Egzamin z logiki i teorii typów, 1 lutego 2023

1. Które z następujących formuł są intuicjonistycznymi twierdzeniami, a które nie? W przypadku odpowiedzi pozytywnej należy skonstruować dowód (czy można to zrobić bez reguły ex falso?), przy odpowiedzi negatywnej można np. wskazać model Kripkego (wystarczy jeden sposób).

   (a) $((\neg\neg p \to p) \to q) \to q$.

   (b) $((\neg\neg p \to p) \to \neg q) \to \neg q$

2. Niech $\gamma = \forall x \exists y (P(y) \wedge (Q(y) \to Q(x)))$, $\beta = \neg\forall x\, Q(x)$ oraz $\delta = \forall x (P(x) \to (Q(x) \vee S))$. Udowodnić, że formuła $\gamma \to \beta \to \delta \to S$ nie jest intuicjonistycznym twierdzeniem, ale jest wymuszana przez wszystkie modele Kripkego o stałych dziedzinach.

3. *Automat z kolejką* ma skończony zbiór stanów i kolejkę (słowo nad ustalonym alfabetem), na której może wykonywać takie operacje:

   – p : `insert` (A); `goto` q (w stanie p wstaw A na koniec kolejki; przejdź do stanu q);
   – p : `delete`; `goto` q (w stanie p usuń literę z początku kolejki; przejdź do q);
   – p : `peek` (A); `goto` q (ze stanu p przejdź do q jeśli na początku kolejki jest A).

   Dla danego automatu z kolejką proszę skonstruować formułę pierwszego rzędu, która jest intuicjonistycznym twierdzeniem wtedy i tylko wtedy, gdy automat uruchomiony w stanie q_0 z kolejką $\#$ zatrzyma się w stanie końcowym q_f. Można zakładać, że kolejka nigdy nie jest pusta.

   Wskazówka: Otoczenie Γ kodujące kolejkę $w = A_1 \ldots A_n$ może się składać z formuł atomowych $S(x_1, x_2), S(x_2, x_3), \ldots, S(x_{n-1}, x_n), A_1(x_1), \ldots, A_n(x_n)$ i pewnej ilości śmieci (niepotrzebnych ale nieszkodliwych innych założeń). Cel dowodowy postaci $p(x_1, x_n)$ wskazuje na bieżący stan p i zawiera wskaźniki do końca i początku kolejki. Zadanie polega na dodaniu do Γ dodatkowych założeń Δ, w ten sposób, aby dowód normalny osądu $\Gamma, \Delta \vdash p(x_1, x_n)$ wymagał dowodu osądu $\Gamma', \Delta \vdash p'(x'_1, x'_n)$, gdzie Γ' i $p'(x'_1, x'_n)$ przedstawiają kolejną konfigurację automatu.

Rozwiązania

1a: Rozpatrzmy model Kripkego o dwóch stanach 0 i 1, przy tym niech $0 \nVdash p, q$ i $1 \Vdash p, q$. Wtedy $0 \Vdash \neg\neg p$, więc $0 \nVdash (\neg\neg p \to p)$. Ale $0 \Vdash (\neg\neg p \to p) \to q$ więc formuła (1a) nie jest wymuszona w stanie 0.

1b: Dowód w Coqu jest w pliku `zad1.v`. Co się tyczy dowodu bez ex falso, przypomnijmy, że negacja $\neg\alpha$ to w istocie skrót implikacji $\alpha \to \bot$. Reguła ex falso nadaje stałej $\bot$ szczególną rolę; jeśli ją usuniemy, to $\bot$ staje się taką samą zmienną zdaniową jak inne. Jeśli więc formuła ma dowód bez reguły ex falso, to ma też dowód formuła otrzymana z niej przez zamianę każdej negacji $\neg\alpha$ na $\alpha \to r$, gdzie r jest nową zmienną. W naszym przypadku byłaby to formuła $(((p \to r) \to r) \to p) \to q \to r) \to q \to r$. Kontrmodel dla tej formuły ma trzy stany 0, 1 i 2, a relację $\Vdash$ określimy tak: $0, 1, 2 \nVdash p$, $0 \nVdash q, r$, $1, 2 \Vdash q$, $2 \Vdash r$, $1 \nVdash r$. Wtedy jedynym stanem, który forsuje formułę $(p \to r) \to r$ jest 2 i żaden stan nie wymusza $((p \to r) \to r) \to p$. Stąd $0 \Vdash (((p \to r) \to r) \to p) \to q \to r$. Ale $0 \nVdash q \to r$.

Inaczej: Dowód formuły $(((p \to r) \to r) \to p) \to q \to r) \to q \to r$, to lambda-term tego typu, który ma postać normalną $\lambda x^{((p\to r)\to r)\to p)\to q\to r} \lambda y^q.M^r$ Term M musi wyglądać tak: $M = xN^{((p\to r)\to r)\to p}Q^q$, gdzie $N = \lambda z^{(p\to r)\to r}.P^p$. Ale w otoczeniu, w którym typy wszystkich zmiennych x, y, z kończą się na r lub q nie istnieje term typu p.

2: Dowód w Coqu przy założeniu schematu Grzegorczyka jest w pliku `zad2.v`.

Weźmy model Kripkego o dwóch stanach $0 < 1$, i strukturach $\mathcal{A}_0 = \langle\{0\}, P^0, Q^0, S^0\rangle$, $\mathcal{A}_1 = \langle\{0, 1\}, P^1, Q^1, S^1\rangle$, gdzie $Q^0 = Q^1 = \{0\}$, $P^0 = \{0\}$, $P^1 = \{0, 1\}$, zeroargumentowa relacja S^0 nie zachodzi, a S^1 zachodzi. W tym modelu atom $P(y)$ jest wymuszany zawsze, więc $0 \Vdash \forall x(P(x) \wedge (Q(x) \to Q(x)))$, i tym bardziej $0 \Vdash \gamma$. Ponieważ $1 \notin Q^1$, więc $0 \nVdash \beta$. Dalej $0 \Vdash \forall x(Q(x) \vee S)$, bo w stanie 0 mamy Q, a w stanie 1 mamy S. Stąd $0 \Vdash \delta$. Ostatecznie $0 \nVdash \gamma \to \beta \to \delta \to S$, bo $0 \nVdash S$.

3: W zbiorze Δ mamy formułę $\forall xy\, q_f(x, y)$ i po jednej formule dla każdej instrukcji automatu:

- $\forall xyz(S(z, y) \to q(x, z) \to p(x, y))$, dla instrukcji p : `delete`; `goto` q;
- $\forall xy(\forall z(S(z, x) \to A(z) \to q(z, y)) \to p(x, y))$, dla instrukcji p : `insert` (A); `goto` q;
- $\forall xy(A(y) \to q(x, y) \to p(x, y))$, dla instrukcji p : `peek` (A); `goto` q.

Powiemy, że otoczenie $\Gamma(x_1, ..., x_n)$ koduje kolejkę $w = A1 \ldots A_n$, gdy należą do niego formuły jak we wskazówce, a wszystkie pozostałe formuły są atomowe i nie zawierają zmiennych $x_1, ..., x_n$, z co najwyżej jednym wyjątkiem postaci $S(x_n, y)$ gdzie $y \neq x_1, ..., x_n$. Wtedy osąd postaci $\Gamma, \Delta \vdash q(x_1, x_n)$ można tylko udowodnić z pomocą formuły związanej z instrukcją dla stanu q. Na przykład jeśli to jest instrukcja p : `insert` (A); `goto` q, to musimy udowodnić $\Gamma, \Delta \vdash \forall z(S(z, x_1) \to A(z) \to p(z, x_n))$, co prowadzi do zadania $\Gamma, S(z, x_1), A(z), \Delta \vdash p(z, x_n)$. Otoczenie $\Gamma\, S(z, x_1), A(z)$ koduje kolejkę $AA1 \ldots A_n$ z pomocą ciagu zmiennych $z, x_1, \ldots, x_n$ (mogą tam też być nowe „śmieci" ale to nic nie szkodzi) i możemy przez indukcję pokazywać, że osąd ma dowód wtedy i tylko wtedy, gdy maszyna uruchomiona w odpowiedniej konfiguracji dojdzie do stanu końcowego. (Dla konfiguracji końcowej dowód jest trywialny.) Automat uruchomiony w stanie q_0 z kolejką # zatrzyma się więc w stanie q_f wtedy i tylko wtedy, gdy istnieje dowód osądu $\Delta, \#(x_1) \vdash q_0(x_1, x_1)$. A zatem jeśli $\Delta = \{\alpha_1, \ldots, \alpha_N\}$, to nasza formuła ma postać $\alpha_1 \to \cdots \to \alpha_N \to \#(x_1) \to q_0(x_1, x_1)$,