

VSL Demonstrační cvičení 3

Ing. Zbyněk Křivka

krivka@fit.vutbr.cz

Konzultace: Čt 11-14h, místnost C48

Příklad 1 – Zadání

39/15a)

Rozhodněte, zda sjednocením **konečného** počtu jazyků s gramatickým základem vznikne jazyk s gramatickým základem.

Úvod: Způsoby popisu jazyka?

- Výčtem ... nekonečný popis
- Gramatikou, Strojem ... konečný popis

Příklad 1 – Řešení

Odpověď: ANO.

Vysvětlení: Jazyky s gramatickým základem $L(G_i)$,
kde $G_i = (N_i, T_i, P_i, S_i)$ pro $\forall i \in \{1, 2, \dots, n\}$.

Sestrojíme $G = (N, T, P, S)$ přijímající $L(G) = \bigcup_{i=1}^n L(G_i)$
takto:

$$N = S \cup N_1 \cup N_2 \cup \dots \cup N_{n-1} \cup N_n,$$

$$T = T_1 \cup T_2 \cup \dots \cup T_{n-1} \cup T_n,$$

$$P = \{S \rightarrow S_i \mid i \in \{1, 2, \dots, n\}\} \cup P_1 \cup P_2 \cup \dots \cup P_n.$$

Konečná kardinalita množin $N, T, P \Rightarrow G$ je OK.

PS: konečný popis nekonečného jazyka, nedeterminismus

Příklad 2 – Zadání

39/15b)

Rozhodněte, zda sjednocením prvků
množiny jazyků s gramatickým základem
vždy vznikne jazyk s gramatickým
základem.

Příklad 2 – Řešení

Odpověď: Obecně NE.

Vysvětlení:

- Jazyků s gramatickým základem – nekonečně mnoho ... jazyků typu L_0 je nekonečně mnoho (viz. minulá D.Ú.)
- G_i ... nekonečně mnoho gramatik
- Sjednocení přes všechna $i \Rightarrow$ nekonečné množiny terminálů, neterminálů a pravidel $\Rightarrow$ není to gramatika (Ta totiž požaduje konečné množiny terminálů, neterminálů a pravidel!)

Příklad výjimky: $L_i = L_j$ pro vš. $i, j \Rightarrow L = L(G_i)$ pro lib. i .

Příklad 3 – Zadání

39/16a)

Rozhodněte, zda **průnik konečného** počtu jazyků s gramatickým základem vytváří opět jazyk s gramatickým základem.

Zdůvodněte.

Příklad 3 – Řešení

Odpověď: ANO.

Vysvětlení: Jazyky s gramatickým základem $\Rightarrow$ existuje odpovídající TS_i pro $\forall i \in \{1, 2, \dots, n\}$, bez újmy na obecnosti předpokládejme, že cyklí pro vstup $w \notin L(TS_i)$.

Sestrojíme vícepáskový TS přijímající $L(TS) = \bigcap_{i=1..n} L(TS_i)$:

Více možných řešení:

1) n pásek (rozkopíruj vstup na všechny pásky; 1 páska pro každý TS_i)
nebo

2) 2 pásky (vstupní+pracovní): Komponenta C obnoví obsah pracovní pásky ze vstupní pásky (kopií) a nastaví se na začátek.

– Kompozice části $TS - TS_i$ pracují nad pracovní páskou:

$$TS = C TS_1 C TS_2 C TS_3 C \dots C TS_n$$

Příklad 4 – Zadání

39/16b)

Rozhodněte, zda **průnik** prvků **množiny jazyků** s gramatickým základem vždy vytváří jazyk s gramatickým základem. Zdůvodněte.

Příklad 4 – Řešení

Odpověď: Obecně NE.

Vysvětlení: vlastně ukáži sporem:

- Varianta 2) Nekonečně mnoho jazyků $\Rightarrow$ nekonečně mnoho TS_i ke kompozici $\Rightarrow$ výsledný TS $\Rightarrow$ porušení požadavku na konečný počet stavů ve výsledném TS $\Rightarrow$ výsledkem není TS = SPOR
- Varianta 1) nekonečně mnoho pásek pro nekonečně mnoho $TS_i \Rightarrow$ porušení požadavku na konečný počet pásek $\Rightarrow$ výsledkem není TS

Příklad výjimky: $L_1 = \emptyset \Rightarrow L = \emptyset \dots L \in L(TS)$

Příklad 5 - Zadání

69/7)

Vypočtete $A(2,2)$, kde A je Ackermannova funkce.

Definice Ackermannovy funkce:

a) $A(0, y) = y + 1$

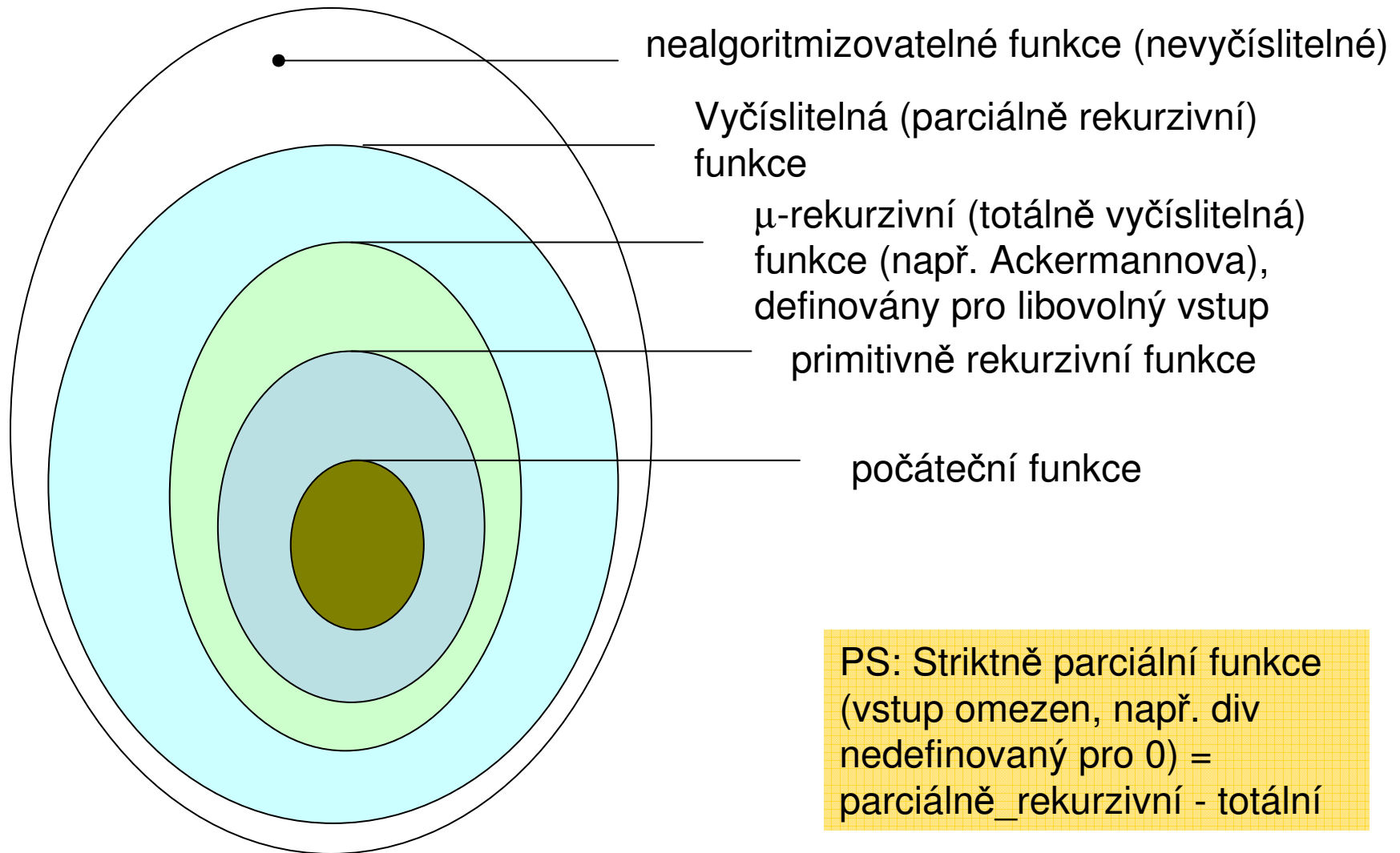
b) $A(x+1, 0) = A(x, 1)$

c) $A(x+1, y+1) = A(x, A(x+1, y))$

- Není primitivně rekurzivní, je to totální funkce.

Příklad 5 – Řešení 1/2

Klasifikace funkcí (viz. přednášky)



Příklad 5 – Řešení 2/2

- $A(2, 2) =^c A(1, A(2, 1)) = \dots$ **Zjisti $A(2,1)$?**

Pomocné: $A(0, 0) =^a 1$; $A(0, 1) =^a 2$; $A(0, 2) =^a 3$; $A(0, 3) =^a 4$; ...

$A(1, 0) =^b A(0, 1) =^a 2$;

$A(2, 0) =^b A(1, 1) =^c A(0, A(1, 0)) = A(0, 2) =^a 3$;

$A(2, 1) =^c A(1, A(2, 0)) = A(1, 3) =^c A(0, A(1, 2)) = A(0, 4) = 5$

- $A(2, 2) =^c A(1, A(2, 1)) = A(1, 5) =^c A(0, A(1, 4)) \dots$ **Zjisti $A(1,4)$?**

$A(1, 4) =^c A(0, A(1, 3)) \dots$ **Zjisti $A(1, 3)$?**

$A(1, 3) =^c A(0, A(1, 2)) \dots$ **Zjisti $A(1, 2)$?**

$A(1, 2) =^c A(0, A(1, 1)) = A(0, 3) =^a 4$

$A(1, 3) =^c A(0, A(1, 2)) = A(0, 4) =^a 5$

$A(1, 4) =^c A(0, A(1, 3)) = A(0, 5) =^a 6$

- **$A(2, 2) = \dots =^c A(0, A(1, 4)) = A(0, 6) =^a 7$**

Příklad 5 – Řešení 2/2 Bonus

$$\begin{aligned} A(2, 2) &=^c A(1, A(2, 1)) =^c A(1, A(1, A(2, 0))) =^b A(1, A(1, A(1, 1))) =^c \\ &A(1, A(1, A(0, A(1, 0)))) =^b A(1, A(1, A(0, A(0, 1)))) =^a \\ &A(1, A(1, A(0, 2))) =^a \\ &A(1, A(1, 3)) =^c A(1, A(0, A(1, 2))) =^c A(1, A(0, A(0, A(1, 1)))) =^c \\ &A(1, A(0, A(0, A(0, A(1, 0))))) =^b A(1, A(0, A(0, A(0, A(0, 1))))) =^b \\ &A(1, A(0, A(0, A(0, 2)))) =^a A(1, A(0, A(0, 3))) =^a A(1, A(0, 4)) =^a \\ &A(1, 5) =^c A(0, A(1, 4)) =^c A(0, A(0, A(1, 3))) =^c \\ &A(0, A(0, A(0, A(1, 2)))) =^c A(0, A(0, A(0, A(0, A(1, 1))))) =^c \\ &A(0, A(0, A(0, A(0, A(0, A(1, 0))))) =^b \\ &A(0, A(0, A(0, A(0, A(0, A(0, 1))))) =^a \\ &A(0, A(0, A(0, A(0, A(0, 2))))) =^a A(0, A(0, A(0, A(0, 3)))) =^a \\ &A(0, A(0, A(0, 4))) =^a A(0, A(0, 5)) =^a A(0, 6) =^a \mathbf{7} \end{aligned}$$

Postřeh: pravidlo a) snižuje počet A, b) nemění, c) zvyšuje počet A

- **$A(2, 2) = 7$**

Příklad 6 – Zadání

69/13

Ukažte, že programovací jazyk Pascal poskytuje prostředky pro výpočet všech parciálně rekurzivních funkcí. V jakém smyslu je mocnost programovacího jazyka Pascal potenciálně omezena v okamžiku, kdy je implementován na konkrétním počítači ?

Příklad 6 – Řešení 1/3

- Počáteční funkce: Nulová, Následníka, Projekce:
 - $\xi()$ 0
 - $\sigma(x), x \in \mathbf{N}$ succ(x); nebo $x := x + 1$;
 - $\Pi_k^n(x), x \in \mathbf{N}^n$ $x[k]$, kde $x : \text{array}[1..n]$ of „typ“
- Primitivně rekurziví funkce - Kombinace:
 - $f: \mathbf{N}^k \rightarrow \mathbf{N}^m$ procedure $f(x_1, x_2, \dots, x_k : \text{integer},$
var $y_1, y_2, \dots, y_m : \text{integer});$ begin ... end;
 - $g: \mathbf{N}^k \rightarrow \mathbf{N}^n$ procedure $g(x_1, x_2, \dots, x_k : \text{integer},$
var $y_{m+1}, \dots, y_{m+n} : \text{integer});$ begin ... end;
 - $f \times g: \mathbf{N}^k \rightarrow \mathbf{N}^{m+n}$ BEGIN
 $f(x_1, \dots, x_k, y_1, \dots, y_m);$
 $g(x_1, \dots, x_k, y_{m+1}, \dots, y_{m+n});$
 END.

Příklad 6 – Řešení 2/3

- Primitivně rekurzivní funkce - Kompozice:
 - $f: N^k \rightarrow N^m$ procedure f($x_1, x_2, \dots, x_k : integer,$
 var $y_1, y_2, \dots, y_m : integer$); begin ... end;
 - $g: N^m \rightarrow N^n$ procedure g($x_1, x_2, \dots, x_m : integer,$
 var $z_1, \dots, z_n : integer$); begin ... end;
 - $f \circ g: N^k \rightarrow N^n$ BEGIN f($x_1, \dots, x_k, y_1, \dots, y_m$);
 g($y_1, \dots, y_m, z_1, \dots, z_n$); END.
- Primitivní rekurze: Rozšířím si definici funkce v Pascalu!
 - $g: N^k \rightarrow N^m$ function g($x_1, x_2, \dots, x_k : integer$) :
 array[1..m] of integer; begin ... end;
 - $h: N^{k+m+1} \rightarrow N^m$ function h($x_1, x_2, \dots, x_{k+m+1} : integer$) :
 array[1..m] of integer; begin ... end;
 - $f(x, 0) = g(x)$ function f($x: array[1..k]$ of integer; $y:integer$) :
 array[1..m] of integer; begin
 - $f(x, y+1)=h(x,y,f(x,y))$ if $y = 0$ then $f := g(x)$
 else $f := h(x, y, f(x, y-1));$ end;
 - , kde $x \in N^k$ resp. $x : array[1..k]$ of integer;

Příklad 6 – Řešení 3/3

- Parciálně rekurzivní funkce - Minimalizace:
 - z $g:N^{n+1} \rightarrow N$ vytvořím $f:N^n \rightarrow N$, že platí:
 - $g(x,y) = 0$, kde x je opět vektor
 - $g(x,z)$ je def. Pro všechna $z < y$, $z \in N$
 - $f(x) = \mu y[g(x,y) = 0]$... „umí hledat, cyklit!“
 - Function $f(\text{vektor } x) : \text{integer}$;
begin $y := 0$;
while($g(x,y) \neq 0$) $y := \text{succ}(y)$;
 $f := y$; end;
- Reálné omezení: konečná paměť $\Rightarrow$ omezení jazyka.
Konečná paměť $\Rightarrow$ lze modelovat konečným automatem.

Příklad 7 – Zadání/Řešení

69/19)

Pro jaké řetězce je definovaná parciální funkce

$$f: \{x, y\}^* \rightarrow \{x, y\}^*,$$

kterou počítá kompozitní Turingův stroj

$$\rightarrow R_x R_y L_{\Delta}.$$

Popište tuto činnost.

Činnost: Najde první x , najde první y a vrátí se na začátek. Tj. přijímá řetězec obsahující alespoň jedno x a za ním alespoň jedno y : $rv = y^*xx^*y(x+y)^*$ pro abecedu $\{x, y\}$.

Funkce f je **parciální identita** tj. nemění pásku.

Příklad 8 – Zadání

69/22)

Předpokládejme, že funkce $g: \mathbb{N} \rightarrow \mathbb{N}$ a

$k: \mathbb{N}^3 \rightarrow \mathbb{N}$ jsou primitivně rekurzivní.

Ukažte, že funkce $f: \mathbb{N} \rightarrow \mathbb{N}$ definovaná

$$f(0, y) = g(y)$$

$$f(x+1, y) = k(f(x, y), y, x)$$

je primitivně rekurzivní.

Příklad 8 – Řešení

- K dispozici máme: **prim. rekurzi, kompozici, kombinaci, nulové**

Primitivní rekurze:	Zadání: $f, g: \mathbb{N} \rightarrow \mathbb{N}, k: \mathbb{N}^3 \rightarrow \mathbb{N}$
$\bar{f}(x, 0) = g(x)$	$f(0, y) = g(y)$
$\bar{f}(x, y+1) = \bar{k}(x, y, \bar{f}(x, y))$	$f(x+1, y) = k(f(x, y), y, x)$

Převod 1. rovnice:

- $f(0, y) = \bar{f}(\Pi_2^2 \times \Pi_1^2 (0, y)) = \bar{f}(y, 0) = g(y)$

Převod 2. rovnice:

- $f(x+1, y) = \bar{f}(\Pi_2^2 \times \Pi_1^2 (x+1, y)) = \bar{f}(y, x+1)$
- $k(f(x, y), y, x) = \bar{k}(\Pi_2^3 \times \Pi_3^3 \times \Pi_1^3 (\bar{f}(\Pi_2^2 \times \Pi_1^2 (x, y)), y, x)) = \bar{k}(y, x, \bar{f}(y, x))$

Nakonec substituce $x/y, y/x$.

- Povolenými úpravami jsme dostali z f, g, k primitivně rekurzivní funkce, takže i f, g, k jsou primitivně rekurzivní funkce.