Úkol 3

1. příklad

Pomocí počátečních funkcí a operátorů kombinace, kompozice a primitivní rekurze vyjádřete funkci počítající odmocninu (zaokrouhlenou dolů na celá čísla):

$$sqrt : \mathbb{N} \to \mathbb{N}, sqrt(x) = z \text{ takové, že } z^2 \leq x \wedge (z+1)^2 > x.$$

Je možné použít funkce $plus(x, y)$, $mult(x, y)$, $monus(x, y)$ a $eq(x, y)$ definované v přednáškách. Kromě nich však nepoužívejte žádné další funkce zavedené na přednáškách mimo funkce počáteční. Nepoužívejte zjednodušenou syntaxi zápisu funkcí – dodržte přesně definiční tvar operátorů kombinace, kompozice a primitivní rekurze.

Řešení:

$rsqrt(x, 0) = \xi()$
$rsqrt(x, y+1) = plus \circ (\pi_3^3(x, y, rsqrt(x, y)) \times (eq \circ ((monus \circ ((mult \circ (\pi_1^1(y) \times \pi_1^1(y))) \times \pi_1^1(x))) \times \xi())))$
$\boldsymbol{sqrt} \equiv rsqrt \circ (\pi_1^1 \times \pi_1^1)$

2. příklad

Mějme následující funkce:

$f(n) = \sqrt{2}n^3$
$g(n) = 10\,000n^2 + 500n + 211.$

Dokažte, že $O(g(n)) \subset O(f(n))$.
Pozn.: Nezapomeňte, že důkaz má dvě části: (i) $O(g(n)) \subseteq O(f(n))$ a (ii) $O(g(n)) \neq O(f(n))$.

Řešení:

- S využitím asymptotických odhadů složitosti můžeme říci, že $O(f(n)) \approx O(n^3)$ a $O(g(n)) \approx O(n^2)$.
- Potom tedy budeme dokazovat ekvivalentní vztah $\boldsymbol{O(g(n)) \subset O(f(n)) \Leftrightarrow O(n^2) \subset O(n^3) \Leftrightarrow (O(n^2) \subseteq O(n^3) \wedge O(n^2) \neq O(n^3))}$.

Důkaz:

(i) Ukážeme, že $\boldsymbol{O(n^2) \subseteq O(n^3)}$.

- Uvažme libovolné $f(n) \in O(n^2)$.
- $\exists c \in \mathbb{R}^+ \; \exists n_0 \in \mathbb{N} \; \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq f(n) \leq cn^2$.
- Zkoumejme nyní rychlost růstu n^3 a cn^2:

$$\lim_{n\to\infty} \frac{n^3}{cn^2} = \left| \begin{array}{c} \lim\limits_{n\to\infty} n^3 = \infty \\ \lim\limits_{n\to\infty} cn^2 = c \cdot \infty \\ \text{lze užít L'Hospitalovo} \\ \text{pravidlo} \end{array} \right| = \lim_{n\to\infty} \frac{3n^2}{2cn} = \left| \begin{array}{c} \lim\limits_{n\to\infty} 3n^2 = \infty \\ \lim\limits_{n\to\infty} 2cn = c \cdot \infty \\ \text{lze užít L'Hospitalovo} \\ \text{pravidlo} \end{array} \right| = \lim_{n\to\infty} \frac{6n}{2c} = \frac{\infty}{\text{sgn(c)}}$$

- n^3 tedy roste rychleji než cn^2 a musí tedy existovat $n_0' \in \mathbb{N}$ takové, že $\forall n \in \mathbb{N} : n \geq n_0' \Rightarrow 0 \leq f(n) \leq cn^2 \leq n^3$. Tedy $\forall n \in \mathbb{N} : n \geq n_0' \Rightarrow f(n) \in O(n^2)$. Potom tedy **platí $\boldsymbol{O(n^2) \subseteq O(n^3)}$**.

(ii) Ukažme, že $\boldsymbol{O(n^2) \neq O(n^3)}$.

- Uvažme libovolné $f(n) \in O(n^3)$.
- $\exists c \in \mathbb{R}^+ \, \exists n_0 \in \mathbb{N} \, \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq f(n) \leq cn^3$.
- Zkoumejme nyní rychlost růstu n^2 a cn^3:

$$\lim_{n\to\infty} \frac{n^2}{cn^3} = \left| \begin{array}{c} \lim\limits_{n\to\infty} n^2 = \infty \\ \lim\limits_{n\to\infty} cn^3 = c \cdot \infty \\ \text{lze užít L'Hospitalovo} \\ \text{pravidlo} \end{array} \right| = \lim_{n\to\infty} \frac{2n}{3cn^2} = \left| \begin{array}{c} \lim\limits_{n\to\infty} 2n = \infty \\ \lim\limits_{n\to\infty} 3cn^2 = c \cdot \infty \\ \text{lze užít L'Hospitalovo} \\ \text{pravidlo} \end{array} \right| = \lim_{n\to\infty} \frac{2}{6cn} = 0$$

- n^2 tedy roste pomaleji než cn^3 a neexistuje tedy $n_0' \in \mathbb{N}$ takové, že $\forall n \in \mathbb{N} : n \geq n_0' \Rightarrow 0 \leq f(n) \leq cn^3 \leq n^2$. Proto **neplatí $\boldsymbol{O(n^3) \subseteq O(n^2)}$**. Z toho plyne, že $\boldsymbol{O(n^2) \neq O(n^3)}$.

$\square$

3. příklad

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 peněžence tak, aby úhrn vitamínu C byl alespoň C. Kromě toho s každým kilem zeleniny přidá zelinář deset deka brokolice zdarma, s obsahem B vitamínu C na kilo.

Formulujte problém tety Květy jako rozhodovací problém, a dokažte, že je NP-úplný. Těžkost dokažte redukcí z některého problému uvedeného v odstavci „NP-complete problems“ zde:
https://en.wikipedia.org/wiki/NP-completeness#NP-complete_problems

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

Řešení:

Formulace problému:

- Problém tety Květy je možné charakterizovat jazykem $K = \{z \# o \# c \mid o \in \mathbb{N}$ je obnos tety Květy v její peněžence, za který nakupuje zeleninu; $c \in \mathbb{N}$ je požadované minimální množství vitamínu C; z je řetězec, který reprezentuje sekvenci dvojic $\mathbb{N} \times \mathbb{N}$, kde prvky dvojic jsou odděleny např. znakem @ a jednotlivé dvojice jsou odděleny např. znakem \$, každá dvojice reprezentuje kus zeleniny v regálu, kde první složka dvojice představuje obsah vitamínu C na kilo a druhá složka představuje cenu na kilo daného druhu zeleniny, jednotlivé dvojice jsou číslovány např od 0$\}$. Znak # je použit jako oddělovač.

- Aby byl v této reprezentaci uvážen fakt, že s každým kilem zeleniny dostane teta Květa deset deka brokolice zdarma s obsahem $B \in \mathbb{N}$ vitamínu C na kilo, je k první složce každé dvojice řetězce z přičtena hodnota $\frac{B}{10}$ (tato hodnota je vhodně zaokrouhlena nebo jsou použita reálná čísla s vhodným kódováním).

- Řešením tohoto problému je sekvence indexů kusů zeleniny z řetězce z taková, že suma cen vybraných kusů je menší nebo rovna dostupnému obnosu o a zároveň je suma množství vitamínu C vybraných kusů větší nebo rovna požadovanému minimálnímu množství c.

Důkaz NP-úplnosti:

1. **Důkaz členství ve třídě NP:**
   - Ukažme, že daný problém lze řešit nedeterministickým Turingovým strojem M pracujícím v polynomiálním čase.
   - Zkonstruujeme M, který pro $w \in K$ přijme v polynomiálním čase.
     - M ověří, zda jeho vstup je platná instance daného problému, tj. řetězec jazyka K. Pokud ne, odmítne. Toto lze ověřit v čase $O(n)$.
     - M nedeterministicky uhádne k indexů kusů zeleniny na pomocnou pásku. Složitost tohoto kroku je $O(n)$.
     - M ověří, zda vybrané index tvoří řešení daného problému, pokud ano, přijme, pokud ne, odmítne. Při tomto ověřování se budou sumy cen a množství vitamínu C zapisovat na další pomocnou pásku. Složitost bude $O(n^3)$.
   - M tedy pracuje v polynomiálním čase.
2. **Důkaz NP-těžkosti:**
   - Důkaz provedeme **polynomiální redukcí z problému batohu (knapsack problem)**, který je NP-úplný.
   - Problém batohu je definován následovně:
     - Je dána konečná množina R, váhová funkce $u : R \longrightarrow \mathbb{N}$ a hodnotová funkce $v : R \longrightarrow \mathbb{N}$.
     - Je dána mezní váha $U \in \mathbb{N}$ a mezní hodnota $V \in \mathbb{N}$.
     - Problém batohu se ptá, zda $\exists R' \subseteq R$ takové, že $\sum_{r \in R'} u(r) \leq U \wedge \sum_{r \in R'} v(r) \geq V$.
   - Tuto redukci lze realizovat úplným deterministickým Turingovým strojem, který pracuje v polynomiálním čas.
   - Každou instanci problému batohu je možné převést na problém tety Květy následovně:
     - Pro každý prvek množiny R vytvoříme příslušnou dvojici v řetězci z, kde první složka této dvojice bude dána hodnotovou funkcí v ($+\frac{B}{10}$ za brokolici zdarma) a druhá složka této dvojice bude dána váhovou funkcí u.
     - Mezní váha U bude zapsána do řetězce o. Mezní hodnota V bude zapsána do řetězce c.
     - Tímto vznikne řetězec jazyka K, tedy instance problému tety Květy.

$\square$

Co znamená pro lidstvo, že synovec Alan má stroj, který tento problém řeší v polynomiálním čase?

- Znamenalo by to, že existuje algoritmus, který tento problém řeší v polynomiálním čase. To by znamenalo, že tento problém patří do třídy P a potom by muselo platit, že **P = NP**.
- Znamenalo by to, že každý problém, u kterého dokáže počítač „rychle" ověřit správnost řešení, dokáže počítač taky dané řešení „rychle" nalézt.

4. příklad

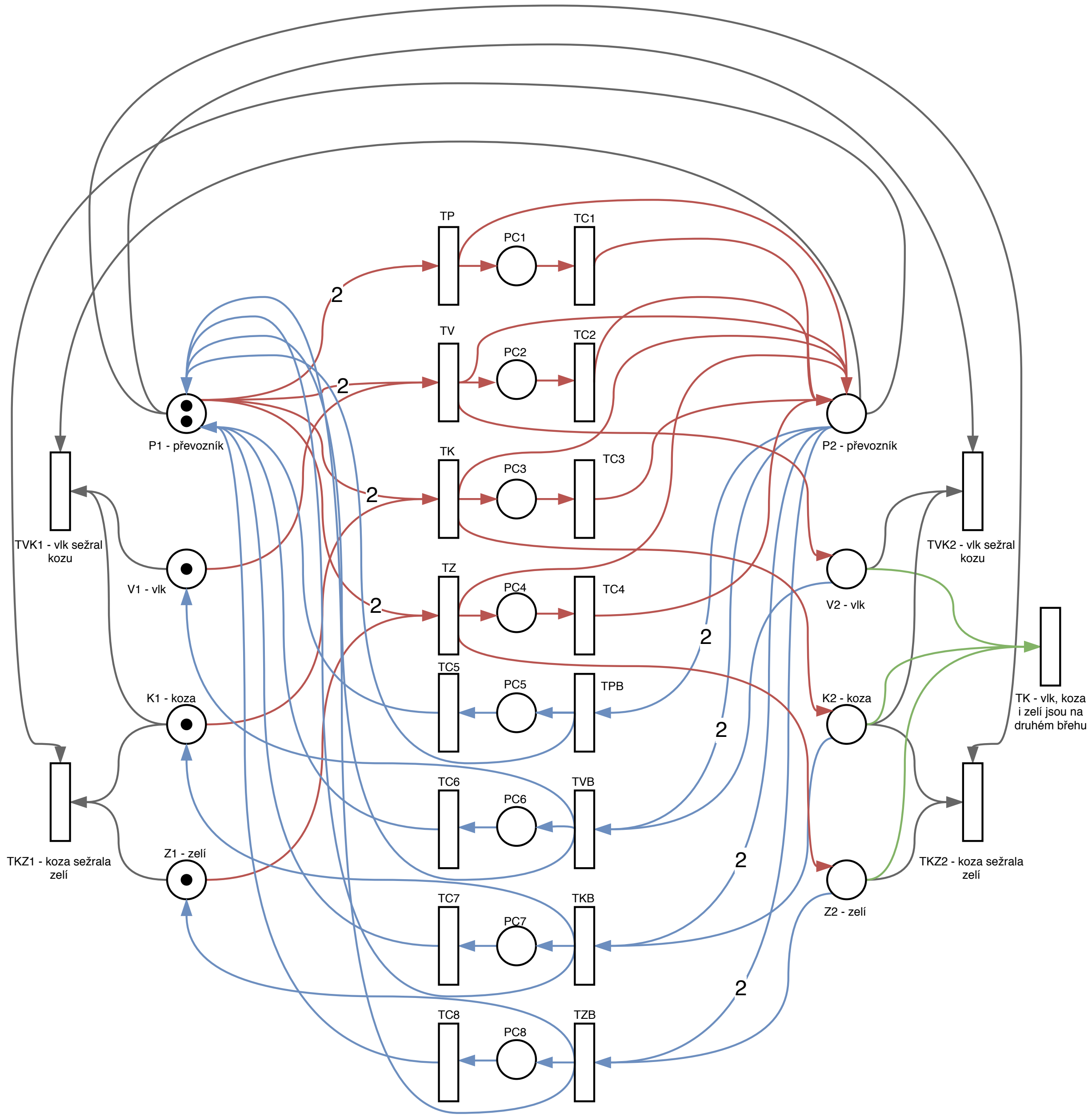
Modelujte následující kritický systém Petriho sítí. Namalujte ji a zapište formálně ve shodě s definicí.

Převozník chce převézt z jednoho břehu na druhý hlávku zelí, kozu a vlka. Do loďky s sebou může vzít buď zelí, nebo kozu, nebo vlka, ale víc se tam nevejde. Nechá-li na břehu hlávku zelí a kozu, koza zelí sežere. Nechá-li na břehu kozu a vlka, pak vlk sežere kozu. Jakým způsobem musí převozník postupovat, aby nedošlo k žádné škodě?

Snažte se o přehlednost a pochopitelnost modelu. Místa vhodně pojmenujte a síť nakreslete přehledně. (příklad ze sbírky úloh Alkuina z Yorku, Úlohy k bystření mladíků, z roku cca 735-804)

Řešení:

Grafická reprezentace Petriho sítě:



Formální zápis Petriho sítě:
$N = (P, T, F, W, K, M_0)$, kde

- $P = \{P1, V1, K1, Z1, P2, K2, Z2, PC1, PC2, PC3, PC4, PC5, PC6, PC7, PC8\}$
- $T = \{TVK1, TKZ1, TVK2, TKZ2, TK, TC1, TC2, TC3, TC4, TC5, TC6, TC7, TC8, TP, TV, TK, TZ, TPB, TVB, TKB, TZB\}$

- $F = \{(P1, TP), (P1, TV), (P1, TK), (P1, TZ), (P1, TVK2), (P1, TKZ2), (P2, TPB), (P2, TVB), (P2, TKB), (P2, TZB), (P2, TVK1), (P1, TKZ1), (V1, TV), (V1, TVK1), (V2, TVB), (V2, TVK2), (K1, TK), (K1, TVK1), (K1, TKZ1), (K2, TKB), (K2, TVK2), (K2, TKZ2), (Z1, TZ), (Z1, TKZ1), (Z2, TZB), (Z2, TKZ2), (PC1, TC1), (PC2, TC2), (PC3, TC3), (PC4, TC4), (PC5, TC5), (PC6, TC6), (PC7, TC7), (PC8, TC8), (TC1, P2), (TC2, P2), (TC3, P2), (TC4, P2), (TC5, P1), (TC6, P1), (TC7, P1), (TC8, P1), (V2, TK), (K2, TK), (Z2, TK), (TP, PC1), (TP, P2), (TV, PC2), (TV, P2), (TV, V2), (TK, PC3), (TK, P2), (TK, K2), (TZ, PC4), (TZ, P2), (TZ, Z2), (TVB, PC6), (TVB, P1), (TVB, V1), (TPB, PC5), (TPB, P1), (TKB, PC7), (TKB, P1), (TKB, K1), (TZB, PC8), (TZB, P1), (TZB, Z1)\}$
- $W = \begin{cases} 2 & \text{pro } (P1, TP) \\ 2 & \text{pro } (P1, TV) \\ 2 & \text{pro } (P1, TK) \\ 2 & \text{pro } (P1, TZ) \\ 2 & \text{pro } (P2, TPB) \\ 2 & \text{pro } (P2, TVB) \\ 2 & \text{pro } (P2, TKB) \\ 2 & \text{pro } (P2, TZB) \\ 1 & \text{jinak} \end{cases}$
- $K = \begin{cases} 2 & \text{pro } P1 \\ 2 & \text{pro } P2 \\ 1 & \text{jinak} \end{cases}$
- $M_0 = \begin{cases} 2 & \text{pro } P1 \\ 1 & \text{pro } V1 \\ 1 & \text{pro } K1 \\ 1 & \text{pro } Z1 \\ 0 & \text{jinak} \end{cases}$