuri de două persoane cu informație completă. Prezentare generală (cu un exemplu comentat) și implementare în Prolog (cu explicarea principalelor predicate din program).

S8. Particularizați algoritmul backprop, forma "off line", îmbunătățit cu metoda momentului, cu funcția eroare "media sumei pătratelor erorilor", pentru o rețea neuronală cu arhitectura 2-1-1 și funcția de transfer, pentru toți perceptronii, liniară.

S9. Se dă gramatica G cu producțiile: $S \rightarrow aSAb \mid Ba$, $A \rightarrow aA \mid c$, $B \rightarrow b$.

- (a) Să se arate, prin construcția tabelului de analiză sintactică, că această gramatică este LL(1).
- (b) Folosind tabela construită la punctul (a), să se analizeze șirul abaacb și să se precizeze care este analiza sa stângă.