

1. Operace rot transformuje libovolný jazyk L na jazyk $rot(L) = \{uv \mid vu \in L\}$. Dokažte, že je třída P uzavřená na operaci rot .

- Třída P je uzavřená na operaci $rot \Leftrightarrow \forall L \in P : rot(L) \in P$.
- Z toho, že $L \in P$, plyne, že ho lze přijímat DTS v polynomiálním čase.
- Víme ovšem, že každý DTS přijímající L v polynomiálním čase lze převést na úplný DTS rozhodující L v polynomiálním čase.
- Nechť tedy M je úplný DTS rozhodující L v polynomiálním čase $P(n)$.
- Sestrojíme 2-páskový úplný DTS M_{rot} rozhodující $rot(L)$ v polynomiálním čase:
 - M_{rot} svůj vstup w systematicky postupně rozdělí všemi možnými způsoby na 2 podřetězce u, v takové, že $w = uv$ (způsobů rozdělení je $|w| + 1$). To má časovou složitost $\mathcal{O}(n)$.
 - Pro každé rozdělení zapíše na 2. pásku řetězec vu (to má časovou složitost $\mathcal{O}(n)$) a odsimuluje na něm úplný DTS M rozhodující L (to má časovou složitost $P(n)$).
 - Pokud M pro nějaké rozdělení přijme, M_{rot} také přijme. Pokud M odmítne pro všechna rozdělení, M_{rot} odmítne.
- M_{rot} zřejmě rozhoduje $rot(L)$ a to s časovou složitostí $\mathcal{O}(n) \cdot (P(n) + \mathcal{O}(n)) \in \mathcal{O}(n^k)$ pro $k \in \mathbb{N}$.
- Ukázali jsme, že pro každý $L \in P$ lze sestrojit úplný DTS M_{rot} rozhodující $rot(L)$ v polynomiálním čase, tudíž platí, že $\forall L \in P : rot(L) \in P$, třída P tedy je uzavřená na operaci rot .

2. Teta Květa stojí před regálem se zeleninou a nehýbá se, protože má těžký rozhodovací problém. Potřebuje sníst co nejvíce vitamínu C, aby ji přešla chřipka. Každý druh zeleniny je charakteristický obsahem vitamínu C na kilo a cenou za kilo. Teta se snaží přijít na to, jestli je možné nakoupit zeleninu za obnos O v její peněženke tak, aby úhrn vitamínu C byl alespoň C . Kromě toho s každým kilem zeleniny přidá zelinář deset deka brokolice zdarma, což obsahuje množství B vitamínu C. Dokažte, že je Květin problém NP-úplný. Těžkost dokažte redukcí z některého problému uvedeného v odstavci „NP-complete problems“ zde:

http://en.wikipedia.org/wiki/NP-complete_problems#NP-complete_problems.

Květin problém je charakterizován jazykem:

$L_K = \{ \{ (c_1, p_1), \dots, (c_n, p_n) \}, C, O \mid \{ (c_1, p_1), \dots, (c_n, p_n) \} \text{ je } n\text{-prvková množina dvojic, pro nějaké } n \geq 0, \text{ kde každá dvojice reprezentuje jeden z druhů zeleniny, } c_i \text{ je součet množství vitamínu C na kilo v zelenině } i \text{ s množstvím vitamínu C v brokolici přidávané zdarma ke každému zakoupenému kilu zeleniny, } p_i \text{ je cena za kilo zeleniny } i, C \text{ je minimální požadované množství vitamínu C, } O \text{ je obnos v Květině peněženke, přitom lze vytvořit } k\text{-prvkovou podmnožinu množiny } \{ (c_1, p_1), \dots, (c_n, p_n) \} \{ (c_1, p_1), \dots, (c_k, p_k) \} \text{ pro nějaké } k, \text{ pro které platí } 0 \leq k \leq n, \text{ tak, že existuje } k\text{-prvková posloupnost přirozených čísel } x_1, \dots, x_k \text{ pro kterou platí: } x_1 \cdot c_1 + \dots + x_k \cdot c_k \geq C \wedge x_1 \cdot p_1 + \dots + x_k \cdot p_k \leq O, \text{ kde } x_i \text{ je počet zakoupených kilogramů zeleniny } i \text{ (předpokládáme, že maximální zakoupitelné množství každého druhu zeleniny není nijak omezeno)} \}$

Množina $\{ (c_1, p_1), \dots, (c_n, p_n) \}$ může být zakódována pomocí $\langle \rangle$ jako posloupnost vhodně oddělených dvojic vhodně oddělených binárně zakódovaných přirozených čísel, C i O lze zakódovat jako vhodně oddělená binárně zakódovaná přirozená čísla. Např. instanci Květin problému $(\{ (0, 1), (2, 3) \}, 2, 3)$ můžeme zakódovat takto: $(00, 01) | (10, 11) | 10 | 11$

I/ Květin problém je v NP

Pro dokázání tohoto tvrzení popíšeme fungování nedeterministického Turingova stroje přijímajícího jazyk L_K v polynomiálním čase. Při rozhodování, zda vstupní řetězec $w \in L_K$ můžeme postupovat takto:

1. Ověříme, zda se jedná o platný kód $\langle \{ (c_1, p_1), \dots, (c_n, p_n) \} C, O \rangle$.
 - Kontrola správného užití oddělovačů (složitost $\mathcal{O}(n)$).
 - Kontrola, zda vstup obsahuje alespoň C a O (regál se zeleninou může klidně být i prázdný) (složitost $\mathcal{O}(n)$).
2. Zjistíme počet prvků n množiny $\{ (c_1, p_1), \dots, (c_n, p_n) \}$. Za tímto účelem procházíme dvojice (c_i, p_i) na vstupu (složitost $\mathcal{O}(n)$) a na pomocné pásce inkrementujeme hodnotu n (složitost $\mathcal{O}(n)$).
3. Pokud $n = 0$ (složitost $\mathcal{O}(n)$): přijmeme, pokud $C = 0$ (složitost $\mathcal{O}(n)$), jinak zamítneme.
4. Jinak náhodně zvolíme nějaké k takové, že $0 < k \leq n$ a označíme k náhodně zvolených dvojic (c_i, p_i) dodáním vhodné značky ke kódu dvojice. Za tímto účelem zapíšeme k na pomocnou pásku (složitost $\mathcal{O}(n)$), následně projdeme vstupní pásku a označíme k náhodně vybraných dvojic (složitost $\mathcal{O}(n)$) při současné dekrementaci k na pomocné pásce (složitost $\mathcal{O}(n)$).
5. Pro každou označenou dvojici náhodně zvolíme celočíselné $x > 0$ a následně zkopírujeme danou dvojici x -krát na 1. pomocnou pásku. Za tímto účelem pro každou označenou dvojici (složitost $\mathcal{O}(n)$) zapíšeme x na 2. pomocnou pásku (složitost $\mathcal{O}(n)$), potom x -krát zkopírujeme danou dvojici ze vstupní pásky na 1. pomocnou pásku (složitost $\mathcal{O}(n)$), přičemž po každém zkopírování dekrementujeme x na 2. pomocné pásce (složitost $\mathcal{O}(n)$).
6. Na 2. a 3. pomocnou pásku zapíšeme 0 (konstantní složitost). Procházíme jednotlivé dvojice zapsané na 1. pomocné pásce (složitost $\mathcal{O}(n)$) a pro každou z nich přičteme k číslu na 2. pomocné

pásce její hodnotu p (složitost $\mathcal{O}(n)$) a k číslu na 3. pomocné pásce její hodnotu c (složitost $\mathcal{O}(n)$).

7. Pokud je číslo na 2. pomocné pásce větší než O (složitost $\mathcal{O}(n)$), odmítneme. Jinak pokud je číslo na 3. pomocné pásce větší nebo rovno C (složitost $\mathcal{O}(n)$), přijmeme, jinak odmítneme.

Celý tento postup má evidentně exponenciální složitost, Květin problém je tedy v NP.

II/ Květin problém je NP-těžký

Použijeme polynomiální redukci z NP-úplného problému Unbounded Knapsack

(http://en.wikipedia.org/wiki/Knapsack_problem),

resp. jeho verze definované zde:

<http://cgm.cs.mcgill.ca/~avis/courses/360/2003/assignments/sol4.pdf>

Unbounded Knapsack problém:

Je dána množina druhů předmětů, každý druh je charakterizován svou vahou a hodnotou. Ptáme se, kolik předmětů od každého druhu vybrat tak, aby celková váha výběru byla menší nebo rovna nějaké konstantě a přitom celková hodnota výběru byla větší nebo rovna nějaké jiné konstantě.

Jazyk specifikující problém Unbounded Knapsack je tedy:

$L_U = \{ \{ (v_1, w_1), \dots, (v_n, w_n) \}, V, W \mid \{ (v_1, w_1), \dots, (v_n, w_n) \} \text{ je } n\text{-prvková množina dvojic, pro nějaké } n \geq 0, \text{ kde každá dvojice reprezentuje jeden z druhů předmětů, } v_i \text{ je hodnota druhu předmětu } i, w_i \text{ je váha druhu předmětu } i, V \text{ je minimální požadovaná hodnota, } W \text{ je maximální váha, přitom lze vytvořit } k\text{-prvkovou podmnožinu množiny } \{ (v_1, w_1), \dots, (v_n, w_n) \} \{ (v_1, w_1), \dots, (v_k, w_k) \} \text{ pro nějaké } k, \text{ pro které platí } 0 \leq k \leq n, \text{ tak, že existuje } k\text{-prvková posloupnost přirozených čísel } x_1, \dots, x_k \text{ pro kterou platí: } x_1 \cdot v_1 + \dots + x_k \cdot v_k \geq V \wedge x_1 \cdot w_1 + \dots + x_k \cdot w_k \leq W, \text{ kde } x_i \text{ je počet vybraných předmětů druhu } i \}$

Ke každé instanci Unbounded Knapsack můžeme sestavit instanci Květin problému takto:

$$\begin{aligned} \{ (c_1, p_1), \dots, (c_n, p_n) \} &= \{ (v_1, w_1), \dots, (v_n, w_n) \} \\ C &= V \\ O &= W \end{aligned}$$

Uvedenou redukci lze implementovat úplným DTS M_δ pracujícím v polynomiálním čase. Stačí, když M_δ zkontroluje, zda má na vstupu korektně zformovanou instanci Unbounded Knapsack problému (postup bude zcela totožný jako výše uvedený postup kontroly pro Květin problém). Pokud ne, vypíše nějakou instanci Květin problému, která nepatří do L_K , pokud ano, vypíše svůj nezměněný vstup, protože instance Unbounded Knapsack problému mají úplně stejný formát jako instance Květin problému.

Zřejmě platí:

M_δ má na vstupu instanci patřící do jazyka $L_U \Leftrightarrow$ výstup M_δ je instance patřící do jazyka L_K .

Je tedy dokázáno, že Květin problém je NP-úplný.

Z dálky na tetu volá synovec Alan, ať nezoufá, že to za chvíli (rozuměj, v polynomiálním čase) vyřeší s pomocí jakéhosi psacího stroje vlastní výroby s nekonečnou páskou. Co to znamená pro lidstvo?

Znamená to, že $P = NP$.

3. Dokažte, že

- (a) $\mathcal{O}(2^n) \subset \mathcal{O}(2^{2^n})$
- (b) $\mathcal{O}(n^3) = \mathcal{O}(n^3 - 10n + 5)$
- (c) $\mathcal{O}(\log_2(5n * (7n)^{2^n})) = \mathcal{O}(n * \log_3(10n))$

V bodech (a), (b) bude třeba dokázat dva směry, \subseteq a \supseteq , \subseteq a \supseteq . Pro (c) si zopakujte pravidla pro počítání s logaritmy.

- (a) $\mathcal{O}(2^n) \subset \mathcal{O}(2^{2^n})$

Z definice $\mathcal{O}(f(n))$:

$$\mathcal{O}(2^n) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * 2^n\}$$

$$\mathcal{O}(2^{2^n}) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * 2^{2^n}\}$$

Vlastní podmnožina:

$$\mathcal{O}(2^n) \subset \mathcal{O}(2^{2^n}) \Leftrightarrow \mathcal{O}(2^n) \subseteq \mathcal{O}(2^{2^n}) \wedge \mathcal{O}(2^n) \not\supseteq \mathcal{O}(2^{2^n})$$

Nejprve dokážeme, že $\mathcal{O}(2^n) \subseteq \mathcal{O}(2^{2^n})$:

$$\mathcal{O}(2^n) \subseteq \mathcal{O}(2^{2^n})$$

\Leftrightarrow

$$\forall f \in \mathcal{F} : (f \in \mathcal{O}(2^n) \Rightarrow f \in \mathcal{O}(2^{2^n}))$$

\Leftrightarrow

$$\forall f \in \mathcal{F} :$$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq f(n) \leq c_1 * 2^n)$$

\Rightarrow

$$(\exists c_2 \in \mathbb{R}^+ \exists n_{02} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{02} \Rightarrow 0 \leq f(n) \leq c_2 * 2^{2^n})$$

Zvolme $c_1 = c_2 = 1, n_{01} = n_{02} = 1$:

$$\forall f \in \mathcal{F} :$$

$$(\forall n \in \mathbb{N} : n \geq 1 \Rightarrow 0 \leq f(n) \leq 1 * 2^n)$$

\Rightarrow

$$(\forall n \in \mathbb{N} : n \geq 1 \Rightarrow 0 \leq f(n) \leq 1 * 2^{2^n})$$

Je zřejmé, že pravá strana implikace musí platit vždy, když platí strana levá, protože platí, že $\forall n \in \mathbb{N} \Rightarrow 1 * 2^{2^n} > 1 * 2^n$. Implikace tudíž musí vždy platit, takže platí, že $\mathcal{O}(2^n) \subseteq \mathcal{O}(2^{2^n})$.

Ještě dokážeme, že $\mathcal{O}(2^n) \not\supseteq \mathcal{O}(2^{2^n})$:

$$\mathcal{O}(2^n) \not\supseteq \mathcal{O}(2^{2^n})$$

\Leftrightarrow

$$\exists g \in \mathcal{F} : g \in \mathcal{O}(2^{2^n}) \wedge g \notin \mathcal{O}(2^n)$$

\Leftrightarrow

$$\exists g \in \mathcal{F} :$$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq g(n) \leq c_1 * 2^{2^n})$$

\wedge

$$(\forall c_2 \in \mathbb{R}^+ \forall n_{02} \in \mathbb{N} \exists n \in \mathbb{N} : n \geq n_{02} \Rightarrow g(n) < 0 \vee g(n) > c_2 * 2^n)$$

Zvolme $g(n) = 2^{2^n}$. Tato funkce splňuje levou část konjunkce např. pro $c_1 = 1, n_{01} = 0$. Zároveň pro ni platí, že pro jakékoli c_2 lze najít takové $x \in \mathbb{N}$, že pro jakékoliv $n > x$ je $g(n) > c_2 * 2^n$. Tím pádem je splněna i pravá strana konjunkce, takže platí, že $\mathcal{O}(2^n) \not\supseteq \mathcal{O}(2^{2^n})$.

Dokázali jsme oba směry, takže musí platit, že $\mathcal{O}(2^n) \subset \mathcal{O}(2^{2^n})$.

(b) $\mathcal{O}(n^3) = \mathcal{O}(n^3 - 10n + 5)$

Z definice $\mathcal{O}(f(n))$:

$$\mathcal{O}(n^3) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * n^3\}$$

$$\mathcal{O}(n^3 - 10n + 5) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * (n^3 - 10n + 5)\}$$

Rovnost množin:

$$\mathcal{O}(n^3) = \mathcal{O}(n^3 - 10n + 5) \Leftrightarrow \mathcal{O}(n^3) \subseteq \mathcal{O}(n^3 - 10n + 5) \wedge \mathcal{O}(n^3) \supseteq \mathcal{O}(n^3 - 10n + 5)$$

Nejprve dokážeme, že $\mathcal{O}(n^3) \subseteq \mathcal{O}(n^3 - 10n + 5)$:

$$\mathcal{O}(n^3) \subseteq \mathcal{O}(n^3 - 10n + 5)$$

\Leftrightarrow

$$\forall f \in \mathcal{F} : (f \in \mathcal{O}(n^3) \Rightarrow f \in \mathcal{O}(n^3 - 10n + 5))$$

\Leftrightarrow

$$\forall f \in \mathcal{F} :$$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq f(n) \leq c_1 * n^3)$$

\Rightarrow

$$(\exists c_2 \in \mathbb{R}^+ \exists n_{02} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{02} \Rightarrow 0 \leq f(n) \leq c_2 * (n^3 - 10n + 5))$$

Zvolme $c_1 = 1, c_2 = 2, n_{01} = n_{02} = 5$:

$$\forall f \in \mathcal{F} :$$

$$(\forall n \in \mathbb{N} : n \geq 5 \Rightarrow 0 \leq f(n) \leq 1 * n^3)$$

\Rightarrow

$$(\forall n \in \mathbb{N} : n \geq 5 \Rightarrow 0 \leq f(n) \leq 2 * (n^3 - 10n + 5))$$

Je zřejmé, že pravá strana implikace musí platit vždy, když platí strana levá, protože platí, že $\forall n \in \mathbb{N} : n \geq 5 \Rightarrow 2 * (n^3 - 10n + 5) > 1 * n^3$. Implikace tudíž musí vždy platit, takže platí, že $\mathcal{O}(n^3) \subseteq \mathcal{O}(n^3 - 10n + 5)$.

Ještě dokážeme, že $\mathcal{O}(n^3) \supseteq \mathcal{O}(n^3 - 10n + 5)$:

$$\mathcal{O}(n^3) \supseteq \mathcal{O}(n^3 - 10n + 5)$$

\Leftrightarrow

$$\forall f \in \mathcal{F} : (f \in \mathcal{O}(n^3 - 10n + 5) \Rightarrow f \in \mathcal{O}(n^3))$$

\Leftrightarrow

$$\forall f \in \mathcal{F} :$$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq f(n) \leq c_1 * (n^3 - 10n + 5))$$

\Rightarrow

$$(\exists c_2 \in \mathbb{R}^+ \exists n_{02} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{02} \Rightarrow 0 \leq f(n) \leq c_2 * n^3)$$

Zvolme $c_1 = c_2 = 1, n_{01} = n_{02} = 1$:

$$\forall f \in \mathcal{F} :$$

$$(\forall n \in \mathbb{N} : n \geq 1 \Rightarrow 0 \leq f(n) \leq 1 * (n^3 - 10n + 5))$$

\Rightarrow

$$(\forall n \in \mathbb{N} : n \geq 1 \Rightarrow 0 \leq f(n) \leq 1 * n^3)$$

Je zřejmé, že pravá strana implikace musí platit vždy, když platí strana levá, protože platí, že

$\forall n \in \mathbb{N} : n \geq 1 \Rightarrow 1 * n^3 > 2 * (n^3 - 10n + 5)$. Implikace tudíž musí vždy platit, takže platí, že $\mathcal{O}(n^3) \supseteq \mathcal{O}(n^3 - 10n + 5)$.

Dokázali jsme oba směry, takže musí platit, že $\mathcal{O}(n^3) = \mathcal{O}(n^3 - 10n + 5)$.

$$(c) \mathcal{O}(\log_2(5n * (7n)^{2n})) = \mathcal{O}(n * \log_3(10n))$$

Funkce lze převést na tyto tvary:

$$\log_2(5n * (7n)^{2n}) = \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}$$

$$n * \log_3(10n) = \frac{n * \ln(10n)}{\ln(3)}$$

Z definice $\mathcal{O}(f(n))$:

$$\mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\}$$

$$\mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c * \frac{n * \ln(10n)}{\ln(3)}\}$$

Rovnost množin:

$$(\mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right) = \mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right)) \Leftrightarrow \forall f \in \mathcal{F} :$$

$$(f \in \mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right)) \Rightarrow f \in \mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right) \wedge (f \in \mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right)) \Rightarrow f \in \mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right))$$

Nejprve dokážeme implikaci na levé straně konjunkce:

$$\forall f \in \mathcal{F} : (f \in \mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right)) \Rightarrow f \in \mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right))$$

\Leftrightarrow

$\forall f \in \mathcal{F} :$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq f(n) \leq c_1 * \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)})$$

\Rightarrow

$$(\exists c_2 \in \mathbb{R}^+ \exists n_{02} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{02} \Rightarrow 0 \leq f(n) \leq c_2 * \frac{n * \ln(10n)}{\ln(3)})$$

$$\text{Zvolme } c_2 = c_1 * \frac{\ln(3) * (\ln(5n) + 2n * \ln(7n))}{\ln(10n) * \ln(2) * n}, n_{02} = n_{01}.$$

Jelikož $c_1 * \frac{\ln(3) * (\ln(5n) + 2n * \ln(7n))}{\ln(10n) * \ln(2) * n} * \frac{n * \ln(10n)}{\ln(3)} = c_1 * \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}$, je zřejmé, že pravá strana implikace musí platit vždy, když platí část levá. Implikace tedy musí být vždy pravdivá.

Ještě dokážeme implikaci na pravé straně konjunkce:

$$\forall f \in \mathcal{F} : (f \in \mathcal{O}\left(\frac{n * \ln(10n)}{\ln(3)}\right)) \Rightarrow f \in \mathcal{O}\left(\frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)}\right))$$

\Leftrightarrow

$\forall f \in \mathcal{F} :$

$$(\exists c_1 \in \mathbb{R}^+ \exists n_{01} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{01} \Rightarrow 0 \leq f(n) \leq c_1 * \frac{n * \ln(10n)}{\ln(3)})$$

\Rightarrow

$$(\exists c_2 \in \mathbb{R}^+ \exists n_{02} \in \mathbb{N} \forall n \in \mathbb{N} : n \geq n_{02} \Rightarrow 0 \leq f(n) \leq c_2 * \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)})$$

$$\text{Zvolme } c_2 = c_1 * \frac{\ln(10n) * \ln(2) * n}{\ln(3) * (\ln(5n) + 2n * \ln(7n))}, n_{02} = n_{01}.$$

Jelikož $c_1 * \frac{\ln(10n) * \ln(2) * n}{\ln(3) * (\ln(5n) + 2n * \ln(7n))} * \frac{\ln(5n) + 2n * \ln(7n)}{\ln(2)} = \frac{n * \ln(10n)}{\ln(3)}$, je zřejmé, že pravá strana implikace musí platit vždy, když platí část levá. Implikace tedy musí být vždy pravdivá.

Dokázali jsme obě strany konjunkce, takže skutečně platí, že $\mathcal{O}(\log_2(5n * (7n)^{2n})) = \mathcal{O}(n * \log_3(10n))$.

4. Pomocí počátečních funkcí a operátorů kombinace, kompozice a primitivní rekurze vyjádřete funkci:

$$\text{tripleplus} : \mathbb{N}^3 \rightarrow \mathbb{N}, \text{tripleplus}(x, y, z) = x + y + z.$$

Nepoužívejte žádné funkce zavedené na přednáškách mimo funkce počáteční, nepoužívejte zjednodušenou syntaxi zápisu funkcí - dodržte přesně definiční tvar operátorů kombinace, kompozice a primitivní rekurze.

$$\begin{aligned} h(x, y, z) &= \sigma \circ \pi_3^3(x, y, z) \\ plus(x, 0) &= \pi_1^1(x, 0) \\ plus(x, y + 1) &= h(x, y, plus(x, y)) \\ g(x, y, z, a) &= \sigma \circ \pi_4^4(x, y, z, a) \\ \text{tripleplus}(x, y, 0) &= plus \circ (\pi_1^3 \times \pi_2^3)(x, y, 0) \\ \text{tripleplus}(x, y, z + 1) &= g(x, y, z, \text{tripleplus}(x, y, z)) \end{aligned}$$