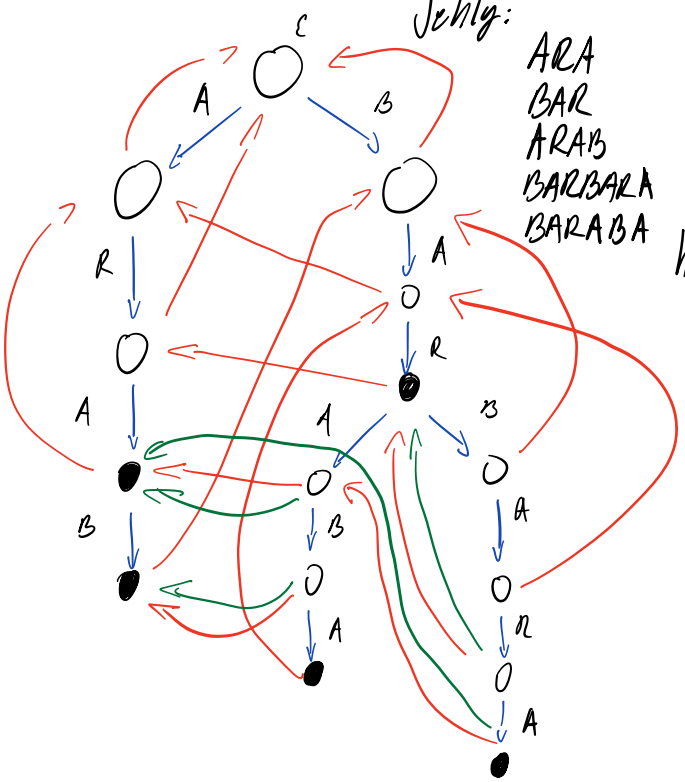
KMP - obecněji (Aho-Corasickí)



Jehly:
ARA
BAR
ARAB
BARBARA
BARABA

$i_1 \_\_ i_n$:= jehly
σ := seno

stavy := prefixy všech jehel

hrany: dopředná → prodloužení prefixu o jedno znaménko
zpětná → zkrácení prefixu na nejdelší vlastní sufix
zkratka → cesta po zpětných, jde na nejbližší koncový stav
- tedy určuje jehlu jíž obsahuje

Reprezentace automatu:

stavy očíslujeme, 0 = kořen (ε)

slovo (i) := které jehle koncí ve stavu "i"

zpět(i) := kam vede zpětná hrana
zkratka (i) := kam vede zkratková hrana
} Vždy maximálně jedna od každé na jednom poli.

dopředu (i, x) := kam vede dopředná hrana pro písmeno x

Krok (s, x):

→ To je, že pro zadané x xx už není součástí řetězce

Dokud dopředu (s, x) = ∅: → pokud už není kam dál jít, chci skákat zpět

Dokud s = kořen: vrátíme s. → pokud jsem již v kořeni

s = zpět(s) → jinak jdu po zpětných

Vrátíme dopředu (s, x) → už existuje příslušná dopředná hrana

Hledej (σ):

s = kořen
Pro j = 0 __ |σ| - 1:

s = Krok (s, σ[j])
t = s — temp. stav
Dokud t ≠ ∅:
Jeli slovo (t) ≠ ∅:
Hlásíme výskyt...
t = zkratka (t) → tohle funguje jen dokud je def.

Lemma: Hledej běží v čase $O(|\sigma| + \#\text{výskytů})$

dopředných je nejvýše tolik co |σ|
zpětných je nejvýše tolik co dopředných
tedy $O(|\sigma|)$

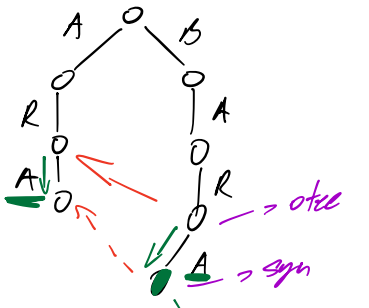
výskytů je tolik, kolikrát se ohlásí.
S tím, že poprvé se může projít symbolem bez ohlášení...

Konstrukce automatu:

Myšlenka: Stejně jako u AMP, nyní však budu počítat všechny jehly paralelně po jednom písmenu a strom budu budovat po instrucích.

Konstrukce $(i_0 \dots i_n)$:

1) Vybudujeme trii pro $i_0 \dots i_n$ -> z toho máme dopředu
2) Zpětná(kořen) = $\emptyset$, Zkratka(kořen) = $\emptyset$
3) Zpětná(∀ syny kořene) = kořen, Zkratka(∀ syny kořene) = $\emptyset$
4) F = fronta se syny kořene:
5) Dokud $F \neq \emptyset$:
6) $\quad v$ = dequeue(F)
7) $\quad$ Pro s syny v:
8) $\quad\quad$ Zpět(s) = Uzel(zpět(v), písmeno na hraně vs)
9) $\quad\quad$ F.Add(s)
10) $\quad\quad$ Dokud Slovo(zpět(s)) $\neq \emptyset$:
11) $\quad\quad\quad$ Zkratka(s) = zpět(s)
12) $\quad\quad$ Jinak Zkratka(s) = Zkratka(zpět(s)).



pokud hledám zpětnou po synu, jdu po zpětné otce a tam přečtu po stejném písmenu, co vede z otce do syna

-> Využití faktu, že se chodí po zpětných a předešlé již máme nalezené a aspoň částečně

Lemma: Konstrukce běží v čase $O(\sum i_i)$

Bez úvodu je vybudování trie asympt. lineární. Zbytek je BFS.

Jehly procházíme paralelně a průchod jednou jehlou je lineární.

A tedy i celý paralelní přístup je lineární. -> Dokonce často stav počítám jen jednou a u dalších jehel ho nemusím počítat.

Věta: Nalezení všech výskytů jehel v seně trvá $O(\sum i_i + |S| + \#\text{výskytů})$.

Důkazy jsou spojení lemmat.

Robinův - Karpův alg.

$h(x_0 \dots x_{J-1}) \to \mathbb{Z}_H$

↳ musím umět přepočítat hash v konstantním čase po posunutí okénka jinak bych nie nemusel tvořit!

$\to h(x_0 \dots x_{J-1}) :=$

$(x_0 r^{J-1} + x_1 r^{J-2} + \dots + x_{J-1} r^0) \bmod H$

posunutí okénka

$h(x_1 \dots x_J) =$

$(x_1 r^{J-1} + x_2 r^{J-2} + \dots + x_J r^0) \bmod H$

Rozdíl?

x_0 zmizelo, x_J přibylo

ostatní se jen vynásobilo r-krát

Řešení:

$h(x_1 \dots x_J) = h(x_0 \dots x_{J-1}) \cdot r - x_0 r^J + x_J$

A tahle je konstantní čas $O(1)$

RK alg.

0) zvolíme r z tělesa náhodně
1) $c = h(\text{jehly})$, $a = h(\sigma[:J])$, spočítám r^J
2) Pro $i = 0, \dots, |\sigma| - J$:
3) Pokud $a = c$ and $\sigma[i:i+J] = i$
4) Hlásím výskyt jehly i na pozici i.
5) $a = (a \cdot r - \sigma[i] \cdot r^J + \sigma[i+J]) \bmod H$.

Pozor: skutečný výskyt je hodně daleko. Vhodné pro nemlčení jehel.

Složitost:

- režie + počítání hashů $:= O(|\sigma|)$
- skutečné výskyty $= O(J \cdot v)$ (v = výskyty)
- falešné výskyty =

↳ Pro jeden (celkem náhodný) h,

$P(\text{falešného výskytu}) = \frac{1}{H}$

→ průměrný čas $= O\left(\frac{|\sigma| \cdot J}{H}\right)$

$\frac{J}{H}$ pro jedno okénko. Celkem $|\sigma|$ okének.

Pokud $H \geq J$: $= O(|\sigma|)$

Mějme $P(x) := p_0 x^0 + p_1 x^1 + \dots + p_{n-1} x^{n-1}$ nad tělesem

Lemma: Pokud $x_1 \dots x_k$ jsou všechny kořeny polynomu P, pak $P(x) = (x - x_1) \cdot (x - x_2) \cdot \dots \cdot (x - x_k) \cdot Q(x)$

kde $Q(x)$ je polynom bez kořenů.

Důsledek: Polynom stupně d má nejvýš d kořenů

Kdyby k bylo větší než stupeň P, vyšel by spor.

Lemma: Nechť P a Q jsou polynomy stupně $< n$, $x_1 \dots x_n$ navzájem různá čísla t.ž. $\forall i: P(x_i) = Q(x_i)$.
Pak $P \equiv Q$.

$R := P - Q$, stupeň opět $< n$ t.ž. $\forall i: R(x_i) = 0$ → Jediný takový polynom je nulový polynom.

$R \equiv 0 \Rightarrow P \equiv Q$

Tedy nyní falešné výskyty dopadnou:

$P(r) = h(i)$
$Q(r) = h(\sigma[i : i+J])$

– Ptám se, že pokud jsou polynomy různé tak pro kterékoliv r jaká je pravděpodobnost, že se budou rovnat.

Tedy $P[P(r) \equiv Q(r)] \leq \frac{J}{H}$ ——> $\frac{1}{H}$ opět z velikosti hash. prostoru, $\leq J$ - kvůli, protože to je takové pozice, kde se dva polynomy mohou rovnat, aby stále nebyly identické.

Tedy H by muselo být alespoň J^2 velké.

Celkový časová složitost je tedy $O\left(\frac{|T| \cdot J^2}{H}\right)$ -- tedy pro $H \geq J^2$ je to $O(|T|)$

$O(|T| \cdot J)$:= ověření jehly v seně

$\cdot O\left(\frac{J}{H}\right)$ počet falešných výskytů (kromě toho správného).