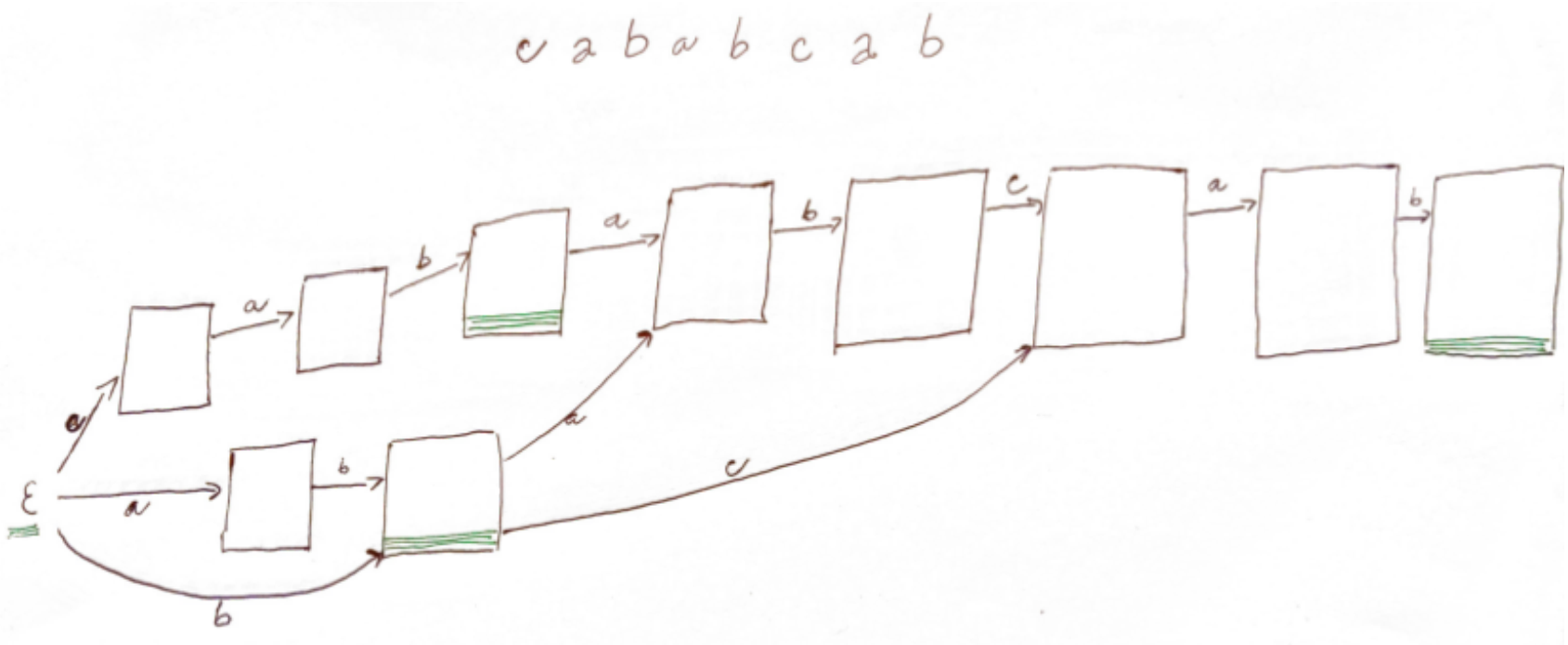
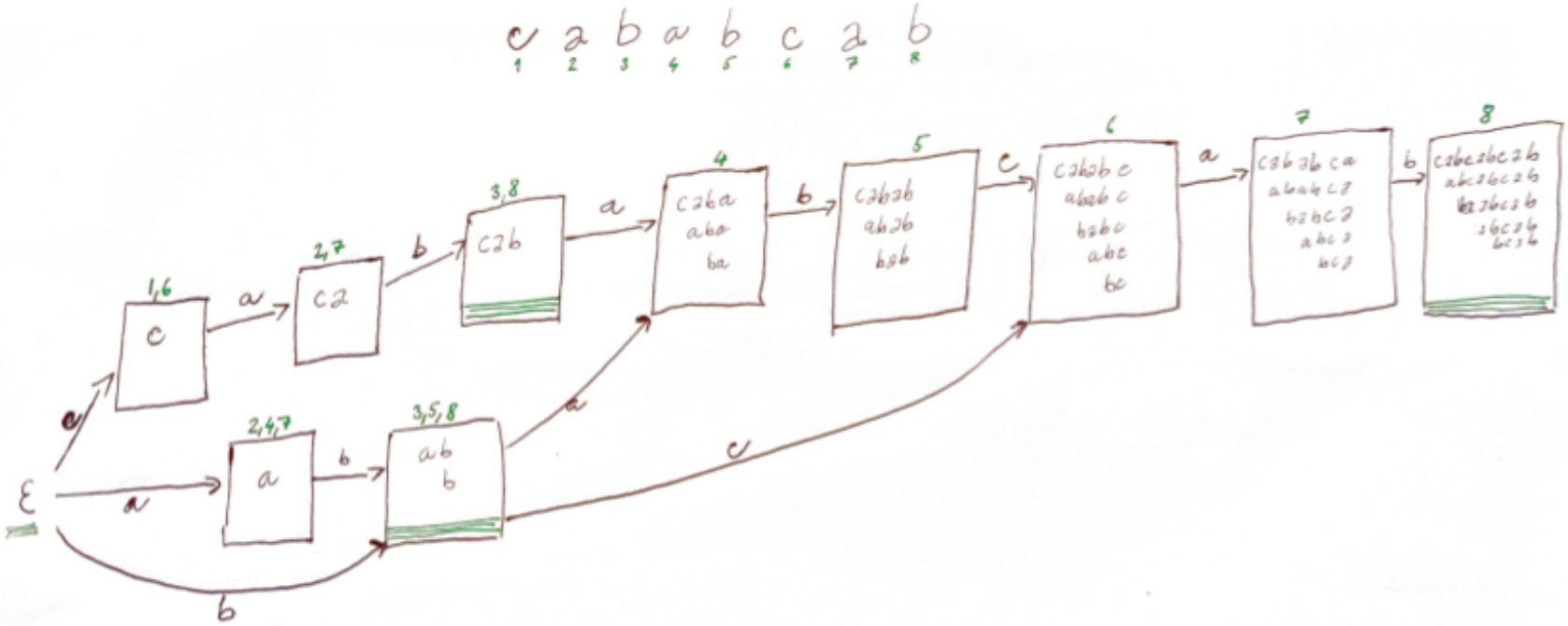
Suffixový automat

Definice

Nejmenší deterministický automat rozpoznávající suffixy vstupního textu.



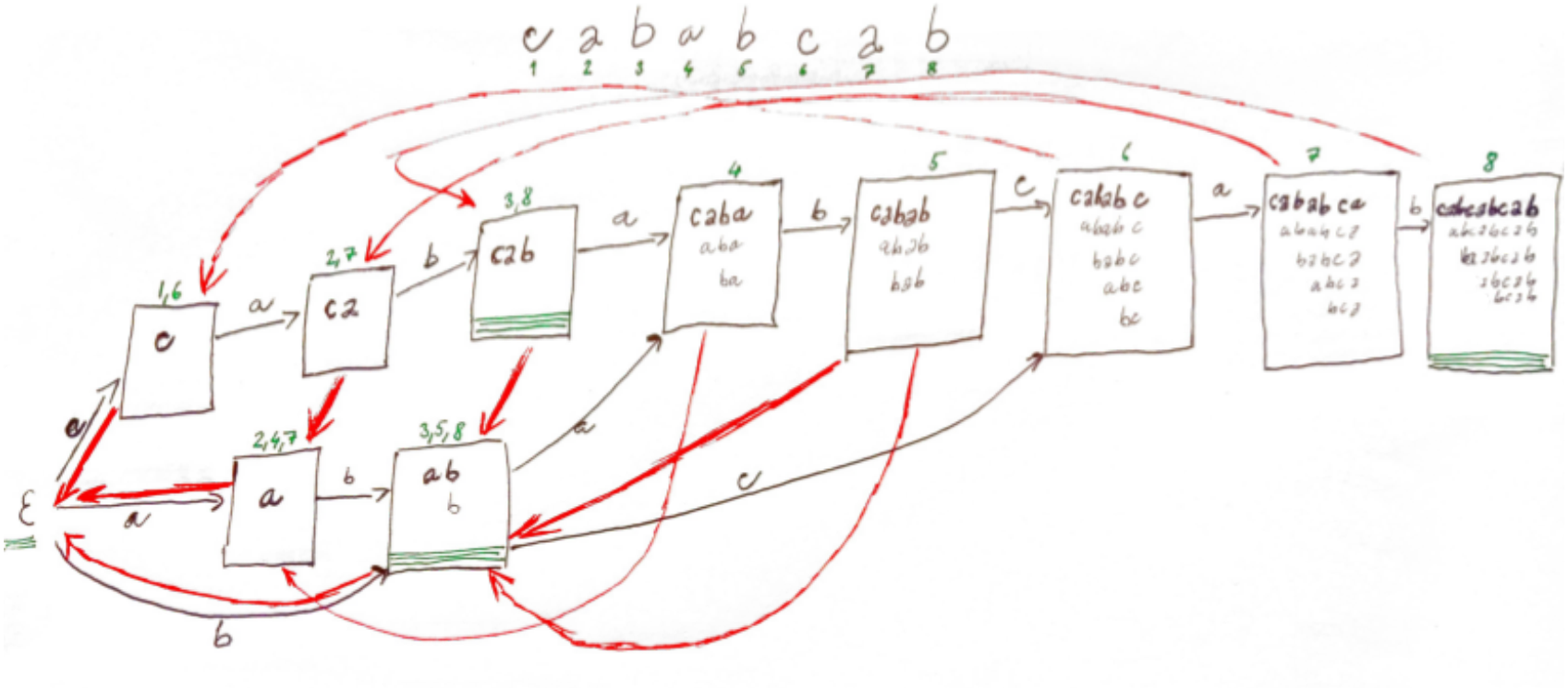
- $[x]$ = stav po načtení řetězce x.
- $P(x)$ = množina pozic, na nichž končí výskyt x v textu.



Pozorování

- $[x] = [y] \Leftrightarrow P(x) = P(y)$
- $[x] = [y] \Rightarrow$ *x je suffix y nebo naopak.*
- *Suffixový automat je acyklický.*

- Reprezentant stavu s = nejdelší x tž. $s = [x]$
- $p(s)$ = stav s' tž. $P(s) \subsetneq P(s')$ a $|P(s')|$ je minimální.

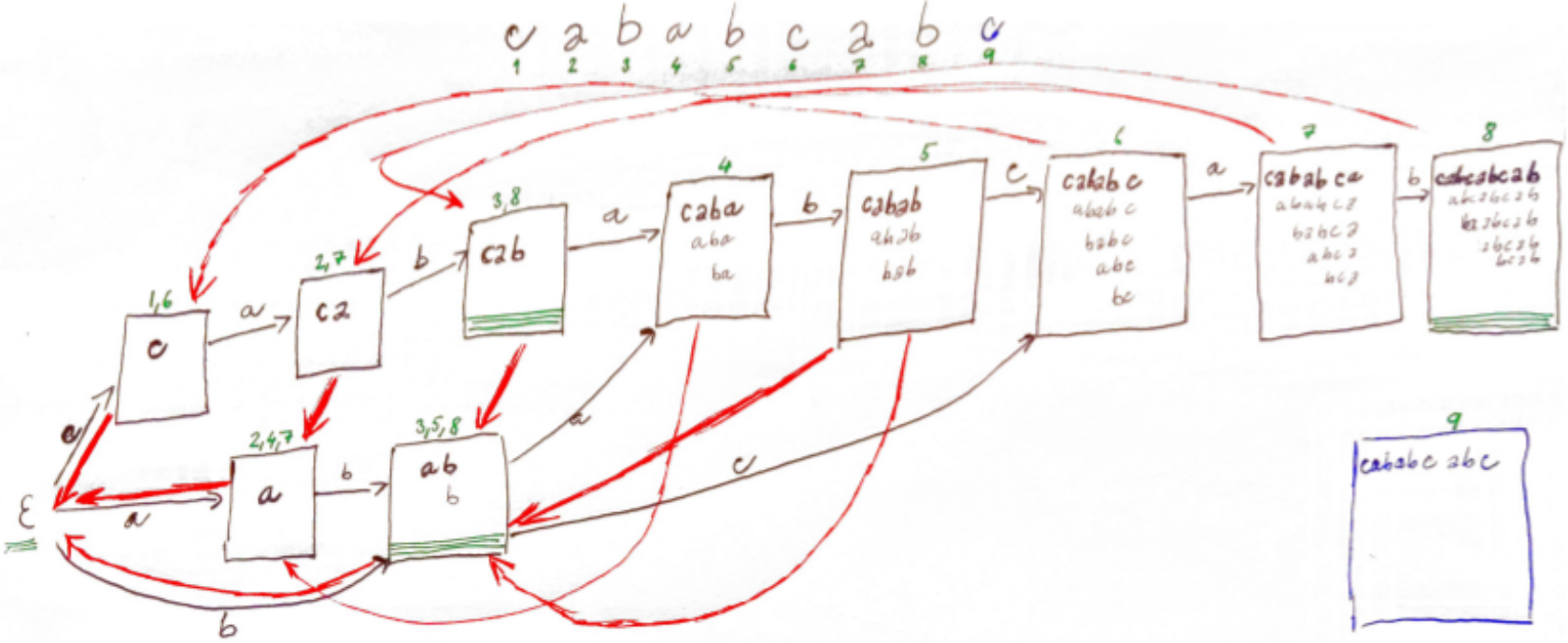


Pozorování

- *x reprezentant* $\Leftrightarrow$ *prefix nebo ho přechází různá písmena.*
- *x repr.:* $[x] = [y] \Leftrightarrow$ *y je suffix x delší než repr.* $p([x])$.
- *Přijímající stavy* $=$ [*text*], p[*text*], ...

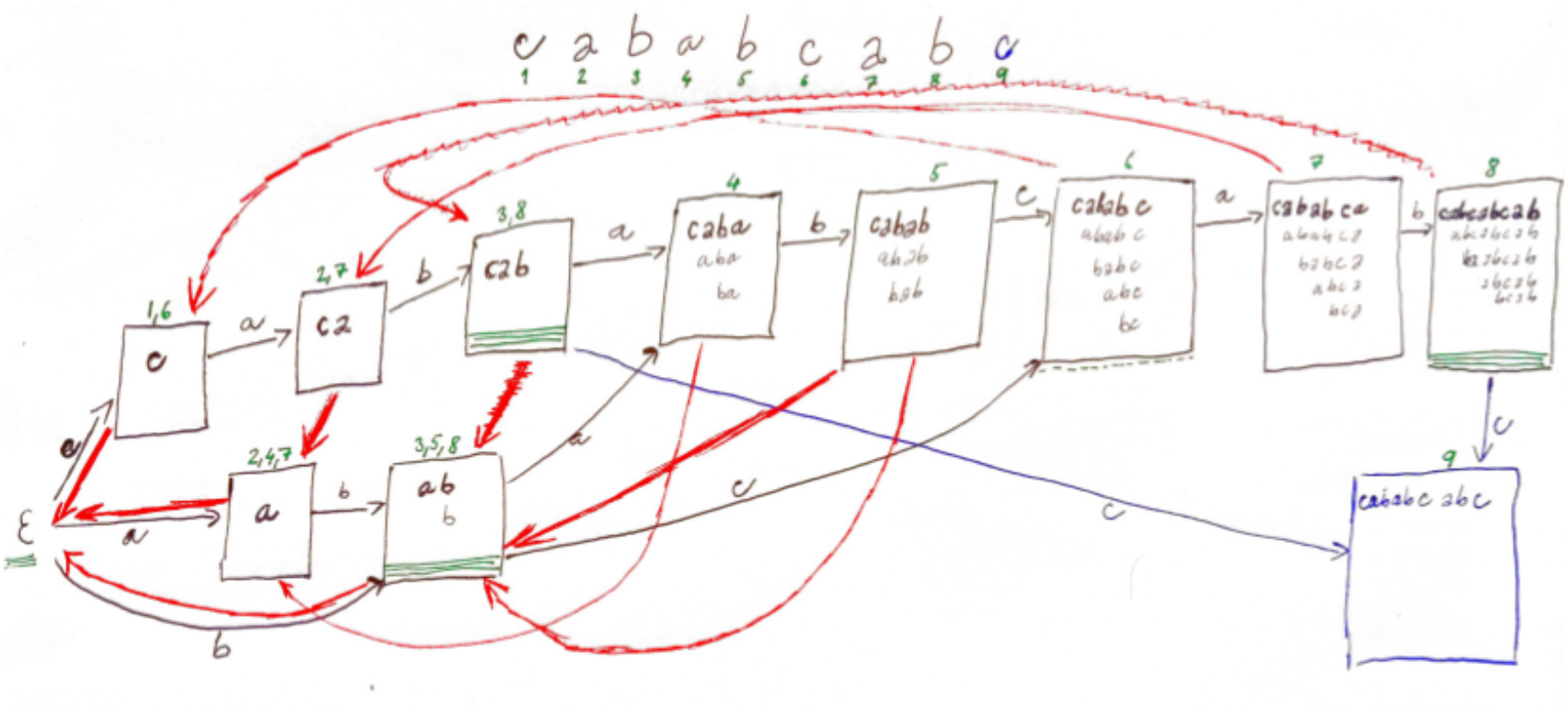
Konstrukce: Přidání písmene c na konec textu ω:

1. Vytvoř nový stav $[\omega c]$.
2. Přidávej c-přechody z $[\omega]$, $p([\omega])$, ... do $[\omega c]$, dokud nenarazíš na c-přechod, z $[\beta]$ do $[\beta c] = [\gamma]$.
3. Jestliže $\beta c = \gamma$, nastav $p([\omega c]) = [\gamma]$ a skonči.
4. Jinak vytvoř stav pro $[\beta c]$, nastav $p([\omega c]) = p(\gamma) = [\beta]$.
5. Zkopíruj přechody $[\gamma]$ do $[\beta c]$, pro $[\mu] = [\beta], [p(\beta)], \ldots,$ dokud existuje c-přechod z $[\mu]$ do $[\gamma]$, přesměruj ho do $[\beta c]$.



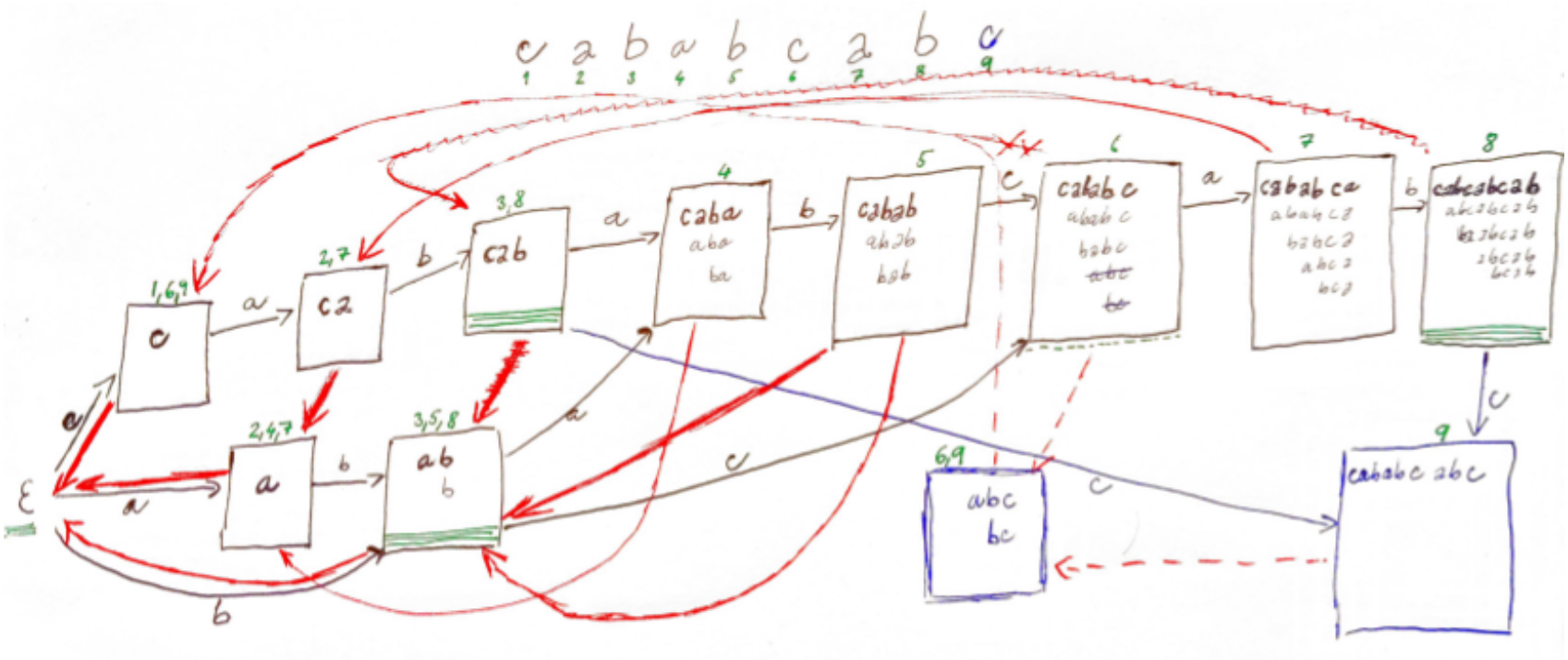
Konstrukce: Přidání písmene c na konec textu ω:

1. Vytvoř nový stav $[\omega c]$.
2. Přidávej c-přechody z $[\omega]$, $p([\omega])$, ... do $[\omega c]$, dokud nenarazíš na c-přechod, z $[\beta]$ do $[\beta c] = [\gamma]$.
3. Jestliže $\beta c = \gamma$, nastav $p([\omega c]) = [\gamma]$ a skonči.
4. Jinak vytvoř stav pro $[\beta c]$, nastav $p([\omega c]) = p(\gamma) = [\beta]$.
5. Zkopíruj přechody $[\gamma]$ do $[\beta c]$, pro $[\mu] = [\beta], [p(\beta)], \ldots,$ dokud existuje c-přechod z $[\mu]$ do $[\gamma]$, přesměruj ho do $[\beta c]$.



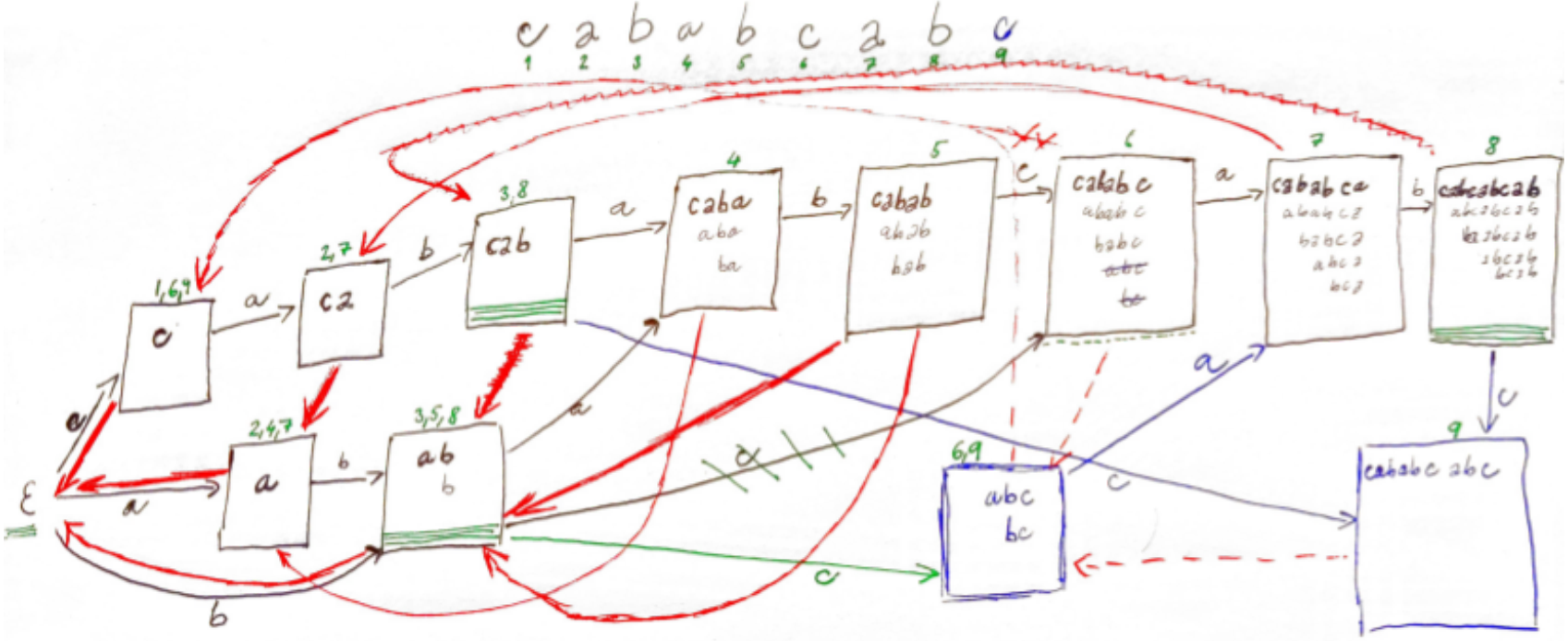
Konstrukce: Přidání písmene c na konec textu ω:

1. Vytvoř nový stav $[\omega c]$.
2. Přidávej c-přechody z $[\omega]$, $p([\omega])$, ... do $[\omega c]$, dokud nenarazíš na c-přechod, z $[\beta]$ do $[\beta c] = [\gamma]$.
3. Jestliže $\beta c = \gamma$, nastav $p([\omega c]) = [\gamma]$ a skonči.
4. Jinak vytvoř stav pro $[\beta c]$, nastav $p([\omega c]) = p(\gamma) = [\beta]$.
5. Zkopíruj přechody $[\gamma]$ do $[\beta c]$, pro $[\mu] = [\beta], [p(\beta)], \ldots,$ dokud existuje c-přechod z $[\mu]$ do $[\gamma]$, přesměruj ho do $[\beta c]$.



Konstrukce: Přidání písmene c na konec textu ω:

1. Vytvoř nový stav $[\omega c]$.
2. Přidávej c-přechody z $[\omega]$, $p([\omega])$, ... do $[\omega c]$, dokud nenarazíš na c-přechod, z $[\beta]$ do $[\beta c] = [\gamma]$.
3. Jestliže $\beta c = \gamma$, nastav $p([\omega c]) = [\gamma]$ a skonči.
4. Jinak vytvoř stav pro $[\beta c]$, nastav $p([\omega c]) = p(\gamma) = [\beta]$.
5. Zkopíruj přechody $[\gamma]$ do $[\beta c]$, pro $[\mu] = [\beta], [p(\beta)], \ldots$, dokud existuje c-přechod z $[\mu]$ do $[\gamma]$, přesměruj ho do $[\beta c]$.



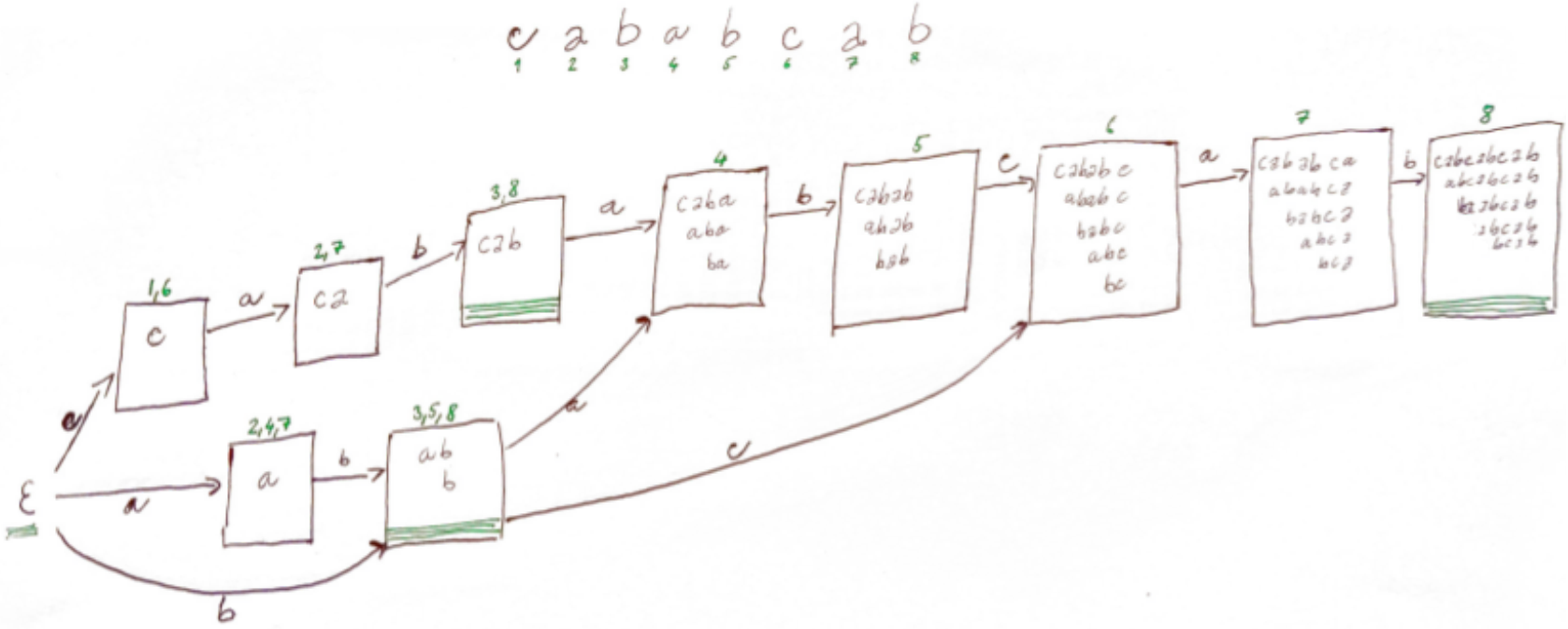
- Časová složitost $O(n)$.
- Relativně jednoduchý algoritmus ($\approx$ 50 řádek kódu).
- Implementace viz stránky praktika

  iuuk.mff.cuni.cz/$\sim$rakdver/index.php?which=uceni&subject=acm

  Můžete použít při řešení úloh.

Pozorování

Suffixový automat je acyklický a cesty z ε $\leftrightarrow$ navzájem různé podřetězce.

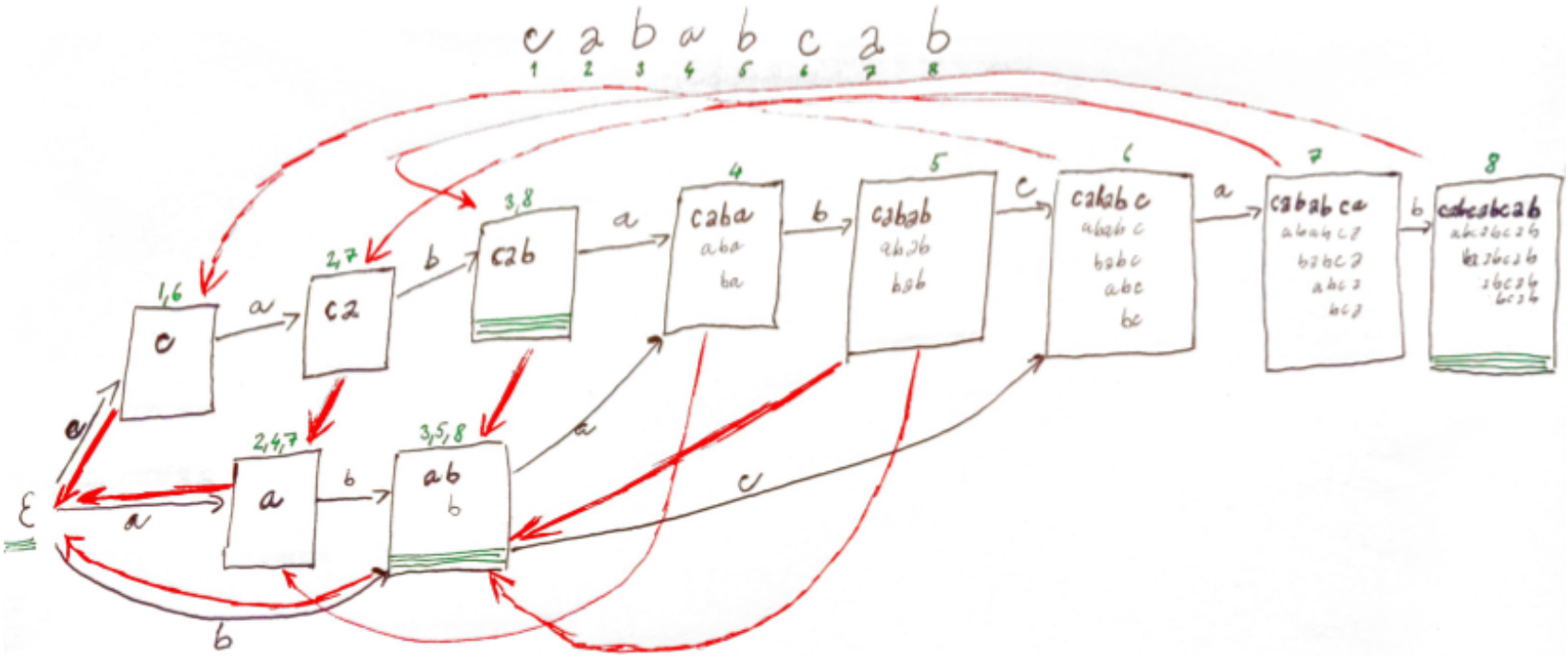


- Je x podřetězcem? Spustíme na x automat.
- Počet výskytů x? Počet cest z $[x]$ do přijímajících stavů (předpočítat dynamickým programováním).
- Počet různých podřetězců $=$ počet cest z ε (dynamické programování).

Pozorování

- *Pro prefix α textu je $|\alpha| \in P([\alpha])$.*
- *Pro libovolné β ostatní hodnoty patří do*

$$\bigcup_{p([\omega])=[\beta]} P([\omega]).$$



- Pozice na nichž se vyskytuje x? Průchod podstromem P zakořeněným v $[x]$.
- Stavy s výskyty končícími na pozicích v podmnožině Z lze předpočítat.
- Nejdelší společný podřetězec, počet společných podřetězců, ... textů t_1 a t_2:
  - Suffixový automat pro $t_1\$t_2$, označit stavy s konci v první a ve druhé části.