FMI, Anul III
Elemente de securitate şi logică aplicată

Examen

Nume: ______________________

Prenume: ____________________

Grupa: ______________________

Modulul 3: Logică pentru cunoaştere şi demonstrare automată

(P1) [2 puncte]

(i) Fie $p, q \in PROP$. Verificaţi dacă următoarele formule sunt valide în clasa tuturor cadrelor Kripke pentru ML_0:

(a) $\Box p \to \Box\Box p$.

(b) $\Diamond(p \wedge q) \to \Diamond p \wedge \Diamond q$.

(ii) Demonstraţi că pentru orice formule φ, ψ ale lui ML_0,

$$\vdash_{\boldsymbol{K}} \neg\varphi \to \neg\psi \quad \text{implică} \quad \vdash_{\boldsymbol{K}} \Box\psi \to \Box\varphi.$$

(iii) Fie $\mathcal{M}_c$ modelul epistemic care descrie jocul de cărţi, definit în curs. Verificaţi dacă următoarea afirmaţie este adevărată:

$$\mathcal{M}_c, (A, B) \Vdash K_2 \neg K_1 2B.$$

Demonstraţie:

(i) (a) Răspunsul este NU. Dăm următorul contraexemplu. Fie cadrul $\mathcal{F} = (W, R)$, unde

$$W = \{0, 1, 2\}, \quad R = \{(0, 1), (1, 2)\}$$

şi $\mathcal{M} = (\mathcal{F}, V)$ cu $V(p) = \{1\}$.

Atunci $\mathcal{M}, 0 \Vdash \Box p$ ddacă $\mathcal{M}, 1 \Vdash p$ (deoarece singurul punct R-accesibil din 0 este 1) ddacă $1 \in V(p)$, ceea ce este adevărat.

Pe de altă parte, $\mathcal{M}, 0 \Vdash \Box\Box p$ ddacă $\mathcal{M}, 1 \Vdash \Box p$ ddacă $\mathcal{M}, 2 \Vdash p$ (deoarece singurul punct R-accesibil din 1 este 2) ddacă $2 \in V(p)$, ceea ce este fals

Prin urmare, $\mathcal{M}, 0 \not\Vdash \Box p \to \Box\Box p$.

(b) Răspunsul este DA. Fie $\mathcal{F}$ un cadru Kripke arbitrar, w o stare din $\mathcal{F}$ şi $\mathcal{M} = (\mathcal{F}, V)$ un model Kripke bazat pe $\mathcal{F}$. Trebuie să arătăm că

$$\mathcal{M}, w \Vdash \Diamond(p \wedge q) \to \Diamond p \wedge \Diamond q.$$

Presupunem că $\mathcal{M}, w \Vdash \Diamond(p \wedge q)$. Atunci există $v \in W$ astfel încât Rwv şi $\mathcal{M}, v \Vdash p \wedge q$, deci $\mathcal{M}, v \Vdash p$ şi $\mathcal{M}, v \Vdash q$.

Avem că

(1) Rwv şi $\mathcal{M}, v \Vdash p$, deci $\mathcal{M}, w \Vdash \Diamond p$;

(2) Rwv şi $\mathcal{M}, v \Vdash q$, deci $\mathcal{M}, w \Vdash \Diamond q$.

Prin urmare, $\mathcal{M}, w \Vdash \Diamond p \wedge \Diamond q$.

(ii) Prezentăm următoarea $\boldsymbol{K}$-demonstraţie:

(1)	$\vdash_{\boldsymbol{K}} \neg\varphi \to \neg\psi$	ipoteză
(2)	$\vdash_{\boldsymbol{K}} (\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$	(Taut)
(3)	$\vdash_{\boldsymbol{K}} \psi \to \varphi$	(MP): (1), (2)
(4)	$\vdash_{\boldsymbol{K}} \Box\psi \to \Box\varphi$	Exemplul 2.31: (3)

(iii) $\mathcal{M}_c, (A, B) \Vdash K_2 \neg K_1 2B$ ddacă agentul 2 ştie că agentul 1 nu ştie că agentul 2 are cartea B.

Avem că $\mathcal{M}_c, (A, B) \Vdash K_2 \neg K_1 2B$ ddacă $\Big($pentru orice $(X, Y) \in W$,

$\mathcal{K}_2(A, B)(X, Y)$ implică $\mathcal{M}_c, (X, Y) \Vdash \neg K_1 2B\Big)$ ddacă

$\Big(\mathcal{M}_c, (A, B) \Vdash \neg K_1 2B$ şi $\mathcal{M}_c, (C, B) \Vdash \neg K_1 2B\Big)$ ddacă

$\Big(\mathcal{M}_c, (A, B) \not\Vdash K_1 2B$ şi $\mathcal{M}_c, (C, B) \not\Vdash K_1 2B\Big)$ ddacă

$\Big(\big($există $(X, Y) \in W$ astfel încât $\mathcal{K}_1(A, B)(X, Y)$ şi $\mathcal{M}_c, (X, Y) \not\Vdash 2B\big)$

şi $\Big($există $(X,Y) \in W$ astfel încât $\mathcal{K}_1(C,B)(X,Y)$ şi $\mathcal{M}_c,(X,Y) \not\Vdash 2B)\Big)$ ddacă

$\Big(\big(\mathcal{M}_c,(A,B) \not\Vdash 2B$ sau $\mathcal{M}_c,(A,C) \not\Vdash 2B\big)$ şi $\big(\mathcal{M}_c,(C,B) \not\Vdash 2B$ sau $\mathcal{M}_c,(C,A) \not\Vdash 2B\big)\Big)$,

ceea ce este adevărat, deoarece $\mathcal{M}_c,(A,C) \not\Vdash 2B$ şi $\mathcal{M}_c,(C,A) \not\Vdash 2B$.

(P2) [2 puncte]

(i) Scrieţi o demonstraţie în Lean pentru teorema `th1` de mai jos.

```
variable {α : Type} (p q : α → Prop)

theorem th1 :
  (∀ x, p x) ∧ (∀ x, q x) → (∀ x, ¬(¬p x ∨ ¬q x))
```

(ii) Definiţi, prin recursie structurală pe numere naturale şi fără a folosi operaţia de înmulţire "`*`" predefinită în Lean, o funcţie `mymul : Nat → Nat → Nat` astfel încât, pentru orice `n m : Nat`, `mymul n m` să returneze produsul numerelor `n` şi `m`.

Rezolvare:

(i)
```
variable {α : Type} (p q : α → Prop)

theorem th1 :
  (∀ x, p x) ∧ (∀ x, q x) → (∀ x, ¬(¬p x ∨ ¬q x)) := by
  intros h a h'
  cases h with
  | intro hp hq =>
  specialize hp a
  specialize hq a
  cases h'
  . contradiction
  . contradiction
```

(ii)
```
def mymul : Nat → Nat → Nat := fun n m => match n with
| 0 => 0
| n + 1 => (mymul n m) + m
```