# 12. Gramatiky typu 0 a 1

## Formální jazyky a automaty

Jiří Balun

# Obsah

# Kontextové gramatiky (CSG)

# Kontextová gramatika (typ 1)

**Intuice:** kontextové gramatiky (oproti CFG) umožňují podmínit aplikaci pravidla podřetězci, které jsou v okolí přepisovaného neterminálu.

## Definice

**Kontextová gramatika** (CSG z *context-sensitive grammar*) má všechna pravidla v jednom z těchto tvarů:

1. $\alpha A \beta \to \alpha \gamma \beta$, kde $A \in N$, $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$ a zároveň $\gamma \neq \varepsilon$,
2. $S \to \varepsilon$, pokud se $S$ nevyskytuje na pravé straně jakéhokoliv pravidla.

- levá strana pravidel může obsahovat kromě přepisovaného neterminálu i tzv. **kontext**:
  - v definici je kontext pravidla $\alpha A \beta \to \alpha \gamma \beta$ dán řetězci $\alpha$ a $\beta$
  - celý řetězec na levé straně pravidla podmiňuje aplikování daného pravidla
  - kontext nemůže být daným pravidlem přepsán (je zachován i na pravé straně pravidla)
- CSG generují **kontextové jazyky** (CSL z *context-sensitive language*):
  - jsou nadtřídou bezkontextových jazyků
  - CSL jsou uzavřené na sjednocení, průnik, zřetězení, Kleeneho uzávěr i doplněk
- problém náležení řetězce do jazyka CSG je rozhodnutelný

# Příklad: kontextová gramatika

**Příklad**

CSG $\mathcal{G}_1 = (\{S, A, B, C, D\}, \{a, b, c\}, P, S)$ s jazykem $L(\mathcal{G}_1) = \{a^m b^m c^n \mid 1 < n < m\}$, který je kontextový (a zároveň není bezkontextový):

$$
\begin{aligned}
S &\to aaABBcc \\
A &\to aAB \mid ab \\
bB &\to bC \\
CB &\to CC \\
bC &\to bD \\
bD &\to bb \\
DC &\to DB \\
BC &\to BB \mid BBc
\end{aligned}
\qquad
\begin{aligned}
S &\Rightarrow aaABBcc \\
&\Rightarrow aaabBBcc \\
&\Rightarrow aaabCBcc \\
&\Rightarrow aaabDBcc \\
&\Rightarrow aaabbBcc \\
&\Rightarrow aaabbCcc \\
&\Rightarrow aaabbDcc \\
&\Rightarrow aaabbbcc
\end{aligned}
\qquad
\begin{aligned}
A &\to ab \\
bB &\to bC \\
bC &\to bD \\
bD &\to bb \\
bB &\to bC \\
bC &\to bD \\
bD &\to bb
\end{aligned}
$$

# Příklady: kontextové jazyky

**Příklad**

Jazyk $L_p = \{a^p \mid p \text{ je prvočíslo}\}$ řetězců, které mají prvočíselnou délku.

**Příklad**

Jazyk $L_{exp} = \{a^{2^n} \mid n \in \mathbb{N}\}$ řetězců, které mají délku mocniny čísla 2.

**Příklad**

Jazyk $L_{mul} = \{a^m b^n c^{mn} \mid m, n \in \mathbb{N}\}$, kde počet $c$ je násobkem počtu znaků $a$ a $b$.

**Příklad**

Jazyk $L_{rep} = \{w^{|w|} \mid w \in \Sigma^*\}$, jehož řetězce jsou $n$-té mocniny všech možných řetězců $w$ nad $\Sigma$, kde $n$ je délka daného řetězce $w$.

# Vztah CSG a ostatních gramatik

**Věta**

Každá regulární gramatika je zároveň kontextová.

**Důkaz**

Z definice regulární gramatiky triviálně platí, že její pravidla zachovávají kontext. □

**Věta**

Ke každé CFG $\mathcal{G}$ existuje ekvivalentní gramatika $\mathcal{G}'$, která je zároveň kontextová

**Důkaz**

Všechna pravidla CFG $\mathcal{G} = (N, \Sigma, P, S)$ jsou ve tvaru $A \to \gamma$, kde $\gamma \in (\Sigma \cup N)^*$:

- pokud $\gamma \neq \varepsilon$, pak pravidlo zachovává kontext triviálně,
- je to případ kontextového pravidla $\alpha A \beta \to \alpha \gamma \beta$, kde $\alpha = \varepsilon$ a $\beta = \varepsilon$,
- jediným problémem jsou proto $\varepsilon$-pravidla, tj. pravidla ve tvaru $A \to \varepsilon$ (kde $A \neq S$),
- $\varepsilon$-pravidla odstraníme algoritmem z převodu CFG do Chomského normální formy. □

# Gramatiky bez omezení

# Gramatika bez omezení (typ 0)

**Intuice:** naše definice gramatiky vyžaduje, aby každé pravidlo bylo **generativní**, tj. pravidlo má na levé straně alespoň jeden neterminál.

*Poznámka:* některé definice gramatik povolují i negenerativní pravidla, ale síla gramatiky se tím nezmění, proto je uvažovat nebudeme.

**Definice**

Každá gramatika, jejíž pravidla jsou generativní, je **gramatika bez omezení** (z našeho pohledu tuto definici splňuje každá korektně definovaná gramatika).

- generují **rekurzivně spočetné jazyky** (REL z *recursively enumerable language*):
  - třída REL je nadtřídou kontextových jazyků
  - jsou uzavřené na sjednocení, průnik, zřetězení a Kleeneho uzávěr
  - nejsou uzavřené na doplněk a množinový rozdíl
- problém náležení řetězce do jazyka gramatiky typu 0 obecně není rozhodnutelný
  - pro řetězec, který gramatika negeneruje, se může ověřování zacyklit
  - problém je rozhodnutelný pouze pro tzv. **rekurzivní jazyky** (jsou podmnožinou REL)

# Příklad: gramatika bez omezení

**Příklad**

Gramatika $\mathcal{G}_2 = (\{S, A, B\}, \{a, b, c\}, P, S)$ s jazykem $L(\mathcal{G}_2) = \{a^n b^n c^n \mid n \geq 1\}$:

$$
\begin{array}{ll}
S \to abc \mid aAbc & S \Rightarrow aAbc \qquad Ab \to bA \\
Ab \to bA & \Rightarrow abAc \qquad Ac \to Bbcc \\
Ac \to Bbcc & \Rightarrow abBbcc \qquad bB \to Bb \\
bB \to Bb & \Rightarrow aBbbcc \qquad aB \to aa \\
aB \to aa \mid aaA & \Rightarrow aabbcc
\end{array}
$$

- *Poznámka:* $L(\mathcal{G}_2)$ je rekurzivně spočetný jazyk, který je zároveň kontextový ($\mathcal{G}_2$ je tzv. **nezkracující**, proto ji lze převést na CSG).

# Příklady: rekurzivně spočetné jazyky

**Příklad**

Jazyk vhodně zakódovaných (například binárně) ekvivalentních deterministických zásobníkových automatů je REL, jehož problém náležení **je rozhodnutelný**:

$$DPDA_{eq} = \{\langle A_1, A_2\rangle \mid A_1 \text{ a } A_2 \text{ jsou DPDA takové, že } L(A_1) = L(A_2)\}\,.$$

**Příklad**

Jazyk neekvivalentních bezkontextových gramatik, jehož náležení **není rozhodnutelné**:

$$CFG_{neq} = \{\langle \mathcal{G}_1, \mathcal{G}_2\rangle \mid \mathcal{G}_1 \text{ a } \mathcal{G}_2 \text{ jsou CFG takové, že } L(\mathcal{G}_1) \neq L(\mathcal{G}_2)\}\,.$$

**Příklad**

Problém zastavení je obecně definovaný pro Turingův stroj (více v kurzu o *vyčíslitelnosti*). Pro jednoduchost proto uvažujme jazyk programů jazyka C nad abecedou `ASCII`, které vždy ukončí svůj výpočet. Náležení do tohoto jazyka **není rozhodnutelné**:

$$HALT_C = \{\langle c, w\rangle \mid c \text{ je zdrojový kód jazyka C, který pro vstup } w \text{ zastaví}\}\,.$$

# Nezkracující gramatiky

## Definice

Pravidlo $\alpha \to \beta$ je **nezkracující**, pokud platí $|\alpha| \leq |\beta|$. Gramatika $\mathcal{G} = (N, \Sigma, P, S)$ je nezkracující, pokud jsou nezkracující všechna její pravidla kromě $S \to \varepsilon$.

## Lemma

Ke každé gramatice $\mathcal{G}$ existuje gramatika $\mathcal{G}'$, ve které jsou všechna pravidla ve tvaru:

1. $N^+ \to N^*$, nebo
2. $A \to a$, kde $A \in N$ a $a \in \Sigma$.

## Důkaz

Pravidla z gramatiky $\mathcal{G}$ transformujeme v $\mathcal{G}'$ tímto způsobem:

1. pro každý terminál $a \in \Sigma$ zavedeme nový neterminál $N_a$ a pomocné pravidlo $N_a \to a$,
2. každý výskyt terminálu $a$ v pravidlech $\mathcal{G}$ nahradíme za $N_a$.

Například pravidlo $Cb \to aBb$ se změní na $CN_b \to N_aBN_b$, kde $N_a \to a$ a $N_b \to b$ jsou nová pomocná pravidla $\mathcal{G}'$. □

# Kontextové a nezkracující gramatiky

**Věta**

Ke každé nezkracující gramatice $\mathcal{G}$ existuje CSG $\mathcal{G}'$ taková, že $L(\mathcal{G}) = L(\mathcal{G}')$.

**Důkaz**

Předpokládejme, že pravidla $\mathcal{G}$ jsou buď ve tvaru $A \to a$ (z předchozího lemma), nebo $A_1A_2 \dots A_m \to B_1B_2 \dots B_n$, kde $A_1, \dots, A_m, B_1, \dots, B_n \in N$ a navíc $m \leq n$. S pomocí nových neterminálů $X_1, \dots, X_m$ je nahradíme za pravidla zachovávající kontext:

$$
\begin{aligned}
A_1A_2 \dots A_m &\to X_1A_2 \dots A_m \\
X_1A_2 \dots A_m &\to X_1X_2 \dots A_m \\
&\vdots \\
X_1X_2 \dots X_{m-1}A_m &\to X_1X_2 \dots X_{m-1}X_mB_{m+1} \dots B_n \\
X_1X_2 \dots X_mB_{m+1} \dots B_n &\to B_1X_2 \dots X_mB_{m+1} \dots B_n \\
&\vdots \\
B_1B_2 \dots B_{m-1}X_mB_{m+1} \dots B_n &\to B_1B_2 \dots B_{m-1}B_mB_{m+1} \dots B_n
\end{aligned}
$$

□

# Příklad: nezkracující gramatika

**Příklad**

CSG $\mathcal{G}_2' = (\{S, A, B, X_1, X_2, X_3, X_4, N_a, N_b, N_c\}, \{a, b, c\}, P, S)$ s jazykem $L(\mathcal{G}_2') = \{a^n b^n c^n \mid n \geq 1\}$, kterou získáme z předchozího příkladu gramatiky bez omezení:

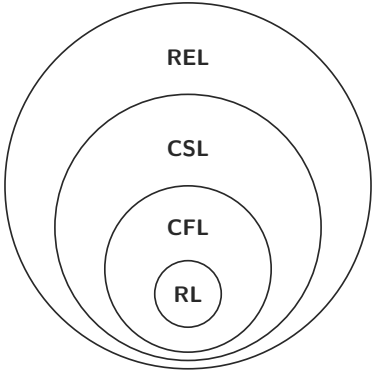
$$
\begin{aligned}
S &\to N_a N_b N_c \mid N_a A N_b N_c \\
A N_b &\to X_1 N_b \\
X_1 N_b &\to X_1 X_2 \\
X_1 X_2 &\to N_b X_2 \\
N_b X_2 &\to N_b A \\
A N_c &\to B N_b N_c N_c \\
N_a B &\to N_a N_a \mid N_a N_a A
\end{aligned}
\qquad
\begin{aligned}
N_b B &\to X_3 B \\
X_3 B &\to X_3 X_4 \\
X_3 X_4 &\to B X_4 \\
B X_4 &\to B N_b \\
N_a &\to a \\
N_b &\to b \\
N_c &\to c
\end{aligned}
$$

**Průběh transformace na CSG:**

- nejprve nahradíme každý výskyt terminálu $\sigma \in \Sigma$ za pomocný neterminál $N_\sigma$,
- poté stačí nahradit dle důkazu předchozí věty pravidla, které nezachovávají kontext.

# Existence dalších jazyků

# Chomského hierarchie




- **Je tu něco?**

**Existuje jazyk, který není rekurzivně spočetný?**

- otázku lze formulovat jako: existuje jazyk, který nelze generovat žádnou gramatikou?
- odpověď je **ano**, například jazyk dvojic ekvivalentních CFG (doplněk $CFG_{neq}$):

$$CFG_{eq} = \{\langle \mathcal{G}_1, \mathcal{G}_2 \rangle \mid \mathcal{G}_1 \text{ a } \mathcal{G}_2 \text{ jsou takové CFG, že } L(\mathcal{G}_1) = L(\mathcal{G}_2)\}$$

# Nespočetnost $2^{\Sigma^*}$

**Věta**

Třída všech jazyků nad abecedou $\Sigma$, kterou značíme $2^{\Sigma^*}$, je nespočetná.

**Důkaz**

Z definice abecedy je $\Sigma$ konečná množina, a proto je Kleeneho uzávěr $\Sigma^*$ spočetně nekonečná množina a lze jej seřadit pomocí shortlex uspořádání, tj. $\Sigma^* = \{w_1, w_2, \ldots\}$.

Důkaz vedeme sporem, proto předpokládejme, že $2^{\Sigma^*}$ je spočetná množina:

- potom by prvky (jazyky) z $2^{\Sigma^*}$ měly jít nějak uspořadat, tj. $2^{\Sigma^*} = \{L_1, L_2, L_3, \ldots\}$,
- $2^{\Sigma^*}$ obsahuje i jazyk definovaný jako $L_{diag} = \{w_i \mid w_i \notin L_i\}$,
- řekněme, že existuje index $j$ takový, že platí $L_{diag} = L_j$,
- pak ale $w_j \in L_{diag}$ právě tehdy, když $w_j \notin L_j$, což je spor, protože $L_{diag} = L_j$.

Všimněte si, že tvrzení platí i pro případ, kdy je $\Sigma$ jednoprvková. □

# Nespočetnost $2^{\Sigma^*}$

| | $w_1$ | $w_2$ | $w_3$ | $\cdots$ | $w_j$ | $\cdots$ |
|---|---|---|---|---|---|---|
| $L_1$ | ✓ | ✗ | ✗ | $\cdots$ | ✓ | $\cdots$ |
| $L_2$ | ✓ | ✗ | ✓ | $\cdots$ | ✗ | $\cdots$ |
| $L_3$ | ✗ | ✓ | ✓ | $\cdots$ | ✓ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | |
| $L_j = L_{diag}$ | ✗ | ✓ | ✗ | $\cdots$ | ?? | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ | |

**Reprezentace důkazu tabulkou:**

- řádek pro $L_j = L_{diag} = \{w_i \mid w_i \notin L_i\}$ je definován jako negace prvků na diagonále
- ve sloupci pro slovo $w_j$ dostáváme spor, protože $w_j \in L_j$ právě tehdy, když $w_j \notin L_j$

# Existence jazyka, který není REL

**Věta**

Nad jednoprvkovou abecedou $\Sigma = \{a\}$ existuje jazyk, který není generován gramatikou.

**Důkaz**

Mějme abecedu $\Gamma = \{a, N, \to, \texttt{\{}, \texttt{\}}, \texttt{(}, \texttt{)}, \texttt{,}\}$ (i symbol čárky je součástí $\Gamma$), pomocí které jsme schopni zakódovat libovolnou gramatiku nad abecedou $\Sigma$:

- neterminály kódujeme unárně jako $N_1 = N$, $N_2 = NN$, $N_3 = NNN, \ldots$
- příklad řetězce nad $\Gamma$ je gramatika $\texttt{(\{}N\texttt{\},\{}a\texttt{\},\{}N \to a\texttt{\},}N\texttt{)}$ popisující jazyk $\{a\}$,
- $\Gamma^*$ je spočetná (stejně jako $\Sigma^*$), proto existuje spočetně mnoho gramatik nad $\Sigma$,
- každá gramatika zakódovaná pomocí $\Gamma$ popisuje právě jeden jazyk nad $\Sigma$,
- proto lze pomocí $\Gamma$ reprezentovat jen spočetně mnoho jazyků nad abecedou $\Sigma$.

Třída jazyků $2^{\Sigma^*}$ je ale nespočetná, proto musí existovat jazyk $L \subseteq \Sigma^*$, který nelze generovat gramatikou z $\Gamma^*$. □