

## Subiecte la proba scrisă a examenului de licență februarie 2013

### Cunoștințe fundamentale – teorie

T1. Să se descrie și să se argumenteze operațiile de rotație simplă și rotație dublă pentru reechilibrarea arborilor binari de căutare AVL (echilibrați după înălțime) la inserția unui nou nod.

T2. Automate push-down. Echivalența cu gramaticile independente de context.

T3. Operatorii LEFT, OUTER, JOIN și DIVISION ai algebrei relaționale: descriere, exemple în limbaj natural, implementare.

Exemplele și implementarea se vor referi la următoarele scheme relaționale:

SALARIAT (cod\_salariat, nume, prenume, cod\_departament)

DEPARTAMENT (cod\_departament, denumire, cod\_manager)

LUCREAZA (cod\_angajat, cod\_proiect, data\_inceput, data\_sfarsit)

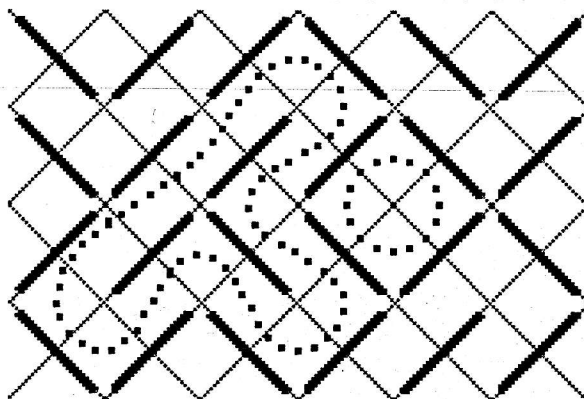
PROIECT (cod\_proiect, nume, buget, data\_start, data\_final)

T4.

Pentru orice algebră Boole  $B = (B, \vee, \wedge, -, 0, 1)$  există o mulțime  $X$  și un morfism injectiv  $d : B \rightarrow \mathcal{P}(X)$ , unde  $(\mathcal{P}(X), \cup, \cap, -, \emptyset, X)$  este algebra Boole a mulțimii părților lui  $X$ .

### Cunoștințe fundamentale – probleme

P1. Prin umplerea unui dreptunghi cu slash-uri (/) și backslash-uri (\) pot fi generate diverse tipuri de labirint. Iată un exemplu:



Este interesant ca un astfel de labirint poate fi privit ca o mulțime de drumuri care nu se intersectează, fiecare drum fiind fie un ciclu fie un drum care are o intrare în

labirint și o ieșire. Aceasta problema privește ciclurile, în exemplul de mai sus sunt exact două astfel de cicluri marcate cu linie punctată.

Scrieti un program care număra ciclurile dintr-un astfel de labirint și găsește lungimea celui mai lung dintre ele. Lungimea unui ciclu este data de numărul de patratele mici strabătute de ciclu. În exemplul de mai sus cel mai lung ciclu are lungimea 16 (constă din 16 patratele), iar celălalt (cel scurt) 4.

Intrarea constă dintr-o linie ce conține dimensiunile labirintului, doi întregi:  $w$  (lățimea) și  $h$  (înălțimea). Următoarele  $h$  linii conțin descrierea labirintului, fiecare din cele  $h$  linii conținând exact  $w$  caractere care pot fi fie "/" fie "\".

La ieșire trebuie afișat numărul de cicluri din labirintul dat la intrare și lungimea celui mai lung ciclu din acest labirint.

Exemplul din figura de mai sus va fi dat la intrare ca:

```
6 4
V/V
V//
//\
V//
```

Ieșirea pentru acest exemplu ar trebui să fie:

```
2 16
```

## DL Informatică

**P2.** Se dă un șir de întregi  $x_1, \dots, x_n$  citați pe rând (în mod dinamic). Se cere să se utilizeze un algoritm /o structură de date, astfel încât, la fiecare pas  $i$ , să se poate răspunde la query-ul "care este valoarea mediana din setul  $x_1, x_2, \dots, x_i$ " în timp  $O(\log i)$ .

**P3. 1.** Să se proiecteze baza de date corespunzătoare activității unui lanț de agenții de turism (diagrama conceptuală – minimum 7 tabele, fără a considera subentitățile).

2. Să se creeze tabelul corespunzător relației `TURIST_participa_la_EXCURSIE`, specificând toate cheile acestuia.

3. Pentru fiecare agenție și fiecare dintre anii 2010, 2011 și 2012, să se afișeze numele agenției și numărul de turiști care au participat la excursiile agenției.

4. Să se aplice forma normală 2 (FN2) pentru un tabel arbitrar al bazei de date proiectate anterior. Comentați.

**P4.** Fie  $\mathcal{L}$  sistemul formal al calculului propozițional clasic și  $Form$  mulțimea formulelor lui  $\mathcal{L}$ . Oricare ar fi  $\Gamma, \Delta \subseteq Form$  și  $\alpha, \beta, \gamma \in Form$  astfel încât  $\Gamma \vdash \alpha \rightarrow \beta$ , demonstrați ca:

(a)  $\Delta \vdash \alpha$  implică  $\Gamma \cup \Delta \vdash \beta$ ,

(b)  $\Gamma \cup \{\alpha \wedge \gamma\} \vdash \beta$ ,

(c) există  $\Gamma_0 \subseteq \Gamma$  astfel încât  $\Gamma_0$  este mulțime finită și  $\Gamma_0 \vdash \alpha \rightarrow \beta$ .

**P5.** Spuneți dacă următoarele afirmații sunt adevărate sau nu, justificați pe scurt răspunsul.

a) Există limbaje regulate care nu sunt independente de context.

b) Fie limbajele  $L_1, L_2$  cu proprietatea că  $L_1 \subseteq L_2$  și  $L_2 \in REG$ . Atunci  $L_1 \in REG$

c) Este decidabil dacă limbajele acceptate de două automate finite deterministe sunt egale sau nu.

d) Există limbaje unare (peste un alfabet cu o singură literă) care nu sunt independente de context.

e) Există un automat finit determinist care să accepte limbajul  $L = \{a^x | x = 7k + 3\}$ ?

**P6.** Se citește un număr natural  $n$  par,  $n < 30$ . Să se genereze și să se afișeze toate combinațiile de  $n$  paranteze rotunde care se închid corect.

De exemplu, pentru  $n=4$  se obțin următoarele combinații:  $(( ))$  și  $(( )) ( )$ .

Se va preciza argumentat complexitatea timp a algoritmului propus și se vor explica informal detaliile de implementare sub formă de program scris într-un limbaj de programare studiat în facultate (C/C++/Java).

**P7.** Se consideră mulțimea  $S = \{1, 2, 3, \dots, n\}$  unde  $n$  este un număr natural mai mare sau egal cu 2 și graful  $G_n = (V_n, E_n)$  definit astfel:

$$V_n = \{X | X \subseteq S\},$$

$$E_n = \{\{X, Y\} | X, Y \in V_n, |X \Delta Y| = 1\}.$$

a) Să se determine cardinalul mulțimii vârfurilor  $V_n$ .

b) Să se determine gradele vârfurilor lui  $G_n$ .

c) Să se determine cardinalul mulțimii muchiilor  $E_n$ .

d) Să se arate că graful  $G_n$  este bipartit.

e) Să se arate că graful  $G_n$  este hamiltonian.

f) Să se arate prin construcție că graful  $G_n$  conține un cuplaj perfect.

g) Să se determine graful  $G_n$  care sunt euleriene.

*Observație:* Prin  $X \Delta Y$  am notat diferența simetrică a mulțimilor  $X$  și  $Y$  adică  $X \Delta Y = (X - Y) \cup (Y - X)$ .

**P8.** Fie un poligon convex cu  $N \geq 4$  laturi, numerotate de la 1 la  $N$ . Poligonul este împărțit în triunghiuri prin  $N-3$  diagonale care nu se intersectează. Diagonalele se citesc de la tastatură sub formă de perechi de vârfuri.

Să se scrie un algoritm eficient care colorează laturile și diagonalele poligonului în roșu și negru astfel încât fiecare dintre triunghiurile în care este împărțit poligonul să aibă atât laturi roșii, cât și laturi negre. Se va afișa fiecare latură cu culoarea ei.

Se va preciza argumentat complexitatea timp a algoritmului propus și se vor explica informal detaliile de implementare sub formă de program scris într-un limbaj de programare studiat în facultate (C/C++/Java).

## Cunoștințe de specialitate

S1. Teorema lui Euler relativă la grafurile planare conexe.

S2a. Implementarea în Prolog a căutării de tip **best-first**.

S2b. Perceptronul cu funcția de transfer Heaviside. Definiție, algoritmul de învățare Rosenblatt, condiții de convergență, demonstrarea convergenței.

S3. Analiza sintactică 1-predictivă, pentru gramatici de tip LL(1). Se vor preciza:

- definiția gramaticilor de tip LL(1);
- caracterizări echivalente ale gramaticilor de tip LL(1);
- definiția mulțimilor First1 și Follow1;
- algoritmi de calcul al mulțimilor First1 și Follow1;
- algoritmul de analiză sintactică 1-predictiv și validitatea sa (formularea proprietății);
- construcția tabelului de analiză sintactică 1-predictivă pentru gramatici de tip LL(1);
- tipuri de conflicte la construirea tabelului de analiză sintactică 1-predictivă și rezolvarea lor.

S4. Algoritmul de cifrare RSA. Definiție. Vulnerabilități și securitate.

S5. Clasa de complexitate SPACE.

S6. 1. Prezențați pe scurt (în câteva rânduri) modelul client/server. Indicați rolurile celor două tipuri de procese precum și scenariul general al interacțiunii client-server.

2. Explicați modul în care se realizează comunicarea între client și server prin intermediul socket-urilor. Precizați cum se obține conexiunea și cum poate fi transmis un mesaj simplu (de exemplu un șir de caractere sau o valoare primitivă) între cele două procese, prezentând și comentând totodată fragmente relevante de cod sursă.

3. Descrieți o abordare bazată pe fire de execuție care să permită unui server să comunice simultan cu mai mulți clienți. Ilustrați atât acțiunea clienților cât și acțiunea serverului prin fragmente de cod sursă.

S7. a. Descrieți constructorul de copiere în C++: ce forme are, când se folosește și cum se comportă la moștenire.

b. Enumerați proprietățile supraîncărcării operatorilor în C++.

c. Descrieți mecanismul prin care se implementează polimorfismul de execuție în C++.

S8. Fie  $(S, \Sigma)$  o semnătură multisortată și  $X$  o mulțime de variabile.

(a) Definiți algebra termenilor  $T_{\Sigma}(X)$ .

(b) Demonstrați că pentru orice  $(S, \Sigma)$ -algebră  $A = (A_S, A_{\Sigma})$  și pentru orice funcție  $S$ -sortată  $a : X \rightarrow A_S$  există un unic  $(S, \Sigma)$ -morfism  $f : T_{\Sigma}(X) \rightarrow A$  astfel încât  $f(x) = a(x)$  oricare  $x \in X$ .