PAL: 9. cvičení

Tomáš Sieger

19. 11. 2020

Opakování z minula

Př. 8/1: skládání automatů 2

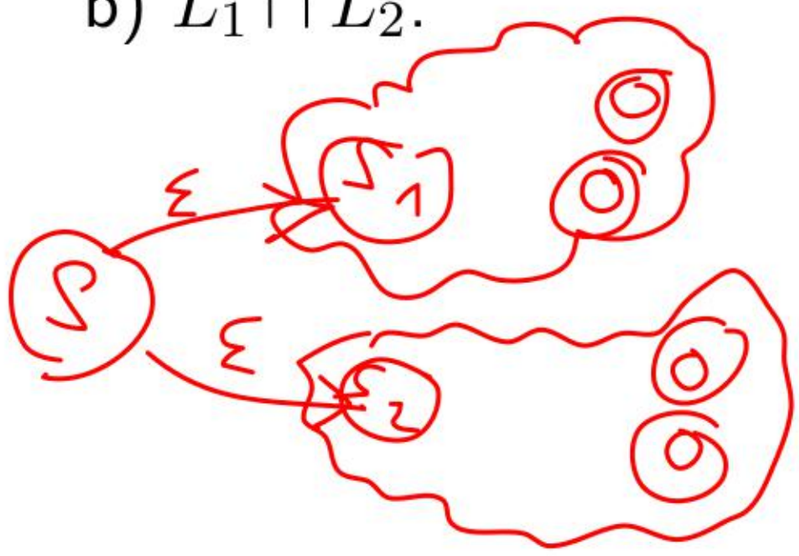
Nad abecedou $\{0, 1\}$, jsou dány dva jazyky L_1 a L_2. Slova L_1 jsou popsána výrazem $0 * 1 * 0 * 1 * 0*$, slova L_2 jsou popsána výrazem $(01 + 10)*$.
Sestrojte konečné automaty rozpoznávající jazyk:
a) $L_1 \cup L_2$,
b) $L_1 \cap L_2$.

Př. 8/1: 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) $L_1 \cap L_2$.

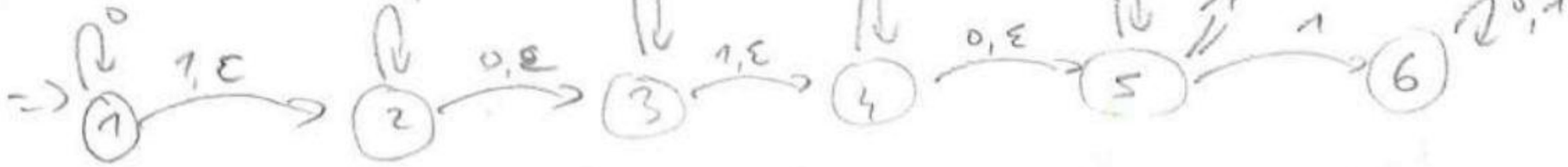


1010
0110
1

(00)*

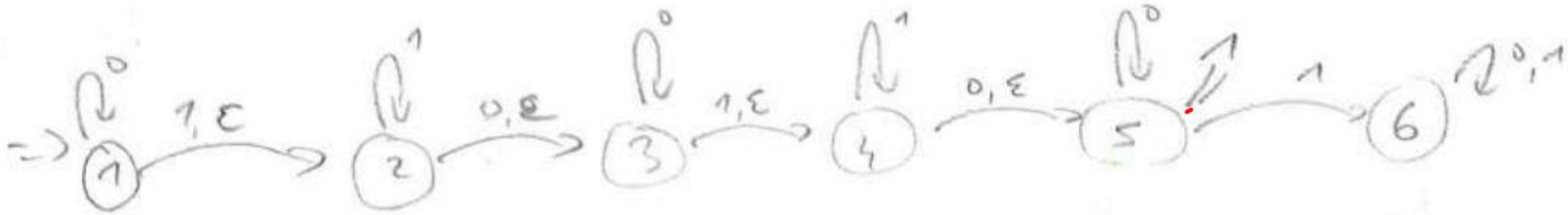
000
000000

(000)*


1
2
3
4
5
6
1, ε
0, ε
1, ε
0, ε
1
0, 1


0, 1
1
0
1
0
1
1
0
1
0, 1

$L_1 \cap L_1$



b)

$L_1 \cup L_2$

$L_1 \cap L_2$

b)

Yes 9 100%

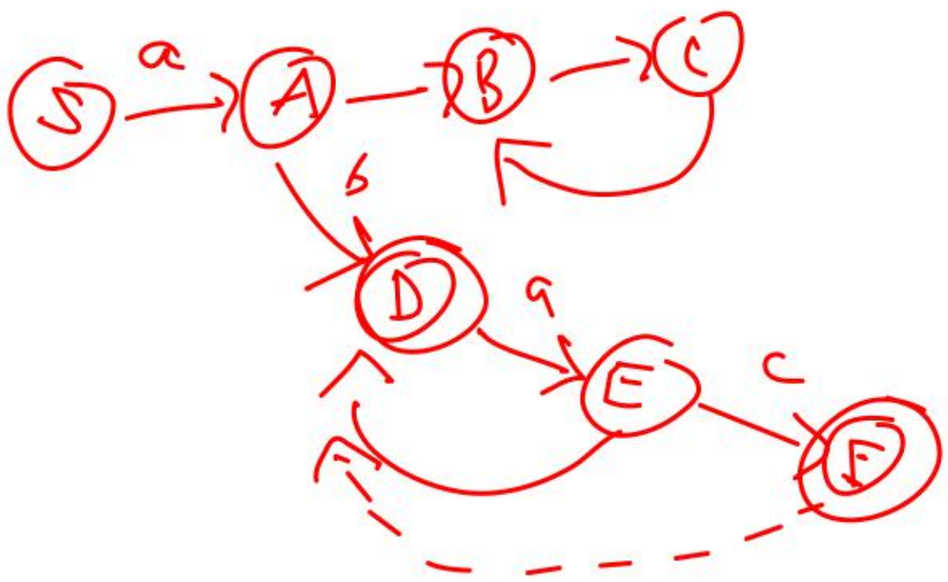
No 0 0% 50%

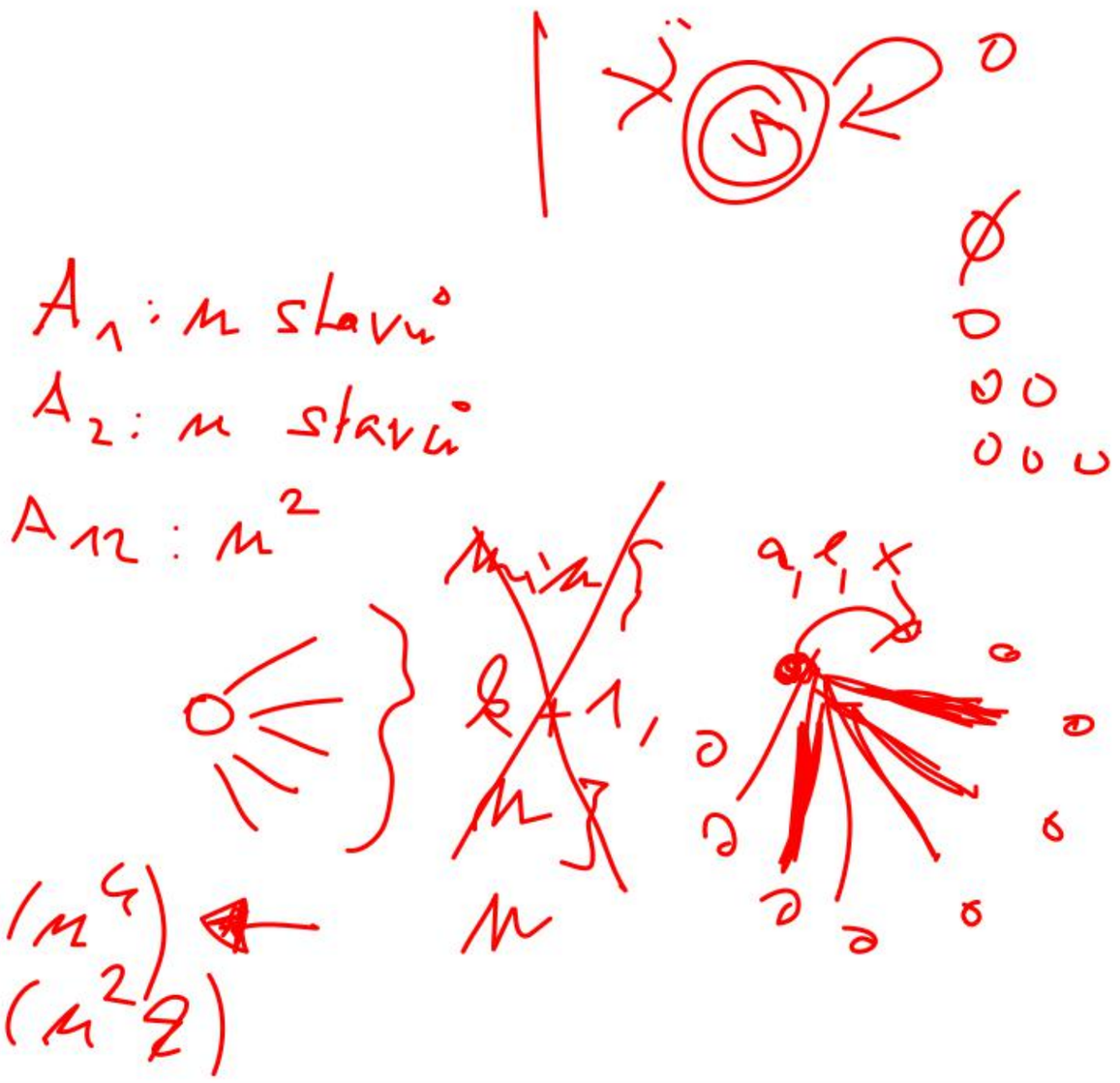
Př. 8/2: konečný průnik

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. 8/2: 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 Hammingovsky 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 $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nederministický automat pro řešení této úlohy.

Př. 8/3: hledání Hammingovsky pozměněného slova

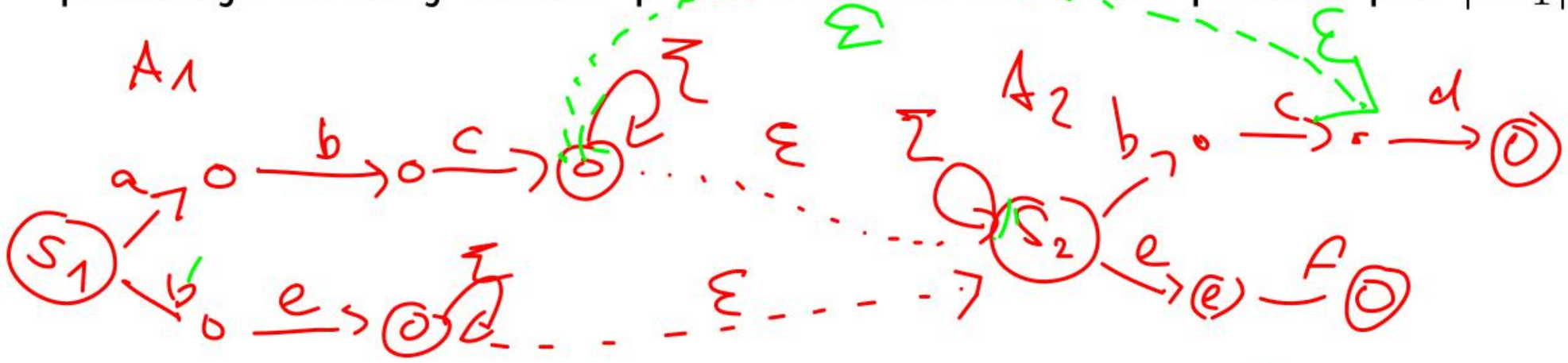
V textu nad abecedou $\{a, b, c, d\}$ máme určit všechny výskyty takových podřetězců, které začínají i končí znakem b a zároveň mají od daného vzorku $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nedeterministický automat pro řešení této úlohy.

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. 8/12: 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 $|M_1| = |M_2| = 2$.



$M_1 = \{abc, be\}$

$M_2 = \{bcd, ef\}$

L_1 M_1 ....

L_2 ... M_2

$L_{12} = L_1 \cap L_2$

$A_{12} = A_1 \cap A_2$

abcd

Yes 7 100%

No 0 0%

11 48

Př. 8/3: hledání Hammingovsky pozměněného slova

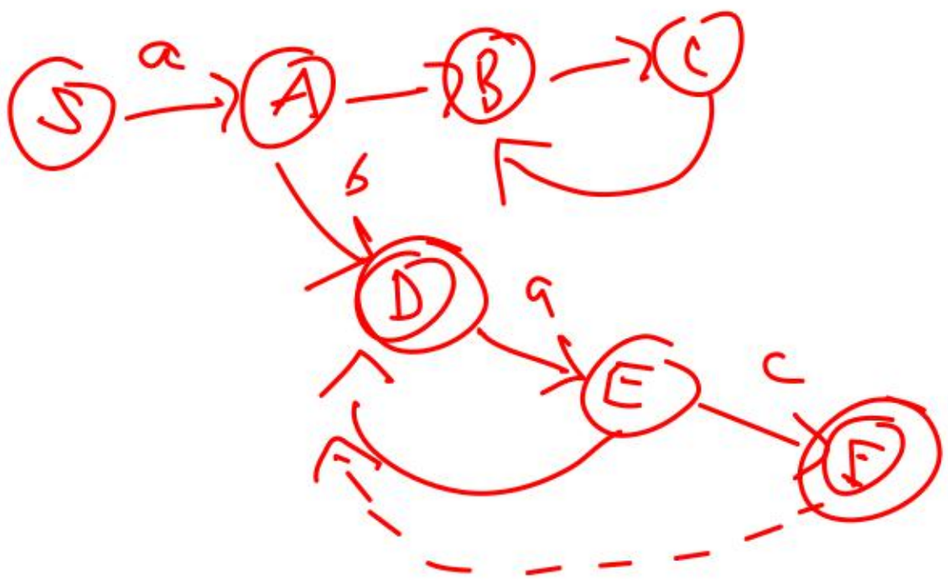
V textu nad abecedou $\{a, b, c, d\}$ máme určit všechny výskyty takových podřetězců, které začínají i končí znakem b a zároveň mají od daného vzorku $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nedeterministický automat pro řešení této úlohy.

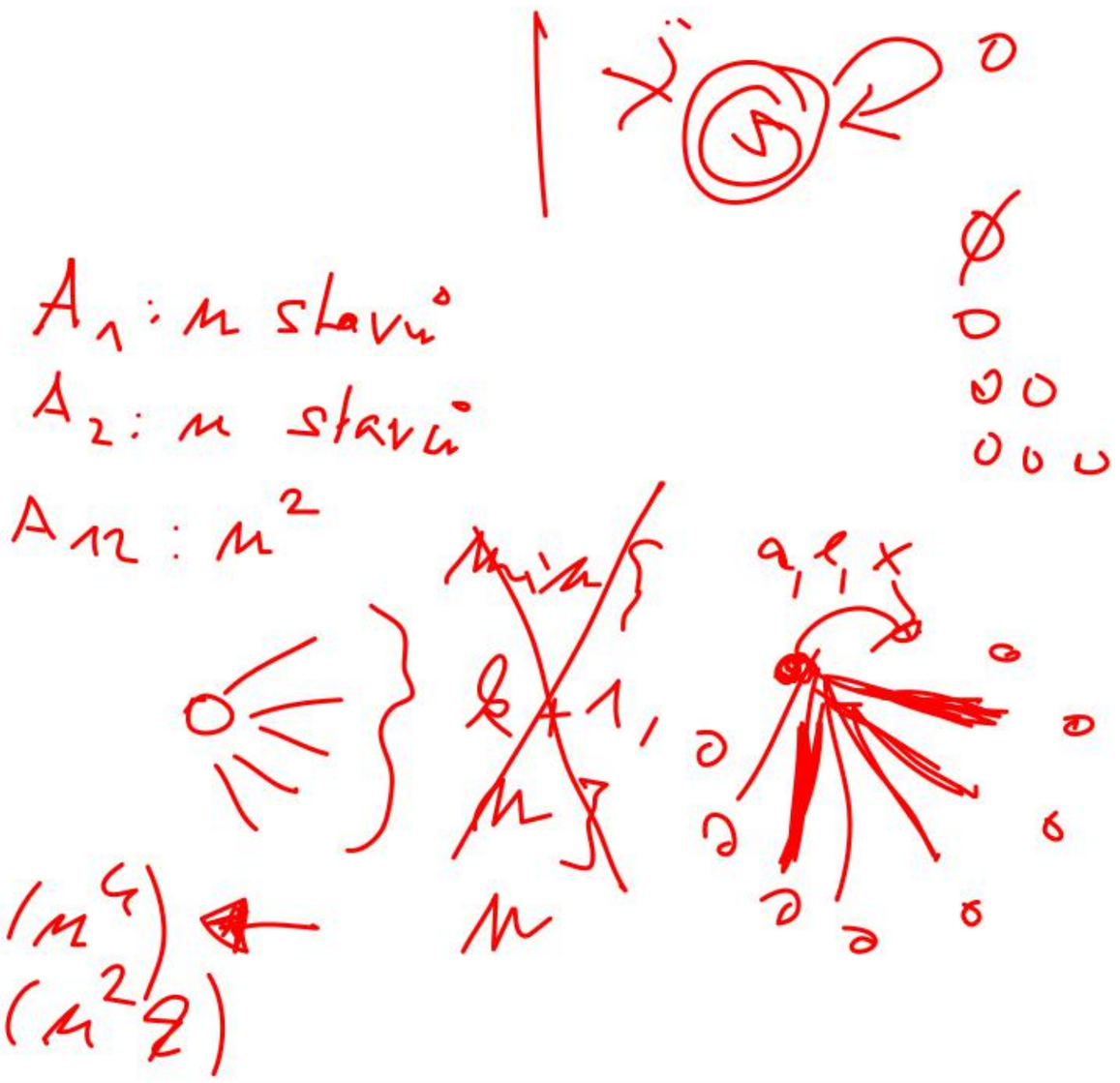
Př. 8/3: hledání Hammingovsky pozměněného slova

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 $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nedeterministický automat pro řešení této úlohy.

Př. 8/2: konečný průnik

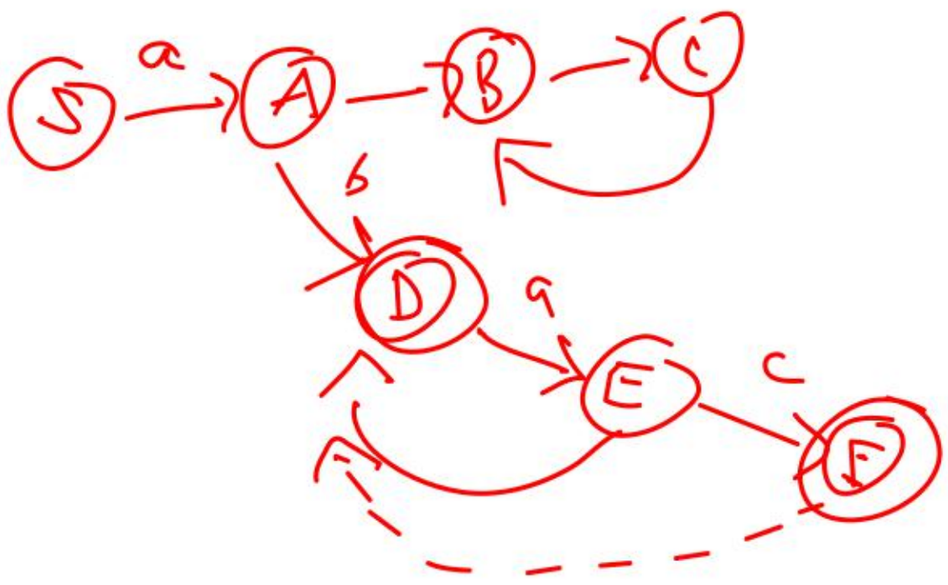
Automat A_1 rozpoznává jazyk L_1, automat A_2 rozpoznává jazyk L_2. Oba automaty mají n stavů. Abeceda pro oba jazyky je shodná a má k znaků. Jaká je asymptotická složitost algoritmu, který efektivně určí, zda jazyk $L_1 \cap L_2$ je konečný?

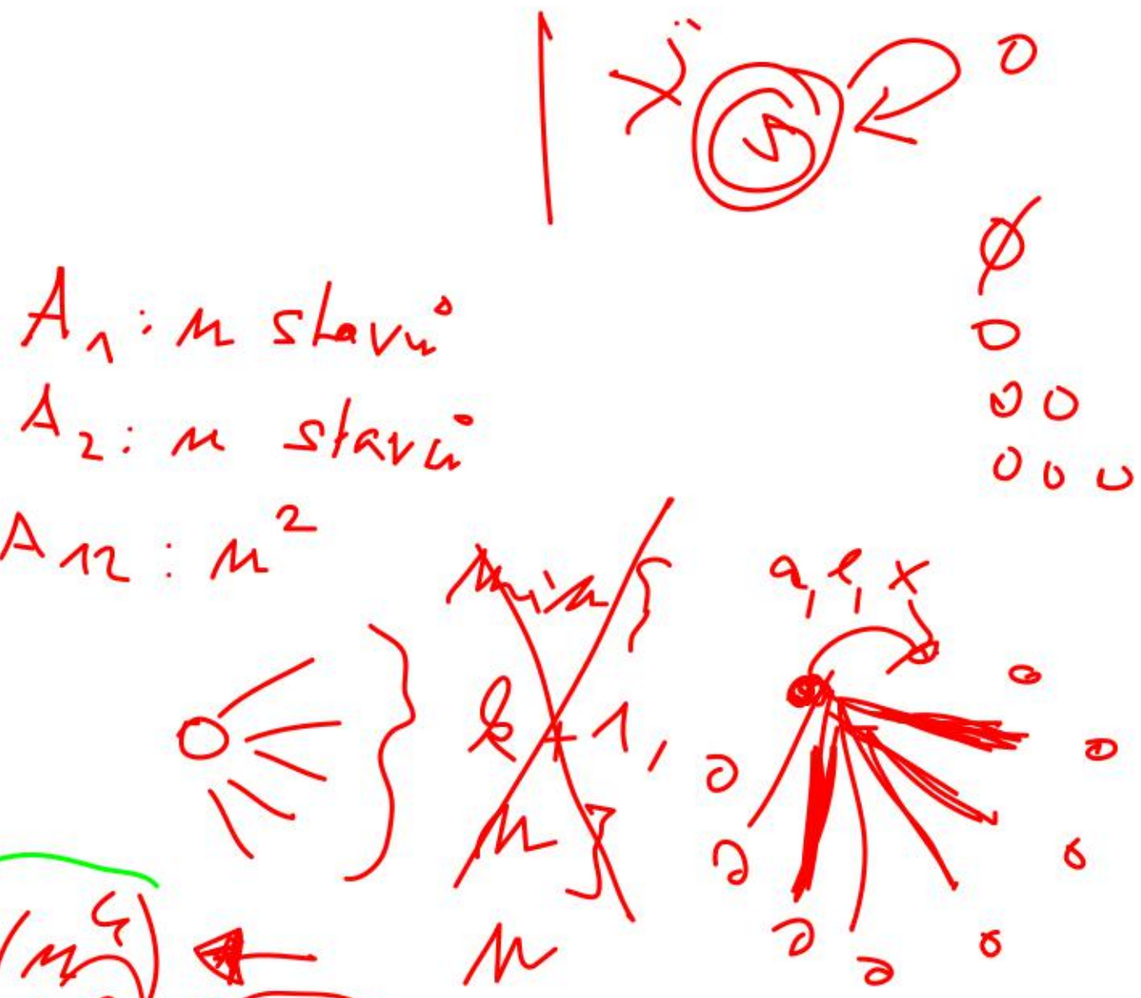




Př. 8/2: konečný průnik

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?





$|V| = n^2$

$|E| = \binom{n^2}{2} = O(n^4)$

$O(n^4)$

$O(|V| + |E|) = O(n^2 + n^4) = O(n^4)$

Př. 8/3: hledání Hammingovsky pozměněného slova

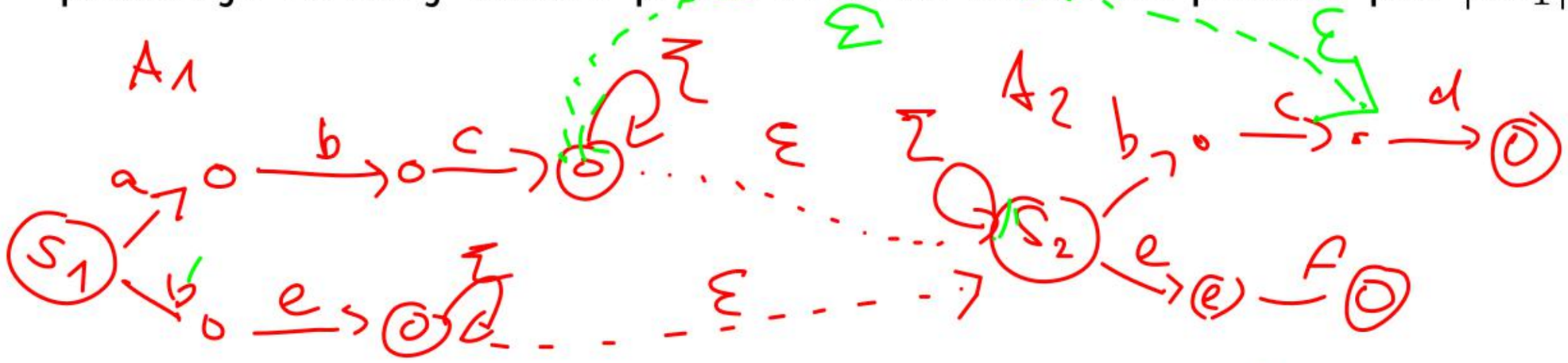
V textu nad abecedou $\{a, b, c, d\}$ máme určit všechny výskyty takových podřetězců, které začínají i končí znakem b a zároveň mají od daného vzorku $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nedeterministický automat pro řešení této úlohy.

Př. 8/3: hledání Hammingovsky 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 $abbbcdabbcdab$ Hammingovu vzdálenost větší než 2. Navrhněte konečný nedeterministický automat pro řešení této úlohy.

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 $|M_1| = |M_2| = 2$.



$M_1 = \{abc, be\}$

$M_2 = \{bcd, ef\}$

L_1 M_1 ....

L_2 ... M_2

$L_{12} = L_1 \cap L_2$

$A_{12} = A_1 \cap A_2$

abcd

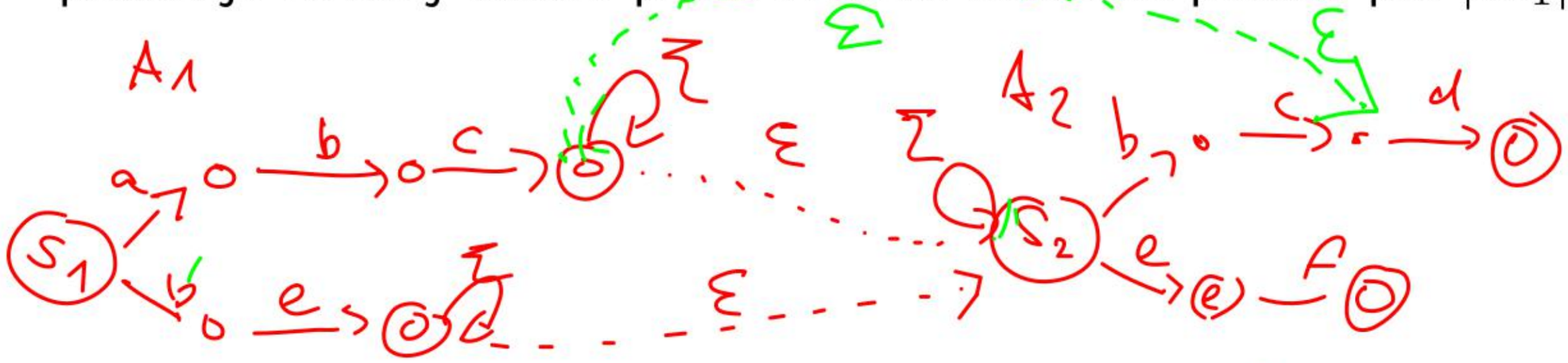
Yes 7 100%

No 0 0%

11 48

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 $|M_1| = |M_2| = 2$.



$M_1 = \{abc, be\}$

$M_2 = \{bcd, ef\}$

L_1 M_1 ....

L_2 ... M_2

$L_{12} = L_1 \cap L_2$

$A_{12} = A_1 \cap A_2$

abcd

Yes 9 100%

No 0 0%

11 48

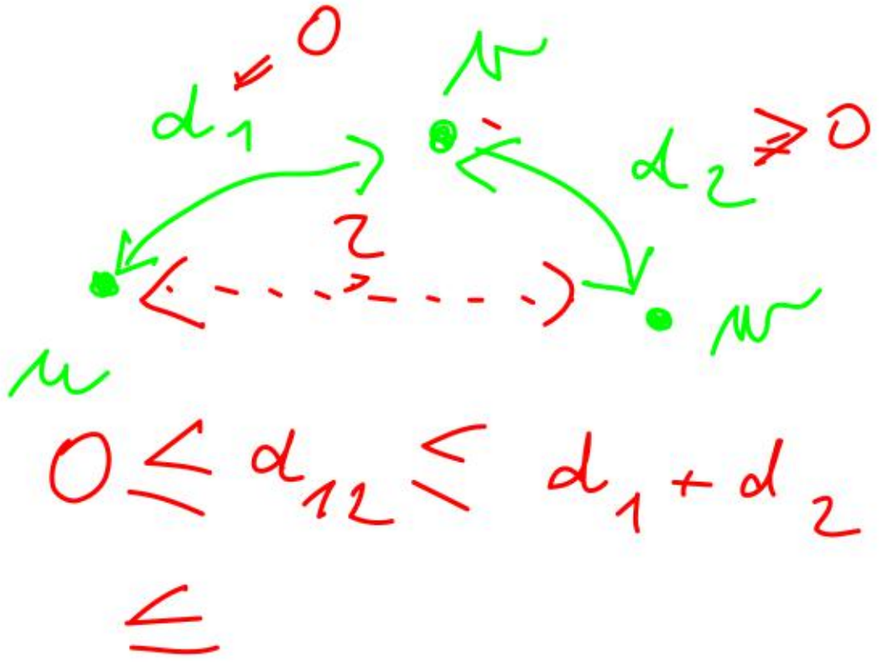
Operace nad jazyky. Přibližné vyhledávání v textu pomocí konečných automatů

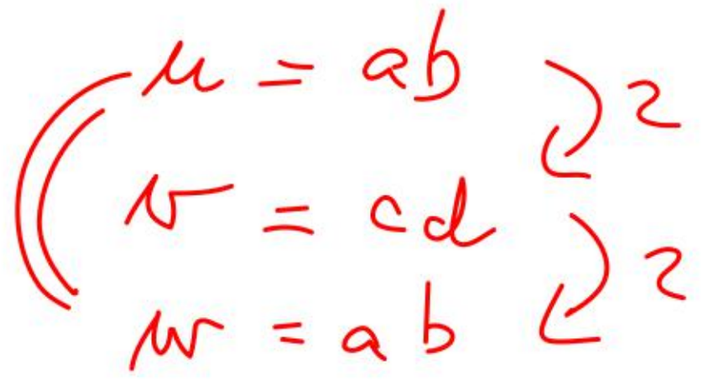
Př. 8/6: vlastnosti Levenshteinovy vzdálenosti

Označme symbolem $d(x, y)$ Levenshteinovu vzdálenost slov x a y. Víme že, pro tři slova u, v, w platí $d(u, v) = d_1$, $d(v, w) = d_2$. Jakých hodnot může nabývat $d(u, w)$ v závislosti na d_1, d_2? Abeceda je pro všechna slova společná.

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 $d(u, v) = d_1$, $d(v, w) = d_2$. Jakých hodnot může nabývat $d(u, w)$ v závislosti na d_1, d_2? Abeceda je pro všechna slova společná.





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 těchže slov. Rozhodněte, který z následujících případů může nastat a pro možné případy uveďte příklad slov v a w délky alespoň 5.

a) $HD(v, w) < LD(v, w)$,

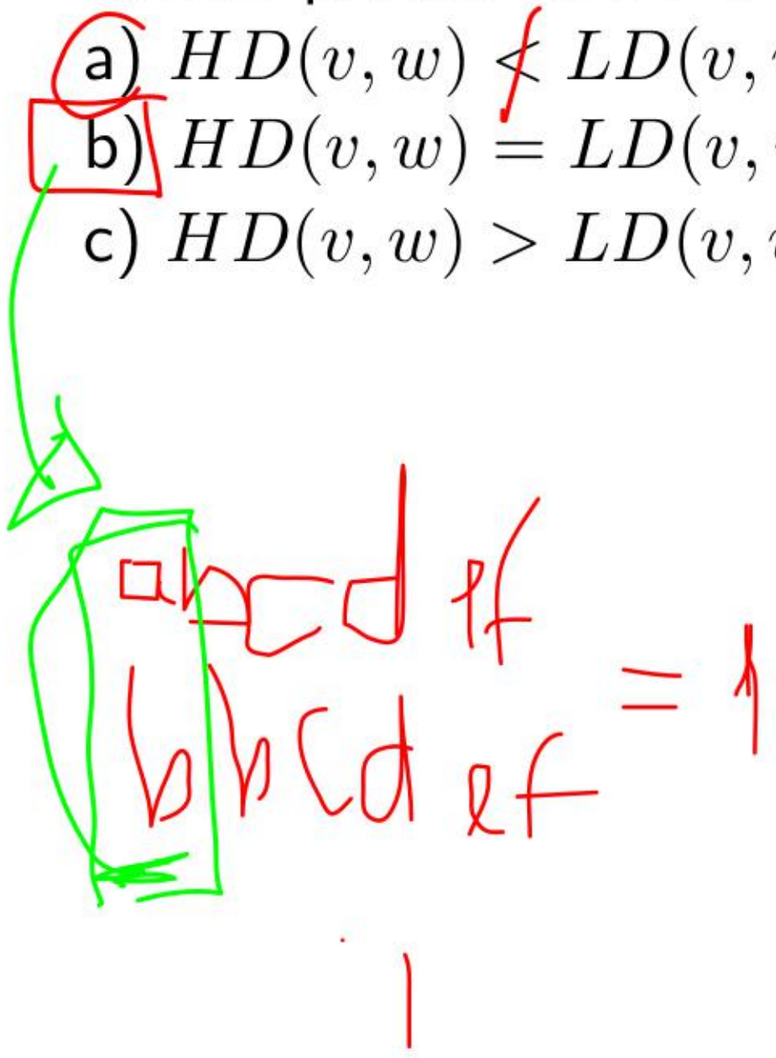
b) $HD(v, w) = LD(v, w)$,

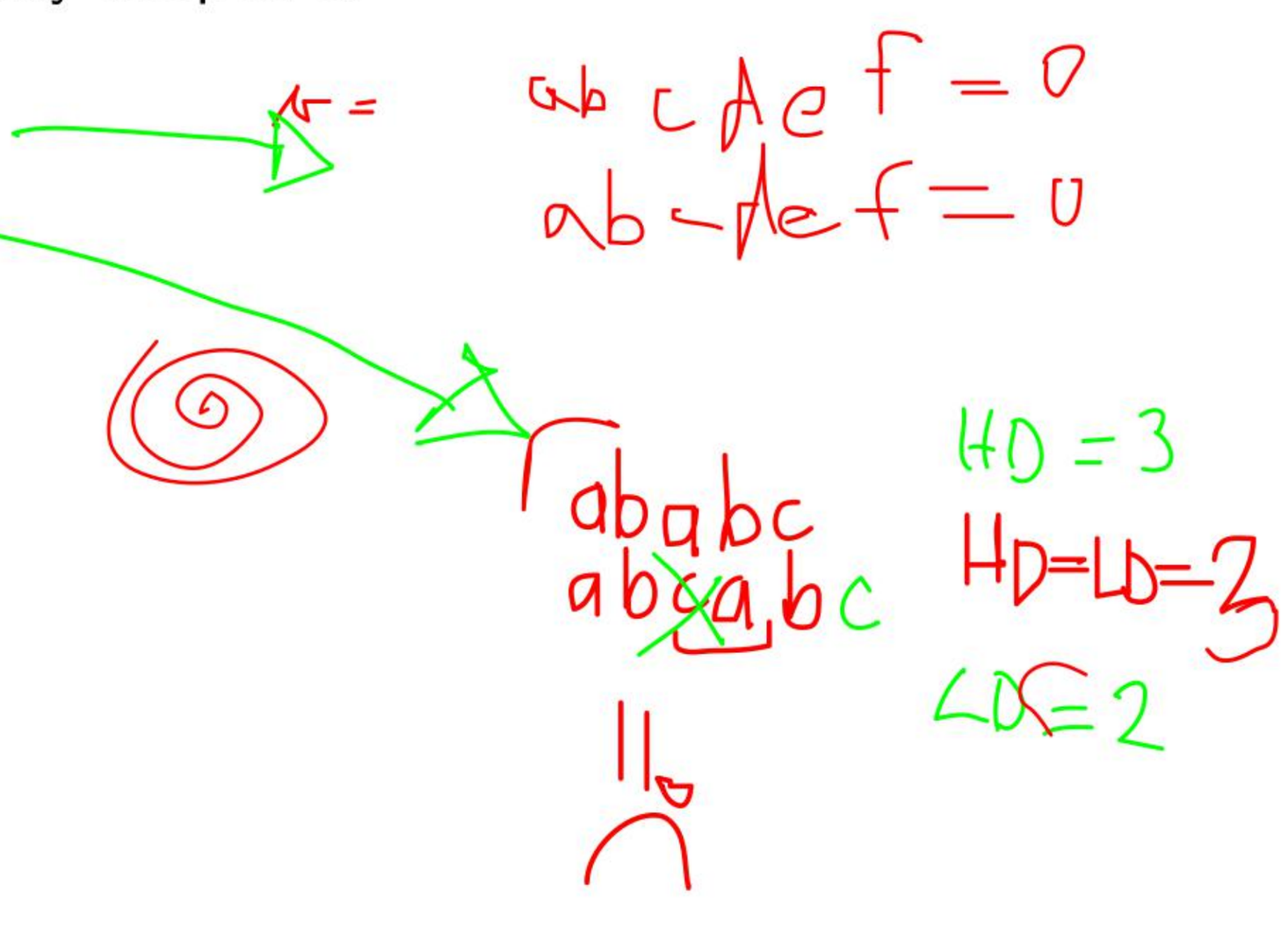
c) $HD(v, w) > LD(v, w)$.

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 těchže slov. Rozhodněte, který z následujících případů může nastat a pro možné případy uveďte příklad slov v a w délky alespoň 5.

a) $HD(v, w) \leq LD(v, w)$,
b) $HD(v, w) = LD(v, w)$,
c) $HD(v, w) > LD(v, w)$.



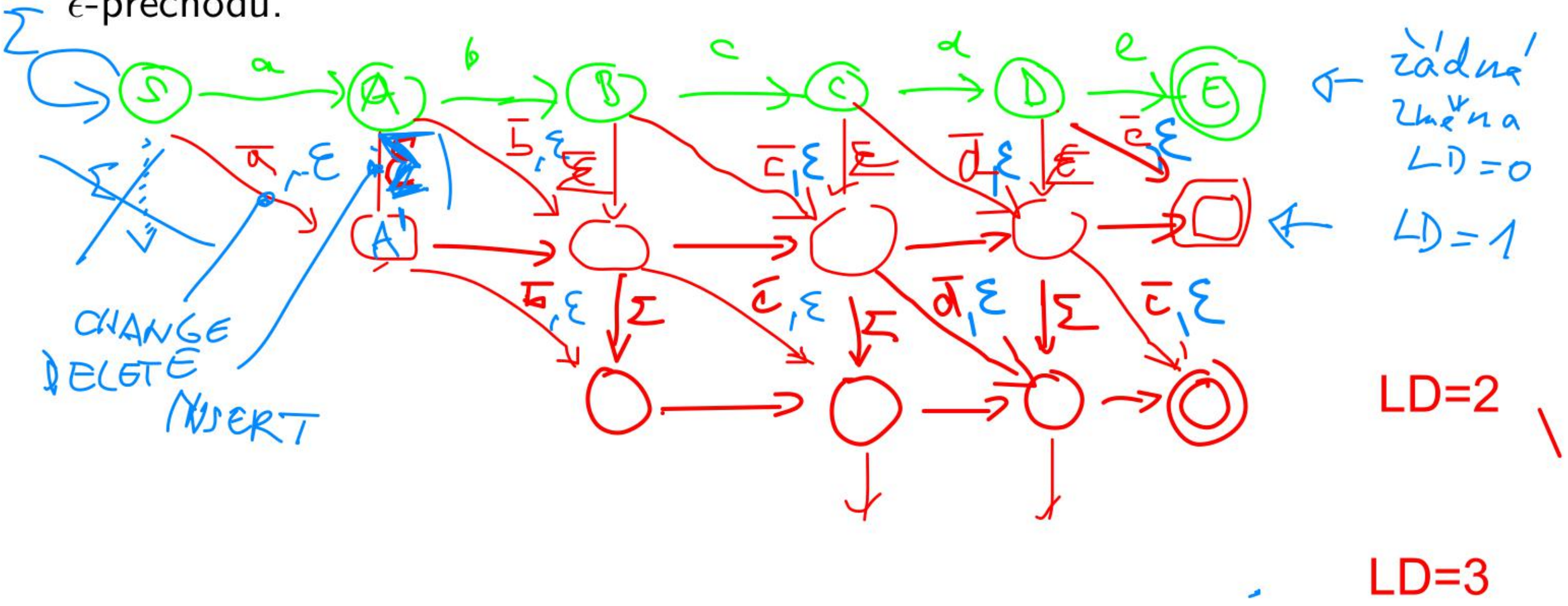


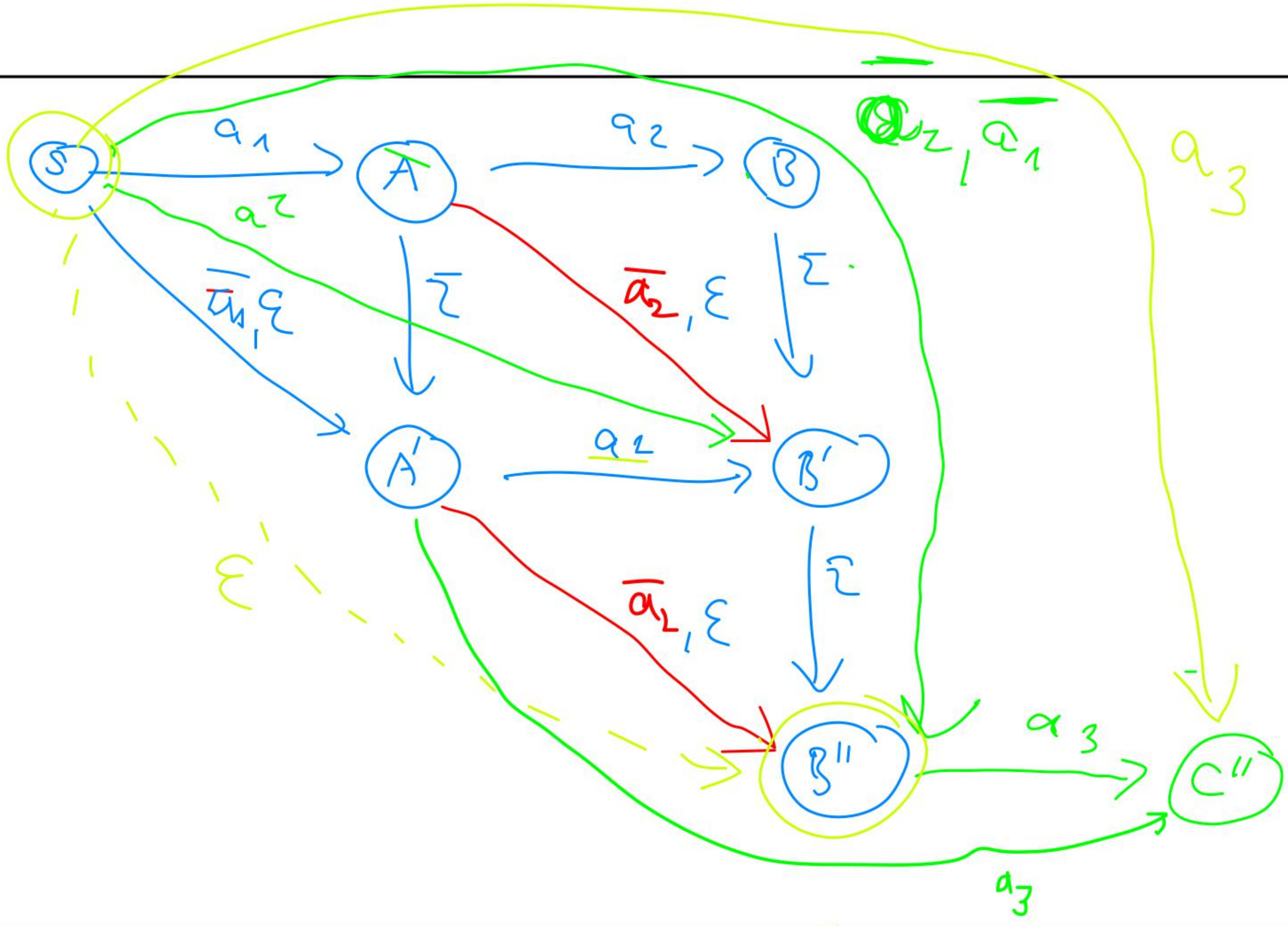
Př. 8/4: hledání Levenshteinovsky změněného slova

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. 8/4: hledání Levenshteinovsky změněného slova

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.



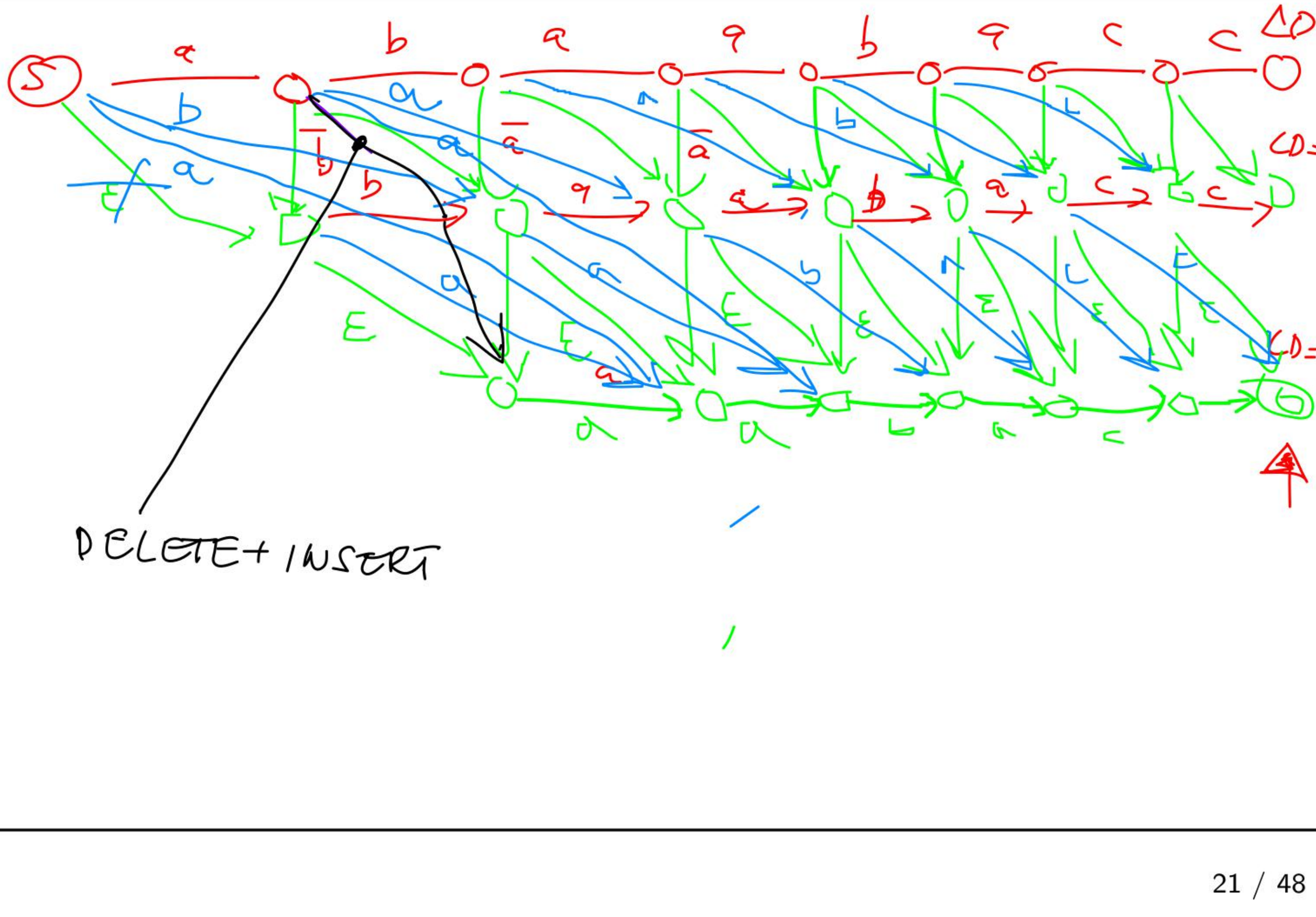

S
A
B
A'
B'
B''
C''
q_1
q_2
q_3
$\bar{q}_1$
$\bar{q}_2$
$\bar{a}_1, \varepsilon$
$\bar{a}_2, \varepsilon$
Σ
a_3

Př. 8/5: hledání Levenshteinovsky změněného slova 2

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. 8/5: hledání Levenshteinovsky změněného slova 2

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.

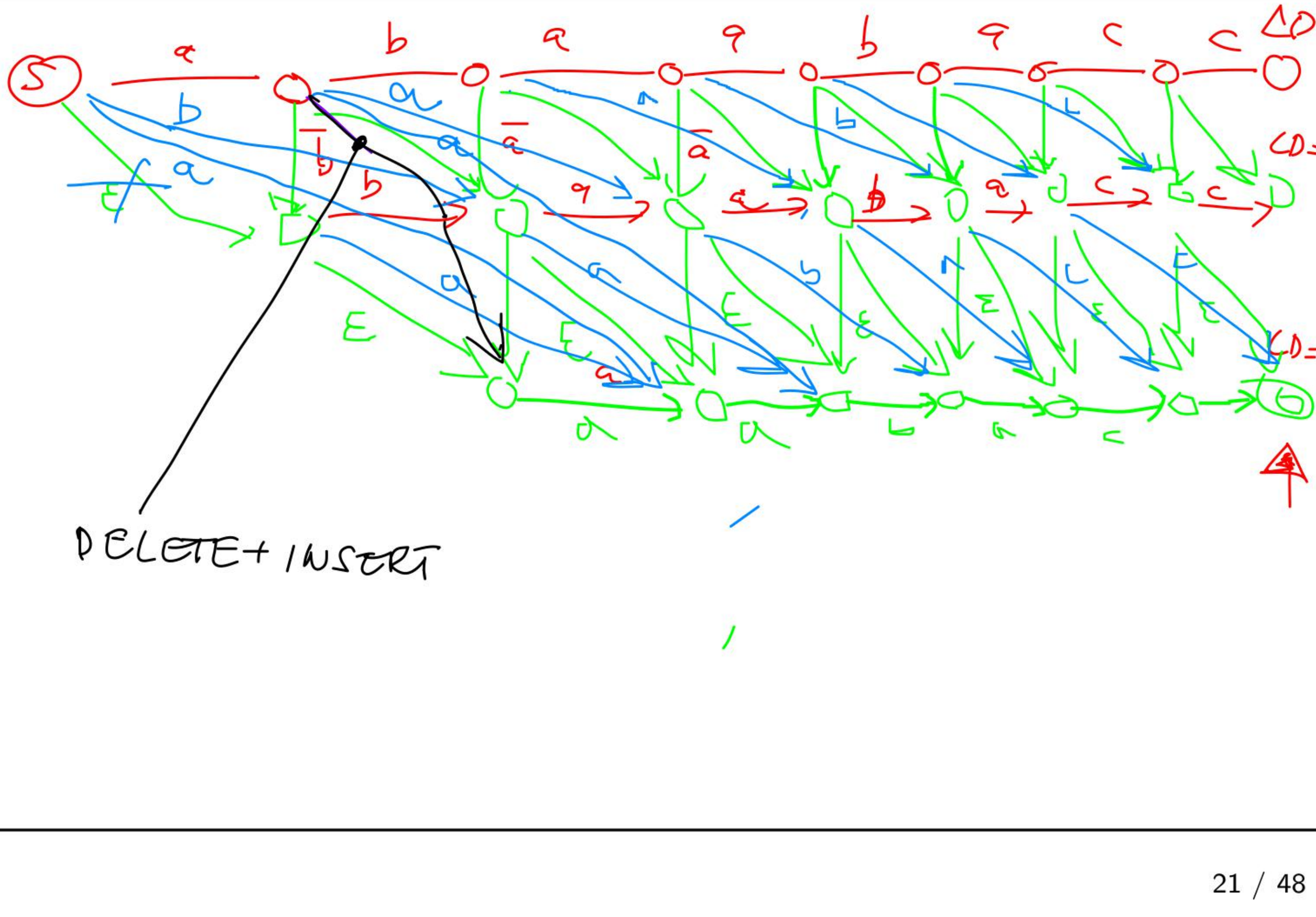

S
a
b
b
a
b
a
b
a
c
c
ED=0
ED=1
ED=2
DELETE+INSERT

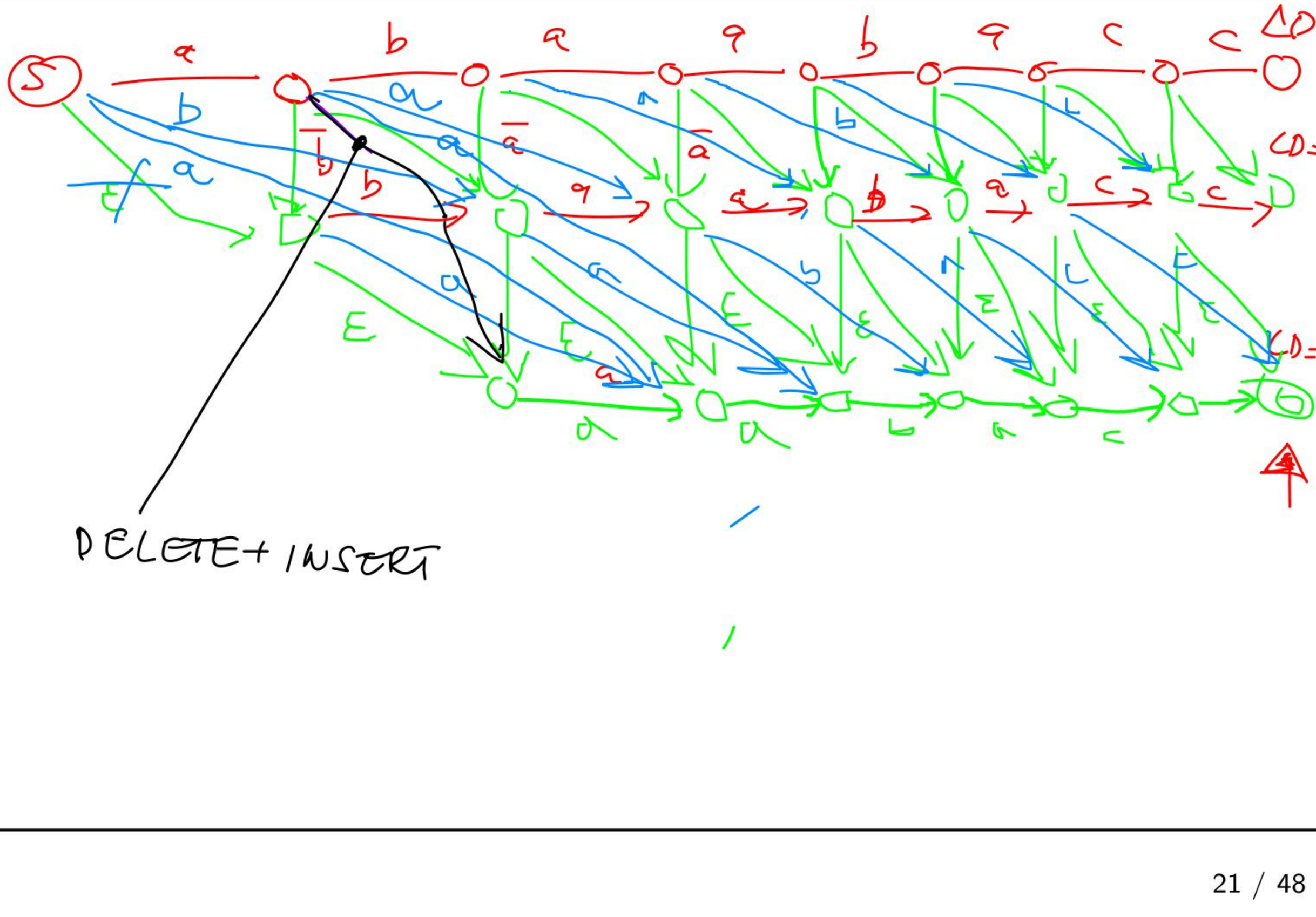
Př. 8/5: hledání Levenshteinovsky změněného slova 2

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. 8/5: hledání Levenshteinovsky změněného slova 2

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.


S
a
b
b
a
b
a
b
a
c
c
ED=0
ED=1
ED=2
DELETE+INSERT

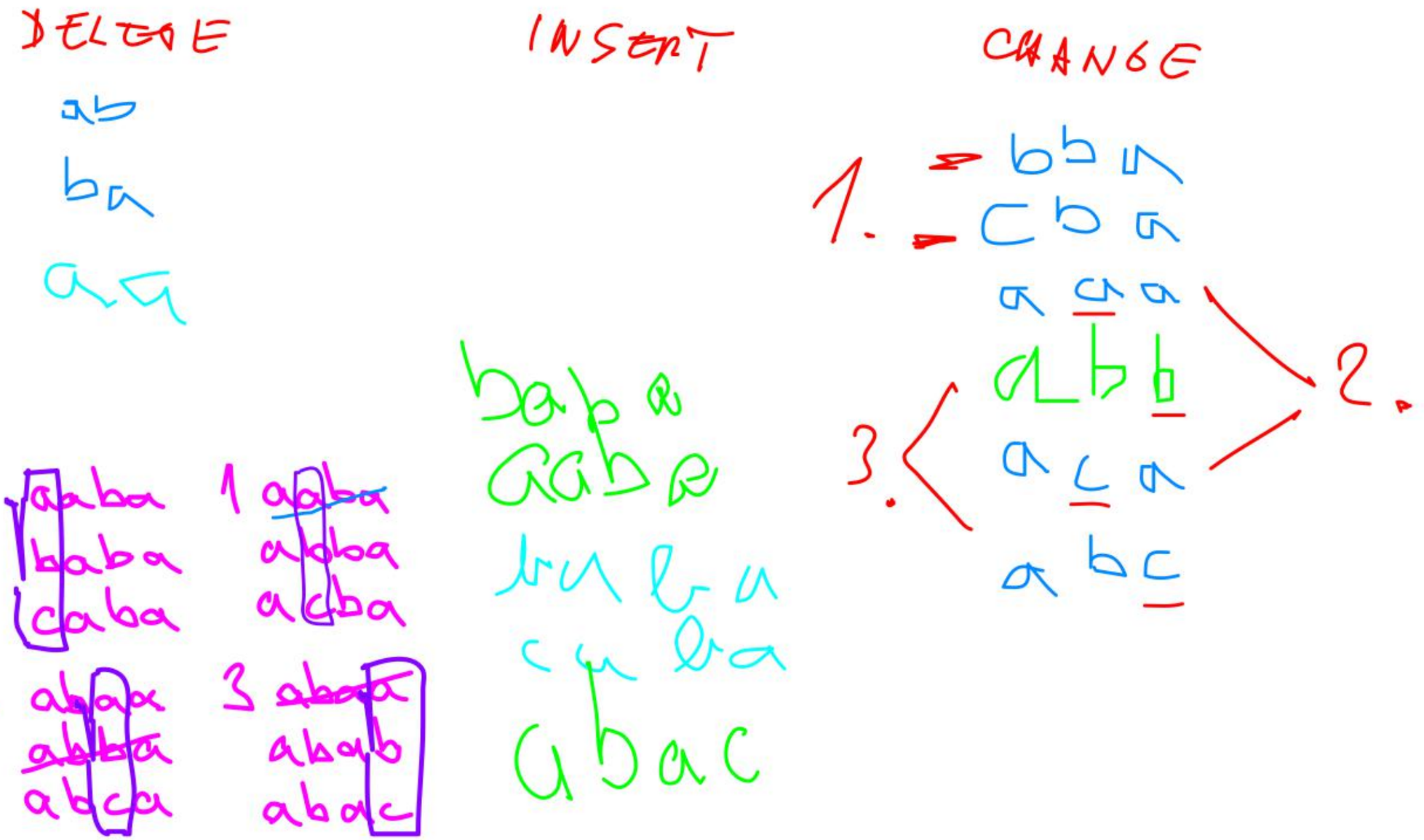

S
a
b
b
a
b
a
b
a
c
c
ED=0
ED=1
ED=2
DELETE+INSERT

Př. 8/9: Levenshtein

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

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 operaci $REWRITE$ (náhrada jednoho znaku jiným znakem abecedy)

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

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?

Slovníkové automaty. Implementace automatů.

Př. 9/5b: bitový paralelizmus

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) $T = accbbaaabcba$**,** $P = acbb$**,** $k = 2$.

Slovníkové automaty. Implementace automatů.

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?

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. 8/10: 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 operaci $REWRITE$ (náhrada jednoho znaku jiným znakem abecedy)

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

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?

Slovníkové automaty. Implementace automatů.

Př. 9/5b: bitový paralelizmus

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) $T = accbbaaabcba$**,** $P = acbb$**,** $k = 2$.

Př. 9/5b: bitový paralelizmus

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) $T = accbbaaabcba$**,** $P = acbb$**,** $k = 2$.

Př. 9/1a: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$, $P = 110010$, $k = 3$.

Př. 9/1b: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$**,** $P = 110010$**,** $k = 3$**.**

Př. 9/2a: Levenshteinovsky blízká slova - dynamicky

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) $T = 0100111010000101011011100$, $P = 11100$, $k = 1$.

Př. 9/1b: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$**,** $P = 110010$**,** $k = 3$**.**

Př. 9/1a: Hammingovsky blízká slova - dynamicky

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

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

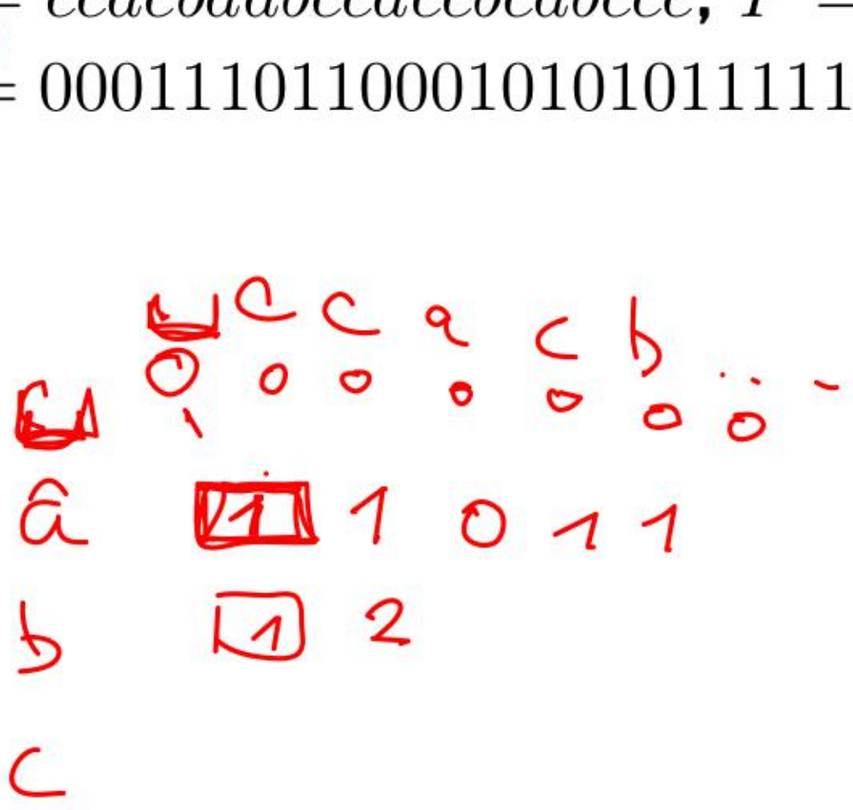
b) $T = 000111011000101010111110$, $P = 110010$, $k = 3$.

Př. 9/1a: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$, $P = 110010$, $k = 3$.



Př. 9/1b: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$**,** $P = 110010$**,** $k = 3$**.**

Př. 9/2a: Levenshteinovsky blízká slova - dynamicky

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) $T = 0100111010000101011011100$, $P = 11100$, $k = 1$.

Př. 9/1b: Hammingovsky blízká slova - dynamicky

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

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

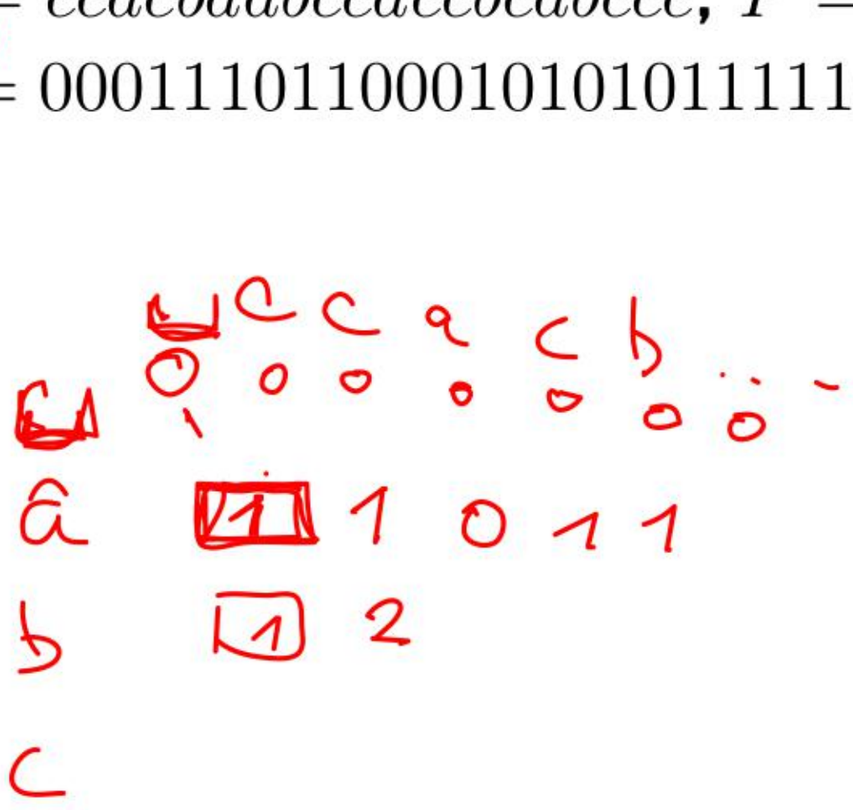
b) $T = 000111011000101010111110$**,** $P = 110010$**,** $k = 3$**.**

Př. 9/1a: Hammingovsky blízká slova - dynamicky

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) $T = 000111011000101010111110$, $P = 110010$, $k = 3$.



Př. 9/1a: Hammingovsky blízká slova - dynamicky

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) $T = 00011101100010101010111110$, $P = 110010$, $k = 3$.