

VSL Demonstrační cvičení 3

ZS 2005

Ing. Zbyněk Křivka

krivka@fit.vutbr.cz

Konzultace: Po 11-12h, místnost C48

Poslední možnost odevzdat

krivka@fit.vutbr.cz

Konzultace: Po 11-12h, místnost C48

Poslední možnost
odevzdat vypracovanou
domácí úlohu
na tento týden !!!

Příklad 3.1 – Zadání

- a) Uvedte příklad věty patřící do jazyka definovaného problémem zastavení L_F (slajdy 22–23) a příklad věty nepatřící do tohoto jazyka.
- b) Lze sestrojit Turingův stroj rozhodující tento jazyk L_F a proč?
- c) Uvedte praktický důsledek odpovědi z bodu b).

Příklad 3.1 – Princip

- Uvažujme jen TS s $\Sigma=\{0,1\}$, $\Gamma=\{0,1, \Delta\}$.
- $\sigma(M) \in \Sigma^*$ je binární zakódování TS M pro univerzální TS
- $L_F = \{w \mid w = \sigma(M), \text{ kde } M \text{ je libovolný TS se vstupní abecedou } \Sigma, \text{ který zastaví pro vstup } w = \sigma(M)\}$

Princip jednoduchého nalezení příkladů TS M :

- kódování TS M do binárního kódu
- přijímání pomocí TS, který nemá příliš velké nároky
např. přijímá jazyk $\Sigma^* = \{0, 1\}^*$ nebo $1\{0, 1\}^*1$
- naopak nepřijímající sama sebe – využijme např.
začátku kódovaného TS, který je vždy neměnný: **1...**

Příklad 3.1a – Řešení

$$M_a = (\{q_0, q_F\}, \{0, 1\}, \{\Delta, 0, 1\}, \delta, q_0, q_F),$$

$$\delta = \{ q_0 0 \rightarrow q_F 0, \quad q_0 1 \rightarrow q_F 1, \quad q_0 \Delta \rightarrow q_F \Delta \}.$$

Poznámka: $0 = 000$, $1 = 0000$, $q_0 = 0$, $q_F = 00$, ...

$w = 1\ 010001001000\ 1\ 01000010010000\ 1\ 011001\ 1$

$$w = \sigma(M_a) \in L_F$$

$$M_b = (\{q_0, q_1, q_F\}, \{0, 1\}, \{\Delta, 0, 1\}, \delta, q_0, q_F),$$

$$\delta = \{ q_0 \Delta \rightarrow q_1 R, \quad q_1 0 \rightarrow q_F 0, \quad q_1 1 \rightarrow q_1 1 \}.$$

$w' = 1\ 011000100\ 1\ 00010001001000\ 1$
 $00010000100010000\ 1$

$$w' = \sigma(M_b) \notin L_F$$

Příklad 3.1bc – Řešení

Nelze, jazyk L_F je nerozhodnutelný !!!

Proč? Důkaz sporem:

- **Předpoklad:** $L_F = L(M_F)$, že M_F rozhoduje L_F , takže rozhoduje i $\sigma(M_0) \in L_F$, kde M_0 je vytvořen s využitím samotného M_F .
- M_F' : modifikace M_F , že 0 $\Leftrightarrow$ No, 1 $\Leftrightarrow$ Yes
- M_0 : modifikace M_F' , že 0 $\Rightarrow M_0$ zastaví, 1 $\Rightarrow M_0$ cyklí
- 2 výlučné odpovědi na otázku $\sigma(M_0) \in L_F$?:
 - $\sigma(M_0) \in L_F \Rightarrow M_F'$ dá 1 $\Rightarrow M_0$ cyklí $\Rightarrow \sigma(M_0) \notin L_F \Rightarrow$ **SPOR**
 - $\sigma(M_0) \notin L_F \Rightarrow M_F'$ dá 0 $\Rightarrow M_0$ zastaví $\Rightarrow \sigma(M_0) \in L_F \Rightarrow$ **SPOR**

$\Rightarrow$ neplatí předpoklad a tedy **M_F nerozhoduje L_F** , ale pouze přijímá.

Důsledky: U obecného algoritmu nelze algoritmicky určit některé vlastnosti. Lze to ale u omezené podmnožiny algoritmů (např. tam, kde známe přesnou horní hranici časové složitosti apod.).

Příklad 3.2 - Zadání

Mějme abecedu A , posloupnost všech unárních primitivně rekurzivních funkcí $f_0, f_1, \dots, f_n, \dots$ nad A a necht' je zadána unární funkce $f : A^* \rightarrow A^*$ takto:

$$f(x) = \begin{cases} f_n(a^n) a & \text{jestli } x = a^n \in \{a\}^* \\ \varepsilon & \text{jinak} \end{cases}$$

kde $a \in A$. Je f primitivně rekurzivní? Dokažte.

Nápověda: a^n je n -tá mocnina řetězce a . Důkaz lze elegantně vést sporem.

Příklad 3.2 – Řešení

- f je efektivně vyčíslitelná (obsahuje jen vyčíslení primitivně rekurzivní f_n a konkatenace)

Důkaz, že f není primitivně rekurzivní:

- **Předpoklad:** f je primitivně rekurzivní
- Pak $f \in \{f_0, f_1, \dots, f_n, \dots\}$ a lze psát $f = f_j$ pro nějaké $j \in \mathbb{N}$,
- pak by ale muselo platit:
 $f(a^j) = f_j(a^j)$ a zároveň $f(a^j) = f_j(a^j)a$, kde $a \in A$, $\Rightarrow$
SPOR (funkce nemůže vracet dva různé výsledky pro stejný vstup!) $\Rightarrow$ neplatí předpoklad, tj.:

Funkce f není primitivně rekurzivní !

Poznámka: Důkaz diagonalizací trochu odlišný od Cantorova důkazu diagonalizací, ale stejný princip.

Příklad 3.3 - Zadání

- a) Ukažte, že funkce počítající libovolný prvek Fibonacciho posloupnosti je primitivně rekurzivní.

$$F_n := F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

- b) Ukažte, že umocňování přirozených čísel je také primitivně rekurzivní funkce.

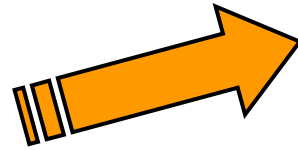
Příklad 3.3a – Řešení

Princip:

$$fib(0) = K_0^1(0) = \xi()$$

$$fib(1) = K_1^1(1) = \sigma \circ \xi()$$

$$fib(n+2) = plus(fib(n+1), fib(n))$$



$$f(\bar{x}, 0) = g(\bar{x}), \text{ kde } \bar{x} \in \mathbb{N}^k$$

$$f(\bar{x}, y+1) = h(\bar{x}, y, f(\bar{x}, y))$$

Příklad řešení bez zpětné rekurze):

$$fib(n) = f(0, 1, n)$$

$$f(a, b, 0) = a$$

$$f(a, b, n+1) = f(b, a+b, n)$$

Příklad 3.3b – Řešení

Mocnina (exp): x, y jsou přirozená čísla

$$- x^0 = 1$$

$$\text{tj. } \exp(x, 0) = 1$$

$$- x^{y+1} = x * x^y$$

$$\text{tj. } \exp(x, y+1) = x * \exp(x, y)$$

$$\exp(x, 0) = K_1^1(x)$$

$$\exp(x, y+1) = \text{mult}(x, \exp(x, y)),$$

kde mult je primitivně rekurzivní funkce.



$$f(\bar{x}, 0) = g(\bar{x}), \text{ kde } \bar{x} \in \mathbb{N}^k$$

$$f(\bar{x}, y+1) = h(\bar{x}, y, f(\bar{x}, y))$$

Příklad 3.4 – Zadání

Mějme parciálně rekurzivní funkce f , g a primitivně rekurzivní funkce p , q . Jsou následující funkce h_a , h_b , h_c , h_d a h_e Turingovsky vyčíslitelné? Svě odpovědi zdůvodněte.

a) $h_a(x) =$

- $f(x)$: jestliže je $f(x)$ definovaná;
- $g(x)$: jestliže je $g(x)$ definovaná a $f(x)$ není definovaná;
- nedef. : jestliže není definována ani $f(x)$ ani $g(x)$.

b) h_b je def. pro ty hodnoty, pro které je definována alespoň jedna z funkcí f a g .

c) h_c je definována pro ty hodnoty, pro které je definována funkce f a zároveň i g .

d) $h_d = p^q$.

e) h_e : Modifikujte a) tak, že f a g budou primitivně rekurzivní funkce.

Příklad 3.4a – Řešení

Odpověď: Parciálně rekurzivní funkce h_a **neexistuje!**

Řešení 3.4a:

- $f(x)$, $g(x)$...reprezentujeme 2 TS (bez abnorm.uk.)
- 1. a 3. odrážka v definici funkce h_a jsou OK.
- **Problém:**
 - $h(x) = g(x)$ if $g(x)$ defined and $f(x)$ not defined.
 - Nelze zjistit, zda je $f(x)$ nedefinovaná !!!
 - „Obecně **nevím**, jestli $f(x)$ ještě stále počítá (v cyklu) a nebo už jenom nekonečně cyklí.“

Příklad 3.4b – Řešení

b) Parciálně rekurzivní funkce h_b **existuje!**

Řešení 3.4b:

- $f(x), g(x) \dots$ 2 TS (bez abnormálního ukončení)
- Při pseudoparalelním výpočtu obou po každém kroku kontroluji, zda **výpočet** některé funkce již **neskončil**,
 - pokud **ano**, tak má být h_b definováno (jak, to nebylo uvedeno; třeba konstantou)
 - pokud **ne**, tak se počítá dále; nevadí pokud již obě funkce nekonečně cyklí (tj. jsou nedefinované), protože pak má být podle definice i h_b nedefinované.

Příklad 3.4c – Řešení

b) Parciálně rekurzivní funkce h_c **existuje!**

Řešení 3.4c:

- $f(x), g(x) \dots$ 2 TS (bez abnormálního ukončení)
- Při pseudoparalelním výpočtu obou po každém kroku kontroluji, zda **výpočet obou** funkcí již **neskončil**,
 - pokud **ano**, tak má být h_c definováno (jak, to nebylo uvedeno; třeba konstantou)
 - pokud **ne**, tak se počítá dále; nevadí pokud některá funkce již nekonečně cyklí (tj. je nedefinované), protože pak má být podle definice i h_c nedefinované.

Příklad 3.4de – Řešení

d)

- Uzavřenost třídy primitivně rekurzivních funkcí na operaci mocnina.
- Každá primitivně rekurzivní funkce Turingovsky vyčíslitelná
- Také mocnina dvou prim. rek. funkcí je Turingovsky vyčíslitelná.

e)

- Primitivně rekurzivní funkce jsou vždy definované
- Výsledkem bude vždy funkce $f(x)$, resp. degradujeme funkci $h_a(x)$ na $h_a(x) = f(x)$ pro libovolné x .

Příklad 3.Bonus - Zadání

„Tiskový problém“ pro Turingův stroj M a nějaký symbol jeho páskové abecedy x je problém určení, zda pro nějaký vstupní řetězec w provede stroj M při přijímání řetězce w alespoň jeden zápis symbolu x na pásku.

Je tento problém rozhodnutelný? Zdůvodněte.

Příklad 3.Bonus – Řešení

- „Tiskový problém“ je NEROZHODNUTELNÝ, protože na něj lze transformovat problém zastavení. Jak?
- **Intuitivně:** Lze identifikovat hrany Turingova stroje M , které provádějí zápis monitorovaného symbolu na pásku, ale nelze obecně analýzou TS M zjistit, jestli bude daná hrana programem použita nebo ne.
- **Přesněji:**
 - Bude na pásku TS M_F zapsán symbol **Y**es, který signalizuje kladnou odpověď na otázku zastavení, když Y není použito v žádné jiné části algoritmu (např. jako pomocný znak na pásce)?
 - Pokud by byl tento problém rozhodnutelný, tak lze určit i kdy odpověď Yes nedá a tedy rozhodovat problém zastavení, což je SPOR.

Dotazy?

Děkuji za pozornost!

Nashledanou!