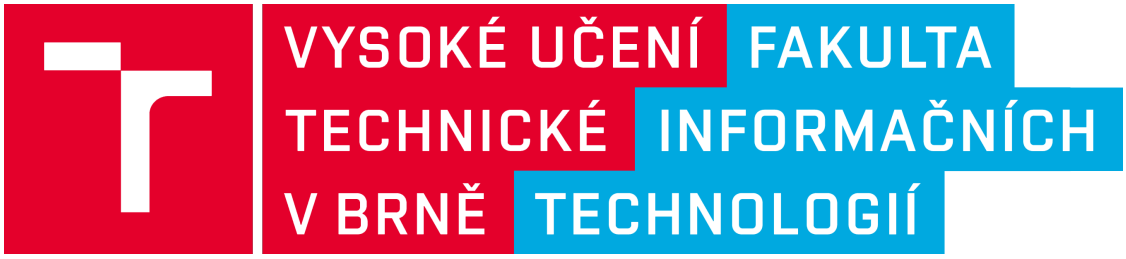

Teoretická informatika 2023/2024

Úkol 2

David Chocholatý (xchoch09)

Brno, 26. listopadu 2023

1 Příklad 1

Rozhodněte a dokažte, zda je jazyk $L_{prime} = \{w \in \Sigma^* : |w| \text{ je prvočíslo}\}$ nad abecedou $\Sigma = \{a, b\}$ bezkontextový. Pro důkaz bezkontextovosti sestavte gramatiku či zásobníkový automat. Pro důkaz nebezkontextovosti použijte pumping lemma nebo uzávěrové vlastnosti.

1.1 Řešení

Jazyk L_{prime} je jazyk, který obsahuje řetězce nad abecedou $\Sigma = \{a, b\}$ takové, že jejich délka je rovna nějakému prvočíslu p, $p \in \mathbb{N} \wedge p > 1$. Na základě definice je prvočíslo takové přirozené číslo větší než 1, které je dělitelné pouze číslem jedna a samo sebou. Abychom o přirozeném čísle p mohli rozhodnout, jestli je či není prvočíslo, je nutné ověřit, zda není dělitelné nějakým přirozeným číslem z rozsahu $\langle 2, \sqrt{p} \rangle$. To by ovšem znamenalo pro zásobníkový automat M, který by přijímal jazyk L_{prime}, že by M musel disponovat i druhým zásobníkem. Dle definice zásobníkového automatu M ovšem obsahuje pouze jeden zásobník[1].

Na základě výše uvedeného je rozhodnuto, že $L_{prime} \notin \mathcal{L}_2$. To bude dokázáno pomocí Pumping lemmatu pro bezkontextové jazyky.

Lemma 1.1. $L_{prime} = \{w \in \{a, b\}^* : |w| \text{ je prvočíslo}\} \notin \mathcal{L}_2$.

Důkaz. Důkaz sporem pomocí Pumping lemmatu pro bezkontextové jazyky. Předpokládejme, že $L_{prime} \in \mathcal{L}_2$. Pak dle Pumping lemmatu pro bezkotextové jazyky platí, že

$$\exists k > 0 : \forall z \in \Sigma^* : z \in L_{prime} \wedge |z| \geq k \implies$$
$$\exists u, v, w, x, y \in \Sigma^* : z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L_{prime}.$$

Uvažujme libovolné $k > 0$ a zvolme $z = a^{p-2}b^2$, kde p, $p \in \mathbb{N} \wedge p > 1$, je prvočíslo takové, že $p \geq k + 2$. Platí, že $z \in L_{prime} \wedge |z| = p - 2 + 2 = p \geq k$. Pro každé rozdělení slova z takové, že $z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k$, nastane následující nezávisle na zastoupení symbolu a nebo b.

Zvolme $i = |uwy|$. Na základě podmínek $|vwx| \leq k$ a $|z| \geq k + 2$ vyplývá, že $i \geq 2$. Platí, že

$$|uv^iwx^iy| = |v^ix^i| + |uwy| = i * |vx| + |uwy|. \tag{1}$$

Z podmínek $i \geq 2$ a $vx \neq \varepsilon$ dohromady vyplývá, že $i * |vx| \geq 2$. Dále lze výsledek rovnice 1 rozepsat ve tvaru

$$i * |vx| + |uwy| = |uwy| * (1 + |vx|), \tag{2}$$

kde $|uwy| \geq 2 \wedge (1 + |vx|) \geq 2$. Na základě rovnice 2 je porušena podmínka, že $|uv^iwx^iy| = p', p' \in \mathbb{N} \wedge p > 1$, kde p' je nějaké prvočíslo. Tudíž $uv^iwx^iy \notin L_{prime}$ a dochází ke sporu s předpokladem, že $L_{prime} \in \mathcal{L}_2$. □

[1] Pokud by zásobníkový automat M obsahoval dva zásobníky, byla by jeho výpočetní síla rovna Turingovým strojům, viz [1].

2 Příklad 2

Uvažujte jazyk $L_{BKG} = \{\langle M\rangle\#\langle G\rangle \ : \ M \text{ je TS, } G \text{ je BKG, } L(G) \subseteq L(M)\}$ nad abecedou $\{0, 1, \#\}$ ($\langle G\rangle$ je kódování BKG G do binárního řetězce). Rozhodněte a dokažte, zda jazyk L_{BKG} (a) je rozhodnutelný, (b) je nerozhodnutelný, ale částečně rozhodnutelný, (c) není ani částečně rozhodnutelný. Pro důkaz (a) popište princip činnosti TS, který jazyk rozhoduje (není třeba sestavovat detailně přechodový diagram), pro důkaz (b) nebo (c) použijte redukci.

2.1 Řešení

Nejprve je nutné uvážit konečnost, případně nekonečnost ověření příslušnosti či nepříslušnosti do jazyka L_{BKG}. Ověření nepříslušnosti do jazyka je konečné (*konečný certifikát nepříslušnosti*), protože stačí najít jediný řetězec $w \in \Sigma^*$ takový, pro který platí, že $w \in L(G) \wedge w \notin L(M)$. Naopak ověření příslušnosti do jazyka je nekonečné (*nekonečný certifikát nepříslušnosti*), protože je nutné pro všechny řetězce, pro které platí, že $w' \in \Sigma^*, w' \in L(G)$, testovat, zda $w' \in L(M)$. Počet všech takových řetězců w' konečné délky může být spočetně nekonečný, a tudíž i ověření příslušnosti do jazyka je nekonečné.

Na základě výše uvedeného je rozhodnuto, že jazyk L_{BKG} není ani částečně rozhodnutelný.

Lemma 2.1. $L_{BKG} = \{\langle M\rangle\#\langle G\rangle \ : \ M \text{ je TS, } G \text{ je BKG, } L(G) \subseteq L(M)\}$ nad abecedou $\{0, 1, \#\}$. Jazyk L_{BKG} není ani částečně rozhodnutelný.

Důkaz. Důkaz pomocí redukce co-$HP \leq L_{BKG}$, kde jazyk co-HP je definován jako

$$\text{co-}HP = \{\langle M_{\text{co-}HP}\rangle\#\langle w\rangle \mid M_{\text{co-}HP} \text{ nezastaví při } w\}.$$

Požadovaná redukce je funkce $\sigma : \{0, 1, \#\}^* \to \{0, 1, \#\}^*$ definovaná následovně:

$$\sigma(\langle M_{\text{co-}HP}\rangle\#\langle w\rangle) = \langle M\rangle\#\langle G\rangle.$$

Funkce σ vrací kód bezkontextové gramatiky (dále jen BKG) G, pro který platí $L(G) = \{a^n b^n \mid n \geq 0\}$. Pokud vstup $\langle M_{\text{co-}HP}\rangle\#\langle w\rangle$ není validní instance co-HP, pak funkce σ vrací kód Turingova stroje (dále jen TS) M takový, že $L(M) = \emptyset$ (tudíž $\langle M\rangle\#\langle G\rangle \notin L_{BKG}$), jinak vrací kód TS M, který pracuje následovně.

Nejdříve TS M spočítá délku řetězce w' definovaného na vlastní vstupní pásce, přičemž $|w'| = m \geq 0$. Tato hodnota je následně uložena na novou pásku. Poté TS M spustí simulaci stroje $M_{\text{co-}HP}$ na w pro maximálně m kroků (na další pásce).

- Pokud stroj $M_{\text{co-}HP}$ v rámci m kroků **zastaví**, pak M začne cyklit.
- Pokud stroj $M_{\text{co-}HP}$ v rámci m kroků **nezastaví**, pak M přijme vlastní vstup.

Tudíž pro M platí následující:

$$\langle M_{\text{co-}HP}\rangle\#\langle w\rangle \notin \text{co-}HP \implies L(M) = \emptyset \implies \langle M\rangle\#\langle G\rangle \notin L_{BKG}.$$
$$\langle M_{\text{co-}HP}\rangle\#\langle w\rangle \in \text{co-}HP \implies L(M) = \Sigma^* \implies \langle M\rangle\#\langle G\rangle \in L_{BKG},$$

Je zřejmé, že uvedenou konstrukci BKG G a TS M lze implementovat pomocí úplného TS, a tedy funkce σ je totální, rekurzivně vyčíslitelná funkce. □

3 Příklad 3

Plný binární strom je graf $T = (V, E_L, E_R)$, kde V je množina vrcholů, $E_L, E_R : V \to V$ jsou injektivní totální zobrazení *levého* a *pravého* následníka taková, že $img(E_L) \cap img(E_R) = \emptyset$, kde $img(f)$ značí *obor hodnot* zobrazení f. Dále platí, že existuje právě jeden vrchol $r \in V$ takový, že $\nexists u \in V : (u, r) \in E_L \cup E_R$ a $\forall v \in V : (r, v) \in (E_L \cup E_R)^{RT}$, kde A^{RT} značí reflexivně-tranzitivní uzávěr relace A.

(a) Dokažte, že množina V je spočetně nekonečná (tj. existuje bijekce mezi V a $\mathbb{N}$).

(b) Nechť *obarvení stromu* T je funkce, která přiřazuje každému vrcholu $v \in V$ barvu $c(v) \in \{red, black\}$. Dokažte pomocí diagonalizace, že množina všech obarvení stromu T je nespočetně nekonečná.

3.1 Řešení

(a)

Teorém 3.1. Nechť $T = (V, E_L, E_R)$ je plný binární strom, pak množina V je spočetně nekonečná.

Důkaz. Nechť $T = (V, E_L, E_R)$ je plný binární strom výšky ω, kde symbol ω značí množinu nezáporných celých čísel $\{0, 1, 2, ...\}$. Dále nechť T_n značí množinu všech uzlů v úrovni $n \in \mathbb{N}, n < \omega$, přičemž počet uzlů v dané úrovni n je dán vztahem 2^n. Tudíž počet uzlů v jednotlivých úrovních $T_0, T_1, ..., T_n$ je konečný. Sjednocením všech úrovní stromu získáme množinu všech uzlů stromu, V, přičemž sjednocení spočetně nekonečně mnoha konečných množin je spočetně nekonečné. □

Dodatek: přirozeně lze označit jednotlivé uzly stromu unikátními identifikátory tak, že každý uzel stromu bude označen přirozeným číslem získaným vztahem $2^n - 1 + m$, kde $n \in \mathbb{N}$ označuje úroveň uzlu, kdy kořen má úroveň 0, a $m \in \mathbb{N}$ označuje pořadí uzlu v dané úrovni při dodržení číslování od 0.

(b)

Teorém 3.2. Nechť *obarvení stromu* T je funkce, která přiřazuje každému vrcholu $v \in V$ barvu $c(v) \in \{red, black\}$. Pak množina všech obarvení stromu T je nespočetně nekonečná.

Důkaz. Důkaz bude proveden pomocí diagonalizace. Obarvení stromu je dále označované jako g a množina všech obarvení stromu T jako S. Předpokládejme, že množina S je spočetná. Podle definice spočetnosti poté musí existovat bijekce $f : \mathbb{N} \longleftrightarrow S$.

Uspořádejme vrcholy stromu do posloupnosti, kterou lze vytvořit konstatováním, že jednotlivé vrcholy, $v_i \in V$, kde $i = 0, 1, 2, ...$, jsou označeny způsobem uvedeným v podúkolu (a). Takto vytvořená posloupnost bude značena jako $v_0, v_1, v_2, ...$ vzestupně dle značení vrcholů. Dále bude využívána posloupnost $g_0, g_1, g_2, ...$ zahrnující všechna obarvení stromu. Potom f může být zobrazena nekonečnou maticí:

$$\begin{array}{rcccccc} & \boldsymbol{v_0} & \boldsymbol{v_1} & \boldsymbol{v_2} & \boldsymbol{\ldots} & \boldsymbol{v_i} & \boldsymbol{\ldots} \\ \boldsymbol{g_0 = f(0)} & a_{00} & a_{01} & a_{02} & \ldots & a_{0i} & \ldots \\ \boldsymbol{g_1 = f(1)} & a_{10} & a_{11} & a_{12} & \ldots & a_{1i} & \ldots \\ \boldsymbol{g_2 = f(2)} & a_{20} & a_{21} & a_{22} & \ldots & a_{2i} & \ldots \\ \boldsymbol{\ldots} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{array},$$

kde

$$a_{ij} = \begin{cases} 0, & \text{jestliže } g_i(v_j) = black, \\ 1, & \text{jestliže } g_i(v_j) = red. \end{cases}$$

Uvažujme obarvení stromu h takové, pro které platí, že pokud $a_{ii} = 0$, pak $h(v_i) = red$, jinak $h(v_i) = black$. Obarvení stromu h se liší od každého obarvení stromu $g_i = f(i)$, kde $i \in \mathbb{N}$, tak, že pokud barva $c(v_i)$ pro g_i je černá barva ($a_{ii} = 0$), pak pro h je stejný uzel (v_i) označen červenou barvou a naopak. Současně ovšem $h \in S$, tudíž f není surjektivní a docházíme ke sporu. □

4 Příklad 4

Navrhněte algoritmus, který pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ a symbol $a \in \Sigma$ spočítá množinu

$$_aN_a = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w \wedge \exists u \in \Sigma^* : w = aua\}.$$

V algoritmu můžete využít množinu $N_\epsilon = \{A \in N \mid A \Rightarrow_G^+ \epsilon\}$. Doporučujeme nadefinovat si další vhodné množiny neterminálů a algoritmicky popsat jejich výpočet (u N_ϵ popis výpočtu není potřeba).

Ilustrijte použití algoritmu na příkladě gramatiky s pravidly

$$\begin{aligned} S &\to aWU \mid UWa \\ W &\to YacU \mid Xa \\ Y &\to X \mid \epsilon \\ X &\to aXa \\ U &\to bcaYY \mid Ucb \end{aligned}$$

4.1 Řešení

Množina $_aN_a$ obsahuje neterminály, ze kterých lze derivovat řetězce terminálů takové, že jejich první a poslední symbol (prefix a sufix o délce 1) je symbol a. Zároveň platí, že minimální délka takových řetězců je 2.

Nejdříve spočítáme množinu neterminálů

$$N_t = \{A \in N \mid A \Rightarrow^+ w \wedge w \in \Sigma^*\}.$$

Taková množina obsahuje neterminály, které vygenerují alespoň jeden řetězec neterminálů, včetně prázdného řetězce. N_t lze vypočítat pomocí následujícího fix-point algoritmu:

1. $N_t^0 := \emptyset, i := 0$,
2. $N_t^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t^i)^*\}$,
3. Pokud $N_t^{i+1} = N_t^i$, potom $N_t := N_t^i$, jinak $i := i + 1$ a `goto` bod 2.

Dále bude v algoritmu využívána poskytnutá množina

$$N_\varepsilon = \{A \in N \mid A \Rightarrow_G^+ \varepsilon\},$$

která obsahuje neterminály, jenž vygenerují alespoň jeden řetězec, který je prázdým řetězcem. Tuto množinu lze vytvořit pomocí fix point algoritmu:

1. $N_\varepsilon^0 := \emptyset, i := 0$,
2. $N_\varepsilon^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon^i)^*\}$,
3. Pokud $N_\varepsilon^{i+1} = N_\varepsilon^i$, potom $N_\varepsilon := N_t^\varepsilon$, jinak $i := i + 1$ a `goto` bod 2.

Nakonec bude zapotřebí zkonstruovat ještě dvě množiny. První z nich je množina

$${}_aN = \{A \in N \mid \exists w, u \in \Sigma^* : A \Rightarrow_G^+ w \wedge w = au\},$$

kde $a \in \Sigma$. Množina ${}_aN$ obsahuje všechny neterminály, které vygenerují alespoň jeden řetězec terminálů takový, který začíná symbolem a (prefix o délce 1 je roven řetězci a). ${}_aN$ lze vypočítat pomocí následujícího fix-point algoritmu:

1. ${}_aN^0 := \emptyset, i := 0$,
2. ${}_aN^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*(\{a\} \cup N_a^i)(\Sigma \cup N_t)^*\}$,
3. Pokud ${}_aN^{i+1} = {}_aN^i$, potom ${}_aN := {}_aN^i$, jinak $i := i + 1$ a `goto` bod 2.

Druhou z množin bude množina

$$N_a = \{A \in N \mid \exists w, u \in \Sigma^* : A \Rightarrow_G^+ w \wedge w = ua\},$$

kde $a \in \Sigma$. Podobně jako množina ${}_aN$, množina N_a obsahuje všechny neterminály, které vygenerují alespoň jeden řetězec terminálů takový, který naopak končí symbolem a (sufix o délce 1 je roven řetězci a). N_a lze vypočítat pomocí fix-point algoritmu:

1. $N_a^0 := \emptyset, i := 0$,
2. $N_a^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup N_a^i)(N_\varepsilon)^*\}$,
3. Pokud $N_a^{i+1} = N_a^i$, potom $N_a := N_a^i$, jinak $i := i + 1$ a `goto` bod 2.

Poté výslednou množinu ${}_aN_a$ lze sestrojit pomocí fix-point algoritmu:

1. ${}_aN_a := \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*(\{a\} \cup {}_aN)(\Sigma \cup N_t)^*(\{a\} \cup N_a)(N_\varepsilon)^*\}, i := 0,$
2. ${}_aN_a^{i+1} := \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^* {}_aN_a^i (N_\varepsilon)^*\} \cup {}_aN_a^i,$
3. Pokud ${}_aN_a^{i+1} = {}_aN_a^i$, potom ${}_aN_a := {}_aN_a^i$, jinak $i := i + 1$ a `goto` bod 2.

Následně bude demonstrována validnost algoritmu na zadané gramatice.

Příklad 4.1. Uvažujme bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ s pravidly

$$\begin{aligned}
&S \to aWU \mid UWa,\\
&W \to YacU \mid Xa,\\
&Y \to X \mid \epsilon,\\
&X \to aXa,\\
&U \to bcaYY \mid Ucb.
\end{aligned}$$

Na této gramatice bude demonstrována funkcionalita navrženého algoritmu. Nejprve dle návrhu algoritmu bude vypočítána množina N_t pomocí vytvořeného fix-point algoritmu:

$$\begin{aligned}
N_t^0 &= \emptyset,\\
N_t^1 &= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t^0)^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup \emptyset)^*\} =\\
&= \{Y\},\\
N_t^2 &= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t^1)^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup \{Y\})^*\} =\\
&= \{U, Y\},\\
N_t^3 &= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t^2)^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup \{U, Y\})^*\} =\\
&= \{U, W, Y\},\\
N_t^4 &= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t^3)^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup \{U, W, Y\})^*\} =\\
&= \{S, U, W, Y\} = N_t^5 = N_t.
\end{aligned}$$

Následně množina N_ε:

$$\begin{aligned}
N_\varepsilon^0 &= \emptyset,\\
N_\varepsilon^1 &= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon^0)^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in \emptyset^*\} =\\
&= \{A \in N \mid \exists(A \to \alpha) \in P \wedge \alpha \in \{\varepsilon\}\} =\\
&= \{Y\} = N_\varepsilon^2 = N_\varepsilon.
\end{aligned}$$

Poté budou určeny obě pomocné množiny, ${}_aN$ a N_a, následovně:

$$
\begin{aligned}
{}_aN^0 &= \emptyset, \\
{}_aN^1 &= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*(\{a\} \cup N_a^0)(\Sigma \cup N_t)^*\} = \\
&= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*(\{a\} \cup \emptyset)(\Sigma \cup N_t)^*\} = \\
&= \{S, W\} = {}_aN^2 = {}_aN.
\end{aligned}
$$

$$
\begin{aligned}
N_a^0 &= \emptyset, \\
N_a^1 &= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup N_a^0)(N_\varepsilon)^*\} = \\
&= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup \emptyset)(N_\varepsilon)^*\} = \\
&= \{S, U\}. \\
N_a^2 &= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup N_a^1)(N_\varepsilon)^*\} = \\
&= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup \{S, U\})(N_\varepsilon)^*\} = \\
&= \{S, U, W\} = N_a^3 = N_a.
\end{aligned}
$$

Výsledná množina, ${}_aN_a$, bude sestavena pomocí s využitím sestaveného fix-point algoritmu:

$$
\begin{aligned}
{}_aN_a^0 &= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*(\{a\} \cup {}_aN)(\Sigma \cup N_t)^*(\{a\} \cup N_a)(N_\varepsilon)^*\} = \\
&= \{S, W\}, \\
{}_aN_a^1 &= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*{}_aN_a^0(N_\varepsilon)^*\} \cup {}_aN_a^0 = \\
&= \{A \in N \mid \exists (A \to \alpha) \in P \wedge \alpha \in (N_\varepsilon)^*\{S, W\}(N_\varepsilon)^*\} \cup \{S, W\} = \\
&= \emptyset \cup \{S, W\} = \{S, W\}, \\
{}_aN_a^0 &= {}_aN_a^1 = {}_aN_a.
\end{aligned}
$$

Zadané gramatice G přísluší množina ${}_aN_a = \{S, W\}$.

Použitá literatura

[1] Hopcroft, J. E.; Ullman, J. D.: *Introduction to automata theory, languages, and computation.* Philippines: Addison-Wesley, 1979, ISBN 0-201-02988-X.