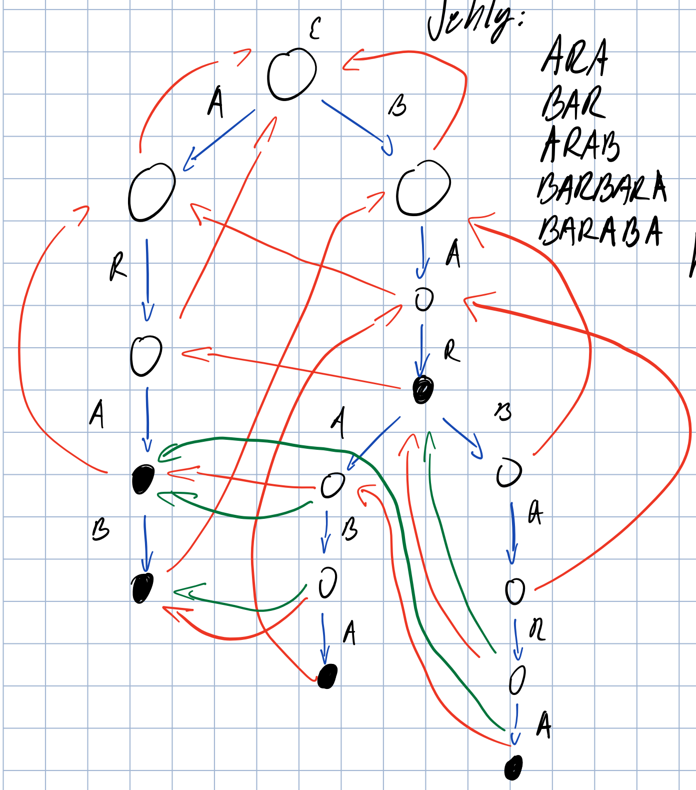
KMP - obecněji (Aho-Corasická)



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 suffix
- zkratka → cesta po zpětných, jde na nejbližší koncový stav – tedy kterou jehlu již obsahuje

Reprezentace automatu:

stavy očíslujeme, 0 = kořen (ε)

slovo(i) := která jehla končí ve stavu „i"

zpět(i) := kam vede zpětná hrana

zkratka(i) := kam vede zkratková hrana

Vždy maximálně jedna od každé na jednom vrcholu.

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, musí skákat zpět

Pokud 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 i = 0 … $|\sigma|-1$:

s = Krok(s, σ[i])

t = s — temp. stav

Dokud t ≠ ∅:

Je-li slovo(t) ≠ ∅:

Hlásíme výskyt…

t = zkratka(t) → tohle funguje jen pokud je def.

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

\#dopředných je nejvýše tolik co $|\sigma|$

\#zpětných je nejvýše tolik co dopředných

tedy $O(|\sigma|)$

\#výskytů je tolik, kolikrát se ohlásí.

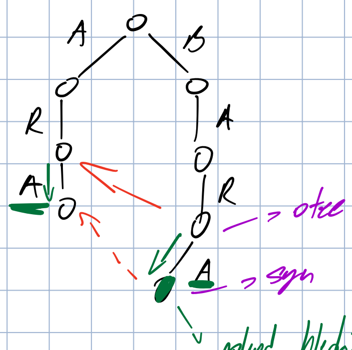
S tím, že popř. se může projít smyčkou bez ohlášení…

Konstrukce automatu:

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

Konstrukce ($i_1 - i_n$):

1) Vybudujeme trii pro $i_1 - i_n$ → z toho máme dopředné
2) Zpětná(kořen) = ∅, Zkratka(kořen) = ∅
3) Zpětná(∀ syn kořene) = kořen, Zkratka(∀ syn kořene) = ∅
4) F = fronta se syny kořene:
5) Dokud F ≠ ∅:
6) v = dequeue(F)
7) Pro s syny v:
8) Zpět(s) = Krok(zpět(v), písmeno na hraně vs)
9) F.Add(s)
10) Dokud Slovo(zpět(s)) ≠ ∅:
11) Zkratka(s) = zpět(s)
12) Jinak Zkratka(s) = Zkratka(zpět(s)).



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

→ Využití faktu, že se dodá po zpětných a předešlé již mám nalezené a pospojované

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

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

Jehly procházíme paralelně a průchod jednou jehlou je lineární.

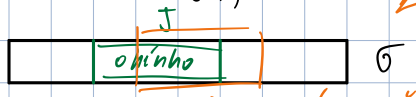
A tedy i celý paralelní přístup je lineární. → Dokonce často stav počítám jen jednou a u dalších jehel ho nemusím počítat.

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

Důkazy jsou spojení lemmat.

Rabinův - Karpův alg.

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



musím umět přepočítat hash v konstantním čase po posunutí okénka

jinak bych nic neušetřil!

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

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

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-kem

Řešení:

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

A tohle 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 $\sigma[i:i+J] = j$
4) Hlásím výskyt jehly j na pozici i.
5) $a = (a \cdot r - \sigma[i] \cdot r^J + \sigma[i+J]) \bmod H$.

Pozor: skutečný výskyt je hodně drahý. Vhodné pro hledání jehel.

Složitost:

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

↳ Pro ideální (dokonale náhodnou) h,

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

$\Rightarrow$ 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 $P(x) = (x - x_1) \cdot (x - x_2) \cdot \ldots \cdot (x - x_k) \cdot Q(x)$ 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 $< n$, $x_1 — x_n$ 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 doopravdy:

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

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

Tedy $P[\,P(r) = Q(r)\,] \leq \frac{J}{H}$ → $\frac{1}{H}$ opět z velikosti hash. prostoru, $\cdot$ J-krát, protože to je tolik pozic, kde se dva polynomy mohou rovnat, aby stále nebyly identické.

Tedy H by muselo být alespoň J^2 velké.

Celková časová složitost je tedy $O\left(\frac{|\sigma| \cdot J^2}{H}\right)$ → tedy pro $H \geq J^2$ je to $O(|\sigma|)$

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

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