3.1.7 Nevypouštěcí gramatiky. Je zřejmé, že každý jazyka typu 1 je také jazyk typu 0. Obdobně, každý jazyk typu 3 je jazyk typu 2. Pro bezkontextové jazyky (jazyky typu 2) již není zřejmé, že jsou také kontextové (typu 1). Je to proto, že kontextové gramatiky mají omezení na pravidla, kde pravá strana je prázdné slovo, což pro bezkontextové gramatiky neplatí.

Navíc, v dalším textu ukážeme, že i v jiných případech (Chomského normální tvar, pumping lemma pro bezkontextové jazyky, algoritmus CYK pro rozhodnutí, zda dané slovo je generováno gramatikou) je potřeba zakázat pravidla s pravou stranou ε. K tomu se hodí pojem nevypouštěcí gramatiky – jedná se o gramatiku, která má pouze pravidle s pravou stranou délky aspoň 1.

Definice. Gramatiku $\mathcal{G} = (N, \Sigma, S, P)$ nazveme *nevypouštěcí*, jestliže neobsahuje žádné pravidlo typu $\alpha \to \varepsilon$. □

3.1.8 Tvrzení. Ke každé bezkontextové gramatice $\mathcal{G}$ existuje nevypouštěcí gramatika $\mathcal{G}_1$ taková, že

$$L(\mathcal{G}_1) = L(\mathcal{G}) \setminus \{\varepsilon\}.$$

Nástin důkazu: Pro danou bezkontextovou gramatiku $\mathcal{G} = (N, \Sigma, S, P)$ nejprve najdeme množinu všech neterminálů U, ze kterých je možné vygenerovat prázdné slovo. To uděláme tak, že postupně vytváříme množiny U_0, $U_1, \dots$ kde:

Do množiny U_0 dáme všechny neterminály, které se přímo přepíší na prázdné slovo, tj.

$$U_0 = \{X\,;\, X \to \varepsilon \in P\},$$

Máme-li zkonstruovánu množinu U_i, pak k množině U_i přidáme všechny neterminály, pro které existuje pravidlo, jehož pravá strana je tvořena jen neterminály z množiny U_i; tj.

$$U_{i+1} = U_i \cup \{X\,;\, X \to \alpha \in P, \text{ pro } \alpha \in U_i^\star\},$$

Protože

$$U_0 \subseteq U_1 \subseteq \dots \subseteq U_i \subseteq \dots N, \quad \subseteq \text{ a } N \text{ je konečná,}$$

existuje k takové, že $U_k = U_{k+1}$. Pak $U = U_k$ je hledaná množina neterminálů.

Nyní definujeme pravidla P_1 gramatiky $\mathcal{G}_1 = (N, \Sigma, S, P_1)$ takto: Je-li $X \to \alpha$ pravidlo z P a $\alpha \neq \varepsilon$, najdeme v α všechny výskyty neterminálů z množiny U; tj přepíšeme pravidlo na tvar

$$X \to \beta_1 X_1 \beta_2 X_2 \dots X_{k-1} \beta_{k-1} X_k \beta_k,$$

kde $X_i \in U$ a slova β_i již neobsahují neterminály z množiny U.

Do P_1 dáme všechna pravidla $X \to \gamma$, kde γ vzniklo z α vynecháním některých (třeba i žádného nebo všech) neterminálů z množiny U.

Pravidla $X \to \varepsilon$ z P do pravidel P_1 nedáme.

O gramatice $\mathcal{G}_1 = (N, \Sigma, S, P_1)$ se dokáže $L(\mathcal{G}_1 = L(\mathcal{G}) \setminus \{\varepsilon\}$, tj. že generuje všechna neprázdná slova jazyka $L(\mathcal{G})$.

3.1.9 Důsledek. Označme $\mathcal{L}_i$ třídu jazyků typu i. Pak platí:

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0.$$

□

Důkaz. Abychom mohli tvrdit, ze každý bezkontextový jazyk je také kontextový, musíme být schopni ke každé bezkontextové gramatice $\mathcal{G}$ sestrojit gramatiku $\mathcal{G}_2$, která je buď nevypouštěcí (když $\mathcal{G}$ negeneruje prázdné slovo) nebo jediné pravidlo s pravou stranou ε je pravidlo, které má na levé straně startovací symbol a tento startovací symbol se již neobjevuje na pravé straně žádného pravidla.

Jestliže tedy $\mathcal{G}$ generuje prázdné slovo, kontextovou gramatiku zkonstruujeme takto: K nevypouštěcí gramatice $\mathcal{G}_1$ přidáme nový startovací symbol S_1 a k pravidlům P_1 přidáme ještě dvě pravidla

$$S_1 \to \varepsilon, \quad S_1 \to S.$$

Gramatika $\mathcal{G}_2$ generuje jazyk $L(\mathcal{G})$ a splňuje podmínky kontextové gramatiky.

Ukážeme si celý postup na příkladě.

3.1.10 Příklad. Je dána gramatika $\mathcal{G} = (\{A, S\}, \{a, b, c\}, S, P)$ pravidly

$$S \to aSc \,|\, A \quad A \to bAc \,|\, \varepsilon.$$

Ke gramatice $\mathcal{G}$ najdeme nevypouštěcí gramatiku $\mathcal{G}_1$, která generuje jazyk $L(\mathcal{G}) \setminus \{\varepsilon\}$ a kontextovou gramatiku $\mathcal{G}_2$, pro kterou $L(\mathcal{G}_2) = L(\mathcal{G})$.

Řešení: Použijeme postup z důkazu věty 3.1.8 a důsledku 3.1.9.

Množina U_0 je rovna $\{A\}$, dále množina $U_1 = U_0 \cup \{S\}$, protože $S \to A$ je pravidlo gramatiky $\mathcal{G}$ a $A \in U_0$. Proto $U = \{A, S\}$ (což je množina všech neterminálů). (Uvědomte si, že v gramatice $\mathcal{G}$ je možné z S vygenerovat ε a to derivací $S \Rightarrow A \Rightarrow \varepsilon$.)

Hledaná gramatika $\mathcal{G}_1$ má pravidla:

$$S \to aSc \,|\, ac \,|\, A \quad A \to bAc \,|\, bc.$$

Všimněte si, že $L(\mathcal{G}_1) = \{a^i b^j c^{i+j} \mid i + j > 0\}$, kdežto $L(\mathcal{G}) = \{a^i b^j c^{i+j} \mid i + j \geq 0\}$.

Kontextová gramatika $\mathcal{G}_2$ je gramatika $(N \cup \{S_1\}, \Sigma, S_1, P_2)$, kde pravidla jsou

$$S_1 \to S \,|\, \varepsilon,\ S \to aSc \,|\, ac \,|\, A,\ A \to bAc \,|\, bc.$$

□

3.1.11 V tomto odstavci ukážeme, že jméno regulární jazyky není zavádějící; tj. že jazyk je přijímán konečným automatem právě tehdy, když je generován gramatikou typu 3.

Tvrzení 1. Každý regulární jazyk je generován některou gramatikou typu 3, tj. některou regulární gramatikou.

Přesněji: je-li L regulární jazyk, pak existuje regulární gramatika $\mathcal{G}$ taková, že $L = L(\mathcal{G})$. □

Nástin důkazu: Je-li L regulární jazyk, pak existuje DFA M, který L přijímá. Definujme gramatiku $\mathcal{G} = (N, \Sigma, S, P)$ takto:

- Množina neterminálů N je množina stavů Q automatu M,
- startovací symbol S je počáteční stav q_0 automatu M,
- P obsahuje pravidla dvou typů:
  1. $q \to ap$ pro $\delta(q, a) = p$,
  2. $q \to \varepsilon$ pro všechny koncové stavy q.

Není těžké ukázat, že takto definovaná gramatika generuje jazyk $L(M)$.

Tvrzení 2. Gramatiky typu 3, tj. regulární gramatiky, přijímají pouze regulární jazyky.

Přesněji, je-li $\mathcal{G}$ gramatika typu 3, pak jazyk $L(\mathcal{G})$ je regulární. □

Nástin důkazu: Ke gramatice s pravidly $X \to wY$ a $X \to w$ pro $w \in \Sigma^\star$ nejprve najdeme gramatiku generující stejný jazyk, která má pravidla pouze typu $X \to aY$ a $X \to \varepsilon$, kde $a \in \Sigma \cup \{\varepsilon\}$. Toho dosáhneme snadno přidáním nových neterminálů (např. pravidlo $X \to abY$ nahradíme pravidly $X \to aZ$ a $Z \to bY$; neterminál Z musí být nový).

K takovéto gramatice sestrojíme automat (obecně NFA s ε-přechody) takto:

- Množina stavů je množina neterminálů.
- Počáteční stav je startovací symbol.
- $Y \in \delta(X, a)$ právě tehdy, když $X \to aY$ je pravidlo.
- Koncové stavy jsou ty neterminály, pro které je $X \to \varepsilon$ pravidlo.

Není těžké ukázat, že takto definovaný automat přijímá jazyk $L(\mathcal{G})$.

3.2 Bezkontextové gramatiky

Připomeňme, že bezkontextová gramatika (CF gramatika) je gramatika $\mathcal{G} = (N, \Sigma, S, P)$, která obsahuje pouze pravidla typu

$$A \to \gamma, \quad \text{kde } \gamma \in (N \cup \Sigma)^\star \text{ a } A \text{ je neterminál.}$$

Dále připomeňme, že ke každé CF gramatice $\mathcal{G}$ existuje nevypouštěcí CF gramatika $\mathcal{G}_1$, která generuje všechna neprázdná slova z $L(\mathcal{G})$, tj. platí $L(\mathcal{G}_1) = L(\mathcal{G}) \setminus \{\varepsilon\}$.

3.2.1 Následující tvrzení je sice jednoduché, ale velmi podstatné pro studium bezkontextových gramatik. Velmi neformálně řečeno, říká, že jednotlivé části odvození jsou nezávislé a „do sebe nezasahují“. Přesněji:

Tvrzení. Máme dánu bezkontextovou gramatiku $\mathcal{G} = (N, \Sigma, S, P)$ a v ní derivaci

$$S \Rightarrow^\star_{\mathcal{G}} \alpha A \beta \Rightarrow^\star_{\mathcal{G}} \gamma,$$

pro $\alpha, \beta, \gamma \in (\Sigma \cup N)^\star$ a $A \in N$.

Pak existují slova $\varphi, \psi, \mu \in (\Sigma \cup N)^\star$ taková, že

$$\gamma = \varphi\psi\mu, \quad \text{kde} \quad \alpha \Rightarrow^\star_{\mathcal{G}} \varphi, \quad A \Rightarrow^\star_{\mathcal{G}} \psi, \quad \beta \Rightarrow^\star_{\mathcal{G}} \mu.$$

3.2.2 Levá derivace, levé odvození. Předchozí tvrzení říká, že pro bezkontextovou gramatiku můžeme pravidla přehazovat, jediné, co musíme ohlídat, je, aby pravidlo bylo možné použít. Můžeme proto pravidla uspořádat (přehodit) tak, abychom přepisovali vždy ten nejvíc levý neterminál. A takové derivaci říkáme levá derivace.

Definice. Přímé odvození se nazývá *levé*, jestliže se přepisuje ten neterminál, který je nejvíc „vlevo“, tj. $u\,A\,\beta \Rightarrow_{\mathcal{G}} u\,\delta\,\beta$, kde $u \in \Sigma^\star$ a $A \to \delta$ je pravidlo gramatiky.

Derivace (odvození) se nazývá *levá derivace* (*levé odvození*), jestliže se skládá pouze z levých přímých odvození. □

Obdobně definujeme pravé přímé odvození a pravou derivaci.

Tvrzení. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Pak pro každou derivaci $S \Rightarrow^\star_{\mathcal{G}} w$ existuje levá derivace terminálního w z S taková, že používá stejná pravidla jako původní derivace (pouze možná v jiném pořadí). □

3.2.3 Derivační strom (parse tree). Derivace v bezkontextové gramatice si můžeme představit také jako stromy, budeme jim říkat derivační stromy. Výhoda derivačních stromů je v tom, že v nich „není zachyceno“ pořadí použití jednotlivých pravidel, jenom jejich struktura.

Definice. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$. *Derivační strom* (též *parse tree*) je kořenový strom, takový, že:

1. Každý vrchol, který není list, je ohodnocen neterminálem.
2. Každý list je ohodnocen terminálem, neterminálem nebo prázdným slovem ε. V případě, že je list ohodnocen prázdným slovem ε, je to jediný následník (svého předchůdce).
3. Jestliže některý vrchol, který není list, je ohodnocen neterminálem A a má následníky (v pořadí zleva doprava) $X_1, X_2, \ldots, X_k$, $X_i \in N \cup \Sigma$, pak $A \to X_1 X_2 \ldots X_k$ je pravidlo gramatiky $\mathcal{G}$.

Řekneme, že derivační strom *dává*, nebo *má za výsledek* slovo γ, jestliže γ je ohodnocení listů derivačního stromu (čteno zleva doprava). □

3.2.4 Je zřejmé, že pro každou derivaci existuje derivační strom, který derivaci odpovídá. Není již pravda, že pro dvě různé derivace jsou derivační stromy také různé; ano, pro dvě derivace, které se liší pouze pořadím použití jednotlivých pravidel, je derivační strom stejný. Ovšem jednoznačnost přece jenom existuje, ale je mezi derivačními stromy a levými derivacemi (pravými derivacemi).

Tvrzení.

1. Pro každou derivaci $S \Rightarrow^\star_{\mathcal{G}} w$ existuje derivační strom s výsledkem w.
2. Ke každému derivačnímu stromu s výsledkem w existuje aspoň jedna derivace $S \Rightarrow^\star_{\mathcal{G}} w$ (takových derivací může být více).
3. Ke každému derivačnímu stromu s výsledkem w existuje právě jedna levá (právě jedna pravá) derivace slova w z S.

3.2.5 Jednoznačné a víceznačné bezkontextové gramatiky. Problému, jak ke slovu generovanému danou CF gramatikou najít jeho derivační strom, se budeme podrobněji věnovat v dalších přednáškách. Pro rekonstrukci daného derivačního stromu je výhodné, když takový derivační strom existuje pouze jediný. A to je případ tzv. jednoznačné gramatiky.

Definice. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Řekneme, že $\mathcal{G}$ je *jednoznačná*, jestliže pro každé slovo w generované gramatikou $\mathcal{G}$ existuje jediný derivační strom s výsledkem w (tj. existuje jediná levá derivace w z S).

V opačném případě mluvíme o *víceznačné* gramatice. □

3.2.6 Víceznačný jazyk. Je zřejmé, že pro jeden bezkontextový jazyk mohou existovat dvě CF gramatiky a to tak, že jedna je víceznačná a druhá je jednoznačná. To jen říká, že v prvním případě jsme si nevybrali tu nejlepší CF gramatiku. Existují ale bezkontextové jazyky, pro které **neexistuje** jednoznačná gramatika.

Definice. Bezkontextový jazyk L se nazývá *víceznačný* (též *podstatně víceznačný*), jestliže **každá** bezkontextová gramatika, která ho generuje, je víceznačná. □

Například jazyk $L = \{a^i b^j c^k d^l \mid i = j \text{ nebo } k = l\}$ je víceznačný.

3.2.7 Redukovaná bezkontextová gramatika. Podobně jako jsme zavedli redukovaný automat, zavedeme též redukovanou CF gramatiku. A to jako gramatiku, která „nemá zbytečné neterminály“. Zbytečné neterminály jsou dvou typů; jsou to jednak neterminály, ze kterých nelze vygenerovat terminální slovo, jednak neterminály, na které „se nedostaneme“ ze startovacího symbolu. Nebude však platit, že každý bezkontextový jazyk má jedinou redukovanou CF gramatiku, která ho generuje. Jinými slovy, existují dvě různé gramatiky (i např. s jiným počtem neterminálů), které generují ten samý jazyk.

Definice. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$, která generuje aspoň jedno slovo. Řekneme, že $\mathcal{G}$ je *redukovaná*, jestliže splňuje tyto dvě podmínky:

1. Ke každému neterminálu A existuje aspoň jedno terminální slovo w takové, že $A \Rightarrow^\star_{\mathcal{G}} w$.
2. Ke každému neterminálu A existují slova $\alpha, \beta \in (N \cup \Sigma)^\star$ tak, že
$$S \Rightarrow^\star_{\mathcal{G}} \alpha\, A\, \beta.$$

□

V příštím odstavci uvedeme postup, který pro každý neprázdný bezkontextový jazyk zajišťuje aspoň jednu redukovanou gramatiku, která ho generuje.

3.2.8 Tvrzení. Ke každé bezkontextové gramatice $\mathcal{G} = (N, \Sigma, S, P)$, pro kterou $L(\mathcal{G}) \neq \emptyset$, existuje redukovaná gramatika $\mathcal{G}_1$ taková, že $L(\mathcal{G}_1) = L(\mathcal{G})$. □

Tvrzení dokážeme algoritmem, který pro danou CF gramatiku rozhodne, zda generuje aspoň jedno slovo; a v případě, že ano, najde redukovanou gramatiku generující stejný jazyk.

Algoritmus redukce CFG. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$.

1. Sestrojíme množinu $V = \{A \mid A \in N, A \Rightarrow^\star_{\mathcal{G}} w, w \in \Sigma^\star\}$; postupujeme indukcí:

   Nejprve položíme
   $$V_1 = \{A \mid \text{existuje } w \in \Sigma^\star \text{ takové, že } A \to w \in P\}.$$
   Pak konstruujeme
   $$V_{i+1} = V_i \cup \{A \mid \text{existuje } \alpha \in (V_i \cup \Sigma)^\star \text{ tak, že } A \to \alpha \in P\}.$$
   Platí
   $$V_1 \subseteq V_2 \subseteq \ \dots \ \subseteq N.$$
   Protože N je konečná, existuje n takové, že $V_n = V_{n+1}$. Položíme $V = V_n$.

   Jestliže $S \notin V$, pak $L(\mathcal{G}) = \emptyset$ a redukovaná gramatika ke gramatice $\mathcal{G}$ neexistuje.

   V opačném případě definujeme $\mathcal{G}' = (V, \Sigma, S, P')$: do P' dáme ta pravidla z P, která obsahují pouze neterminály z množiny V (a to jak na levé, tak na pravé straně).

2. Pro gramatiku $\mathcal{G}' = (V, \Sigma, S, P')$ sestrojíme (opět indukcí) množinu
   $$U = \{A \mid A \in V, \text{ existují } \alpha, \beta \in (V \cup \Sigma)^\star \text{ tak, že } S \Rightarrow^\star_{\mathcal{G}} \alpha\, A\, \beta\}.$$
   Nejprve položíme
   $$U_0 = \{S\},$$
   pak konstruujeme
   $$U_{i+1} = U_i \cup \{A \mid \text{existují } B \in U_i, \alpha, \beta \in (V \cup \Sigma)^\star \text{ tak, že } B \to \alpha\, A\, \beta \in P'\}.$$
   Tj. do množiny U_{i+1} dáme všechny neterminály, které se objevují na pravé straně některého pravidla z P, kde na levé straně pravidla jsou neterminály pouze z U_i.

   Platí
   $$U_0 \subseteq U_1 \subseteq U_2 \subseteq \ \dots \ \subseteq V.$$
   Proto existuje m takové, že $U_m = U_{m+1}$. Položíme $U = U_m$.

   Hledaná gramatika je gramatika $\mathcal{G}_1 = (U, \Sigma, S, P_1)$, kde P_1 je množina všech pravidel z P' (a tedy i z P), které obsahují neterminály pouze z množiny U.

Platí: gramatika $\mathcal{G}_1 = (U, \Sigma, S, P_1)$ je redukovaná a generuje stejný jazyk jako původní gramatika $\mathcal{G} = (N, \Sigma, S, P)$.

3.2.9 Poznámky.

- Je obtížné ke dvěma bezkontextovým gramatikám zjistit, zda generují stejný jazyk. Redukce gramatik nám k rozhodnutí nepomůže.
- Kroky předchozího postupu nelze zaměnit. Kdybychom nejprve hledali množinu neterminálů U a pak teprve z ní vybírali ty neterminály, ze kterých je možné odvodit terminální slovo, výsledná gramatika by nemusela splňovat druhou podmínku z 3.2.7 a tudíž být redukovaná.