

Prednáška 5 - Regulárne výrazy, ich vzťah ku konečným automatom

Ing. Viliam Hromada, PhD.

C-510

Ústav informatiky a matematiky
FEI STU

`viliam.hromada@stuba.sk`

20.10.2020



Regulárne výrazy

- Na posledných 3 prednáškach sme si ukázali 3 rôzne spôsoby, ako sa dajú popísať rôzne jazyky: DKA, NKA, ε -NKA
- Iný spôsob sú tzv. **regulárne výrazy**.
- Ukážeme si, že sa jedná len o iný formálny zápis **práve** tých jazykov, ktoré sú popísateľné pomocou konečných automatov.
- Konečné automaty sa často označujú ako **akceptačný / výpočtový** typ popisu nejakého jazyka, regulárne výrazy sú skôr **algebraický popis**.
- Typická aplikácia regulárnych výrazov je pri vyhľadávaní reťazcov spĺňajúcich nejaký vzor (typicky v search engineoch, textových procesoroch, atď.).



Opakovanie

Pre pripomenutie:

- Uvažujme nejakú abecedu A , nad ktorou vieme vytvárať rôzne reťazce.
- Množinu **všetkých** reťazcov vytvorených nad abecedou A označujeme A^*
- **Lubovoľnú** množinu reťazcov nad nejakou abecedou A nazývame **jazykom** L nad abecedou A , t.j. $L \subseteq A^*$.



Operácie s jazykmi

S jazykmi L, M nad nejakou abecedou vieme vykonávať rôzne operácie. Z hľadiska regulárnych výrazov sú dôležité tieto 3:

1. **Zjednotenie** jazykov $L \cup M$ - tvorí ho množina tých reťazcov, ktoré patria alebo do L , alebo do M , alebo do oboch.
2. **Zreťazenie** jazykov LM - tvoria ho všetky reťazce, ktoré vzniknú zreťazením reťazca z jazyka L s reťazcom z jazyka M .
3. **Iterácia** jazyka L (nazývaná aj Kleeneho uzáver), ozn. L^* označuje množinu všetkých reťazcov, ktoré vzniknú zreťazením ľubovoľných kombinácií reťazcov z jazyka L , vrátane ich opakovaní.

Iterácia L^* sa dá definovať aj pomocou zjednotenia a zrežazenia jazyka L ako

$$L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup LLLL \cup \dots$$

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

kde L^i je i -násobné zreťazenie jazyka L .



Operácie s jazykmi - Príklad

Nech abeceda $A = \{0, 1\}$. Nech 2 jazyky $L = \{001, 10, 111\}$, $M = \{\varepsilon, 001\}$ sú nad abecedou A . Potom:

- (zjednotenie) $L \cup M = \{001, 10, 111, \varepsilon\}$.
- (zreťazenie) $LM = \{001, 10, 111, 001001, 10001, 111001\}$
- (zreťazenie) $ML = \{001, 10, 111, 001001, 00110, 001111\}$
- (zreťazenie) $MM = \{\varepsilon, 001, 001001\}$
- (iterácia) $L^* = L^0 \cup L \cup L^2 \cup L^3 \cup \dots = \{\varepsilon, 001, 10, 111, 001001, 00110, 001111, 10001, 1010, 10111, 111001, 11110, 111111, 001001001, 00100110, 001001111, 00110001, 0011010, 00110111, \dots\}$



Definícia regulárneho výrazu

- Regulárne výrazy sú spôsob popisu jazykov.
- Každý regulárny výraz teda reprezentuje nejakú množinu reťazcov.
- Nech regulárny výraz je označený E a jazyk, ktorý popisuje, je označený $L(E)$.
- Regulárne výrazy sú definované podobne ako algebraické systémy - pozostávajú z nejakých základných elementov, nad ktorými sú definované rôzne operácie. Uved'me si, aké základné elementy a operácie uvažujeme nad regulárnymi výrazmi.



Definícia regulárneho výrazu

Základné elementy regulárneho výrazu:

1. "Konštanty" ε a $\emptyset$ sú regulárne výrazy reprezentujúce jazyky $\{\varepsilon\}$ a $\emptyset$. T.j. $L(\varepsilon) = \{\varepsilon\}$ a $L(\emptyset) = \emptyset$
2. Ak a je symbol abecedy, tak potom regulárny výraz $\mathbf{a}$ reprezentuje jazyk $\{a\}$, t.j. $L(\mathbf{a}) = \{a\}$.
3. Premenná L popisujúca nejaký jazyk.

Základné elementy regulárnych výrazov sú teda prázdny jazyk $\emptyset$, prázdny reťazec ε a symboly abecedy. Z týchto základných elementov vieme vytvárať komplikovanejšie regulárne výrazy pomocou operátorov a zátvoriek.



Príklady

Nech abeceda $A = \{a, b\}$. Regulárne výrazy popisujúce rôzne jazyky nad touto abecedou (E je regulárny výraz, $L(E)$ je jazyk, ktorý popisuje):

- $E = \emptyset, L(E) = \emptyset$
- $E = \varepsilon, L(E) = \{\varepsilon\}$
- $E = \mathbf{a}, L(E) = \{a\}$
- $E = \mathbf{a+b}, L(E) = \{a, b\}$
- $E = \mathbf{ab}, L(E) = \{ab\}$
- $E = \mathbf{ba}, L(E) = \{ba\}$
- $E = \mathbf{(ab)+(ba)}, L(E) = \{ab, ba\}$
- $E = \mathbf{a^*}, L(E) = \{\varepsilon, a, aa, aaa, aaaa, \dots\}$
- $E = \mathbf{(a+b)^*} = \{\varepsilon, a, b, aa, ab, ba, bb, \dots\} = A^*$



Príklady

Nech abeceda $A = \{a, b\}$. Regulárne výrazy popisujúce rôzne jazyky nad touto abecedou (E je regulárny výraz, $L(E)$ je jazyk, ktorý popisuje):

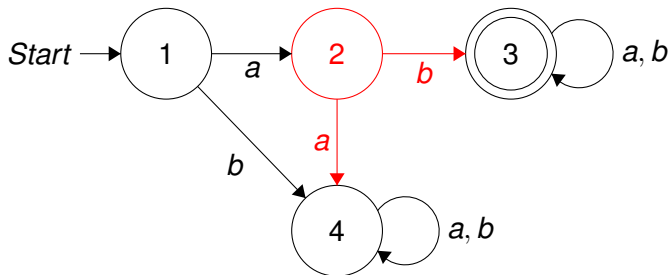
- $E = \mathbf{a(b^*)}$, $L(E) = \{a, ab, abb, abbb, abbbb, \dots\}$
- $E = \varepsilon + \mathbf{a(b^*)}$, $L(E) = \{\varepsilon, a, ab, abb, abbb, abbbb, \dots\}$
- $E = \mathbf{a(a+b)^*}$, $L(E)$ sú všetky reťazce z písmen a a b začínajúce a
- $E = \mathbf{b(a+b)^*b}$, $L(E)$ sú všetky reťazce z písmen a a b začínajúce b a končiace b
- $E = \mathbf{(ab + ba)^*}$, $L(E)$ sú reťazce z a a b , v ktoré pozostávajú z opakujúcich sa reťazcov ab alebo ba , $L(E) = \{\varepsilon, ab, ba, abab, abba, baab, baba, \dots\}$



Ekvivalencia regulárnych výrazov a KA

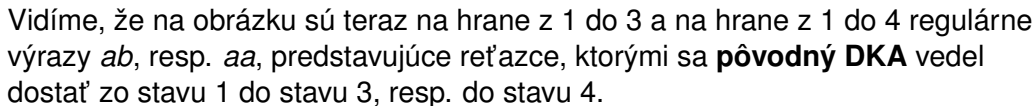
- Regulárne výrazy teda vieme použiť na popis jazykov.
- Ako sme spomínali posledné týždne, na popis jazykov vieme použiť aj konečné automaty.
- Dá sa ukázať, že regulárne výrazy popisujú **práve tie jazyky**, ktoré sú popísateľné konečnými automatmi.
- Teda každý jazyk popísaný regulárnym výrazom sa dá akceptovať nejakým konečným automatom.
- A každý jazyk akceptovaný konečným automatom sa dá popísať nejakým regulárnym výrazom.

Uvažujme nasledovnú situáciu:

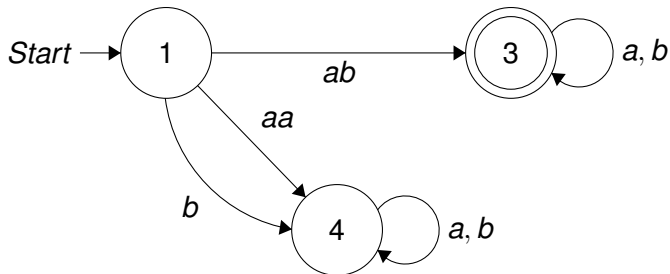


Je zrejmé, že automat akceptuje reťazce dané regulárnym výrazom $ab(a + b)^*$. Chceme odstrániť stav 2 (červený), pričom vzniknutý "automat" by však stále mal akceptovať jazyk $ab(a + b)^*$.

- Ak odstránime stav 2, musíme v automate zachovať možnosť prejsť zo stavu 1 do stavu 3 prečítaním *ab* zo vstupu.
- Rovnako musíme zachovať možnosť prejsť zo stavu 1 do stavu 4 prečítaním *aa* zo vstupu (t.j. prechodom cez stav 2).
- Na tento účel začneme uvažovať špeciálny prechodový diagram, v ktorom na hranách nebudú len **symbolsy vstupnej abecedy**, ale celé **regulárne výrazy**.
- Význam však zostane zachovaný - aby v prechodovom diagrame ("automate") existovala možnosť prejsť zo stavu 1 do stavu 3, musí sa prečítať zo vstupu reťazec zodpovedajúci regulárnemu výrazu na hrane z 1 do 3.

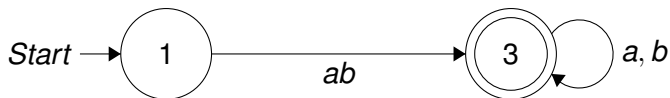


Čo keby sme teraz z tohto diagramu odstránili stav 4 ?

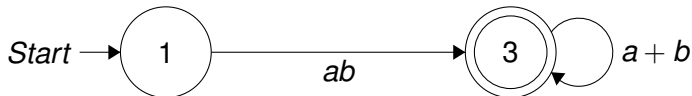


Keďže z tohto stavu **nevychádza** ďalej žiadna hrana a tento stav **nie je akceptačný**, jeho odstránením **nezmeníme jazyk**, ktorý akceptuje príslušný prechodový diagram. Preto ho môžeme odstrániť bez zmeny akceptovaného jazyka.

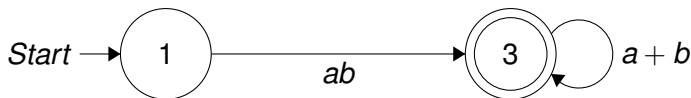
Dostávame teda prechodový diagram:



- Zo stavu 1 do stavu 3 máme hranu ohodnotenú **regulárnym výrazom** **ab**.
- V automate sú 2 hrany, ktoré majú rovnaký začiatok a koniec - slučky pre *a* a *b* v stave 3. Tieto 2 slučky vieme nahradiť **jednou** slučkou, ohodnotenou **regulárnym výrazom** pre ich zjednotenie, t.j. $(a + b)$



Dostávame teda prechodový diagram

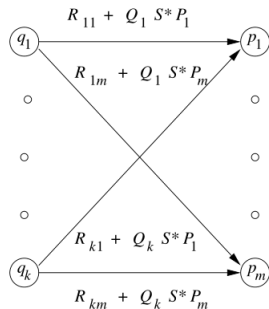
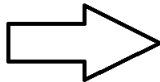


- Vznikol z pôvodného DKA, pričom sme odstraňovali stavy a dopĺňali prechody ohodnotené **regulárnymi výrazmi**, aby sme nezmenili akceptovaný jazyk.
- Z tohto prechodového diagramu je zrejmé, že **práve tie reťazce**, ktoré vie akceptovať, sú v tvare **$ab(a+b)^*$** , t.j. začínajú prefixom ab , za ktorým sa môže nachádzať ľubovoľná postupnosť a, b , čo je **práve jazyk**, ktorý akceptoval **pôvodný DKA**.
- Tým sa nám podarilo popísať jazyk, ktorý akceptoval pôvodný DKA pomocou regulárneho výrazu.

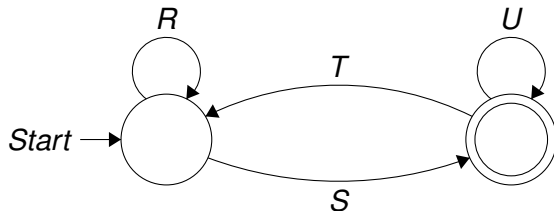
Konverzia DKA na regulárny výraz

- Základ konverzie DKA na regulárny výraz spočíva v postupnej eliminácii stavov automatu, až kým nezostane len počiatočný stav a 1 akceptačný stav.
- Následne sa tento automat s 2 stavmi popíše pomocou regulárnych výrazov.

Uvažujme všeobecnú situáciu, že chceme z automatu odstrániť stav s :



Automat, ktorý má 2 stavy: akceptačný a počiatočný, má vo všeobecnosti nasledovný tvar, kde R, S, T, U sú regulárne výrazy:



Jazyk, ktorý takýto automat akceptuje sa dá **vždy** popísať regulárnym výrazom:

$$E = (R + SU^*T)^*SU^*$$

(samozrejme, ak v reálnom príklade S, T neexistujú, tak ich vieme nahradiť $\emptyset$. Navyše v každom automate je možné ako súčasť R a U uvažovať ϵ).

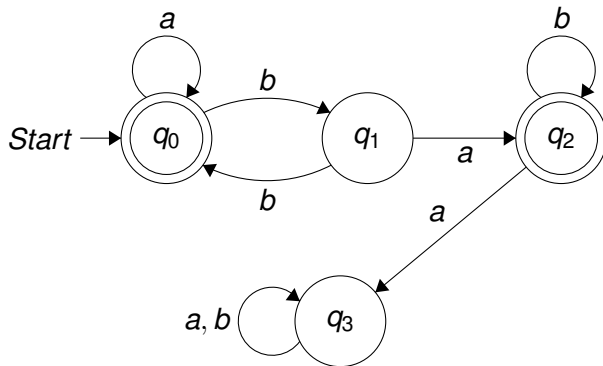
Konverzia DKA na regulárny výraz

Celkový postup je teda nasledovný: Pre daný DKA

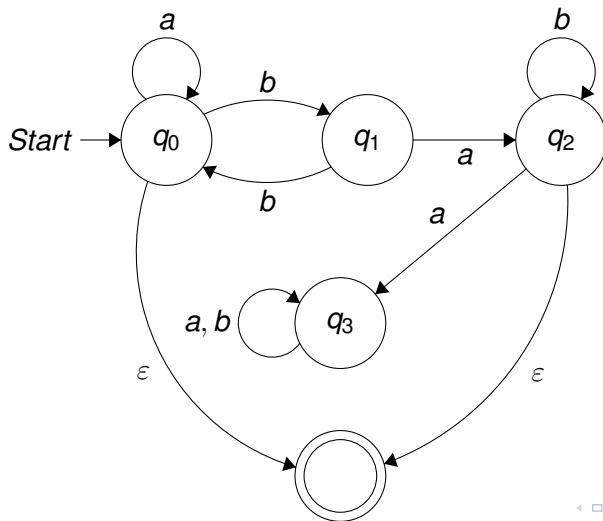
- Zjednotíme mená symbolov na hranách so spoločným začiatkom a koncom.
- Vytvoríme nový akceptačný stav do ktorého vedieme ε -prechody z pôvodných akceptačných stavov, ktoré zároveň označíme ako neakceptačné.
- Odstránime všetky neakceptačné a nepočiatočné stavy a doplníme hrany podľa slajdu č. 21.
- Výsledný automat s 1 počiatočným a 1 akceptačným stavom popíšeme regulárnym výrazom podľa slajdu č. 22.

Príklad

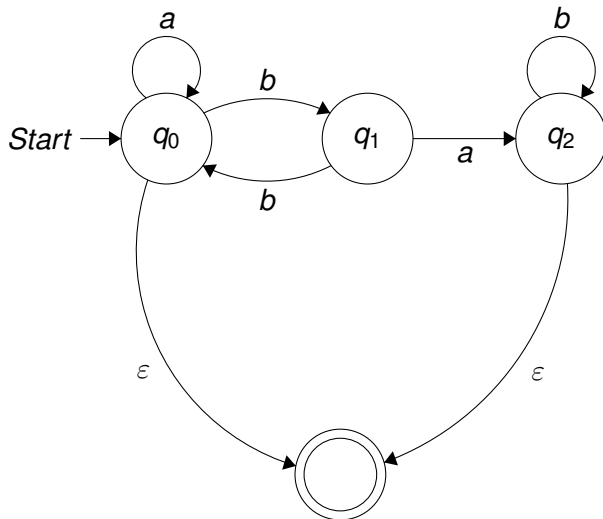
Popíšte regulárnym výrazom jazyk konečného automatu:



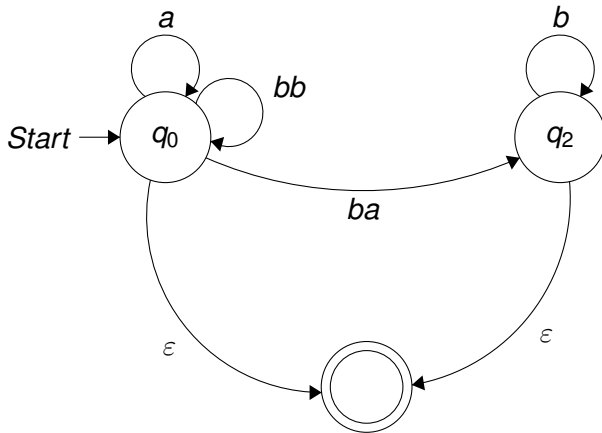
Pridáme nový akceptačný stav, z pôvodných akceptačných stavov q_0, q_2 vedieme doň ε -prechody a zároveň q_0, q_2 označíme ako neakceptačné:



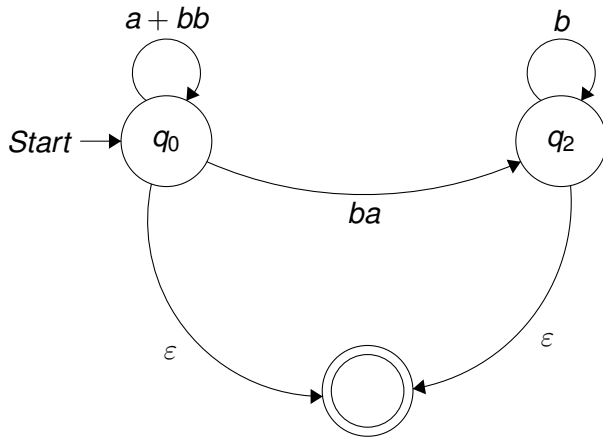
Odstránime stav q_3 :



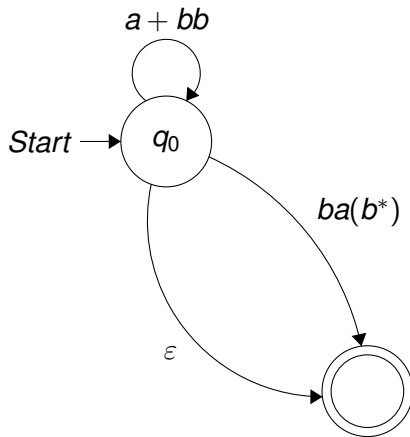
Odstránime q_1 :



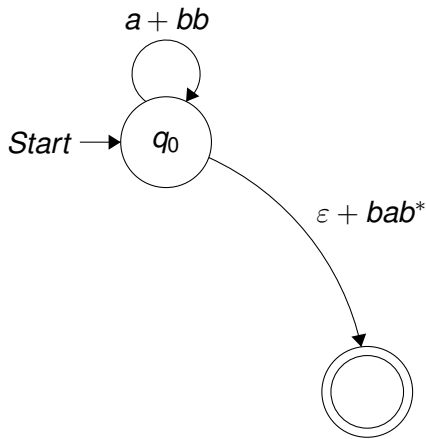
Zjednotíme výrazy na hranách, ktoré majú spoločný začiatok a koniec (v tomto prípade slučky v q_0)



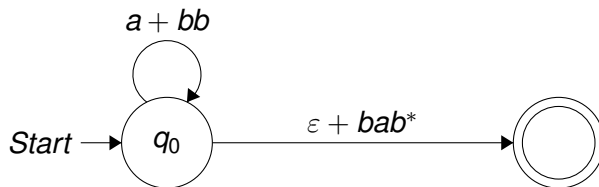
Odstránime q_2 :



Zjednotíme výrazy na hranách, ktoré majú spoločný začiatok a koniec (v tomto prípade hrany z q_0 do akceptačného stavu)



Dostali sme prechodový diagram s 1 počiatočným stavom a 1 akceptačným stavom. Z neho teraz poľahky určíme akceptovaný jazyk:



$$R = (\mathbf{a+bb})^*(\varepsilon + \mathbf{bab}^*)$$

Konverzia regulárneho výrazu na konečný automat

- Každý regulárny výraz sa dá konvertovať na konečný automat.
- Úloha konverzie regulárneho výrazu na konečný automat má praktický význam v programoch, ktorých úlohou je v dátach vyhľadávať reťazce zodpovedajúce zadanému regulárnemu výrazu, keďže sa skonštruuje príslušný konečný automat, ktorý sa následne použije na vstupné dáta.

- Platí, že každý regulárny výraz sa dá previesť na konečný automat.
- Tvrdenie dokážeme pomocou konštruktívneho dôkazu, ktorý bude použiteľný aj vždy, keď budeme chcieť v praxi daný KA zostrojiť.
- Dôkaz zároveň bude pomocou tzv. štruktúrálnej indukcie, keďže využijeme rekurzívnu definíciu regulárnych výrazov.
- T.j. najprv dokážeme, že každý elementárny regulárny výraz sa dá akceptovať nejakým konečným automatom. Následne ukážeme, že aj zjednotenie, zreťazenie a iterácia sa dajú akceptovať nejakým konečným automatom.

Dôkaz

Nech R je regulárny výraz a $L = L(R)$. Ukážeme, že vždy existuje ε -NKA E , taký, že $L(E) = L = L(R)$, ktorý:

- Má presne 1 akceptačný stav.
- Do počiatočného stavu nevchádzajú žiadne hrany.
- Z akceptačného stavu žiadne hrany nevychádzajú.



Na obrázku sú 3 ϵ -NKA pre základné regulárne výrazy: ϵ , $\emptyset$, **a**



$$R = \varepsilon$$



$$R = \emptyset$$



$$R = a$$

Nach D und θ reguläre Abbildung. Dann: $N(KA)$ ist die kleinste Teilmenge von A



Dôkaz pre rekurzívne definované regulárne výrazy

4 rekurzívna definícia bola, že ak R je regulárny výraz, potom aj (R) je regulárny výraz. Konečný automat pre jazyk popísaný výrazom (R) je rovnaký, ako automat pre R .

Každá konverzia regulárneho výrazu na konečný automat teda prebieha nasledovne:

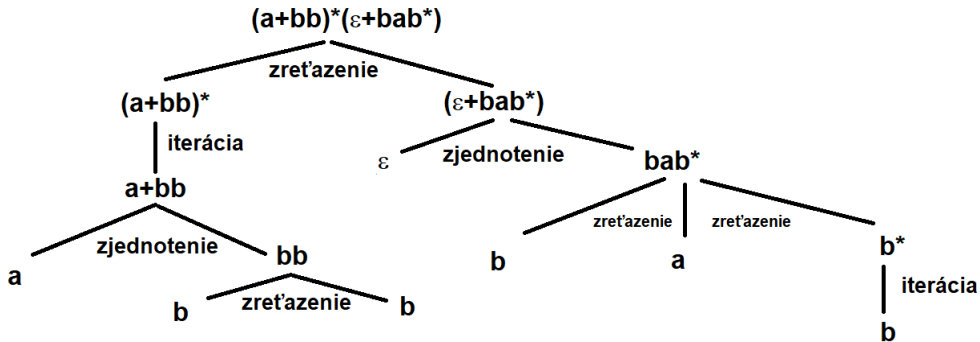
1. Postupne sa zostrojuje ε -NKA pre zadaný regulárny výraz podľa schém zo slajdov 35,36.
2. Teda najprv sa zostroja ε -NKA pre elementárne "regulárne podvýrazy" zadaného výrazu, na ktoré sa následne aplikujú operácie zjednotenia, zreťazenia a iterácie, kým sa nezíska požadovaný výsledok.
3. Výsledný ε -NKA sa následne môže konvertovať na ekvivalentný DKA pre praktické použitie.

Príklad

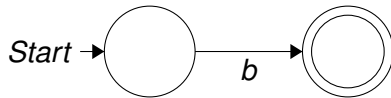
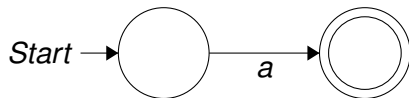
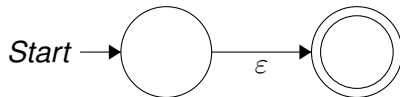
Zostrojte ε -NKA akceptujúci jazyk daný regulárnym výrazom:

$$R = (\mathbf{a+bb})^*(\varepsilon + \mathbf{bab}^*)$$

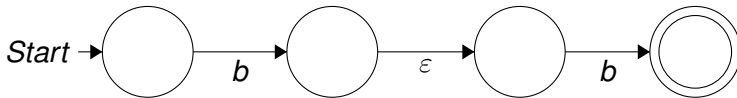
Daný výraz $(\mathbf{a+bb})^*(\varepsilon + \mathbf{bab}^*)$ si musíme rozdeliť na postupné aplikácie operátorov:



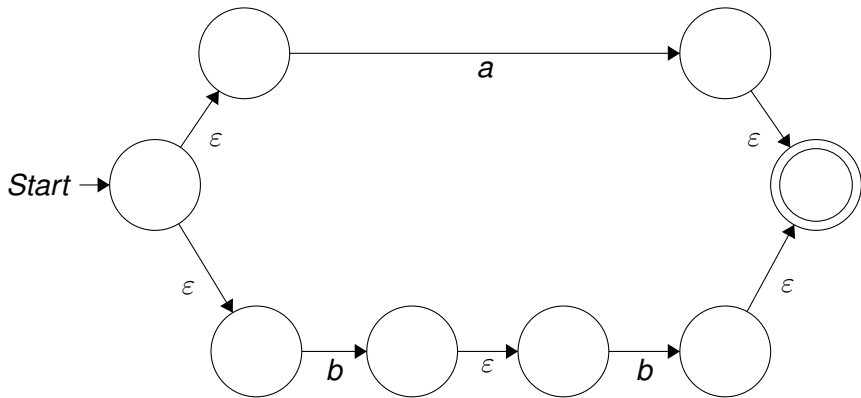
Začneme teda s elementárnymi konečnými automatmi pre ε , **a** a **b**:



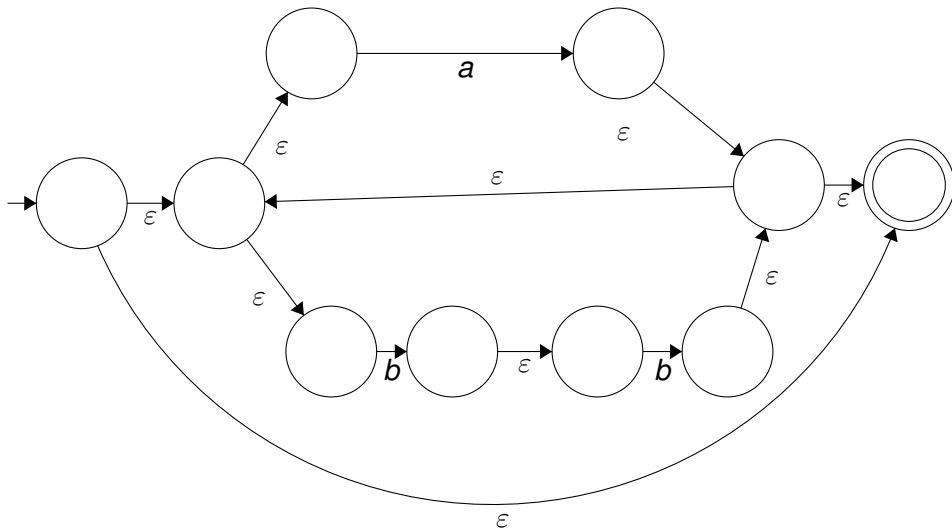
Automat pre **bb**, t.j. zreťazenie 2 automatov pre **b**:



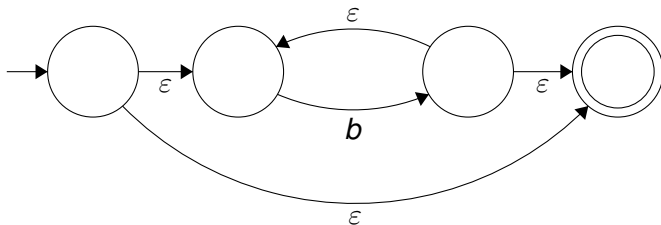
Automat pre **a + bb**, t.j. zjednotenie automatu pre **a** a automatu pre **bb** (ktorý už máme!):



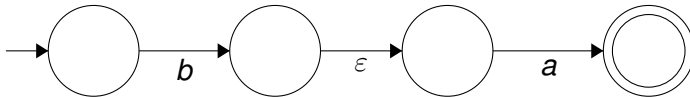
Automat pre **(a+bb)***, t.j. iterácia automatu pre **(a+bb)**:



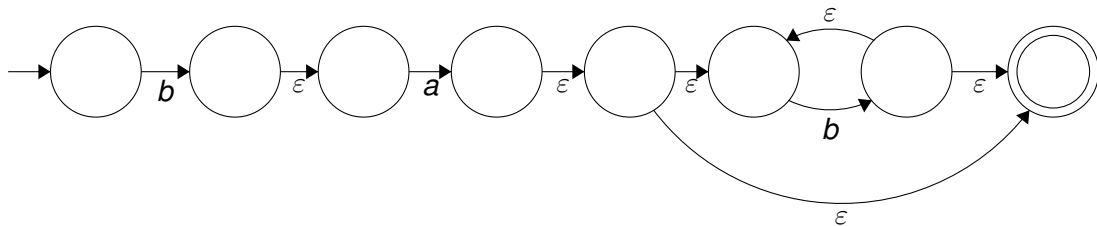
Automat pre $\mathbf{b}^*$, t.j. iterácia automatu pre $\mathbf{b}$:



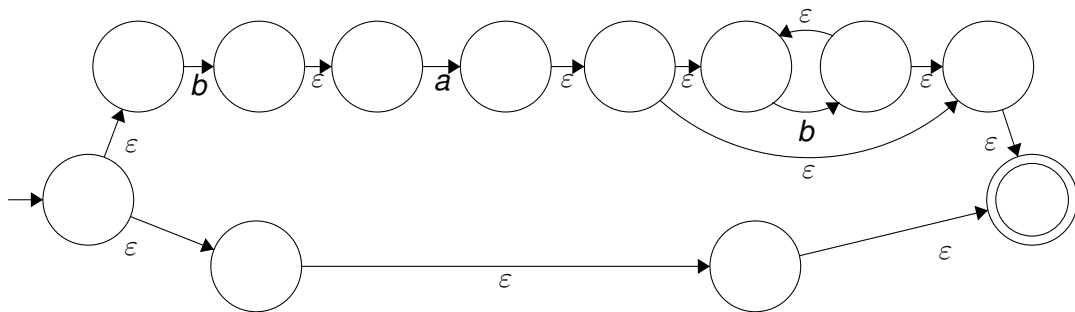
Automat pre **ba**, t.j. zrežazenie automatov pre **b** a **a**:



Automat pre **bab***, t.j. zrežazenie automatov pre **ba** a **b***



Automat pre $(\epsilon + \mathbf{bab}^*)$, t.j. zjednotenie ϵ a $\mathbf{bab}^*$



Praktické aplikácie regulárnych výrazov

- Typické použitie regulárnych výrazov je, že sa pomocou nich popisujú výrazy, ktoré sa hľadajú v texte, pričom sa prevedú na DKA, ktorý sa potom na samotné vyhľadávanie použije.
- Typicky sa používajú pri kompilácii programov - na predchádzajúcej prednáške som spomínal, že sa tam využívajú konečné automaty pre rozpoznávanie kľúčových slov / celočíselných konštánt, atď'. Tieto konečné automaty sú práve **výsledkom** konverzie z príslušných regulárnych výrazov, t.j. najprv sa vytvoria regulárne výrazy pre konštanty, identifikátory, kľúčové slová a tie sa následne prevedú na KA.
- Často je jednoduchšie napísať **regulárny výraz** a ten previesť algoritmom na DKA, než rovno písať DKA.



Použitá literatura

- 1 Hopcroft, Motwani, Ullman - Introduction to Automata Theory, Languages and Computations, 3rd Ed.

