**BI-SPOL-1 Přehled Chomského hierarchie formálních jazyků a gramatik. Turingovy stroje. Třídy problémů P, NP, NP-těžký, NP-úplný**

BI-AAG

### Chomského hierarchie

Str. 7 – 11

**Abeceda**

* konečná množina symbolů (značíme Σ nebo T)
* např. {0,1}, {a,b,c,d,e}

**Řetězec** nad abecedou

* konečná posloupnost symbolů abecedy, používáme pojmenování slovo nebo věta
* ε – **prázdný řetězec**, prázdná posloupnost symbolů
* - množina všech řetězců (včetně prázdného) nad abecedou
* - množina všech neprázdných řetězců nad abecedou , tedy = ∪ {ε}
* např. je-li = {a, b, +}, pak ε, a, b, aa, a+b jsou řetězce nad touto abecedou

**Zřetězení** – operace spojující dva řetězce

**Délka řetězce** – počet symbolů řetězce

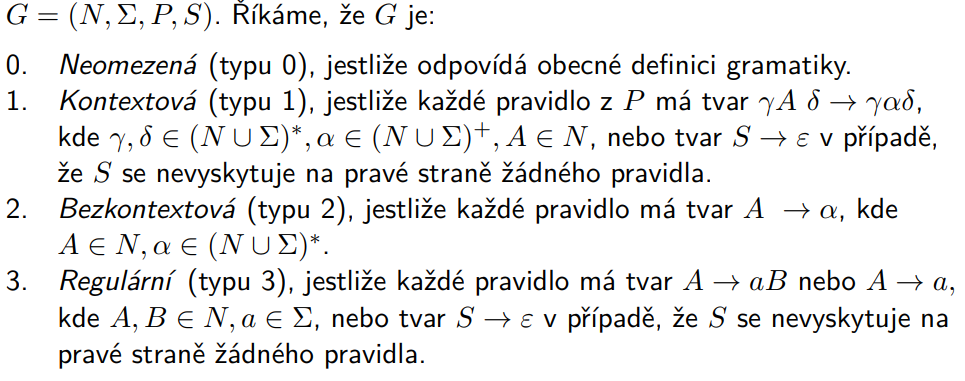
**Formální jazyk**

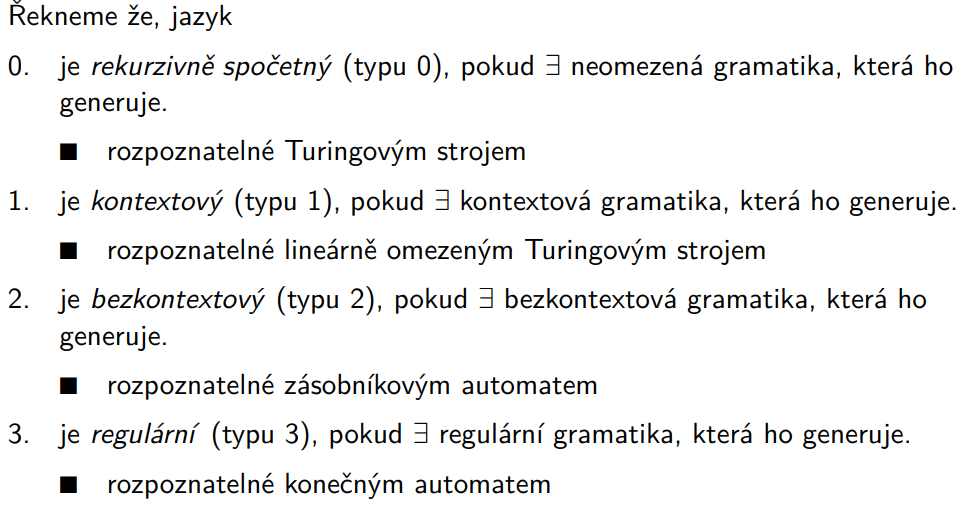
* značíme L
* Libovolná podmnožina množiny všech možných řetězců nad danou abecedou
* Zapisujeme buď výčtem nebo charakteristickou vlastností

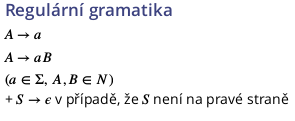
**Gramatika**

* čtveřice G = (N, Σ, P, S)
* N je konečná množina neterminálních symbolů
* Σ je konečná množina terminálních symbolů (Σ ∩ N = ∅, značíme též T)
* P je množina (přepisovacích) pravidel. Je to konečná podmnožina množiny (N ∪ Σ)∗.N.(N ∪ Σ)∗ × (N ∪ Σ)∗, (element (α, β) z P zapíšeme α → β a nazveme pravidlo)
* S ∈ N je počáteční symbol gramatiky (též větný nebo startovací symbol)

**Klasifikace gramatik**

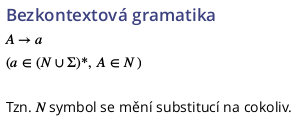


**Klasifikace jazyků**

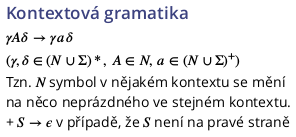


**Regulární jazyk**

* nejjednodušší množina formálních jazyků
* přijmout **(ne)deterministickým konečným automatem**
* generovat **regulární gramatikou**
* popsat **regulárním výrazem**
* např. L = {a^n : n ∈ N}

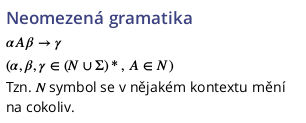
**Bezkontextový jazyk**

* přijmout **nedeterministickým zásobníkovým automatem**
* generovat **bezkontextovou gramatikou**
* např. L = {a^n b^n : n ∈ N}



**Kontextový jazyk**

* přijmout **nedeterministickým lineárně omezeným Turingovým strojem**
* generovat **kontextovou** nebo **nezkracující gramatikou**
* např. L = {a^n b^n c^n : n ∈ N}



**Rekurzivně spočetné jazyky**

* přijmout **(ne)deterministickým Turingovým strojem**
* generovat **neomezenou gramatikou**
* např. L = Halting problem (Turingův stroj *R* se pro vstup *w* zastaví)

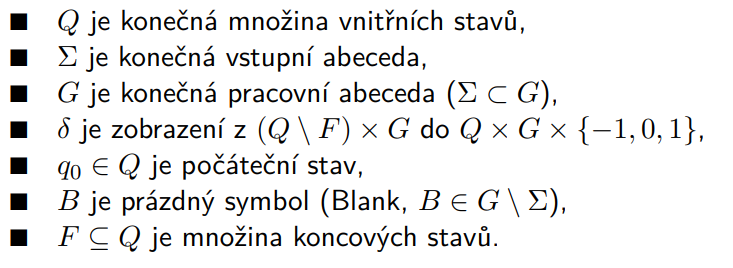
****

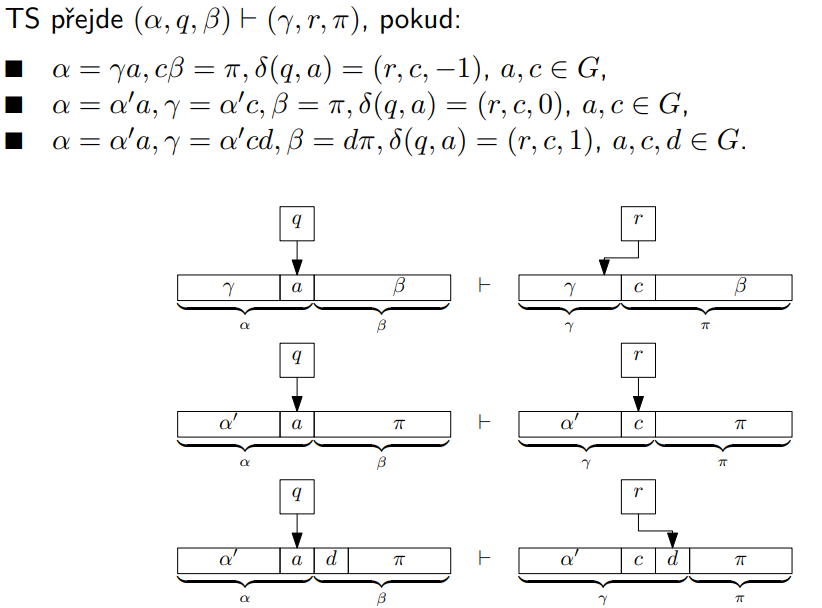
### Turingův stroj

Turingův stroj se skládá z řídící jednotky, neomezené čtecí pásky rozdělené do buněk a čtecí hlavy.

Čtecí hlava se umí pohybovat oběma směry, či posečkat na stejném místě.

Formálně je TS sedmice *R = (Q, Σ, G, δ, q0, B, F)*, kde:





**Deterministický TS**

* TS může zapisovat na vstupní pásku a libovolně se po ní pohybovat.

**Nedeterministický TS**

* Na rozdíl od DTS může mít v daném stavu několik přechodů pro daný symbol. NTS si tedy může vybrat, jakým způsobem bude ve výpočtu pokračovat.
* přijímají právě rekurzivně spočetné jazyky

Jestliže MN je NTS, pak existuje DTS MD takový, že L(MN) = L(MD)

**Lineárně omezený TS**

* TS je lineárně omezený, pokud nemůže překročit délku k-násobku vstupního slova pro nějaké k ≥ 1.
* Pro každou nezkracující gramatiku G existuje ekvivalentní kontextová gramatika
* Gramatiky generují právě rekurzivně spočetné jazyky. Kontextové jazyky jsou přijímány právě lineárně omezenými TS.

**Univerzální TS**

* Turingův stroj je univerzální, právě když přijímá všechny dvojice *(kód(R); α)* takové, že TS. R přijímá slovo α.
* přijímá jako vstup zakódovaný TS a vstup. Simuluje TS

**Přijetí vstupu**

* DTS příjme vstup pokud přejde do konečného stavu a páska je prázdná (pouze *B* symboly).
* NTS přijme vstupní řetězec, pokud existuje alespoň jedna posloupnost přechodů, kdy TS přejde do koncového stavu a páska je na konci výpočtu prázdná.
* vstup není přijat, pokud:

TS je v koncovém stavu, ale páska není prázdná

TS se zacyklil

TS nemůže pokračovat (nedefinovaný přechod)

### Třídy problémů P, NP, NP-těžký, NP-úplný

**Polynomiální redukce** ( ≤p )

* proces, který převádí problém A → B v polynomiálním čase ( a^n )

se stejnou pravdivostní hodnotou.

**Třída P**

* třída problémů, které lze řešit v polynomiálně omezeném čase na DTS
* např. GCD

**Třída NP**

* třída problémů, které lze řešit v polynomiálně omezeném čase na NTS
* všechny P patří do NP
* faktorizace přirozených čísel

**Třída NP-těžký**

* B je NP-těžký, jestliže A ≤p B pro každé A ∈ NP
* NP-těžké problémy jsou takové problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP. (Alespoň tak těžké, jako nejtěžší problémy v NP.)
* nemusí spadat do třídy NP

**Třída NP-úplný**

* Říkáme, že B je NP-úplný, jestliže B je NP-těžký a B ∈ NP
* NP-úplné problémy jsou takové nedeterministicky polynomiální problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP.
* „nějtěžší problémy v NP“
* využití v kryptografii
* Pokud by byl nalezen polynomiálně deterministický algoritmus pro libovolnou NP-úplnou úlohu, všechny NP by byly vyřešeny.
* např. Cook’ theorem

