Egzamin z logiki i teorii typów, 19 czerwca 2019

1. Która z następujących formuł zdaniowych jest twierdzeniem intuicjonistycznym?

   (a) $(p \to q \to r) \to ((p \to r) \to s) \to ((q \to r) \to s) \to s$;

   (b) $(p \to q \to r) \to ((p \to r) \to s) \to ((q \to r) \to s) \to (s \to r) \to r$.

2. Która z następujących formuł pierwszego rzędu jest twierdzeniem intuicjonistycznym?

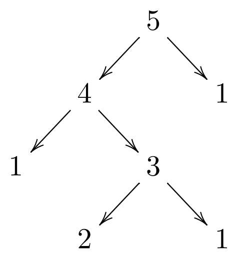
   (a) $\neg\forall x\, P(x) \to \neg\forall x\, \neg\neg P(x)$;

   (b) $\neg\exists x\, P(x) \to \neg\exists x\, \neg\neg P(x)$.

3. Typ $\tau = \forall p\, ((p \to p \to \omega \to p) \to (\omega \to p) \to p)$ jest w systemie $\mathbf{F}$ odpowiednikiem typu indukcyjnego drzew binarnych etykietowanych liczbami naturalnymi:

   ```
   Inductive T : Set := kons : T -> T -> nat -> T | leaf : nat -> T.
   ```

   Na przykład term $T = \Lambda p \lambda f^{p \to p \to \omega \to p} \lambda x^{\omega \to p}.\, x\underline{3}$ to samotny liść oznaczony trójką, a drzewo $\lambda f^{p \to p \to \omega \to p} \lambda x^{\omega \to p}.\, f(f(x\underline{1})(f(x\underline{2})(x\underline{1})\underline{3})\underline{4})(x\underline{1})\underline{5}$ widać na obrazku.

   (a) Jakie operacje na drzewach są reprezentowane wyrażeniami:[1][2]
   – $\lambda t{:}\tau.\, t\omega(\lambda p{:}\omega \lambda q{:}\omega \lambda x{:}\omega.\, p + q + x)\mathbf{I}$?
   – $\lambda t{:}\tau.\, t\tau(\lambda p{:}\tau \lambda q{:}\tau \lambda x{:}\omega.\, p)(\lambda x{:}\omega.\, T)$?

   (b) Napisać termy typu $\tau \to \omega$ reprezentujące takie funkcje:
   – liczba liści drzewa;
   – wysokość drzewa.

   (c) Napisać term typu $\tau \to \tau$ reprezentujący operację wyzerowania wszystkich etykiet.

4. Dany jest automat dwulicznikowy o zbiorze stanów Q, stanie początkowym q_0 i końcowym q_f. Napisać formułę logiki pierwszego rzędu, która jest twierdzeniem intuicjonistycznym wtedy i tylko wtedy, gdy automat akceptuje konfigurację początkową $(q_0, 0, 0)$. Formuła powinna być efektywnie konstruowalna z automatu. Wskazówka: symbole relacyjne mogą reprezentować stany automatu. Można użyć symboli funkcyjnych dla następnika i zera.

[1] Operacje na liczbach naturalnych są takie jak zwykle.

[2] Podpowiedź: pierwszą z funkcji w części 3a można w Coqu zdefiniować tak:

```
Fixpoint F(t:T) : nat := match t with (kons t1 t2 k) => (F t1)+(F t2)+k | (leaf k) => k end.
```

Rozwiązania

1a: Nie. Kontrmodel Kripkego składa się z trzech stanów a, b, i c, gdzie $a \leq b, c$ ale stany b i c są nieporównywalne. Stan a nie wymusza żadnych atomów, natomiast $b \Vdash q, s$ oraz $c \Vdash p, s$. Wtedy formuła $p \to r$ jest wymuszona tylko w stanie b a formuła $q \to r$ jest wymuszona tylko w stanie c. A więc stan a wymusza wszystkie trzy założenia i nie wymusza konkluzji s.

Inny sposób: przyjmijmy $v(p) = (-\infty, 0)$, $v(q) = (0, \infty)$, $v(r) = \varnothing$, $v(s) = \mathbb{R} - \{0\}$ w algebrze otwartych podzbiorów prostej $\mathbb{R}$. Wtedy $[\![p \to r]\!]_v = [\![\neg p]\!]_v = (0, \infty) \subseteq [\![s]\!]_v$ i podobnie $[\![q \to r]\!]_v \subseteq [\![s]\!]_v$. Ponadto $[\![p \to q \to r]\!]_v = [\![p \wedge q \to r]\!]_v = \mathbb{R}$, tymczasem $[\![s]\!]_v \neq \mathbb{R}$.

1b: Tak, bo to jest typ termu: $\lambda x^{p\to q\to r} \lambda y^{(p\to r)\to s} \lambda z^{(q\to r)\to s} \lambda u^{s\to r}.\, u(y(\lambda v^p.\, u(z(\lambda w^q.xvw))))$.

Inny sposób: rozpatrzmy stan c modelu Kripkego, w którym wymuszone są wszystkie założenia, ale nie konkluzja r. Skoro $c \Vdash s \to r$, to $c \nVdash s$. Zatem $c \nVdash p \to r$, jest więc taki stan $d \geq c$, że $d \Vdash p$ oraz $d \nVdash r$, skąd też $d \nVdash s$. Wtedy także $d \nVdash q \to r$, jest więc taki stan $e \geq d$, że $e \Vdash q$ i $e \nVdash r$. Ale $e \Vdash p$ i mamy sprzeczność z powodu pierwszego założenia.

2a: Nie. Rozpatrzmy $\mathcal{O}(\mathbb{R})$-strukturę $\mathcal{A} = \langle \mathbb{R}, P^{\mathcal{A}} \rangle$, gdzie $P^{\mathcal{A}}(r) = \mathbb{R} - \{r\}$ dla dowolnego $r \in \mathbb{R}$. Wtedy $[\![\forall x\, P(x)]\!] = \varnothing$, ale $[\![\forall x\, \neg\neg P(x)]\!] = \mathbb{R}$, bo $\sim(\sim(\mathbb{R} - \{r\})) = \mathbb{R}$ dla każdego $r \in \mathbb{R}$. Zatem znaczeniem formuły $\neg\forall x\, P(x)$ jest cała prosta a znaczeniem formuły $\neg\forall x\, \neg\neg P(x)$ jest zbiór pusty.

Inny sposób: rozpatrzmy model Kripkego o zbiorze stanów $\mathbb{N}$ i stałych dziedzinach $D_n = \mathbb{N}$. Relację P zdefiniujemy w dziedzinie D_n jako zbiór $\{k \mid k \leq n\}$. Nie ma takiego stanu, który wymusza $\forall x\, P(x)$, bo zawsze jest element, który nie należy do relacji. Za to formuła $\forall x\, \neg\neg P(x)$ jest wymuszona w każdym stanie, bo każdy element prędzej czy później do relacji wpadnie.

2b: Tak. To jest typ takiego termu:

$$\lambda X^{\neg\exists x\, P(x)} \lambda Y^{\exists x\, \neg\neg P(x)}.\, \texttt{unpack}\ Y\ \texttt{as}\ [a, V^{\neg\neg P(a)}]\ \texttt{in}\ V(\lambda Z^{P(a)}.\, X[a, Z]^{\exists x\, P(x)}.$$

Inny sposób: rozpatrzmy dowolny model Kripkego i dowolny stan c. Jeśli $c \Vdash \neg\exists x\, P(x)$, to znaczy, że w każdym stanie $d \geq c$ relacja P jest pusta. Tymczasem $c \Vdash \exists x\, \neg\neg P(x)$ oznacza, że dla pewnego elementu a dziedziny D_c zachodzi $c, \mathsf{a} \Vdash \neg\neg P(x)$, w szczególności $c, \mathsf{a} \nVdash \neg P(x)$. Inaczej: element a musi w jakimś przyszłym stanie należeć do relacji P.

3a: Pierwsza operacja to suma wszystkich etykiet drzewa, druga to funkcja stale równa 3. Można to udowodnić przez indukcję, zauważając, że termy zamknięte M typu τ są (z dokładnością do $\beta\eta$-konwersji) generowane przez taką pseudo-gramatykę:

$$M ::= \Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.\, x\underline{k} \ \mid\ \Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.\, f(M_1 p f x)(M_2 p f x)\underline{k}.$$

3b: Term $\lambda t{:}\tau.\, t\omega(\lambda p{:}\omega \lambda q{:}\omega \lambda x{:}\omega.\, p{+}q)(\lambda x{:}\omega.\, \Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.x\underline{1})$ liczy liście, natomiast wysokość dostaniemy z pomocą termu $\lambda t{:}\tau.\, t\omega(\lambda p{:}\omega \lambda q{:}\omega \lambda x{:}\omega.\, 1 + \max\{p, q\})(\lambda x{:}\omega.\, \Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.x\underline{0})$

3c: Zdefiniujmy najpierw funkcje $\mathsf{kons} = \lambda t{:}\tau \lambda s{:}\tau \lambda n{:}\omega.,\Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.\, f(tpfx)(spfx)(xn)$ i $\mathsf{leaf} : \lambda n{:}\omega \Lambda p \lambda f^{p\to p\to\omega\to p} \lambda x^{\omega\to p}.\, xn$. Zerowanie etykiet można teraz napisać tak:

$$\lambda t{:}\tau.\, t\tau(\lambda y{:}\tau \lambda z{:}\tau \lambda u{:}\omega.\, \mathsf{kons}\, y\, z\, \underline{0})(\lambda u : \omega.\, \mathsf{leaf}\underline{0}).$$

4: Formuła ma postać $\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \bot$, a jej przesłanki α_i są następujące:

– $q_0(0,0) \to \bot)$ oraz $\forall xy(q_f(x,y))$";
– $\forall xy(p(s(x), y) \to q(x,y))$, dla instrukcji postaci „$q : c_1 := c_1 + 1;$ `goto` p";
– $\forall xy(p(x, s(y)) \to q(x,y))$, dla instrukcji postaci „$q : c_2 := c_2 + 1;$ `goto` p";
– $\forall xy(p(x,y) \to q(s(x), y))$, dla instrukcji postaci „$q : c_1 := c_1 - 1;$ `goto` p;"
– $\forall xy(p(x,y) \to q(x, s(y)))$, dla instrukcji postaci „$q : c_2 := c_2 - 1;$ `goto` p";
– $\forall xy(p(0,y) \to q(0,y))$, oraz $\forall xy(r(s(x), y) \to q(s(x), y))$,
dla instrukcji postaci „$q :$ `if` $c_1 = 0$ `then goto` p `else goto` r";
– $\forall xy(p(x,0) \to q(y,0))$, oraz $\forall xy(r(x, s(y)) \to q(x, s(y)))$,
dla instrukcji postaci „$q :$ `if` $c_2 = 0$ `then goto` p `else goto` r".