Automaty a formální jazyky

Přednáška I.
Úvodní motivace
Základní pojmy
Konečný automat

Kontakt

- Lenka Kosková Třísková
- lenka@kosek.cz
- 604 510 877 (když se zpozdím o cca 10 minut, volejte, zda dorazím)
- Stránky předmětu: http://lenka.kosek.cz/afj

Organizace a pravidla

- Přednáška ani cvičení nejsou povinné
- Bodování:
  - Za účast na každém cvičení 2 body
  - Za zpracování úloh do cvičení body dle typu úlohy
  - Za aktivitu ve cvičeních a chytré nápady lze získat také body
- Pro zápočet je nutné získat nejméně 2n bodů, kde n je počet cvičení za semestr.
  - Omluvenky mne nezajímají – chybějící cvičení si nahradíte aktivitou.
- Zkouška písemná, celkem 3 části, lze získat maximálně 30 bodů
- Hodnocení:
  - „Výborně“ – celkem 2n plus nejméně 25 bodů
  - „Velmi dobře“ – celkem 2n plus nejméně 20 bodů
  - „Dobře“ – celkem 2n plus nejméně 10 bodů
  - „Ještě jednou“ – méně než 2n plus 10 bodů

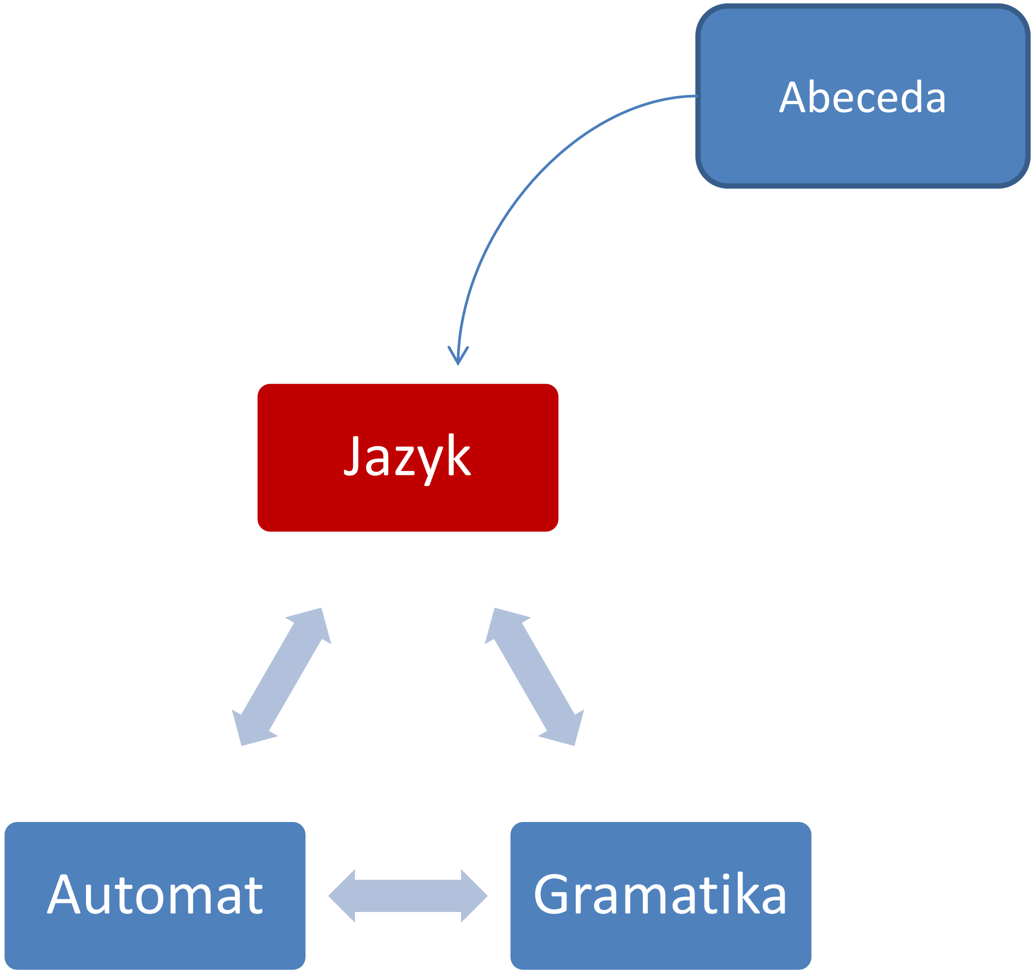
Literatura a další zdroje

- Chytil M.: *Automaty a gramatiky,* SNTL Praha, 1984
- Automata theory, Languages and Computation, Addison Wesley (Hopcroft, Motwani, Ullman)
- Automaty a formální jazyky (I. Černá, M. Křetínský, A. Kučera, Učební text FI MUNI)
- Případně další – bude odkazováno na stránkách předmětu

Proč takový předmět

- Informatická teorie (řešitelnost úloh, návrh jazyků)
- Zpracování umělých jazyků a všeho, co se jim podobá (od assembleru přes XML až k elektronickým fakturám)
- Modelování světa vůkol (zejména systémů)
- Zpracování a hledání pravidelnosti v jazycích přirozených
- Zábava a cvičení mozkové kůry

Jak to spolu souvisí



Abeceda a slovo I.

- **Abeceda:** Libovolná konečná množina.
  - Příklady symbolů: písmena, znaky, obrázky, elektrické signály…
  - Dohoda: Abecedu budeme označovat znakem ∑
- **Symboly (písmena):** Prvky nějaké abecedy.
- **Slovo nad abecedou ∑:** Libovolná konečná posloupnost prvků z množiny ∑. (Prvky se mohou i opakovat, na pořadí symbolů ve slově záleží.)
  - Příklad: ∑ = {a, b, c}
  - Slova: abc, aab, aba, aca, caab atd.

Abeceda a slovo II.

Dohoda – označení slov a symbolů:

- Není-li potřeba jinak, potom symboly označujeme malými písmeny a používáme začátek abecedy (*a, b, c*..).
- Slova označujeme také malými písmeny, ale od konce abecedy (*u, v, w…*).

Příklady:

- – Abeceda $\Sigma = \{a, b, c\}$
- – Slova: $u = ababab$, $v = caca$, $w = cba$
- – Množina slov $L = \{u, v, w\} = \{ababab, caca, cba\}$

Spojování slov

- **Zřetězení:** spojení slov za sebe.
  Máme-li slovo $u = a_1a_2\ldots a_n$ a slovo $v = b_1b_2\ldots b_m$,
  potom slovo $uv=a_1a_2\ldots a_nb_1b_2\ldots b_m$ označujeme jako zřetězení *u* a *v*.
  (Analogie k násobení, proto se někdy se píše s tečkou: *u.v*)
  Opakované n-násobné zřetězení stejného slova (analogie k mocnině) se označuje: u^n
- **Prefix:** Slovo *u* je prefixem slova *w*, pokud *w = uv* pro nějaké *v*.
- **Sufix:** Slovo *v* je sufixem slova *w*, pokud *w = uv* pro nějaké *u*.
- **Podslovo:** Slovo *u* je podslovem slova *w*, pokud *w = vuz* pro nějaká slova *v*.
- Poznámka: V definicích předpokládáme, že všechna slova *u,v,w,z* jsou sestavena nad stejnou abecedou. Abecedy nemícháme.

Jazyk I.

- **Jazyk nad abecedou ∑** je libovolná množina slov sestavených nad danou abecedou. Každý jazyk sestavený nad abecedou ∑ je podmnožinou ∑*.
- **Dohoda:** Pro jazyk používáme označení *L*, případně s dolním indexem, rozlišujeme-li mezi více jazyky (L_1, L_2 atd.)

Jazyk a jeho definice II.

- **Výčtem**: všechna slova vypíšeme.

  Příklad: $\Sigma=\{0,1\}, L=\{00,11,01,10\}$
- **Formálním popisem:** s pomocí množinové konstrukce formulujeme podmínku pro společnou vlastnost slova.

  Příklad: $\Sigma=\{0,1\}, L=\{u \in \Sigma^*;\ |u| = 2\}$
- **Rozsáhlým popisem, gramatikou či syntaktickými pravidly**: pro „složitější“ tedy v praxi skoro pro všechny případy.

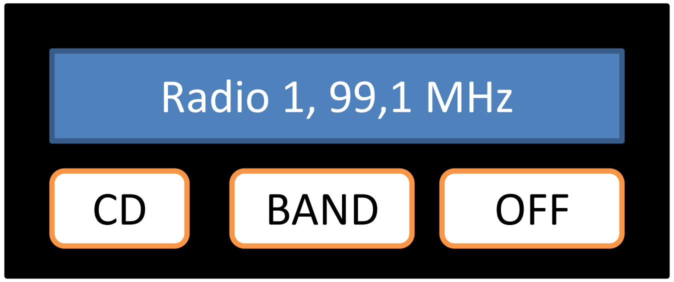
  Příklad: Definice programovacího jazyka. Vyjmenují se klíčová slova plus pravidla pro jejich řetězení.

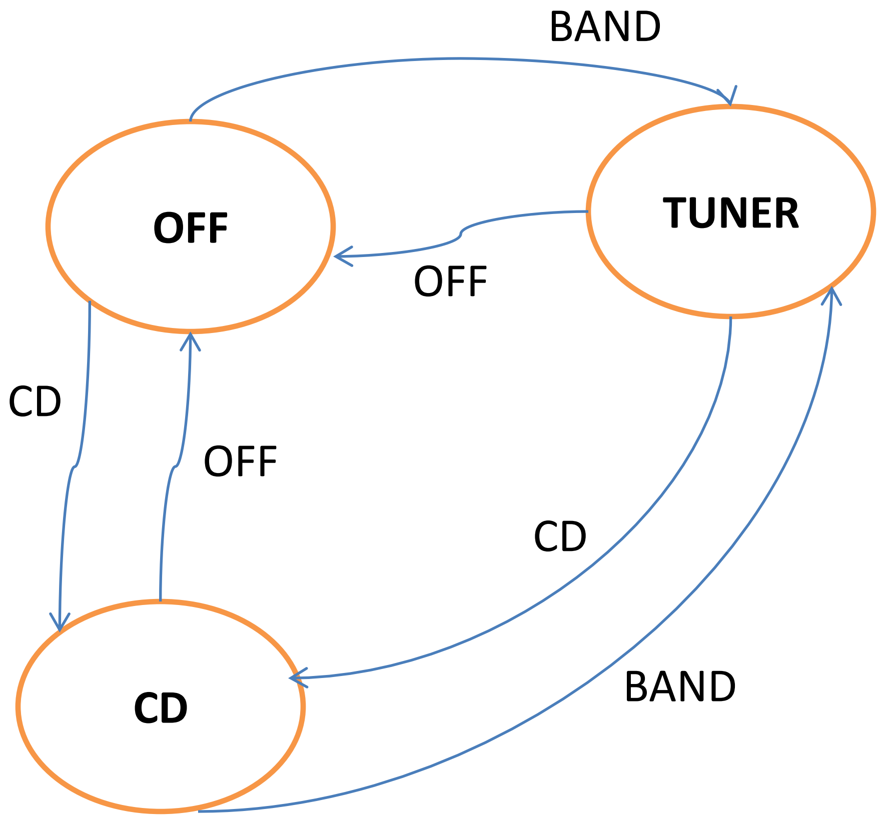
Konečný automat

- Modelování světa vůkol.
- Pro systém (jev, proces, děj) nalezneme:
  - **Stavy**, v nichž se může nacházet.
  - **Vstupy (události)**, jež vedou k přechodu mezi stavy.
  - **Výstupy** – zpravidla odpovídají nějakému stavu.
- Příklad stavového automatu:
  - Model obchodního procesu
  - Model autorádia

Příklad automatu - autorádio

- Přehrává CD, hraje rádio
- Stavy: vypnuto *(OFF)* – rádio *(TUNER)* – přehrává CD *(CD)*
- Události/vstupy – stisky tlačítek (*CD, BAND, OFF)*
- Počáteční stav: OFF





Konečný automat - definice

- **Konečný automat** je každá uspořádaná pětice $A = \{Q, \sum, \delta, q_0, F\}$ kde:
  - Q je konečná neprázdná množina (stavy)
  - $\sum$ je konečná neprázdná množina (vstupní abeceda)
  - δ je zobrazení $Q \times \sum \rightarrow Q$ (přechodová funkce)
  - $q_0 \in Q$ (počáteční stav)
  - $F \subseteq Q$ je konečná množina (rozpoznávací nebo koncové stavy).

Přijímání slova, jazyka - neformálně

- Definice: Slovo je **přijímáno (rozpoznáváno) automatem** *A*, pokud jej převede z počátečního stavu do některého ze stavů rozpoznávacích (koncových).
- Definice: **Jazyk přijímaný (rozpoznávaný) automatem *A*** je množina slov přijímaných automatem *A*, formálně označujeme *L(A).*

Přijímání jazyka - formálně

- Definice: **Rozšířená přechodová funkce** δ^*:

  $\delta^*(q, \varepsilon) = q$

  $\delta^*(q, aw) = \delta^*(\delta(q,a), w)$

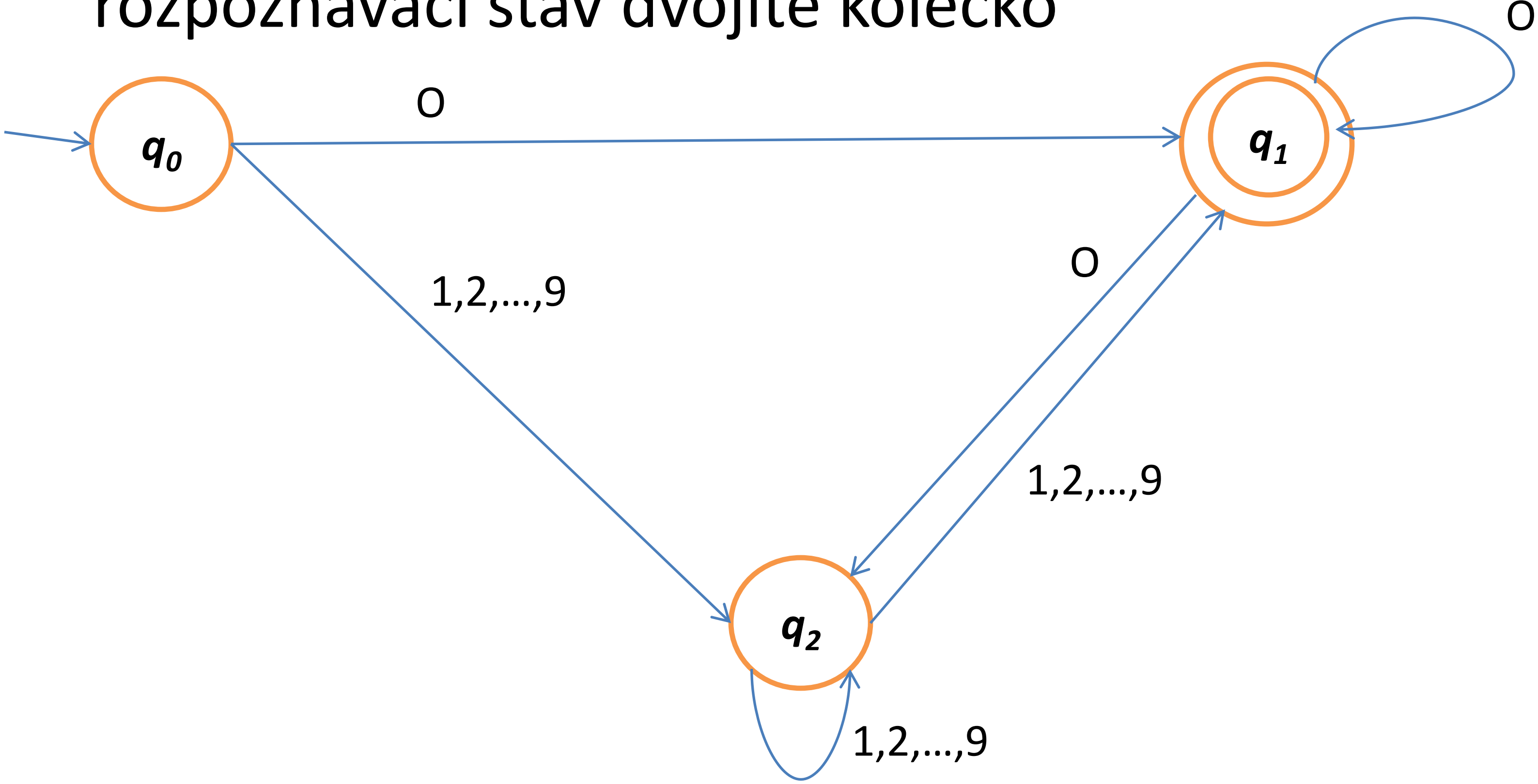
  Platí: $q \in Q$, $a \in \Sigma$, $w \in \Sigma^*$
- Rozšířená přechodová funkce definuje, co se děje s posloupností symbolů ze vstupní abecedy.
- „Skáčeme“ stav po stavu pro jednotlivé symboly od počátku slova.
- Definice: Slovo w je přijímáno (rozpoznáváno) automatem A právě, když $\delta^*(q_0, w) \in F$. Jazyk L je přijímán (rozpoznáván) automatem A právě, když $\delta^*(q_0, w) \in F$ pro všechna $w \in L$.

Metody popisu automatu

- Schéma:
  - Do grafu malujeme stavy a přechody mezi nimi („bubliny“ a „čáry“)
  - Musí se označit počáteční stav a rozpoznávací stavy
  - Vypisuje se množina stavů a abeceda
- Strom
  - „Jiné schéma“, přechodová funkce se maluje do stromového grafu
  - Vypsat množinu stavů, abecedu, označit rozpoznávací stavy
- Tabulka
  - Vypsat: Stavy a abecedu, počáteční stav a rozpoznávací stavy
  - Tabulkou popsat přechodovou funkci
- Výpis
  - Vypisuje všechno, tedy i jednotlivé přechody

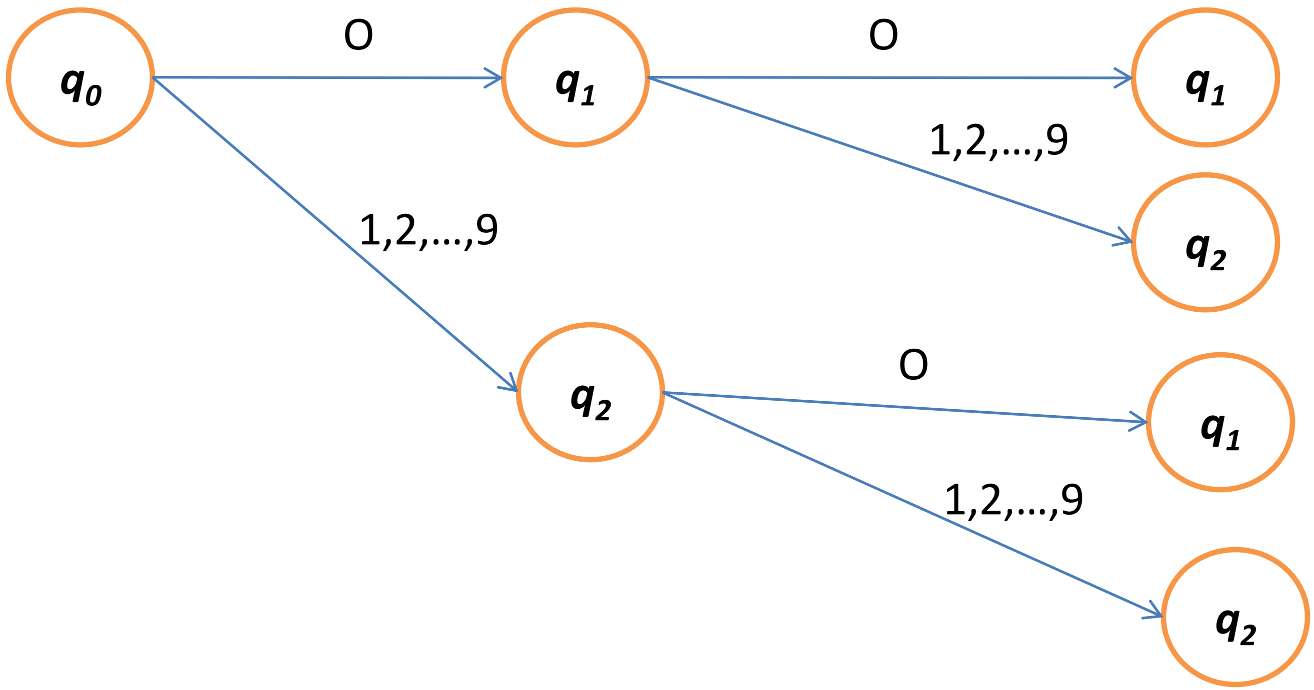
Příklad: Čísla dělitelná 10, schéma

- Abeceda $\sum = \{0,1,2,3,4,5,6,7,8,9\}$, jazyk L = čísla dělitelná deseti
- Stavy $Q = \{q_0,q_1,q_2\}$, počáteční stav šipka dovnitř, rozpoznávací stav dvojité kolečko



Příklad: Čísla dělitelná 10, strom

- Abeceda $\sum = \{0,1,2,3,4,5,6,7,8,9\}$, jazyk L = čísla dělitelná deseti
- Stavy $Q = \{q_0,q_1,q_2\}$, počáteční stav a rozpoznávací stav označen šipkami



Příklad: Čísla dělitelná 10, tabulka

- Abeceda $\sum = \{0,1,2,3,4,5,6,7,8,9\}$, jazyk L = čísla dělitelná deseti
- Stavy $Q = \{q_0,q_1,q_2\}$, počáteční stav a rozpoznávací stav označen šipkami

	0	**1, 2,…,9**
$\boldsymbol{q_0}$	q_1	q_2
$\boldsymbol{q_1}$	q_1	q_2
$\boldsymbol{q_2}$	q_1	q_2

Jazyk rozpoznatelný KA

- Existuje-li alespoň jeden konečný automat, pro který platí $L = L(A)$, potom říkáme, že jazyk L je rozpoznatelný konečným automatem.
- Definice: Jazyky rozpoznatelné konečnými automaty označujeme jako **regulární jazyky.**
- Typický případ jazyku, který není regulární, je jazyk $\{0^n,1^n;n\geq 0\}$ pro $\sum=\{0,1\}$.
- Kritérium pro rozlišení regulárních a neregulárních jazyků definuje Nerodova věta.

Nerodova věta II. – Znění

- **Nerodova věta:** Nechť L je jazyk nad konečnou abecedou $\sum$. L je rozpoznatelný KA, právě tehdy když existuje pravá kongruence konečného indexu taková, že L je sjednocením jistých tříd rozkladu $\sum^*/\approx$.
- **Důkaz**:
  - **Princip:** Hledáme pro daný jazyk automat a obrácené cestu od automatu k jazyku, opíráme se přitom o třídy rozkladu s pomocí operátoru $\approx$.

Nerodova věta III. – Důkaz

1. Máme jazyk L a předpokládáme, že je rozpoznatelný KA, hledáme vhodnou relaci, definujeme ji takto:

   $u \approx v$ právě když $\delta(q_0,u) \approx \delta(q_0,v)$

2. Máme jazyk a relaci, sestavujeme KA:

   Třídy rozkladu získané díky relaci tvoří hledaný KA.

Nerodova věta IV. – Praktické dopady

- Již zmíněný jazyk $\{0^n,1^n;n\geq 0\}$ pro $\sum=\{0,1\}$ není regulární, neboť pro něj nelze nalézt rozklad:
  - Předpokládáme konečný počet tříd *m.* Potom by muselo platit, že alespoň dvě slova ze skupiny 0, 00, 000, …, 0^{m+1}leží ve stejné třídě.
  - Muselo by platit že $0^i \approx 0^j$ *pro* $1 \leq i \leq j \leq m+1$.
  - Jenže když pak zprava přidáme 1^i dostaneme slova 0^i1^i *a* 0^j1^i která by si měla být ekvivalentní, ale druhé slovo nepatří do L!
  - A takhle nám do toho vždycky „vleze“ slovo, které do L nepatří, konečný rozklad prostě nenajdeme.

Další kritéria – „Pumping“ lemma

- **Formálně - Věta:** Nechť L je regulární jazyk. Pak existuje přirozené číslo *n* takové, že libovolné slovo $z \in L$, jehož délka je větší než n, lze psát ve tvaru *uvw*, kde $|uv| \leq n$, $1 \leq |v|$ a pro všechna $0 \leq i$ je $uv^iw \in L$.
- Neformálně: „Nekonečnost“ regulárních jazyků je dána opakujícími se řetězci (jinak nenajdeme rozklad…).
- U regulárního jazyka najdeme vždycky „konečně blízko“ k počátku slova vzorec, který můžeme smazat nebo zopakovat a výsledek zase patří do jazyka.