16. Teorie jazyků a gramatik

1. Základní pojmy

- *abecedu* definujeme jako konečnou množinu znaků - značíme Σ
- jakýkoliv *řetězec* znaků abecedy Σ je konečná posloupnost znaků abecedy Σ
- příkladem abecedy může být $\Sigma = \{0, 1\}$, poté řetězce z ní vytvořené jsou například 0010101 nebo 11010101 atd.
  - projekt Human Genome Project zkoumající lidský genom bude využívat abecedu $\Sigma = \{A, C, G, T\}$
- pakliže uméstíme tyto řetězce do množiny, budeme jí značit Σ^+, poté se hodí zadefinovat Σ^+ jako množinu neprázdných řetězců, prázdný řetězec budeme označovat ϵ
  - s řetězci potom můžeme provádět různé operace, $x \in \Sigma^+$, potom $x = x\epsilon = \epsilon x$
- definice *formálního jazyka*: opět se jedná o množinu, označíme ji L, tato množina bude množinou řetězců nad abecedou Σ, takže $L \subseteq \Sigma^*$, jinými slovy, formálním jazykem se rozumí soubor všech řetězců, které lze nad danou abecedou vytvořit
  - tato definice nám umožňuje vytvářet formální jazyky, které jsou generované nějakým pravidlem
  - příkad $\Sigma = \{0, 1\}$ a $L = \{x \in \Sigma^*; |0| = |1|\}$
- posledním důležitým pojmem jsou *formální gramatiky*, je to jiný způsob definice množin, které jsou vytvořené nad abecedou Σ, gramatiky mají ale výhody oproti definicím pomocí formálních jazyků, gramatiky umožňují zjišťovat, zda-li daný řetězec patří do formálního jazyka
- Gramatika je opět množina, která je definovaná pomocí čtyř pravidel
  - $G = (N, \Sigma, P, S)$
  - N je množina všech neterminálních symbolů
  - Σ je množina všech terminálních symbolů, to znamená, že $\Sigma \cap N = \emptyset$
  - P je množina všech přepisovacích pravidel
    - přepisovací pravidlo zapisujeme $\rightarrow$, můžeme si to představit jako přepisování, nahrazení původního řetězce za nový
  - S je počáteční symbol gramatiky
- gramatika je tím pádem jednoznačně definovaná
- terminály značíme $a, b, c, ...$
- neterminly značíme $A, B, C, ...$
- S značíme počáteční znak
- příklad definujme gramatiku $G = (\{S, A\}, \{0, 1\}, P, S)$, kde P obsahuje $\{S \rightarrow 0A1, A \rightarrow 0A, A \rightarrow 0\}$, a jazyk, který tato gramatika generuje označíme L a bude obsahovat nekonečné množství řetězců $\{001, 0001, 0000001, ...\}$
- každý řetězec, který daná gramatika vygeneruje je tvořen pouze terminálními symboly
- jedna gramatika generuje pouze jeden jazyk jednoznačně, ale jeden jazyk může být generován vícero gramatikami

2. Použití

- základní stavební kámen Computer Science

- teorie složitosti, pomocí formálních jazyků je jednoduché uspořádat určité množiny a podle jejich velikosti zjistit jejich složitost, algoritmus počítá vlastně určitou část formálního jazyka
- výpočetní modely RAM a Turingův stroj
- pochopení DNA, zkoumání lidského genomu
- tvorba překladačů mezi jazyky
- komprese dat, verifikace komunikačních protokolů (tam to bude fungovat na proncipu, zda je daný řetězec generovatelný určitou gramatikou)

3. Klasifikace - Chomského hierarchie

- jelikož může mít gramatika velice komplikovaná pravidla, přepisování velice dlouhých řetězců za jiné, obtížný úkol pro výpočetní stroj, bylo by dobré gramatiky tedy klasifikovat podle složitosti jejich pravidel
- Chomského hierarchie:
- Rozlišujeme základní typy:
  - **Neomezené**: do této množiny spadají všechny gramatiky dále řečené, tyto gramatiky, postačí, když pravidla gramatiky vyhoví obecné definici
  - **Kontextové**: obecné pravidlo, kterým se řídí množina P $\alpha A\beta \rightarrow \alpha\gamma\beta$, kde A je neterminální symbol a $\{\alpha, \beta, \gamma\}$ jsou řetězce terminálů i neterminálů, pokud se terminál S nevyskytuje na pravé straně žádného pravidla, potom musí množina P obsahovat $S \rightarrow \epsilon$; takže při přepisování záleží na kontextu přepisovaného řetězce
  - **Bezkontextová**: speciální případ kontextové gramatiky, řetězce terminálů a neterminálů α, β jou prázdné, takže pravidlo množiny P má obecný tvar $A \rightarrow \gamma$, kde A je opět neterminál; jednoduchý příklad: P: $S \rightarrow aSb|ab$, gramatika generuje $\{a^n b^n,\, n \geq 1\}$
  - **Regulární**: každé pravidlo z P má tvar $A \rightarrow \alpha B$, nebo $A \rightarrow \alpha$, kde A, B jsou neterminály a α řetězec z abecedy Σ, zvláštní případ je $\Sigma \rightarrow \epsilon$
- u tohoto dělení je důležité si uvědomit, že čím hlouběji v hierarchii sestupujeme, tím jsou gramatiky chudší na rozmanitost pravidel, bezkontextové a regulární umožňují přepisovat pouze neterminály
- poté podle příslušné gramatiky rozlišujeme jazyky L (rekurzivně spoštený, kontextový, bezkontextový a regulární), přitom každý regulární jazyk je zároveň kontextový apod.

4. Výpočetní modely

- pro každý typ gramatiky G byl identifikován model stroje, který dokáže určit zda řetězec $\alpha \in \Sigma^*$ náleží jemu příslušného jazyka L, který je generovaný G
- **Turingův sroj -** nekonečná páska a konečná množina stavů, a čtecí hlava, v každém kroku je přečten znak z pásky a stav TS změněn, *gramatika neomezená*
- **Lineárně omezený** TS - má omezenou pásku, *gramatika kontextová*
- **Zásobníjový automat -** konečná množina stavů, jeho program je zadán jako množina přechodů závisející na obsahu zásobníku a čteném znaku, *bezkontextová gramatika*

- **Konečný automat** - viz další otázka, nicméně je navržen pro *regulární gramatiku*