Lista zadań z matematycznych podstaw informatyki nr 5.

Przyjmijmy (jak na wykładzie), że

$$U = \{\langle a, b\rangle \in N^2 : \exists \varphi \in \mathcal{F} \;\; a = \lceil \varphi \rceil \wedge T \vdash \varphi[x \leftarrow \underline{b}]\}$$

gdzie x to ustalona zmienne, φ to formuła, w której najwyżej zmienna x jest wolna, a T – ω-niesprzeczne rozszrzenie Q, oraz

$$U_a = \{b \in N : \langle a, b\rangle \in U\}.$$

Zad. 1. (Raz jeszcze.) Jeżeli T jest ω-niesprzecznym rozszerzeniem teorii Q, to dla dowolnej rekurencyjnie przeliczalnej relacji $R \subseteq N$ jest taka liczba $a \in N$, że $R = U_a$.

Zad. 2. Relacja U jest relacją uniwersalną dla rekurencyjnie przeliczalnych zbiorów liczb naturalnych. Oznacza to, że relacja U jest rekurencyjnie przeliczalna i każdy rekurencyjnie przeliczalny zbiór liczb naturalnych jest postaci U_a dla pewnego $a \in N$.

Zad. 3. Pokaż, że U nie jest relacją rekurencyjną. Wskazówka: oczywiście należy skorzystać z metody przekątniowej i poprzedniego zadania.

Zad. 4. Rozstrzygalność zbioru twierdzeń teorii w pewnym stopniu jest równoważna zupełności teorii.
Rozważmy następujący algorytm: dane jest zdanie φ, pytamy się, czy φ jest twierdzeniem (pewnej ustalonej) teorii T?

1) $d := \varepsilon$; (wartościami zmiennej d są napisy)
2) while true do begin
   (a) $d :=$ następny po d,
   (b) if d jest dowodem φ w T, then return 'φ jest twierdzeniem T',
   (c) if d jest dowodem $\sim \varphi$ w T, then return 'φ nie jest twierdzeniem T'.
3) end

W tym algorytmie $\sim \varphi$ oznacza $\neg\varphi$, jeżeli φ nie jest negacją, oraz ψ, jeżeli $\varphi = \neg\psi$ ($\sim \varphi$ to negacja φ, ale gdyby miała zaczynać się podwójną negacją, to bez tych dwóch negacji). Natomiast procedura badająca dowody nie może o żadnym napisie twierdzić, że jest jednocześnie dowodem zdania i jego negacji (np. zakłada, że udowodniona może zostać tylko ostatnia formuła dowodu)

Pokaż, że

1) Podany algorytm jest poprawny wtedy i tylko wtedy, gdy teoria T jest niesprzeczna.
2) Podany algorytm zatrzymuje się po uruchomieniu z dowolnym zdaniem wtedy i tylko wtedy, gdy teoria T jest zupełna.

Tak więc najprostszy algorytm szukania dowodu rozstrzyga zbiór twierdzeń T wtedy i tylko wtedy, gdy teoria T niesprzeczna i zupełna.

Zad. 5. (Jeszcze raz poprzednie zadanie, bardziej formalnie.) Jeżeli T jest aksjomatyzowalnym, zupełnym i niesprzecznym rozszerzenie teorii Q, to relacja U jest rekurencyjna. Wskazówka: Zdefiniujmy dwa zbiory

$$U_0 = \{\langle \lceil \varphi \rceil, b\rangle \in N^2 : \varphi \in \mathcal{F} \wedge T \vdash \varphi[x \leftarrow \underline{b}]\}$$

oraz

$$U_1 = \{\langle \ulcorner \varphi \urcorner , b \rangle \in N^2 : \varphi \in \mathcal{F} \wedge T \vdash \neg\varphi[x \leftarrow \underline{b}]\}.$$

Są to rozłączne zbiory rekurencyjnie przeliczalne, a ich suma jest zbiorem rekurencyjnym.

Zad. 6. Rachunkiem kwantyfikatorów nazywamy teorię bez aksjomatów pozalogicznych zapisywaną w języku zawierającym przeliczalne zbiory symboli funkcyjnych i relacyjnych wszystkich możliwych arności. Rachunkiem kwantyfikatorów z równością nazywamy teorię, której aksjomatami oprócz aksjomatów logicznych są także aksjomaty równości. Pokaż, że zbiory (numerów) twierdzeń obu rachunków kwantyfikatorów są rekurencyjnie przeliczalne, ale nie są rekurencyjne. Wskazówka: problem, czy dana formuła jest twierdzeniem teorii Q redukuje się do problemu, czy dana formuła jest twierdzeniem rachunku kwantyfikatorów. Wobec tego rekurencyjność rachunku kwantyfikatorów pociąga za sobą rekurencyjność zbioru twierdzeń teorii Q. Aby skonstruować redukcję, korzystamy z twierdzenia o dedukcji i ewentualnie o generalizacji.

Zad. 7. W tym zadaniu T oznacza niesprzeczną arytmetykę zawierającą arytmetykę Q. Niech

$$f_a(x) = y \Leftrightarrow \exists\varphi \ \ulcorner\varphi\urcorner = a \wedge T \vdash (\varphi[u \leftarrow \underline{x}] \Leftrightarrow v = \underline{y}).$$

(dokładniej: istnieje formuła φ z dwoma zmiennymi wolnymi u i v taka, że ..., gdzie $\ulcorner\varphi\urcorner$ oznacza numer formuły φ.) Udowodnij, że $f_0, f_1, \ldots$ jest akceptowalnym systemem programowania, a więc w szczególności w ciągu $f_0, f_1, \ldots$ występują tylko funkcje rekurencyjne, ponadto występują wszystkie takie funkcje, oraz dla niego istnieje całkowita i rekurencyjna funkcja c taka, że $f_a(f_b(x)) = f_{c(a,b)}(x)$. Wbrew pozorom, w akceptowalnych systemach programowania można programować. Warunek z definicji można tak oto zinterpretować: mając programy a i b pozwalające na obliczanie wartości funkcji f_a i f_b potrafimy w sposób efektywny utworzyć program $c(a, b)$ obliczający złożenie tych funkcji, a nawet program ten potrafimy wyliczyć za pomocą komputera.