\* 1.Najděte v textu *T* všechny výskyty řetězců, které mají od vzorku *P* Hammingovu vzdálenost rovnou nejvýše *k*. Použijte metodu dynamického programování.

a) *T* = ccacbaabccaccbcabccc, *P* = abcba, *k* = 2

b) *T* = 000111011000101010111110,  *P* = 110010, *k* = 3

\* 2. Najděte v textu *T* všechny výskyty řetězců, které mají od vzorku *P* Levenshteinovu vzdálenost rovnou nejvýše *k*. Použijte metodu dynamického programování

a) *T* = aacacacbaabbbcbbcacc, *P* = cbbba, *k* = 3.

b) *T* = 010011101000010101011100 , *P* = 11100, *k* = 1

\* 3.Sestrojte nedeterministický automat, který v textu nad abecedou A vyhledá právě každé slovo množiny M.

a) A = {a, b, c}, M = {a, b, ba, bc, aaa, bab, ccc, abbc, abcc } .

b) A = {0, 1}, M = {10, 11, 101, 111, 1011, 1101, 10001, 10011, 10111, 11101, 11111 }.

\* 4. Sestrojte deterministický automat, který v textu nad abecedou A vyhledá právě každé slovo množiny M z předchozí úlohy.

\* 5. Sestavte tabulky pro simulaci činnosti vyhledávacího automatu metodou bitového paralelizmu pro daný text *t*, vzorek *p* a Hammingovu vzdálenost *k*

a) *t* = abcbcaaccbbaa, *p* = bbac, *k* = 2

b) *t* = accbbaaabcba, *p* = acbb, *k* = 2

6. Dva řetězce X a Y nad danou abecedou mají redukovanou Levenshteinovu vzdálenost rovnou *k*, právě tehdy, když *k* je minimální počet editačních operací, pomocí nichž lze z X vytvořit Y. Přitom za editační operace se považují pouze Insert nebo Delete. Popište algoritmus, jímž se pomocí dynamického programování bude počítat redukovaná Levenshteinova vzdálenost X a Y.

7. Popište algoritmus, jímž se v textu budou hledat metodou dynamického programování všechny řetězce, jejichž redukovaná Levenshteinova vzdálenost (viz předch. úlohu) od zadaného řetězce bude nejmenší možná.

8. Řekneme, že Insert-vzdálenost dvou řetězců X a Y je právě *k*, pokud *k* je minimum operací Insert, které aplikujeme na pouze jeden z obou řetězců, a po jejichž provedení budou oba řetězce shodné. Pokud shody nelze dosáhnout, vzdálenost X a Y bude plus nekonečno. Sestavte NFA, který přijme všechny řetězce R, které mají od daného vzorku P Insert-vzdálenost rovnou nejvýše *k*.

9. Navrhněte metodu založenou na dynamickém programování, která v textu najde výskyt všech řetězců, jež mají od daného vzorku Insert-vzdálenost rovnou dané hodnotě *k*.

10. Popište změny, které bude nutno provést, pokud v předchozí úloze zaměníme Insert-vzdálenost za Delete-vzdálenost. Definujte nejprve Delete-vzdálenost, analogicky jako Insert-vzdálenost.