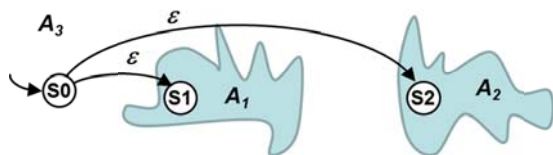


1. Nad abecedou $\{0, 1\}$, jsou dány dva jazyky L_1 a L_2 . Slova L_1 jsou popsána výrazem $0^*1^*0^*1^*0^*$, slova L_2 jsou popsána výrazem $(01+10)^*$. Sestrojte konečný automat bez ε -přechodů A_1 rozpoznávající jazyk $L_1 \cup L_2$.

Minimální počet přechodů (přechod = šipka v přechodovém diagramu ohodnocená případně více symboly) v A_1 :

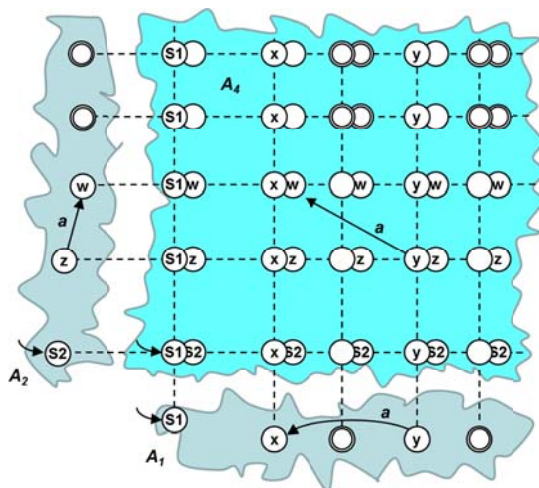
- A. 10 nebo méně
- B. 11 až 15
- C. 16 až 20
- D. 21 až 25
- E. 26 nebo více



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 konečný automat bez ε -přechodů A_2 rozpoznávající jazyk $L_1 \cap L_2$.

Minimální počet přechodů (přechod = šipka v přechodovém diagramu ohodnocená případně více symboly) v A_2 :

- A. 10 nebo méně
- B. 11 až 20
- C. 21 až 30
- D. 31 až 40
- E. 41 nebo více



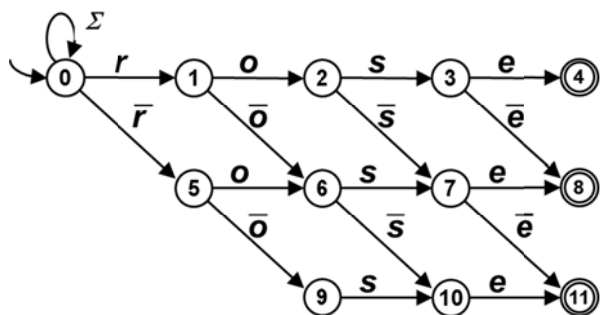
3. Automat A_1 rozpoznává jazyk L_1 , automat A_2 rozpoznává jazyk L_2 . Oba automaty mají n stavů. Abeceda pro oba jazyky je shodná a má k znaků. Jaká je asymptotická složitost algoritmu, který efektivně určí, zda jazyk $L_1 \cap L_2$ je konečný?

- A. $O(n^2k)$
- B. $O(n^3k)$
- C. $O(n^4k)$
- D. $O(n^4k^2)$
- E. Exponenciální vůči n .

4. 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 $bacdb$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nederministický automat pro řešení této úlohy.

Počet stavů automatu:

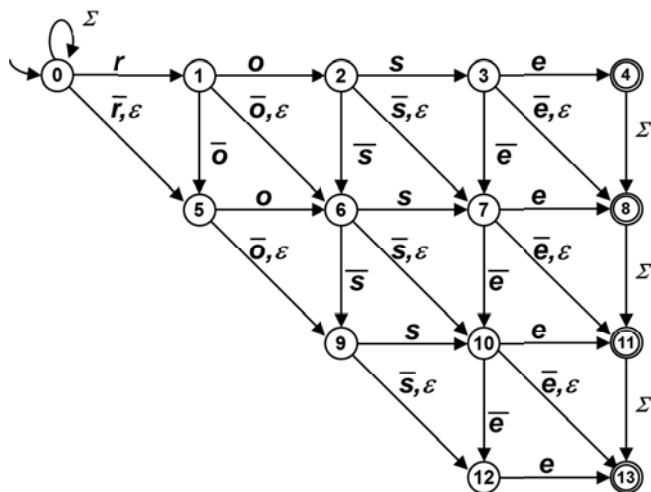
- A. 7 nebo méně
- B. 8 až 10
- C. 11 až 13
- D. 14 až 16
- E. 17 nebo více**



5. Konečný automat pro hledání v textu všech podřetězců, které mají od daného vzorku Levenshteinovu vzdálenost rovnou nejvýše danému k , obsahuje epsilon-přechody. Sestrojte tento automat pro vzorek $ccbbaa$ (nad abecedou $\{a, b, c\}$) a hodnotu $k = 3$.

Počet stavů automatu:

- A. 10 nebo méně
- B. 11 až 16
- C. 17 až 22
- D. 23 až 28
- E. 29 nebo více



6. Konečný automat pro hledání v textu všech podřetězců, které mají od daného vzorku Levenshteinovu vzdálenost rovnou nejvýše danému k . Sestrojte tento automat pro vzorek $ccbbaa$ (nad abecedou $\{a, b, c\}$) a hodnotu $k = 3$, přičemž z automatu jsou odstraněny všechny epsilon-přechody.

Počet stavů automatu:

- A. 10 nebo méně
- B. 11 až 16
- C. 17 až 22
- D. 23 až 28
- E. 29 nebo více

7. Sestrojte NKA pro nalezení v textu libovolného neprázdného řetězce z množiny všech souvislých podřetězců vzorku *klmnop*.

Počet stavů automatu:

- A. 10 nebo méně
- B. 11 až 16
- C. 17 až 22
- D. 23 až 28
- E. 29 nebo více

8. Napište všechna slova, která mají od vzorku abc nad abecedou $\{a, b, c\}$ Levenshteinovu vzdálenost rovnu 1. Jejich počet je:

- A. 10 nebo méně
- B. 11 až 16
- C. 17 až 22
- D. 23 až 28
- E. 29 nebo více

9. 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 epsilon-přechodů, který v textu určí všechny výskyty řetězců, které mají od daného vzorku *abaabacc* redukovanou Levenshteinovu vzdálenost rovnou právě 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 operací 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 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 . Pripomeňme, že celé slovo se považuje za svůj vlastní prefix i suffix. Sestavte příklad pro $|M_1| = |M_2| = 2$.

12. 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?

13. 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?