

Egzamin poprawkowy z logiki i teorii typów, 19 lutego 2025

1. Które z następujących formuł są intuicjonistycznymi twierdzeniami, a które nie?

- (a) $((\neg\neg p \rightarrow p) \rightarrow \neg\neg p) \rightarrow \neg\neg p$;
- (b) $((\neg\neg p \rightarrow p) \rightarrow \neg p) \rightarrow \neg p$;
- (c) $((\neg\neg p \rightarrow p) \rightarrow p \vee \neg p) \rightarrow \neg p \vee \neg\neg p$.

2. Udowodnić, że jeśli formuła postaci $(\neg\neg\alpha \rightarrow \alpha) \rightarrow (\neg\alpha \vee \neg\neg\alpha)$ jest twierdzeniem zdaniowej logiki intuicjonistycznej, to także formuła $\neg\alpha \vee \neg\neg\alpha$ jest twierdzeniem.

3. Która z następujących formuł jest intuicjonistycznym twierdzeniem? W przypadku odpowiedzi negatywnej: czy istnieje kontrmodel Kripkego o stałych dziedzinach i czy istnieje kontrmodel Kripkego o skończonej liczbie stanów?

- (a) $(\forall a (P(a) \vee Q(a))) \rightarrow (\exists a P(a)) \vee (\forall a Q(a))$;
- (b) $(\forall a (\neg P(a) \rightarrow Q(a))) \rightarrow (\exists a P(a)) \vee (\forall a Q(a))$;
- (c) $(\forall a (P(a) \vee Q(a))) \rightarrow (\neg\exists a P(a)) \rightarrow (\forall a Q(a))$.

4. *Automat z dwoma stosami* ma skończony zbiór stanów i dwa stosy (słowa nad ustalonym alfabetem), na których może wykonywać operacje **push** i **pop**. Zmiana stanu może zależeć od symboli na szczycie stosów, albo od tego, czy któryś stos jest pusty. (Można zakładać, że na spodzie leży zawsze znak $\square$, którego się nigdy nie zdejmuje.) W konfiguracji początkowej oba stosy są puste.

Dla danego automatu z dwoma stosami proszę opisać formułę pierwszego rzędu, która jest intuicjonistycznym twierdzeniem wtedy i tylko wtedy, gdy automat uruchomiony w konfiguracji początkowej zatrzyma się w stanie końcowym q_f . Formuła powinna być obliczalna w czasie wielomianowym.

Rozwiązania

1a: Formuła $((\neg\neg p \rightarrow p) \rightarrow \neg\neg p) \rightarrow \neg\neg p$; jest twierdzeniem. Dowód zapiszemy w postaci lambda-termu $\lambda x^{(\neg\neg p \rightarrow p) \rightarrow \neg\neg p} \lambda y^{\neg p}. x(\lambda z^{\neg\neg p}. zy[p])y$. (Tutaj $zy : \perp$.)

1b: Formuła $((\neg\neg p \rightarrow p) \rightarrow \neg p) \rightarrow \neg p$ też jest twierdzeniem. Dowód $\lambda x^{(\neg\neg p \rightarrow p) \rightarrow \neg p} \lambda y^p. x(\lambda z^{\neg\neg p}. y)y$.

1c: Formuła $(\neg\neg p \rightarrow p) \rightarrow p \vee \neg p \rightarrow \neg p \vee \neg\neg p$ nie jest twierdzeniem. Istotnie, rozpatrzmy model Kripkego o czterech stanach 0,1,2,3, uporządkowany w ten sposób, że $0 < 1 < 3$, oraz $0 < 2$, natomiast element 2 jest nieporównywalny z 1 i z 3. Atom p jest wymuszony wyłącznie w stanie 3. Zatem tylko stany 2 i 3 wymuszają warunek $\neg\neg p \rightarrow p$, skąd $0 \Vdash (\neg\neg p \rightarrow p) \rightarrow p \vee \neg p$. Tymczasem $0 \nVdash \neg p \vee \neg\neg p$.

2: Skoro $(\neg\neg\alpha \rightarrow \alpha) \rightarrow (\neg\alpha \vee \neg\neg\alpha)$ jest twierdzeniem, to ma dowód normalny, czyli w rozszerzonym rachunku lambda istnieje term zamknięty tego typu w postaci normalnej. Taki term M musi mieć postać $\lambda x^{\neg\neg\alpha \rightarrow \alpha}. M$, gdzie $x : \neg\neg\alpha \rightarrow \alpha \vdash M : \neg\alpha \vee \neg\neg\alpha$. Są trzy możliwości: term M jest albo konstruktorem postaci $\text{in}_1(P)$ lub $\text{in}_2(P)$ albo eliminatorem zaczynającym się od aplikacji xP . W każdym przypadku mamy albo $x : \neg\neg\alpha \rightarrow \alpha \vdash P : \neg\alpha$ albo $x : \neg\neg\alpha \rightarrow \alpha \vdash P : \neg\neg\alpha$. Stąd $\vdash (\neg\neg\alpha \rightarrow \alpha) \rightarrow \neg\alpha$ albo $\vdash (\neg\neg\alpha \rightarrow \alpha) \rightarrow \neg\neg\alpha$. Z zadań 1a i 1b natychmiast wynika, że albo $\vdash \neg\neg\alpha$ albo $\vdash \neg\alpha$.

3a: Formuła $\forall a (P(a) \vee Q(a)) \rightarrow (\exists a P(a)) \vee (\forall a Q(a))$ nie jest twierdzeniem. Ma kontrmodel Kripkego postaci $\langle \{0,1\}, \leq, \{A_0, A_1\} \rangle$, gdzie $0 < 1$, oraz $A_1 = \langle \{1\}, P^1, Q^1 \rangle$, $A_2 = \langle \{1,2\}, P^2, Q^2 \rangle$. Relacje są takie: $P^1 = \emptyset$, $Q^1 = Q^2 = \{1\}$, $P^2 = \{2\}$. Ale ta formuła jest wymuszona we wszystkich modelach o stałych dziedzinach. Jeśli bowiem $c \Vdash \forall a (P(a) \vee Q(a))$, to $c, v \Vdash P(a) \vee Q(a)$ dla każdego wartościowania v . Czyli każdy element dziedziny (tej samej w każdym stanie) musi już w stanie c należeć do jednej z dwóch relacji. Reszta jest oczywista.

3b: Formuła $(\forall a (\neg P(a) \rightarrow Q(a))) \rightarrow (\exists a P(a)) \vee (\forall a Q(a))$ też nie jest twierdzeniem i ma nawet kontrmodel o skończonej liczbie stanów i stałych dziedzinach. Na przykład $\langle \{0,1\}, \leq, \{A_0, A_1\} \rangle$, gdzie $0 < 1$, oraz $A_1 = \langle \{1\}, P^1, Q^1 \rangle$, $A_2 = \langle \{1\}, P^2, Q^2 \rangle$. Tym razem relacje są takie: $Q_1 = Q_2 = P_1 = \emptyset$, $P_2 = \{1\}$. Ponieważ $\neg P(x)$ nigdzie nie jest wymuszone, więc $0 \Vdash \forall a (\neg P(a) \rightarrow Q(a))$. Tymczasem oczywiście $0 \nVdash (\exists a P(a)) \vee (\forall a Q(a))$.

3c: Formuła $(\forall a (P(a) \vee Q(a))) \rightarrow (\neg\exists a P(a)) \rightarrow (\forall a Q(a))$ jest twierdzeniem, a jej dowód można zapisać tak: $\lambda X^{\forall a (P(a) \vee Q(a))} \lambda Y^{\neg\exists a P(a)} \lambda a. Xa [U:P(a). Y[a,U][Q(a)]; V:Q(a). V]$. (Tutaj $Y[a,U] : \perp$.)

4: Można użyć dwuargumentowego symbolu relacyjnego $q(x,y)$ do reprezentowania konfiguracji ze stanem q , przy czym zmienne x i y służą jako wskaźniki do wierzchołków dwóch stosów. Symbol relacyjny S , podobnie jak w zadaniu 32, odgrywa rolę następnika i wiąże ze sobą sąsiednie pozycje na stosie. Jednoargumentowy symbol $\square$ reprezentuje dno stosu. Przykład: otoczenie Γ kodujące stosy ABC i BD może się składać z formuł:

$$\square(x_0), \square(y_0), S(x_0, x_1), S(x_1, x_2), S(x_2, x_3), A(x_1), B(x_2), C(x_3), S(y_0, y_1), S(y_1, y_2), B(y_1), D(y_2).$$

Celem dowodowym dla tak zakodowanej konfiguracji $\langle q, ABC, BD \rangle$ będzie formuła $q(x_3, y_2)$.

Instrukcje maszyny (dla stanu p) można kodować na przykład tak:

pop₂; **goto** q : $\forall xyz(S(z, y) \rightarrow q(x, z) \rightarrow p(x, y));$
push₁(B); **goto** q : $\forall xy(\forall z(S(x, z) \rightarrow B(z) \rightarrow q(z, y)) \rightarrow p(x, y));$
peek₁(A); **goto** q : $\forall xy(A(y) \rightarrow q(x, y) \rightarrow p(x, y)).$

Szukana formuła ma postać $\alpha_1 \rightarrow \dots \rightarrow \alpha_N \rightarrow \forall xy q_f(x, y) \rightarrow \square(x_0) \rightarrow \square(y_0) \rightarrow q_0(x_0, y_0)$, gdzie założenia $\alpha_1, \dots, \alpha_N$ kodują instrukcje jw.