Logică Matematică şi Computaţională – Subiecte de Examen

Claudia MUREŞAN

cmuresan@fmi.unibuc.ro, claudia.muresan@g.unibuc.ro

Universitatea din Bucureşti, Facultatea de Matematică şi Informatică

Timp de lucru: 3 ore.

Materiale permise: orice material tipărit sau scris de mână.

Este interzisă folosirea dispozitivelor electronice.

Este interzisă părăsirea sălii de examen timp de o oră şi jumătate din momentul primirii subiectelor.

Punctaj (maxim **11,5 puncte**; nota: $\min\{10, \text{punctaj}\}$; observaţi că **nota 10** poate fi obţinută, în cazul unui punctaj maxim pentru TEMELE COLECTIVE, cu două *cerinţe reduse* de programare în Prolog, pentru **jumătate de punctaj**, dacă rezolvările sunt aproape perfecte):

- **1 punct** din oficiu;
- **3 puncte** pentru TEMELE COLECTIVE;
- fiecare dintre cele două cerinţe *numerotate* ale fiecărui exerciţiu: **1,25 puncte**.

Pentru cerinţele de programare în Prolog se poate folosi orice predicat predefinit, precum şi orice predicat scris la LABORATOR sau într–o TEMĂ COLECTIVĂ, utilizând directiva pentru includerea bazelor de cunoştinţe *labNrlmcVer.pl* şi *temeleNr.pl* în cea curentă, cu condiţia respectării **denumirilor predicatelor** din FIŞIERELE .PL de la LABORATOR şi din ENUNŢURILE TEMELOR COLECTIVE. Toate celelalte predicate auxiliare necesare pentru a defini predicatele cerute trebuie scrise pe lucrarea de examen. Atenţie la cerinţa ca predicatele auxiliare să poată fi aplicate pentru orice argumente de tipurile specificate: menţiunea ca acestea să fie **arbitrare**!

Fiecare coală (nu pagină) din lucrare va fi semnată cu numele şi prenumele în clar. Pe prima pagină vor fi scrise şi numărul grupei şi data examenului.

Toate subiectele vor fi tratate pe lucrarea de examen. Dacă nu aveţi nicio idee de abordare a unui exerciţiu, atunci veţi scrie noţiuni şi/sau proprietăţi teoretice din curs şi/sau predicate din laborator legate de acel exerciţiu pe lucrare. Dar **numai** în cazul în care nu ştiţi să abordaţi exerciţiul. Orice încercare de abordare a unui exerciţiu valorează mai mult, ca punctaj, decât o astfel de tratare minimală, scriind teorie sau predicate făcute la laborator, iar acestea nu se punctează în plus în cazul în care abordaţi acel exerciţiu.

1 Lista 1 de subiecte

Exerciţiul 1. Să se determine toate funcţiile strict descrescătoare $f : \mathcal{L}_2 \times \mathcal{L}_3 \to \mathcal{L}_2 \oplus \mathcal{L}_2^2$ de la produsul direct al lanţului cu exact 2 elemente cu lanţul cu exact 3 elemente la suma ordinală a lanţului cu exact 2 elemente cu, rombul şi să se arate că niciuna nu e injectivă:

① matematic;

② prin următoarele predicate în Prolog:

- un predicat unar $fctL2xL3laL2plusL2xL2(-ListaFctStrDesCresc)$, care determină în argumentul său $ListaFctStrDesCresc$ lista funcţiilor strict descrescătoare de la $\mathcal{L}_2 \times \mathcal{L}_3$ la $\mathcal{L}_2 \oplus \mathcal{L}_2^2$, folosind un predicat auxiliar care determină lista funcţiilor strict descrescătoare între două poseturi finite arbitrare;
- un predicat zeroar $niciunainj$ care întoarce $true$ ddacă toate funcţiile strict descrescătoare $f : \mathcal{L}_2 \times \mathcal{L}_3 \to \mathcal{L}_2 \oplus \mathcal{L}_2^2$ sunt neinjective, care apelează predicatul $fctL2xL3laL2plusL2xL2(-ListaFctStrDesCresc)$, apoi aplică listei $ListaFctStrDesCresc$ un predicat care testează dacă toate funcţiile între două mulţimi finite arbitrare dintr–o listă arbitrară de funcţii între aceste mulţimi sunt neinjective.

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie doar predicatul unar $fctL2xL3laL2plusL2xL2$ definit ca mai sus.

Exerciţiul 2. Fie V mulţimea variabilelor propoziţionale, iar E mulţimea enunţurilor logicii propoziţionale clasice. Să se demonstreze că, pentru orice $\Sigma, \Delta \in \mathcal{P}(E)$ şi orice $\alpha, \beta, \gamma \in E$, are loc următoarea regulă de deducţie:

$$\frac{\Sigma \vdash \alpha \vee (\beta \to \gamma),\ \Delta \vdash \gamma \to \alpha}{\Sigma \cup \Delta \vdash \beta \to \alpha}$$

① matematic;

② prin următoarele predicate în Prolog:

• un predicat ternar $ipoteza1(+Alfa, +Beta, +Gama)$, care întoarce valoarea de adevăr a enunţului compus $\alpha \vee (\beta \to \gamma)$ în funcţie de valorile de adevăr $Alfa$, $Beta$, respectiv $Gama$ ale enunţurilor α, β, respectiv γ într–o interpretare arbitrară;
• un predicat binar $ipoteza2(+Alfa, +Gama)$, care întoarce valoarea de adevăr a enunţului compus $\gamma \to \alpha$ în funcţie de valorile de adevăr $Alfa$, respectiv $Gama$ ale enunţurilor α, respectiv γ intr–o interpretare arbitrară;
• un predicat binar $concluzia(+Alfa, +Beta)$, care întoarce valoarea de adevăr a enunţului compus $\beta \to \alpha$ în funcţie de valorile de adevăr $Alfa$, respectiv $Beta$ ale enunţurilor α, respectiv β intr–o interpretare arbitrară;
• un predicat zeroar *regded*, care întoarce *true* ddacă din cele două ipoteze de mai sus se deduce concluzia de mai sus pentru orice triplet de valori booleene ale enunţurilor α, β, γ într–o interpretare (astfel că, dacă acest predicat e satisfăcut, atunci, în particular, considerând interpretările care satisfac pe $\Sigma \cup \Delta$, avem o demonstraţie semantică, prin tabel de adevăr, pentru regula de deducţie de mai sus).

Desigur, la fel ca în lecţiile de laborator, prin **valoare de adevăr** a unui enunţ $\varepsilon \in E$ într–o interpretare $h : V \to \mathcal{L}_2$ ne referim la valoarea $f(\widetilde{h}(\varepsilon))$ a funcţiei $f \circ \widetilde{h} : E \to \{false, true\}$ în enunţul ε, unde $\widetilde{h} : E \to \mathcal{L}_2$ este prelungirea lui h la mulţimea E a enunţurilor care transformă conectorii logici în operaţii booleene, iar $f : \mathcal{L}_2 \to \{false, true\}$ este izomorfismul boolean: $f(0) = false$, $f(1) = true$.

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie unul dintre predicatele *ipoteza*1, *ipoteza*2, *ipoteza*3 şi *concluzia*.

Exerciţiul 3. Considerăm signatura de ordinul I: $\tau = (1; 2; \emptyset)$, simbolul de operaţie unară f şi simbolul de relaţie binară R, o mulţime $A = \{a, b, c, d\}$ având $|A| = 4$ şi o structură de ordinul I de signatură τ: $\mathcal{A} = (A, f^{\mathcal{A}}, R^{\mathcal{A}})$, cu mulţimea suport A, iar $f^{\mathcal{A}} : A \to A$ şi $R^{\mathcal{A}} \subseteq A^2$, astfel încât, dacă $(A, \leq)$ este posetul având relaţia de succesiune $\prec = \{(a, b), (b, d)\}$:

- $f^{\mathcal{A}}$ este izomorfism de poseturi de la $(A, \leq)$ la dualul său $(A, \geq)$;
- $R^{\mathcal{A}}$ este relaţia de ordine $\geq$ a posetului dual lui $(A, \leq)$.

Considerăm două variabile distincte $x, y \in Var$ şi enunţul:

$$\varepsilon = \forall\, x\, \forall\, y\, [R(x, f(y)) \to \neg\, R(f(x), f(f(y)))].$$

Să se determine funcţia $f^{\mathcal{A}}$ şi relaţia binară $R^{\mathcal{A}}$, apoi să se determine dacă $\mathcal{A} \vDash \varepsilon$:

① matematic;

② prin următoarele predicate în Prolog, pentru care mulţimea $A = \{a, b, c, d\}$ va fi introdusă ca listă de constante şi relaţia de succesiune $\prec = \{(a, b), (b, d)\}$ ca listă de perechi de constante, iar restul argumentelor vor fi calculate în aceste predicate:

• un predicat binar $posetA(-MultElemA, -OrdA)$, care instanţiază variabila $MultElemA$ cu lista de constante care dă mulţimea A, iar în argumentul $OrdA$ determină relaţia de ordine pe $MultElemA$ având $\prec = \{(a, b), (b, d)\}$ ca relaţie de succesiune asociată (direct cu închiderea reflexiv–tranzitivă a lui $\prec$, nu selectând dintre relaţiile de ordine pe $MultElemA$ pe aceea a cărei relaţie de succesiune asociată este egală ca mulţime cu lista corespunzătoare lui $\prec$; deci procedaţi aşa cum am construit, la laborator, orice poset din mulţimea sa de elemente şi relaţia sa de succesiune);
• un predicat unar $detf(-Fctf)$, care întoarce în argumentul său $Fctf$ funcţia de la A la A care este bijectivă, crescătoare şi cu inversa crescătoare de la posetul $(MultElemA, OrdA)$ determinat de predicatul $posetA(-MultElemA, -OrdA)$ şi dualul $(MultElemA, InvOrdA)$ al acestui poset, calculând inversa $InvOrdA$ a relaţiei $OrdA$, apoi aplicând poseturilor $(MultElemA, OrdA)$ şi $(MultElemA, InvOrdA)$ un predicat care determină izomorfismele între oricare două poseturi finite;
• un predicat unar $detR(-RelR)$, care calculează în argumentul său $RelR$ inversa relaţiei $OrdA$ determinate de predicatul $posetA(-MultElemA, -OrdA)$;
• un predicat zeroar $verifAsatepsilon$, care întoarce *true* ddacă $\mathcal{A} \vDash \varepsilon$, efectuând o demonstraţie semantică, prin testarea perechilor de valori din mulţimea A pentru variabilele x, y într–o interpretare arbitrară (**indicaţie:** atenţie la reprezentarea unei funcţii ca relaţie binară funcţională totală, deci ca listă de perechi, care poate fi furnizată unui predicat unar într–o singură clauză, versus reprezentarea ei prin atâtea clauze pentru un predicat binar câte elemente are domeniul acelei funcţii; a doua reprezentare e suficientă pentru cerinţa redusă de mai jos, dar predicatele pentru generare şi colectare de funcţii scrise la laborator folosesc prima dintre aceste reprezentări).

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie doar predicatul zeroar $verifAsatepsilon$, introducând direct operaţia unară $f^{\mathcal{A}}$ şi relaţia binară $R^{\mathcal{A}}$ în baza de cunoştinţe.

2 Lista 2 de subiecte

Exerciţiul 4. Să se determine toate funcţiile strict crescătoare $f : \mathcal{L}_2 \times \mathcal{L}_3 \to \mathcal{L}_2^2 \oplus \mathcal{L}_2$ de la produsul direct al lanţului cu exact 2 elemente cu lanţul cu exact 3 elemente la suma ordinală a rombului cu lanţul cu exact 2 elemente şi să se arate că toate păstrează relaţia de succesiune, i.e. satisfac, pentru orice $x, y \in \mathcal{L}_2 \times \mathcal{L}_3$: $x \prec y \implies f(x) \prec f(y)$, unde am notat cu $\prec$ atât relaţia de succesiune a lui $\mathcal{L}_2 \times \mathcal{L}_3$, cât şi pe cea a lui $\mathcal{L}_2^2 \oplus \mathcal{L}_2$:

① matematic;

② prin următoarele predicate în Prolog:

• un predicat unar $fctL2xL3laL2xL2plusL2(-ListaFctStrCresc)$, care determină în argumentul său $ListaFctStrCresc$ lista funcţiilor strict crescătoare de la $\mathcal{L}_2 \times \mathcal{L}_3$ la $\mathcal{L}_2^2 \oplus \mathcal{L}_2$, folosind un predicat auxiliar care determină lista funcţiilor strict crescătoare între două poseturi finite arbitrare;
• un predicat zeroar $toatepastrsucc$ care întoarce $true$ ddacă toate funcţiile strict crescătoare $f : \mathcal{L}_2 \times \mathcal{L}_3 \to \mathcal{L}_2^2 \oplus \mathcal{L}_2$ păstrează relaţia de succesiune, care apelează predicatul $fctL2xL3laL2xL2plusL2(-ListaFctStrCresc)$, apoi aplică listei $ListaFctStrCresc$ un predicat care testează dacă toate funcţiile între două poseturi finite arbitrare dintr–o listă arbitrară de funcţii între aceste poseturi păstrează relaţia de succesiune, i.e. duc relaţia de succesiune de pe domeniul lor în relaţia de succesiune de pe codomeniul lor.

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie doar predicatul unar $fctL2xL3laL2xL2plusL2$ definit ca mai sus.

Exerciţiul 5. Fie V mulţimea variabilelor propoziţionale, iar E mulţimea enunţurilor logicii propoziţionale clasice. Să se demonstreze că, pentru orice $\Sigma, \Delta \in \mathcal{P}(E)$ şi orice $\alpha, \beta, \gamma \in E$, are loc următoarea regulă de deducţie:

$$\frac{\Sigma \vdash \alpha \to (\beta \vee \gamma),\ \Delta \vdash \beta \to \neg\,\alpha,\ \Sigma \cap \Delta \vdash \neg\,\alpha \to (\neg\,\beta \wedge \gamma)}{\Sigma \cup \Delta \vdash \neg\,\beta \wedge \gamma}$$

① matematic;

② prin următoarele predicate în Prolog:

- un predicat ternar $ipoteza1(+Alfa, +Beta, +Gama)$, care întoarce valoarea de adevăr a enunţului compus $\alpha \to (\beta \vee \gamma)$ în funcţie de valorile de adevăr $Alfa$, $Beta$, respectiv $Gama$ ale enunţurilor α, β, respectiv γ într–o interpretare arbitrară;
- un predicat binar $ipoteza2(+Alfa, +Beta)$, care întoarce valoarea de adevăr a enunţului compus $\beta \to \neg\,\alpha$ în funcţie de valorile de adevăr $Alfa$, respectiv $Beta$ ale enunţurilor α, respectiv β intr–o interpretare arbitrară;
- un predicat ternar $ipoteza3(+Alfa, +Beta, +Gama)$, care întoarce valoarea de adevăr a enunţului compus $\neg\,\alpha \to (\neg\,\beta \wedge \gamma)$ în funcţie de valorile de adevăr $Alfa$, $Beta$, respectiv $Gama$ ale enunţurilor α, β, respectiv γ intr–o interpretare arbitrară;
- un predicat binar $concluzia(+Beta, +Gama)$, care întoarce valoarea de adevăr a enunţului compus $\neg\,\beta \wedge \gamma$ în funcţie de valorile de adevăr $Beta$, respectiv $Gama$ ale enunţurilor β, respectiv γ intr–o interpretare arbitrară;
- un predicat zeroar $regded$, care întoarce $true$ ddacă din cele trei ipoteze de mai sus se deduce concluzia de mai sus pentru orice triplet de valori booleene ale enunţurilor α, β, γ într–o interpretare (astfel că, dacă acest predicat e satisfăcut, atunci, în particular, considerând interpretările care satisfac pe $\Sigma \cup \Delta$, avem o demonstraţie semantică, prin tabel de adevăr, pentru regula de deducţie de mai sus).

Desigur, la fel ca în lecţiile de laborator, prin **valoare de adevăr** a unui enunţ $\varepsilon \in E$ într–o interpretare $h : V \to \mathcal{L}_2$ ne referim la valoarea $f(\widetilde{h}(\varepsilon))$ a funcţiei $f \circ \widetilde{h} : E \to \{false, true\}$ în enunţul ε, unde $\widetilde{h} : E \to \mathcal{L}_2$ este prelungirea lui h la mulţimea E a enunţurilor care transformă conectorii logici în operaţii booleene, iar $f : \mathcal{L}_2 \to \{false, true\}$ este izomorfismul boolean: $f(0) = false$, $f(1) = true$.

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie unul dintre predicatele $ipoteza1$, $ipoteza2$, $ipoteza3$ şi $concluzia$.

Exerciţiul 6. Considerăm signatura de ordinul I: $\tau = (1; 2; \emptyset)$, simbolul de operaţie unară f şi simbolul de relaţie binară R, o mulţime $A = \{a, b, c, d\}$ având $|A| = 4$ şi o structură de ordinul I de signatură τ: $\mathcal{A} = (A, f^{\mathcal{A}}, R^{\mathcal{A}})$, cu mulţimea suport A, iar $f^{\mathcal{A}} : A \to A$ şi $R^{\mathcal{A}} \subseteq A^2$, astfel încât, dacă $(A, \leq)$ este posetul având relaţia de succesiune $\prec = \{(a, b), (c, d)\}$:

- $f^{\mathcal{A}}$ este izomorfism de poseturi de la $(A, \leq)$ la dualul său $(A, \geq)$ care satisface $f^{\mathcal{A}}(\{a, b\}) \cap \{a, b\} = \emptyset$;
- $R^{\mathcal{A}}$ este relaţia de echivalenţă pe A generată de $\prec$: $R^{\mathcal{A}} = \mathcal{E}(\prec)$.

Considerăm două variabile distincte $x, y \in Var$ şi enunţul:

$$\varepsilon = \forall\, x\, \exists\, y\, [R(x, y) \to R(f(x), f(f(y)))].$$

Să se determine funcţia $f^{\mathcal{A}}$ şi relaţia binară $R^{\mathcal{A}}$, apoi să se determine dacă $\mathcal{A} \vDash \varepsilon$:

① matematic;

② prin următoarele predicate în Prolog, pentru care mulţimea $A = \{a, b, c, d\}$ va fi introdusă ca listă de constante şi relaţia de succesiune $\prec = \{(a, b), (c, d)\}$ ca listă de perechi de constante, iar restul argumentelor vor fi calculate în aceste predicate:

• un predicat binar $posetA(-MultElemA, -OrdA)$, care instanţiază variabila $MultElemA$ cu lista de constante care dă mulţimea A, iar în argumentul $OrdA$ determină relaţia de ordine pe $MultElemA$ având $\prec = \{(a, b), (c, d)\}$ ca relaţie de succesiune asociată (direct cu închiderea reflexiv–tranzitivă a lui $\prec$, nu selectând dintre relaţiile de ordine pe $MultElemA$ pe aceea a cărei relaţie de succesiune asociată este egală ca mulţime cu lista corespunzătoare lui $\prec$; deci procedaţi aşa cum am construit, la laborator, orice poset din mulţimea sa de elemente şi relaţia sa de succesiune);
• un predicat unar $detf(-Fctf)$, care întoarce în argumentul său $Fctf$ funcţia de la A la A care este bijectivă, crescătoare şi cu inversa crescătoare de la posetul $(MultElemA, OrdA)$ determinat de predicatul $posetA(-MultElemA, -OrdA)$ şi dualul $(MultElemA, InvOrdA)$ al acestui poset şi are proprietatea că $f(a) \notin \{a, b\}$, $f(b) \notin \{a, b\}$, calculând inversa $InvOrdA$ a relaţiei $OrdA$, apoi aplicând poseturilor $(MultElemA, OrdA)$ şi $(MultElemA, InvOrdA)$ un predicat care determină izomorfismele între oricare două poseturi finite, iar, dintre izomorfismele determinate astfel, selectându–l pe cel cu proprietatea $f(a) \notin \{a, b\}$ şi $f(b) \notin \{a, b\}$;
• un predicat unar $detR(-RelR)$, care calculează în argumentul său $RelR$ relaţia de echivalenţă generată de $\prec = \{(a, b), (c, d)\}$, folosind un predicat care determină echivalenţa generată de o relaţie binară arbitrară pe o mulţime finită;
• un predicat zeroar $verifAsatepsilon$, care întoarce $true$ ddacă $\mathcal{A} \vDash \varepsilon$, efectuând o demonstraţie semantică, prin testarea perechilor de valori din mulţimea A pentru variabilele x, y într–o interpretare arbitrară (**indicaţie:** atenţie la reprezentarea unei funcţii ca relaţie binară funcţională totală, deci ca listă de perechi, care poate fi furnizată unui predicat unar într–o singură clauză, versus reprezentarea ei prin atâtea clauze pentru un predicat binar câte elemente are domeniul acelei funcţii; a doua reprezentare e suficientă pentru cerinţa redusă de mai jos, dar predicatele pentru generare şi colectare de funcţii scrise la laborator folosesc prima dintre aceste reprezentări).

Pentru **jumătate din punctajul** de la **această a doua cerinţă**, puteţi scrie doar predicatul zeroar $verifAsatepsilon$, introducând direct operaţia unară $f^{\mathcal{A}}$ şi relaţia binară $R^{\mathcal{A}}$ în baza de cunoştinţe.