\* 1. 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

a)*A*1 rozpoznávající jazyk *L*1 ∪ *L*2,b)*A*2 rozpoznávající jazyk *L*1 ∩ *L*2.

\* 2. Automat A1 rozpoznává jazyk *L*1, automat A2 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 ∩ *L*2 je konečný?

\* 3. 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 *abbbcdabbcdab* Hammingovu vzdálenost *větší* než 2. Navrhněte konečný nederministický automat pro řešení této úlohy.

\* 4. Konečný automat pro hledání v textu všech podřetězců, které mají od daného vzorku Levenshteinovu vzdálenost menší než dané *k*, obsahuje epsilon-přechody. Napište přiklad tohoto automatu pro délku vzorku 6 a hodnotu *k* = 3. Dále napište, jak bude tento automat vypadat po odstranění všech epsilon-přechodů.

\* 5. 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 totmto případě pouze operace Insert a Delete. Sestavte nedeterministický automat bez epsilon-přechodů, který v textu určí všechny výskyty řetězců, které mají od daného vzorku *abaabacc* redukovanou Levenshteinovu vzdálenost rovnou právě 2.

6. Označme symbolem d(*x*, *y*) Levenshteinovu vzdálenost slov *x* a *y*. Víme že, pro tři slova *u*, *v*, *w* platí d(*u*, *v*) = *d*1,

d(*v*, *w*) = *d*2. Jakých hodnot může nabývat d(*u*, *w*) v závislosti na *d*1, *d*2? Abeceda je pro všechna slova společná.

7. Sestrojte NKA pro nalezení v textu libovolného prvku množiny všech souvislých podřetězců vzorku

*abcdefghijklmnopqrstuvwxzy*.

8. Označme symbolem HD(*v*, *w*) Hammingovu vzdálenost slov *v* a *w* nad abecedou A, symbolem LD(*v*, *w*) Levenshteinovu vzdálenost těchže 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*).

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

a) 1,b) 2.

10. 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 operaci REWRITE (náhrada jednoho znaku jiným znakem abecedy)

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

11. Nad abecedou A = {0, 1} je dán jazyk *L*1 všech slov, která obsahují souvislý podřetězec 00 nejvýše jednou. Dále je dán jazyk *L*2 všech slov nad A, která obsahují podřetězec 11 právě jednou. Sestavte konečný automat, který v textu nad abecedou A najde výskyty všech slov jazyka a) *L*1 ∩ *L*2, b) *L*1 ∪ *L*2.

12. Nad abecedou A jsou dány dvě konečné množiny řetězců, M1 a M2. 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 M1 a alespoň jeden suffix w leží v množině M2. Připomeňme, že celé slovo se považuje za svůj vlastní prefix i suffix. Sestavte příklad pro |M1| = |M2| = 2.

13. 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?

14. 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?