

10. Vlastnosti bezkontextových jazyků

Formální jazyky a automaty

Jiří Balun

Obsah

1 Vlastnosti bezkontextových jazyků

- Uzávěrové vlastnosti
- Rozhodnutelnost vlastností
- Vlastnosti deterministických bezkontextových jazyků

2 Pumping lemma (PL) pro bezkontextové jazyky

- Definice a vysvětlení
- Důkaz PL
- Příklady

Vlastnosti bezkontextových jazyků

Základní uzávěrové vlastnosti

Věta

Nechť L , L_1 a L_2 jsou bezkontextové jazyky, pak jsou bezkontextové i jazyky:

- 1 L^*
- 2 $L_1 \cdot L_2$
- 3 $L_1 \cup L_2$

Důkaz

- 1 Mějme CFG $\mathcal{G} = (N, \Sigma, P, S)$ s jazykem $L(\mathcal{G}) = L$, pak CFG $\mathcal{G}'$ s jazykem $L(\mathcal{G}') = L^*$ definujeme jako $\mathcal{G}' = (N \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow SS' \mid \varepsilon\}, S')$.

Pro důkazy 2 a 3 mějme CFG $\mathcal{G}_1 = (N_1, \Sigma_1, P_1, S_1)$ a $\mathcal{G}_2 = (N_2, \Sigma_2, P_2, S_2)$ takové, že $L(\mathcal{G}_1) = L_1$ a $L(\mathcal{G}_2) = L_2$ a předpokládejme, že $N_1 \cap N_2 = \emptyset$ (v opačném případě symboly přejmenujeme). Navrhneme CFG $\mathcal{G}'$ jako:

- 2 $\mathcal{G}' = (N_1 \cup N_2 \cup \{S'\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S' \rightarrow S_1 S_2\}, S')$ s jazykem $L(\mathcal{G}') = L_1 \cdot L_2$.
- 3 $\mathcal{G}' = (N_1 \cup N_2 \cup \{S'\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S' \rightarrow S_1 \mid S_2\}, S')$ s jazykem $L(\mathcal{G}') = L_1 \cup L_2$.



Uzavřenost na reverz a homomorfismus

Věta

Nechť L je bezkontextový jazyk a $h : \Sigma \rightarrow \Sigma'$ je homomorfismus, pak jsou bezkontextové i tyto jazyky:

- 1 L^R
- 2 $h(L)$
- 3 $h^{-1}(L)$

Důkaz

Mějme CFG $\mathcal{G} = (N, \Sigma, P, S)$ s jazykem $L(\mathcal{G}) = L$.

- 1 Navrhne CFG $\mathcal{G}'$ s jazykem $L(\mathcal{G}') = L^R$ jako $\mathcal{G}' = (N, \Sigma, P^R, S)$, kde pro každé pravidlo $A \rightarrow \alpha$ z P definujeme reverzní pravidlo $A \rightarrow \alpha^R$ v P^R .
- 2 Navrhne CFG $\mathcal{G}'$ s jazykem $L(\mathcal{G}') = h(L)$ jako $\mathcal{G}' = (N, \Sigma', P^h, S)$, kde pro každé pravidlo $A \rightarrow \alpha$ z P definujeme pravidlo $A \rightarrow \beta$ v P^h , kde β získáme z α nahrazením každého výskytu terminálu za jeho obraz dle h , například $a \in \Sigma$ nahradíme za $h(a)$.
- 3 důkaz vynecháme. □

Uzavřenost na průnik s regulárním jazykem

Věta

Nechť L je bezkontextový jazyk a R je regulární jazyk, pak jazyk $L \cap R$ je bezkontextový.

Idea důkazu

Nechť $A = (Q_A, \Sigma, \delta_A, q_{0,A}, F_A)$ je DFA s $L(A) = R$, a $P = (Q_P, \Sigma, \Gamma, \delta_P, q_{0,P}, F_P)$ je PDA s $L(P) = L$, z nichž sestavíme PDA P' , který je obdobou součinného automatu:

$$P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, \langle q_{0,P}, q_{0,A} \rangle, Z_0, F_P \times F_A),$$

kde $\delta(\langle p, q \rangle, \hat{a}, X)$ pro $\hat{a} \in (\Sigma \cup \{\varepsilon\})$ obsahuje všechny dvojice $(\langle r, s \rangle, \gamma)$ takové, že:

1 $(r, \gamma) \in \delta_P(p, \hat{a}, X)$

2 $\delta_A(q, \hat{a}) = s$

P' tedy může provést všechny přechody jako P , ale navíc simuluje běh A v druhé složce svých stavů. Koncové stavy P' jsou ty dvojice stavů, kde přijme A i P .

Zbytek důkazu spočívá v indukci přes počet kroků výpočtu: $\langle q_{0,P}, w, Z_0 \rangle \vdash^* \langle p, \varepsilon, \gamma \rangle$ a $\hat{\delta}_A(q_{0,A}, w) = q$ platí právě tehdy, když $\langle \langle q_{0,P}, q_{0,A} \rangle, w, Z_0 \rangle \vdash^* \langle \langle p, q \rangle, \varepsilon, \gamma \rangle$.



Uzavřenost na prefix a sufix

Věta

Nechť L je bezkontextový jazyk, pak jsou bezkontextové i jazyky $Pfx(L)$ a $Sfx(L)$.

Důkaz

V důkazu využijeme uzavřenosti CFL na homomorfismy a jejich inverzi:

- definujeme kopii abecedy Σ jako $\tilde{\Sigma} = \{\tilde{a} \mid a \in \Sigma\}$,
- dále definujeme homomorfismus $h: (\tilde{\Sigma} \cup \Sigma) \rightarrow \Sigma$, který odstraňuje vlnku ze symbolu, tj. $h(\tilde{a}) = a$ pro všechny $a \in \tilde{\Sigma}$ a $h(a) = a$ pro všechny $a \in \Sigma$,
- dále definujeme homomorfismus $g: (\tilde{\Sigma} \cup \Sigma) \rightarrow \Sigma$, který odstraňuje symboly s vlnkou, tj. $g(\tilde{a}) = \varepsilon$ pro všechny $a \in \tilde{\Sigma}$ a $g(a) = a$ pro všechny $a \in \Sigma$,
- připomeňme, že $h^{-1}(L)$ je množina všech řetězců z L , ve kterých nahradíme libovolné symboly za jejich variantou s vlnkou,
- výsledné bezkontextové jazyky získáme (díky uzavřenosti CFL na průnik s regulárním jazykem) jako $Pfx(L) = g(h^{-1}(L) \cap \Sigma^* \tilde{\Sigma}^*)$ a $Sfx(L) = g(h^{-1}(L) \cap \tilde{\Sigma}^* \Sigma^*)$. □

Neuzavřenost na průnik a doplněk

Věta

Třída bezkontextových jazyků není uzavřená na průnik.

Důkaz

Mějme bezkontextové jazyky $L_1 = \{a^k b^k c^\ell \mid k, \ell \geq 0\}$ a $L_2 = \{a^\ell b^k c^k \mid k, \ell \geq 0\}$, pak:

$$L_1 \cap L_2 = \{a^k b^k c^k \mid k \geq 0\},$$

který **není bezkontextový** (ukážeme později pomocí PL). □

Věta

Třída bezkontextových jazyků není uzavřená na doplněk.

Důkaz

Plyne z uzavřenosti na sjednocení a DeMorganova zákona $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$, proto pokud by CFL byly uzavřeny na doplněk, pak musí být uzavřeny i na průnik – spor. □

Souhrn vlastností bezkontextových jazyků

Uzávěrové vlastnosti

- sjednocení
- zřetězení a Kleeneho uzávěr
- zobrazení pomocí homomorfismu a inverzního homomorfismu
- reverzní jazyk
- jazyk prefixů, sufixů a infixů

Bezkontextové jazyky nejsou uzavřené na...

- množinové operace: průnik, doplněk a rozdíl

Rozhodnutelnost vlastností CFL

Rozhodnutelné vlastnosti

- 1 test náležení řetězce w do jazyka CFG, tj. test na $w \in L(\mathcal{G})$
 - algoritmus CYK nebo pomocí PDA
- 2 test na neprázdnot jazyka CFG, tj. test na $L(\mathcal{G}) = \emptyset$
 - pokud $L(\mathcal{G}) = \emptyset$, pak gramatika po převedení do $\mathcal{G}$ do CNF nemá žádný neterminál
- 3 test na konečnost jazyka CFG
 - po převedení gramatiky $\mathcal{G}$ do CNF otestujeme zda pro nějaký neterminál platí $A \Rightarrow^* \alpha A \beta$, kde $\alpha, \beta \in (\Sigma \cup N)^*$, tedy jestli umíme z A odvodit další A (proč to funguje?)

Nerozhodnutelné vlastnosti

- ekvivalence dvou CFG, tj. test na $L(\mathcal{G}_1) = L(\mathcal{G}_2)$
- inkluze dvou CFG, tj. test na $L(\mathcal{G}_1) \subseteq L(\mathcal{G}_2)$
- prázdnot průniku dvou CFG, tj. test na $L(\mathcal{G}_1) \cap L(\mathcal{G}_2) = \emptyset$
- univerzalita, tj. test zda $L(\mathcal{G}) = \Sigma^*$
- jednoznačnost CFG a existence jednoznačné CFG pro daný CFL
- test na regularitu jazyka CFG

Vlastnosti deterministických CFL

Uzavřenost operací

- DCFL jsou **uzavřené na doplněk**
- ale nejsou uzavřené na zřetězení, sjednocení, průnik, Kleeneho uzávěr a ani reverz

Rozhodnutelnost vlastností

- ekvivalence dvou DCFL je **rozhodnutelná**
 - pro dvě DPDA P_1 a P_2 , lze rozhodnout platnost $L(P_1) = L(P_2)$
- test na regularitu DCFL a univerzalita jsou rozhodnutelné
- inkluze dvou DCFL a jednoznačnost jsou pořád nerozhodnutelné

Uzavřenost DFCL na doplněk: pomocné lemma

Intuice: pro doplněk DCFL L chceme použít podobnou konstrukci jako u DFA, tj. prohodit koncové a nekconcové stavy DPDA pro jazyk L . Nejprve ale musíme vyřešit dva problémy:

- 1 zamítající výpočty, které nedočtou celý vstup, protože skončí mimo všechny stavy (například kvůli neúplné přechodové funkci nebo nedostatku symbolů na zásobníku),
- 2 DPDA po přečtení celého řetězce může ještě provést ε -přechody (bez čtení vstupu) přes koncové i nekconcové stavy, tj. i po jejich prohození by takový DPDA slovo přijmul.

Lemma 1

Mějme DPDA D , pak existuje DPDA D' takový, že $L(D) = L(D')$ a zároveň:

- 1 D' vždy přečte celý svůj vstup,
- 2 pokud D' přejde do koncového stavu, tak zůstane v koncovém stavu dokud nepřečte další znak ze vstupu.



Uzavřenost DFCL na doplněk

Věta

Nechť L je DCFL, pak jeho doplněk $\bar{L}$ je také DCFL.

Idea důkazu

Mějme DPDA $D = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ s $L(D) = L$ a vlastnostmi z Lemma 1 (bez újmy na obecnosti uvažujeme modifikaci DPDA, který nemusí číst symbol zásobníku).

Nejprve upravíme D na D' tak, aby mohl vstoupit do koncového stavu jen jednou za každý přečtený znak ze vstupu a navíc jen těsně před přečtením dalšího znaku:

- stav označíme jako **čtecí**, pokud čte symbol pouze ze vstupu (a žádný ze zásobníku),
- každý stav, který čte zároveň ze vstupu i zásobníku, upravíme společně s jeho přechody:
 - přechod $\delta(q, a, Z) = (p, \gamma)$ nahradíme novým čtecím stavem q' , který je koncový pokud byl q koncový v D , a dvěma přechody: $\delta(q, \varepsilon, Z) = (q', \varepsilon)$ a dále $\delta(q', a, \varepsilon) = (p, \gamma)$,
- pokud zvolíme jako koncové stavy D' jen koncové čtecí stavy, pak platí $L(D') = L(D)$.

Výsledný DPDA $\bar{D}$, který rozpoznává doplněk L , tj. $L(\bar{D}) = \bar{L}$, získáme prohozením koncových a nekonečných čtecích stavů D' .



Pumping lemma (PL) pro bezkontextové jazyky

Pumping lemma (PL) pro CFL

Intuice: pokud je řetězec bezkontextového jazyka dostatečně dlouhý, pak v jeho derivaci musí nastat situace, kdy z nějakého neterminálu A odvodíme kromě dalších symbolů ten samý neterminál A . Tuto část derivace lze tedy libovolně opakovat.

Pumping lemma

Nechť L je bezkontextový jazyk, pak existuje číslo $n \in \mathbb{N}$ takové, že každý řetězec $s \in L$ délky aspoň n lze rozdělit na pět částí $s = uvxyz$, které splňují:

- 1 $|vy| > 0$,
- 2 $|vxy| \leq n$,
- 3 $uv^ixy^iz \in L$ pro všechna $i \geq 0$.

- tvrzení je tvaru implikace: pokud je L bezkontextový pak musí něco splňovat. . .
 - pokud jazyk není bezkontextový, pak z pro něj z lemma nic konkrétního nevyplývá
 - kontrapozicí dostaneme: pokud L nesplňuje podmínky z PL, pak L není bezkontextový
 - existuje jazyk, který není bezkontextový a zároveň splňuje všechny podmínky z lemma
- postup použití PL pro CFL při důkazu sporem je stejný jako pro RL

Vztah délky řetězce a výšky jeho derivačního stromu

Lemma 2

Nechť $\mathcal{G}$ je CFG a k je maximální délka pravé strany jejích pravidel, pak platí:

- 1 libovolný derivační strom v $\mathcal{G}$ s výškou h derivuje řetězec délky nejvýše k^h ,
- 2 všechny derivační stromy řetězce délky $k^h + 1$ mají výšku nejméně $h + 1$.

Důkaz

Mějte libovolný derivační strom T v $\mathcal{G} = (N, \Sigma, P, S)$ nějakého řetězce $s \in L(\mathcal{G})$:

- kořen T má nejvýše k potomků (přepsání pravidlem $S \rightarrow \alpha$, kde $|\alpha| \leq k$),
- proto je nejvýše k listů se vzdáleností 1 od kořene,
- každý z těchto potomků může mít také nejvýše k svých potomků,
- analogicky tedy platí, že pro vzdálenost 2 od kořene můžeme mít nejvýše k^2 listů atd.,
- pokud tedy T má výšku h , pak má nejvýše k^h listů,
- proto platí 1, tj. $|s| \leq k^h$, jelikož derivace T odpovídá zřetězení všech jeho listů,
- důkaz 2 je analogický.



Důkaz PL

Důkaz (část 1/3)

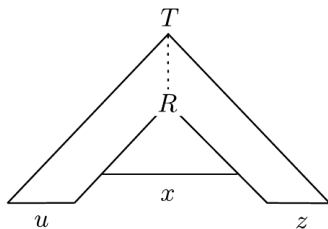
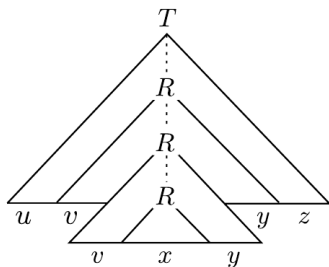
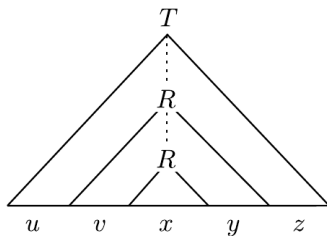
Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG generující jazyk L .

- Zvolíme konstantu z PL jako $n = k^{|N|+1}$, kde k je maximální délka pravé strany pravidel $\mathcal{G}$,
- dále zvolíme libovolný řetězec $s \in L$, pro který platí $|s| \geq n$,
- každý derivační strom řetězce s má výšku nejméně $|N| + 1$ (dle Lemma 2).

Ze všech derivačních stromů řetězce s zvolíme strom T , který má nejméně uzlů:

- označme ρ nejdelší cestu v T z kořene do listu (má délku nejméně $|N| + 1$),
- proto má ρ nejméně $|N| + 2$ uzlů (obsahují jeden terminál a zbytek neterminálů),
- $\mathcal{G}$ má ale jen $|N|$ neterminálů, proto se v ρ nějaký neterminál opakuje,
- vybereme neterminál R , který se opakuje v posledních $|N| + 1$ uzlech ρ .

Rozdělíme $s = uvxyz$ dle obrázku a ukážeme, že platí podmínky **1**, **2** a **3** z PL.



Nahrazení v derivačním stromu.
 (převzato z *M. Sipser: Introduction to the Theory of Computation*)

Důkaz (část 2/3)

Nejprve ukážeme platnost podmínky 3, tedy že $uv^ixy^iz \in L$, pro $i \geq 0$:

Každý výskyt R v T má vlastní podstrom, který generuje část s :

- první výskyt R od kořene generuje (větší) podstrom řetězce vxy ,
- poslední výskyt R v daném podstromu generuje (menší) podstrom řetězce x ,

Protože oba podstromy generuje stejný neterminál, tak je možné jeden za druhý vzájemně nahradit a pořád dostaneme validní derivační strom v $\mathcal{G}$:

- opakovaným nahrazením menšího podstromu tím větším dostaneme strom řetězce uv^ixy^iz , pro libovolné $i > 0$,
- nahrazením většího podstromu tím menším dostaneme strom řetězce uxz (tedy případ kdy $i = 0$).

Důkaz (část 3/3)

Pro platnost podmínky 1 musíme ukázat, že v nebo y je neprázdný řetězec:

- předpokládejme, že v i y jsou prázdné, a tedy $s = uvxyz = uxz$,
- pokud v takovém případě nahradíme větší podstrom (generující vxy) tím menším (generující x), pak by výsledný strom měl méně uzlů než T a pořád by generoval s ,
- to je ale ve sporu s tím, že T je strom řetězce s , který má nejméně uzlů,
- proto je v nebo y neprázdný řetězec.

Pro platnost podmínky 2 musíme ukázat, že $|vxy| \leq n$:

- jako ρ jsme zvolili nejdelší cestu v T , ale nyní uvažujme jen posledních $N + 1$ uzlů,
- R jsme zvolili tak, že se opakuje v posledních $|N| + 1$ uzlech ρ ,
- přičemž první výskyt R (v této části ρ) generuje podstrom řetězce vxy ,
- tento podstrom má výšku nejvýše $|N| + 1$ (délka dané části ρ),
- proto může generovat řetězec s délkou nejvýše $k^{|N|+1} = n$ (z Lemma 2).



Příklad: jazyk $a^k b^k c^k$ a PL

Věta

Jazyk $L = \{a^k b^k c^k \mid k \geq 0\}$ není bezkontextový.

Důkaz

Použijeme důkaz sporem: předpokládáme, že L je bezkontextový a najdeme spor s PL.

- Dle PL existuje konstanta $n \in \mathbb{N}$,
- zvolíme řetězec $s = a^n b^n c^n$, který patří do jazyka L a $|w| \geq n$,
- nyní ukážeme, že s nelze rozdělit na $s = uvxyz$ tak, aby byly splněny podmínky z PL,
- z podmínky **1** z PL dostáváme, že alespoň jeden z řetězců v a y je neprázdný,
- uvažujme dva případy podle rozložení znaků ve v a y :
 - 1** v a y obsahují pouze znaky nejvýše jednoho druhu, například $v = a^\ell$ a $y = b^m$ (tak aby byly splněny **1** a **2** z PL), potom ale $uv^2xy^2z \notin L$, protože obsahuje méně znaků c ,
 - 2** pokud v nebo y obsahuje více než jeden druh znaků, pak řetězec $uv^i xy^i z$ může obsahovat stejný počet všech znaků a , b a c , ale znaky nebudou ve správném pořadí, a proto také $uv^2 xy^2 z \notin L$ (navíc neplatí **2** z PL, protože $|vxy| = |a^\ell b^n c^m| > n$),
- jeden z těchto případů musí nastat, přičemž v obou dostáváme spor s PL. □

Příklad: jazyk $a^k b^\ell c^k d^\ell$ a PL

Věta

Jazyk $L = \{a^k b^\ell c^k d^\ell \mid k, \ell \geq 0\}$ není bezkontextový.

Důkaz

Použijeme důkaz sporem: předpokládáme, že L je bezkontextový a najdeme spor s PL.

- Dle PL existuje konstanta $n \in \mathbb{N}$, a nejprve zvolíme řetězec $s' = a^n b c^n d$,
- s' sice splňuje podmínku $|s'| \geq n$, ale **spor na něm ukázat nelze**,
- takový řetězec s' lze totiž rozdělit následově:

$$u = a^{n-1} \quad v = a \quad x = b \quad y = c \quad z = c^{n-1} d$$

- přičemž jsou splněny všechny podmínky z PL, **1** a **2** platí triviálně, a pro **3** dostaneme, že pro každé $i \geq 0$ platí: $a^{n+i-1} b c^{n+i-1} d \in L$.
- Nyní zvolíme řetězec $s = a^n b^n c^n d^n$, na kterém **lze ukázat spor** (postupujeme podobně jako u přechozího příkladu, přičemž využijeme podmínku **2** z PL).

