

Reper matricas

- ad vepuri & automahtinu, algebrin formu, slozifiki at HW ar korpereivosh.

- Uzrasine saklodu DV : priiter zapisane (zvachy (a,b,A))  
 operatory +  $1 \cdot 0^*$  a vachny -  
 $\uparrow$   $\swarrow$   $\nwarrow$   
 dazice vachabavos izenage

- Tot shchitine posomol izla dany itelide u poln'ide jizgla  $L(n)$  reprezentiravetse DV n?

- Tipnizy 1. lens: prierevone v na automal (KA)

- Jedna vazivost, jib p'irost v na KA p' zlostrukural vajivine DCA (nie p'edvashy).

- Tot velky DCA vavilko z DV v  $1 \cdot p \cdot k$   $|n|$  dazic potem mudi v  $r$  p'irinev. d'ika jzha itelide  $(|a,b| = 3)$ .

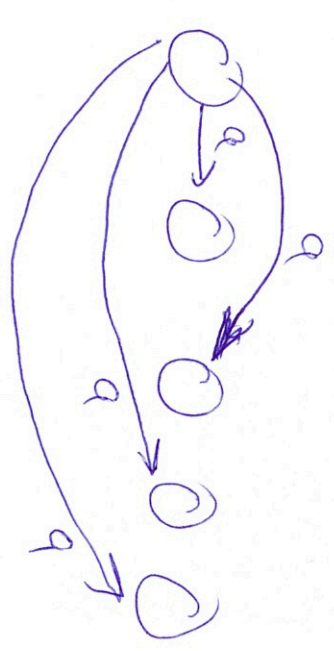
- kaziki operace komsulete DCA p'ide komsuldu p'evet stavu / m'edvodi & komsul v'povodent

DFA pro pokračování: velikost DFA pro  
 slovíčko  $uv$  je  $uv$  stavů.  $Tedy$   
 stavů DFA, který má  $O(|uv|)$  stavů  
 a  $O(|uv|)$  přechodů.

- regex malý,  $\gamma$  možná menší, přitom nad DFA s využitím backtrackingu a "přeskalování"  $\epsilon$  přechodů — neděle se příliš nepohybují!
- Obvykle se odstraňují složení  $\epsilon$ -přechodů. Dej se realizovat bez nutnosti dělat udívací
- reorganizování  $\alpha$ -přechodů (pro  $\alpha \in \Sigma$ ) do usídli stavů, ab-glance se lze dostat po přímém přechodu přes  $\epsilon$  přechody:



$\rightsquigarrow$



Matic je zapotřebí analogicky  
 upravit  $\epsilon$  přechody z  $q_0$ .



- Jaký je dopad výše uvedené konstrukce na počet stavů a přechodů?

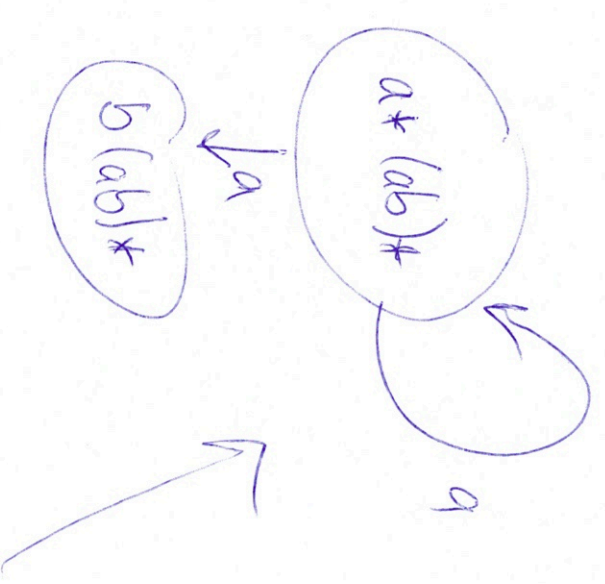
- stavů - více dvojic:  $O(n^2)$ .
- počet přechodů více dramaticky narůstá - lze označit  $O(n^3)$

Vědomí rychlosti stavů x vědomí o čase stavů  $O(n^2)$

- Existují ale i jiné možnosti přechodů - např. různé typy dělení DV, jít např.

Antiunitární dělení:

- Považujeme koncept "particiální dělení" DV do symbolů a abecedy.
- Bere, der, rep. myšlenka - uvažujeme  $a, 1, \bar{a}(r)$  - množina DV, abeceda reprezentující "symbol" řešení popsaných  $r$  pro celou  $a$ .
- např.  $\bar{a}(a^* (ab)^*) = \sum a^* (ab)^* | b(ab)^* |$



- výsledné prvky vzájemně jsou
- Stejný stav ovlivňuje
- Výsledné NFA budou mít:
  - $O(n)$  stavů, kde  $n$  je počet ústředních
  - zobrazí se  $\Sigma^n$  - dlouhý DV r
  - (tedy  $n \leq r$ ) - pro  $r = a^+(ab)^*$
  - $O(n^2)$  přechodů - pro  $r = a^+(ab)^*$
  - pro  $n = 3$ .

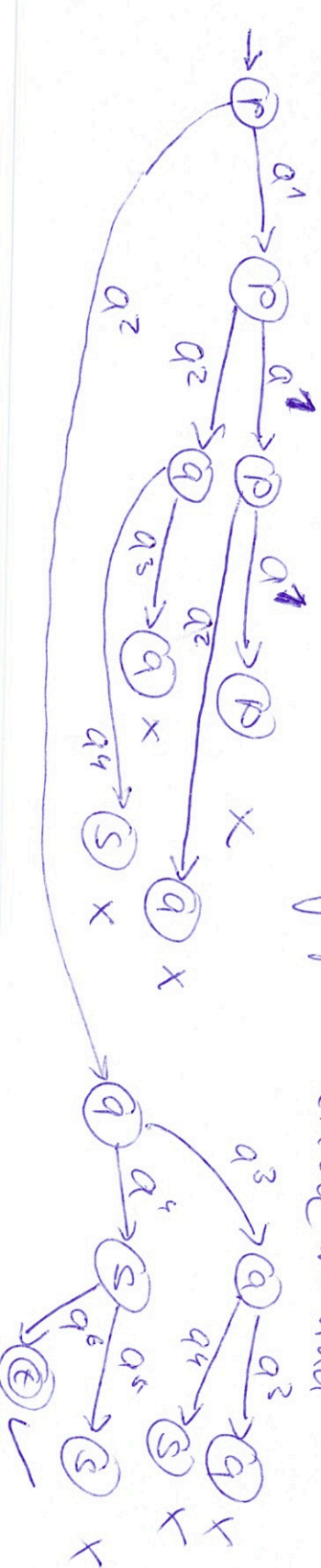
- Nad NFA lze provést regeer uvolnění pomocí 'backtrackingu'.

- Jaka bude složitost?
- Uvažme např. DV  $r = (a+ab)^*a+(a+ab)^*$
- NFA pro  $r$  může vypadat takto:



označení!   
 pomocná funkce   
 pro přechody

- Uvažme pattern uvolnění pro řetězec  $w = a^n a^n$





- Pri 15 slavi bih teženina pri glavi neke prešnja vslednik,

- Obena: - stran a vjse  $|w|$  a jedena vtreri!

- 2 soha dostavene preed ypoatrhnik Emla's  
—  $O(|w|)$

$O(|w|^{|\w|+1})$  — preed vsli? no skromet  
ygy  $|w|$  a amle  $|w|$ .

- Velni drake! Ptoha se pravitni n zvikomach  
uneta žizlye, pretože se snodna dvubivny s  
uneta prstirivni DV a operaci s nivi.

- Alle pass: uelbe vstl n bezpečnostne žmihlyk  
aplikacik - krosi tr. DeDOS - regox devidl  
of service.

- Jedna e vovirli! jez se zhorit boeltraedivny ži  
delemivizae.

- OVSEM DVA! žlang' vovirne, nviže vyl  
aveta ži  $2^{O(|w|)}$  slov<sup>o</sup> (a  $O(|w|) \cdot 2^{O(|w|)} = 2^{d(|w|)}$  prebreri<sup>o</sup>)  
v regum. Poed mala n abeadi poed vjdi. stranu

- Pokud výsledky DNA mají  $2^{O(n-1)}$  stavů/přechodů!  
Jelikož buďle stavů bud neexistuje neexistuje přechodů!  
relace  $w$  ?

$$- O(|w|) !$$

- Celková cena neexistuje vzhledem k tomu DNA pro  
neexistuje  $r$  a relace  $w$  je kedy:  
 $2^{O(n)} + O(|w|) -$  to předp. že

DNA jsou schopni abstrahovat v rozsahu aplikací!  
Algoritmy struktury (včetně přechodů) — ne  
semantika přechodů!

- Poznámka se vedle toho, že neexistuje vzhledem  
se pokusí o přechod na DNA a vzhledem!  
Pokud vygeneruje více než jeden počet stavů  
(jinak přechod na DNA)

- Jde o jinou možnost ?

→ Dobro víme že se lihu (on-the-fly) k Thompson.



- První jine klasickém subbed construction!  
 ale pruse pro stary dosazitelne přes testování!  
 řešení.

- Pro větš jstředlostevne,

→  $\{p_i\} \xrightarrow{a}$

$\{p_i, q_i\} \xrightarrow{a}$

$\{p_i, q_i, s_i\} \xrightarrow{a}$

$\{p_i, q_i, s_i, t_i\} \checkmark$

- Složitost:  $O(|w|^2) + O(|w| \cdot |w| \cdot |w|)$

↑  
 konstrukce NFA

↑  
 pro každý znak

↑  
 pro každý stav ne  
 vyhledá všechny  
 unik začít

$$= O(|w| \cdot |w|^2)$$

↑  
 pro každý stav ne  
 vyhledá všechny  
 unik začít  
 1 stav

- Dá se vyřešit uvedené zlepost?

Dá!

- HW implementace (regr. n. FFA) - každý regr se  
 přeloz na obvod, který používá každý veloz o velik.  
 lane' početu slavi' NFA; O(1) odpovídaj' formu, zela  
 u lane' unesitve slavi' jsm' nejsem přislušiv' stang!  
 a ž' & disponici obvod, který jstředlostevne  
 řešení.

vypráše vysledky brány webst: složitost  $O(|w|)$   
za dan obrovského objemu (časy) ziskují vypráv  
vysledky -

- Další možnost - Zavedeme cache! Příklad možná  
rešerá (Pomůže při zábled nepř., n. GNU grep, Google B2):

```
state_cache = { 5 ; 1 // mapuje na množiny stavů
```

```
trans_cache = { 3 ; 1 // na identifikování
```

```
for (i = 1 .. |w|) :
```

mapuje dvojice lid. množiny stavů (symbol) na id. alfove množiny stavů

```
if (trans_cache (i state, a_i)) = 9 ≠ 1 )
```

```
state = 9  
else :
```

```
ms = state_cache   . get (state) ;
```

```
next_ms = { 9 | 9 ∈ T(P, a_i), P ∈ ms } ;
```

```
next_state = state_cache (next_ms) ;
```

```
trans_cache . add ( (state, a_i), next_state ) ;
```

```
state = next_state
```

```
return ( state_cache . get (state) | 0 ≠ 0 ) .
```

state =  
state\_cache (3903) ;



- složitost při řešení 1 zadaní:

-  $O(1)$  - celé int

-  $O(|V^2|)$  - celé uvis

- pro dostatečně dlouhý řetězec při úspěšné u proužití  
celé dostane složitost  $O(|W|)$  pro  
malou rege. udeliny (j. jde třeba seřadit  $|W|$ ).

- Pro krátké regey obvykle není problém (převážně  
distygní vyřešení / alternativy - uapř. Interval  
Hypercube).

- Druhá možnost nastavení s rešitřením regey

- konkrétní se používá uapř. na mezence  
regey - tj. konstrukce  $r \in \mathbb{N}_1, n_2 \mathbb{Z}$

uapř.:  $\bullet * a \in \mathbb{Z}, 100 \mathbb{Z}$

- klasický přístup: rekurzivní de regey s zabudování  
operací - uapř.:  $\bullet * a \in \mathbb{Z} \mid a \in \mathbb{Z} \mid a \in \mathbb{Z} \mid a \in \dots$  ))

- velikost rekurzivní :

- rekurzivní řešení :  $O(|V| \cdot \max\_bound)$

↑  
post-řezení, velikost  
aplikace řešení  
režijní řešení

- završení řešení :  
vzhledem k tomu, že  
rekurzivní řešení završení  
(velikost vstupní úlohy) :  
 $O(|V| \cdot \max\_bound)$

- ALG PROR: max-bound je započítán  
hlavou a plati, že max-bound  $\in 2^{O(|V|)}$   
(konečně  $1001 = 311$ )

- pro rekurzivní řešení tedy dostáváme  
lep. složitost

- pro završení - 2-  
lep. složitost

- Potenciál pro DPAS



- JeZ lze levle proklein ksil?

- Zašledekí vysvětlende: vevravnípl kveluřy ctkerí pvedu, de řídícík skruí, ale rozířít VA a dookledeí ctkerí pteřogíal za kélu poel repelí.
- Ukuve zekeríř uselky žoustrake, šleer, vyřy vříř uveluř: derívee, vřpael vřel. šlouu adl.

- Doplně - Tulel Hyperscan

- obzledně zřejě vejřřkruvířá ualeler.
- su-še-řřy deleruvířtze bez cele
- řada delřířk teekruí:
- vee. vřvřěkí Tulel HNU - velleuří iustrake
- předuvostkuí vřledeí vřřřřkří žoustrakruířk pteřogruvří a oř pael pouvřřří vřř uveluřřřk teekruí uveří velleuřřř - vřřřřřřř.

- mapst. pro heger \* ahey \* a 21,000 3 meter  
bardo ulelad vyshy ahey a waster fiber  
prnu!

- Doplně 2 - existují průmyslové podniky,  
železná infrastruktura dopravní infrastruktura -  
mapst. • NET 7 ulelad pruvstina on-the-fly  
denimace (včetně podpory pruvstina gita  
voting a ahoce)