
PAL: 8. cvičení

Tomáš Sieger

12. 11. 2020

Opakování z minula

Př. 7/2a: automaty

Nakreslete stavový diagram automatu přijímajícího právě všechna slova nad abecedou $\{0, 1\}$, která

- a) **obsahují podposloupnost 1010 alespoň jednou,**
- b) **neobsahují podposloupnost 1010,**
- c) **obsahují podposloupnost 1010 právě jednou,**
- d) **obsahují podposloupnost 1010 nejvýše dvakrát.**

Př. 7/2b: automaty

Nakreslete stavový diagram automatu přijímajícího právě všechna slova nad abecedou $\{0, 1\}$, která

- a) obsahují podposloupnost 1010 alespoň jednou,
- b) neobsahují podposloupnost 1010,**
- c) obsahují podposloupnost 1010 právě jednou,
- d) obsahují podposloupnost 1010 nejvýše dvakrát.

Př. 7/2c: automaty

Nakreslete stavový diagram automatu přijímajícího právě všechna slova nad abecedou $\{0, 1\}$, která

- a) obsahují podposloupnost 1010 alespoň jednou,
- b) neobsahují podposloupnost 1010,
- c) obsahují podposloupnost 1010 právě jednou,**
- d) obsahují podposloupnost 1010 nejvýše dvakrát.

Př. 7/2d: automaty

Nakreslete stavový diagram automatu přijímajícího právě všechna slova nad abecedou $\{0, 1\}$, která

- a) obsahují podposloupnost 1010 alespoň jednou,
- b) neobsahují podposloupnost 1010,
- c) obsahují podposloupnost 1010 právě jednou,
- d) obsahují podposloupnost 1010 nejvýše dvakrát.**

Př. 7/3a: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) **jehož slova obsahují pouze nuly**
- b) jehož každé slovo obsahuje právě jedinou jedničku
- c) jehož každé slovo obsahuje alespoň jednu jedničku
- d) jehož každé slovo obsahuje alespoň dvě jedničky
- e) jehož slova obsahují sudý počet jedniček
- f) jehož slova obsahují lichý počet jedniček

Př. 7/3b: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) jehož slova obsahují pouze nuly
- b) jehož každé slovo obsahuje právě jedinou jedničku**
- c) jehož každé slovo obsahuje alespoň jednu jedničku
- d) jehož každé slovo obsahuje alespoň dvě jedničky
- e) jehož slova obsahují sudý počet jedniček
- f) jehož slova obsahují lichý počet jedniček

Př. 7/3c: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) jehož slova obsahují pouze nuly
- b) jehož každé slovo obsahuje právě jedinou jedničku
- c) jehož každé slovo obsahuje alespoň jednu jedničku**
- d) jehož každé slovo obsahuje alespoň dvě jedničky
- e) jehož slova obsahují sudý počet jedniček
- f) jehož slova obsahují lichý počet jedniček

Př. 7/3d: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) jehož slova obsahují pouze nuly
- b) jehož každé slovo obsahuje právě jedinou jedničku
- c) jehož každé slovo obsahuje alespoň jednu jedničku
- d) jehož každé slovo obsahuje alespoň dvě jedničky**
- e) jehož slova obsahují sudý počet jedniček
- f) jehož slova obsahují lichý počet jedniček

Př. 7/3e: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) jehož slova obsahují pouze nuly
- b) jehož každé slovo obsahuje právě jedinou jedničku
- c) jehož každé slovo obsahuje alespoň jednu jedničku
- d) jehož každé slovo obsahuje alespoň dvě jedničky
- e) **jehož slova obsahují sudý počet jedniček**
- f) jehož slova obsahují lichý počet jedniček

Př. 7/3f: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

- a) jehož slova obsahují pouze nuly
- b) jehož každé slovo obsahuje právě jedinou jedničku
- c) jehož každé slovo obsahuje alespoň jednu jedničku
- d) jehož každé slovo obsahuje alespoň dvě jedničky
- e) jehož slova obsahují sudý počet jedniček
- f) **jehož slova obsahují lichý počet jedniček**

Př. 7/4: nedeterministický automat: nuly a jedničky

Navrhněte NKA nad abecedou $\{0, 1, 2\}$, který v textu vyhledá všechny řetězce obsahující tři nuly a dvě jedničky.

Př. 7/5: nedeterministický automat: abcd

Navrhněte NKA nad abecedou $\{a, b, c, d\}$, který v textu vyhledá všechny řetězce ve tvaru $\#ba\#\#b\#$, kde symbol $\#$ představuje právě jeden libovolný znak z množiny $\{a, b, d\}$. Automat musí být schopen zpracovat celý text libovolné délky, tj. octnout se v koncovém stavu po přečtení posledního znaku každého výskytu hledaného řetězce.

Př. 7/6: automat z regulárního výrazu

Sestavte automat, který v textu nad abecedou $\{a, b, c\}$ vyhledává všechna slova popsaná regulárním výrazem $(ac^* + bb)^* a$.

Př. 7/8: řetězce se stejným počtem znaků

Sestavte NKA nad abecedou $\{0, 1, 2\}$, který v textu vyhledá všechny řetězce obsahující stejný počet znaků 0, 1 i 2.

Př. 7/11: regulární soupeři

Rozhodněte, zda regulární výrazy $(01 + 0)^* 0$ a $0(10 + 0)^*$ popisují stejný regulární jazyk.

Operace nad jazyky. Přibližné
vyhledávání v textu pomocí
konečných automatů

Př. 8/0: skládání automatů

Ze dvou NKA A_1 a A_2 přijímajících jazyky L_1 resp. L_2 vytvořte dva NKA, které budou přijímat jazyk $L_1 \cup L_2$ resp. $L_1 \cap L_2$. (*Doporučení: Stavů A_1 si označte písmeny, stavy A_2 čísly, aby bylo skládání přehledné.*)

a) $L_1 = \{0\}, L_2 = \{1\},$

b) $L_1 = \{(00)^*\}, L_2 = \{(000)^*\}.$

Př. 8/1: skládání automatů 2

Nad abecedou $\{0, 1\}$, jsou dány dva jazyky L_1 a L_2 . Slova L_1 jsou popsána výrazem $0 * 1 * 0 * 1 * 0^*$, slova L_2 jsou popsána výrazem $(01 + 10)^*$.

Sestrojte konečné automaty rozpoznávající jazyk:

- a) $L_1 \cup L_2$,
- b) $L_1 \cap L_2$.

Př. 8/2: konečný průnik

Automat A_1 rozpoznává jazyk L_1 , automat A_2 rozpoznává jazyk L_2 . Oba automaty mají n stavů. Abeceda pro oba jazyky je shodná a má k znaků. Jaká je asymptotická složitost algoritmu, který efektivně určí, zda jazyk $L_1 \cap L_2$ je konečný?

Př. 8/3: hledání pozměněného slova

V textu nad abecedou $\{a, b, c, d\}$ máme určit všechny výskyty takových podřetězců, které začínají i končí znakem b a zároveň mají od daného vzorku *abbbcdabbbcdab* Hammingovu vzdálenost větší než 2. Navrhněte konečný nederministický automat pro řešení této úlohy.

Př. 8/4: hledání pozměněného slova 2

Konečný automat pro hledání v textu, který hledá všechny podřetězce mající od daného vzorku Levenshteinovu vzdálenost menší než dané k , obsahuje ϵ -přechody. Nakreslete příklad tohoto automatu pro délku vzorku 6 a hodnotu $k = 3$. Dále nakreslete, jak bude tento automat vypadat po odstranění všech ϵ -přechodů.

Př. 8/5: hledání pozměněného slova 3

Dvě slova V , W nad abecedou A mají redukovanou Levenshteinovu vzdálenost rovnu k , pokud k je nejmenší možný počet editačních operací, po jejichž provedení ze slova V vznikne slovo W . Za editační operace považujeme v tomto případě pouze operace *Insert* a *Delete*. Sestavte nedeterministický automat bez ϵ -přechodů, který v textu určí všechny výskyty řetězců, které mají od daného vzorku *abaabacc* redukovanou Levenshteinovu vzdálenost rovnou právě 2.

Př. 8/6: vlastnosti Levenshteinovy vzdálenosti

Označme symbolem $d(x, y)$ Levenshteinovu vzdálenost slov x a y . Víme že, pro tři slova u, v, w platí $d(u, v) = d1$, $d(v, w) = d2$. Jakých hodnot může nabývat $d(u, w)$ v závislosti na $d1, d2$? Abeceda je pro všechna slova společná.

Př. 8/8: Hamming vs. Levenshtein

Označme symbolem $HD(v, w)$ Hammingovu vzdálenost slov v a w nad abecedou A , symbolem $LD(v, w)$ Levenshteinovu vzdálenost těchto slov. Rozhodněte, který z následujících případů může nastat a pro možné případy uveďte příklad slov v a w délky alespoň 5.

- a) $HD(v, w) < LD(v, w)$,
- b) $HD(v, w) = LD(v, w)$,
- c) $HD(v, w) > LD(v, w)$.

Př. 8/9: Levenshtein

Napište všechna slova, která mají od vzorku aba nad abecedou $\{a, b, c\}$ Levenshteinovu vzdálenost rovnu 1.

Př. 8/10: SWAP & REWRITE

V textu hledáme podřetězec Q , který se od daného vzorku P může lišit právě jedním z následujících způsobů:

- Q vznikl z P právě jednou operací *SWAP* (vzájemné prohození dvou sousedních znaků),
- Q vznikl z P právě jednou operací *REWRITE* (náhrada jednoho znaku jiným znakem abecedy)

Sestavte NKA pro hledání Q , když víme, že $P = abbaac$, abeceda je $\{a, b, c\}$.

Př. 8/12: předpona a přípona

Nad abecedou A jsou dány dvě konečné množiny řetězců M_1 a M_2 . Popište, jak sestavíte konečný automat, který přijímá všechna taková slova w nad abecedou A , pro která platí, že alespoň jeden prefix slova w leží v množině M_1 a alespoň jeden suffix w leží v množině M_2 . Připomeňme, že celé slovo se považuje za svůj vlastní prefix i suffix. Sestavte příklad pro $|M_1| = |M_2| = 2$.

Př. 8/13: generování podobných textů: Hamming

Navrhněte algoritmus pro vypsání všech slov nad abecedou A , která mají od daného vzorku p Hammingovu vzdálenost právě $k > 0$. Hodnota k je pevně dána. Jaká bude asymptotická složitost tohoto algoritmu?

Př. 8/14: generování podobných textů: Levenshtein

Navrhněte algoritmus pro vypsání všech slov nad abecedou A , která mají od daného vzorku p Levenshteinovu vzdálenost nejvýše $k > 0$. Hodnota k je pevně dána. Jaká bude asymptotická složitost tohoto algoritmu?

