

Předmět KIV/TI - přednáška 3

# Chomského klasifikace gramatik a jazyků

Ing. Václav Vais, Ph.D.

[vais@kiv.zcu.cz](mailto:vais@kiv.zcu.cz)

# Závěry z minulé přednášky

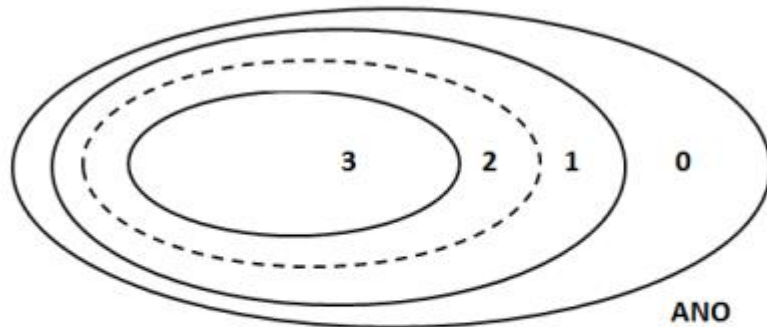
- Existují jazyky, které lze rozpoznávat konečným automatem, tj.
  - zpracováním řetězce  $w \in L$  automat přejde do koncového stavu
  - zpracováním řetězce  $w \notin L$  automat přejde do nekoncového stavu
- Existují jazyky, které rozpoznávat konečným automatem nelze.
- Jako nástroj, který nám má umožnit rozpoznat, zda jazyk rozpoznat konečným automatem lze, jsme zavedli popis jazyka pomocí gramatiky

# Jak souvisí gramatiky s automaty?

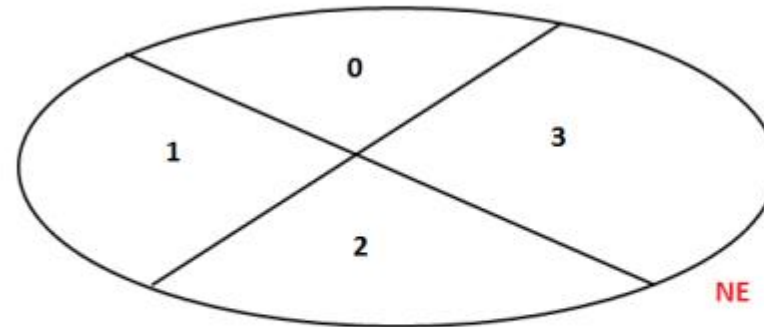
- Výzkumy v oblasti teorie jazyků ukázaly, že „složitost rozpoznávání“ konkrétního jazyka závisí na **tvaru přepisovacích pravidel gramatiky**, která tento jazyk generuje.
- Pod pojmem „složitost rozpoznávání“ rozumíme „výpočetní sílu“ automatu, který jazyk rozpoznává, respektive je modelem syntaktického analyzátoru tohoto jazyka.
- „Nejslabším“ modelem syntaktického analyzátoru je konečný automat.

# Chomského klasifikace gramatik

- **! POZOR !**
  - není to klasifikace do disjunktních tříd, ale
  - je to klasifikace **hierarchická**, založená na **vnořování** tříd



Princip Chomského klasifikace



Princip klasifikace do disjunktních tříd

# Chomského klasifikace gramatik

- gramatiky jsou klasifikovány do čtyř tříd
- značení tříd - 0, 1, 2, 3
- určující pro zařazení gramatiky je nejvyšší třída, jejímž pravidlům gramatika vyhovuje
- všechny gramatiky vyhovují pravidlům pro gramatiky třídy 0
- každá další (tj. vyšší) třída zpřísní formální požadavky na tvar přepisovacích pravidel
- nejužší třída 3 obsahuje gramatiky generující jazyky rozpoznatelné KA

# Chomského klasifikace – gramatiky typu 0

- gramatiky typu 0 (G0)
- všechna pravidla jsou ve tvaru  $\alpha \rightarrow \beta$ , kde

$$\alpha \in (N \cup T)^* N (N \cup T)^*$$

(na levé straně musí být alespoň jeden neterminální symbol)

$$\beta \in (N \cup T)^*$$

(na pravé straně může být i prázdný řetězec)

(nic nového, jen nejobecnější popis tvaru pravidel libovolné gramatiky)

# Chomského klasifikace – gramatiky typu 1

- gramatiky typu 1 (G1)
- všechna pravidla jsou ve tvaru  $\alpha X \beta \rightarrow \alpha \gamma \beta$ , kde

$$\alpha, \beta \in (N \cup T)^*$$

$$X \in N$$

$$\gamma \in (N \cup T)^+$$

Výjimka: v gramatice může být pravidlo  $S \rightarrow e$ , pak se ovšem  $S$  nesmí vyskytnout na pravé straně přepisovacích pravidel

- používané názvy – kontextová gramatika, context sensitive grammar (CSG), nevypouštěcí gramatika

# Chomského klasifikace – gramatiky typu 2

- gramatiky typu 2 (G2)
- všechna pravidla jsou ve tvaru  $X \rightarrow \gamma$ , kde
$$X \in N$$
$$\gamma \in (N \cup T)^*$$
- používané názvy – bezkontextová gramatika (BKG), context free grammar (CFG)
- nejpoužívanější gramatiky jsou typu 2
- nejpracovnější metody syntaktické analýzy



# Chomského klasifikace gramatik

- Srovnání kontextových a bezkontextových gramatik
  - kontextová -  $\alpha X \beta \rightarrow \alpha \gamma \beta$  (pravidlo 1)
  - bezkontextová -  $X \rightarrow \gamma$  (pravidlo 2)
- Zdá se, že „efekt“ obou pravidel je stejný, ale **POZOR**
  - podle pravidla 1 může dojít k substituci  $X$  za  $\gamma$  pouze ve „správném kontextu“ (tj. „zleva  $\alpha$ , zprava  $\beta$ “)
  - podle pravidla 2 může dojít k substituci  $X$  za  $\gamma$  kdykoli

# Chomského klasifikace – gramatiky typu 3

- gramatiky typu 3 pravé (G3P)
- všechna pravidla jsou ve tvaru  $X \rightarrow w$  nebo  $X \rightarrow wY$  kde  
 $X, Y \in N, w \in T^*$
- gramatiky typu 3 levé (G3L)
- všechna pravidla jsou ve tvaru  $X \rightarrow w$  nebo  $X \rightarrow Yw$  kde  
 $X, Y \in N, w \in T^*$
- používané názvy - pravá lineární gramatika, levá lineární gramatika

# Chomského klasifikace gramatik - poznámky

- gramatika je typu  $i$  , jsou-li všechna pravidla typu  $i$  nebo vyššího  
⇒ o typu gramatiky rozhoduje „nejhorší pravidlo“ (pravidlo s nejnižším typem)
- vyskytují-li se v gramatice současně pravidla typu G3P a G3L, nejedná se o gramatiku typu G3, ale (v nejlepším případě) o typ G2
- pravidla, která se mohou vyskytnout v G3P i G3L, označujeme jako pravidla typu G3 (symetrická)
- tvar pravidel u G1 je takový, že existují gramatiky typu G2 a G3, které mu nevyhovují (mohou obsahovat pravidla  $X \rightarrow e$  , kde  $X \in N$  ; proto je v obrázku třída G1 vyznačena jen čárkovaně, zakreslení není zcela korektní

# Chomského klasifikace gramatik - příklady

- Jakého typu je následující gramatika?

$$\begin{array}{l} G_1: \quad S \xrightarrow{3P \quad 3} abA \mid ab \\ \quad \quad A \xrightarrow{3P \quad 3} aaB \mid ba \\ \quad \quad B \xrightarrow{2 \quad 3} AB \mid e \end{array}$$

- Gramatika je typu G2.

# Chomského klasifikace gramatik - příklady

- Jakého typu je následující gramatika?

$$\begin{array}{l} G_2 : \quad \overset{3P}{S} \rightarrow \overset{3}{abA} \mid ab \\ \quad \quad \quad \overset{3L}{A} \rightarrow \overset{3}{Bab} \mid e \\ \quad \quad \quad \overset{3P}{B} \rightarrow \overset{3}{bbB} \mid b \end{array}$$

- Gramatika je typu G2

# Hierarchické uspořádání tříd jazyků

- Podobně, jako se klasifikují gramatiky, budeme klasifikovat i jazyky
- Jazyk  $L$  je typu  $i$  jestliže existuje gramatika  $G$  typu  $i$  taková, že  $L = L(G)$ .
- **! POZOR !** Mohou v tom být záludnosti, viz příklad:

$G: S \rightarrow AB$

$A \rightarrow aaA \mid e$

$B \rightarrow bbb$

$G$  je gramatika typu G2

Jakého typu je jazyk?

**NELZE MECHANICKY KONSTATOVAT** „ $L(G)$  je jazyk typu 2.“

# Hierarchické uspořádání tříd jazyků

- Příklad:

$G: S \rightarrow AB$

$A \rightarrow aaA \mid e$

$B \rightarrow bbb$

$G$  je gramatika typu  $G_2$

Jakého typu je jazyk?

Provedeme odvození několika řetězců:

$$S \xRightarrow{1} AB \xRightarrow{3} B \xRightarrow{4} bbb$$

$$S \xRightarrow{1} AB \xRightarrow{2} aaAB \xRightarrow{3} aaB \xRightarrow{4} aabbb$$

$$S \xRightarrow{1} AB \xRightarrow{2} aaAB \xRightarrow{2} aaaaAB \xRightarrow{3} aaaaB \xRightarrow{4} aaaabbb$$

$$S \xRightarrow{1} AB \xRightarrow{2} aaAB \xRightarrow{2} aaaaAB \xRightarrow{2} aaaaaaAB \xRightarrow{3} aaaaaaB \xRightarrow{4} aaaaaabbb$$

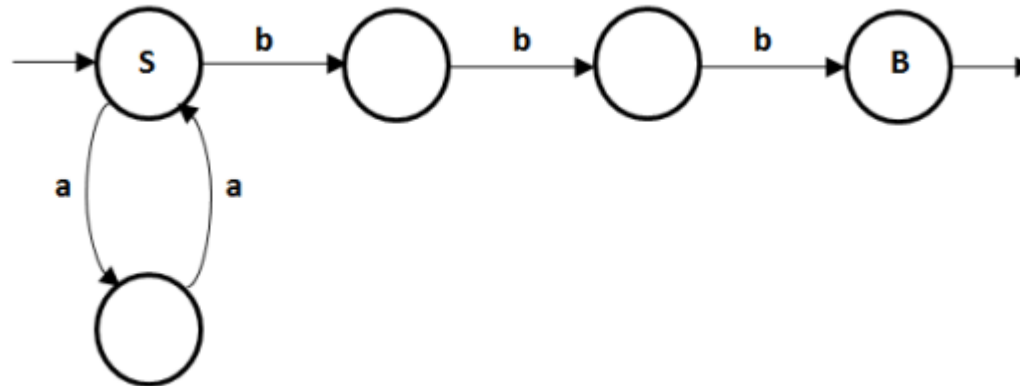
# Hierarchické uspořádání tříd jazyků

- Příklad:

$G: S \rightarrow AB$

$A \rightarrow aaA \mid e$

$B \rightarrow bbb$



Zřejmě platí

$$L(G) = \{ w \mid w = (aa)^n bbb \wedge n \geq 0 \}$$

Jazyk lze popsat i jinou (ekvivalentní) gramatikou:  $G_1: S \rightarrow aaS \mid bbb$

Platí  $L(G) = L(G_1)$   $G_1$  je typu G3P, proto je jazyk  $L(G)$  typu 3.

Výchozí gramatika  $G$  byla „zbytečně složitá“.



# Hierarchické uspořádání tříd jazyků

- Závěry z předchozího příkladu
  - Jazyk může být generován ekvivalentními gramatikami různých typů
  - Nelze mechanicky konstatovat „Každá gramatika typu  $i$  generuje jazyk typu  $i$ .“
  - Typ jazyka je typ nejvyšší gramatiky (tedy gramatiky s nejpřísnějšími pravidly), která jazyk generuje.

# Vztahy mezi třídami jazyků

- Poznatky z teorie bezkontextových jazyků:
  - Ke každé BKG  $G$  existuje ekvivalentní BKG  $G'$  bez pravidel typu  $X \rightarrow e$  (výjimka - může být pravidlo  $S \rightarrow e$ , ale pak  $S$  nesmí být na pravé straně).
  - Tato  $G'$  tedy vyhovuje tvaru pravidel pro gramatiky typu 1 (kontextové).
- Označíme-li tedy  $\mathcal{L}_i$  třídu všech jazyků typu  $i$ , pak platí relace
$$\mathcal{L}_0 \supseteq \mathcal{L}_1 \supseteq \mathcal{L}_2 \supseteq \mathcal{L}_3$$
- Kdybychom tuto relaci znázorňovali graficky (analogie vztahu mezi typy gramatik), hranice mezi  $\mathcal{L}_1$  a  $\mathcal{L}_2$  už by mohla být plná, zanoření je korektní.

# Použití jazyků různých typů

- Největší praktické použití mají jazyky typu 2 (bezkontextové jazyky)
  - Metody překladu jsou rozpracované do detailů
  - Pro vhodně navržené jazyky je syntaktická analýza deterministická
  - Moderní programovací i specifikační jazyky jsou vesměs jazyky typu 2.
- Použití jazyků typu 3 (jsou označovány jako **regulární jazyky**)
  - K popisu objektů při rozpoznávání scény, akustických signálů apod.  
(jednoduché úlohy z oblasti umělé inteligence).
  - Lexikální analýza (rozpoznávání klíčových slov, identifikátorů a konstant ve zdrojovém kódu programů ve vyšších programovacích jazycích).

# Syntaktické analyzátory podle typu jazyka

| <b>Třída jazyků</b> | <b>Používaný název</b>         | <b>Model syntaktického analyzátoru</b> |
|---------------------|--------------------------------|--|
| 0                   | rekurzivně vyčíslitelné jazyky | Turingův stroj                         |
| 1                   | kontextové jazyky              | lineárně omezený Turingův stroj        |
| 2                   | bezkontextové jazyky           | nedeterministický zásobníkový automat  |
| 3                   | regulární jazyky               | konečný automat                        |

## Předmět KIV/TI

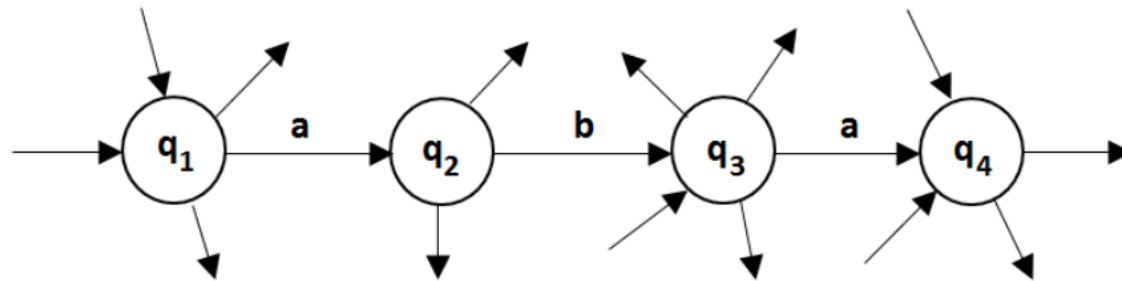
# Regulární jazyky a jejich souvislost s konečnými automaty

Ing. Václav Vais, Ph.D.

[vais@kiv.zcu.cz](mailto:vais@kiv.zcu.cz)

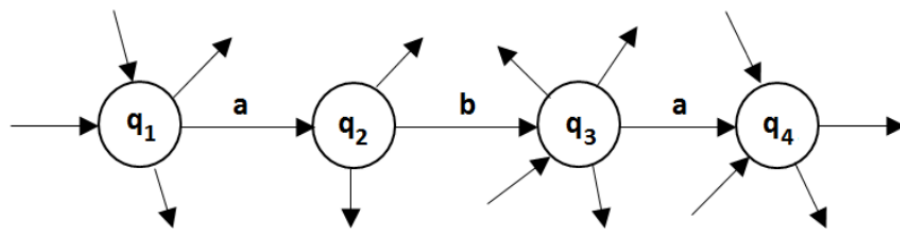
# Zobecněná přechodová funkce KA

- Připomenutí: přechodová funkce  $\delta : Q \times \Sigma \rightarrow Q$
- Příklad (výsek z přechodového grafu KA)



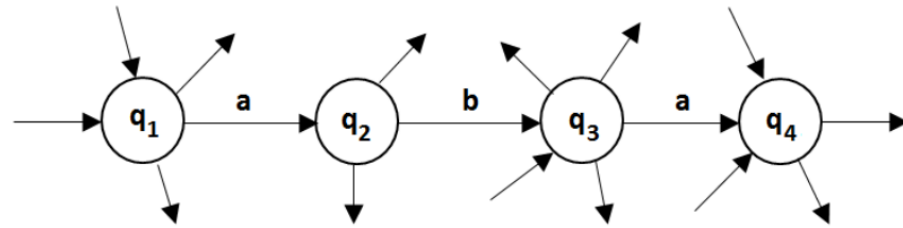
- Jak zareaguje automat na vstupní řetězec *aba* ?

# Zobecněná přechodová funkce KA



- Jak zareaguje automat na vstupní řetězec *aba* ?
- Proveďte tři přechody
$$\delta(q_1, a) = q_2$$
$$\delta(q_2, b) = q_3$$
$$\delta(q_3, a) = q_4$$
- Všechny tyto přechody jsou jednoznačné, jednoznačné tedy bude i rozšíření přechodové funkce tak, že bude definovat nejen následující stav po zpracování jednoho písmene, ale i stav po zpracování řetězce z více písmen

# Zobecněná přechodová funkce KA



- Jak zareaguje automat na vstupní řetězec *aba* ?

- Proveďte tři přechody

$$\delta(q_1, a) = q_2$$

$$\delta(q_2, b) = q_3$$

$$\delta(q_3, a) = q_4$$

- Tedy

$$\delta^*(q_1, aba) = q_4$$

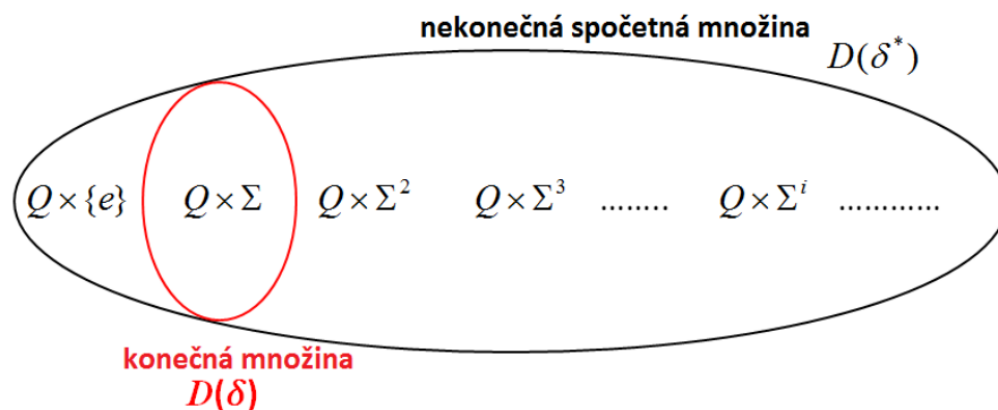


# Zobecněná přechodová funkce KA

- Zobecněná přechodová funkce:  $\delta^* : Q \times \Sigma^* \rightarrow Q$
- Definiční obor zobecněné přechodové funkce:

$$D(\delta^*) = Q \times \Sigma^* = Q \times (\{e\} \cup \Sigma \cup \Sigma^2 \cup \dots) = Q \times \bigcup_{i=0}^{\infty} \Sigma^i$$

Vztah mezi definičními obory obou přechodových funkcí:



$$D(\delta) \subseteq D(\delta^*)$$

$$\delta^*(q, a) = \delta(q, a) \quad \forall q \in Q, \forall a \in \Sigma$$

# Zobecněná přechodová funkce KA

- Zobecněná přechodová funkce je jednoznačně určena přechodovou funkcí
- Lze ji vyjádřit pomocí přechodové funkce rekurzivně:

$$\begin{aligned}\delta^*(q, wa) &= \delta(\delta^*(q, w), a) & \forall q \in Q, \forall w \in \Sigma^*, \forall a \in \Sigma \\ \delta^*(q, e) &= q & \forall q \in Q\end{aligned}$$

- Důsledek:

$$(\delta^*(q, u) = \delta^*(q, v)) \Rightarrow (\delta^*(q, uw) = \delta^*(q, vw)) \quad \forall q \in Q, \forall u, v, w \in \Sigma^*$$

# Jazyk rozpoznávaný automatem

- Jazykem rozpoznávaným konečným automatem

$$A = (Q, \Sigma, \delta, q_0, F)$$

nazýváme jazyk

$$L(A) = \{w \mid w \in \Sigma^* \wedge \delta^*(q_0, w) \in F\}$$

- Verbálně: Jazykem rozpoznávaným konečným automatem rozumíme množinu všech vstupních řetězců, které převedou automat z počátečního stavu do některého ze stavů koncových. (již jsme slyšeli)

# Ekvivalentní rozpoznávací automaty

- Za ekvivalentní automaty považujeme takové, které rozpoznávají stejný jazyk.
- Exaktně:  $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$  a  $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$   
jsou ekvivalentní, právě když  $L(A_1) = L(A_2)$  .
- **! POZOR !** Ekvivalentní mohou být i automaty, které nemají stejný počet stavů.
- Redukovaný automat = takový reprezentant třídy ekvivalentních automatů, který má minimální počet stavů. Existují algoritmy pro minimalizaci KA.