# 1 Postův korespondenční problém

Postův korespondenční problém (PKP) je úloha, jejímž vstupem jsou seznamy slov $r_1, \ldots, r_n$ a $s_1, \ldots, s_n$ pro nějaké přirozené číslo $n \geq 1$. Úlohou je zjistit, zda existuje neprázdná posloupnost indexů $t_1, ..., t_k \in \{1, \ldots, n\}$ taková, že zřetězení příslušných slov splňují

$$r_{t_1} \ldots r_{t_k} = s_{t_1} \ldots s_{t_k}$$

Pokud taková posloupnost indexů existuje, budeme ji nazývat řešením příslušné instance PKP. Pro analýzu instancí PKP budeme uvažovat také částečná řešení, což je libovolná posloupnost indexů $t_1, \ldots, t_k$, která je řešením, nebo kratší ze slov $r_{t_1} \ldots r_{t_k}$ a $s_{t_1} \ldots s_{t_k}$ je prefixem druhého. Pokud je $t_1, \ldots, t_k$ řešením PKP, pak pro každé $1 \leq l \leq k$ je posloupnost $t_1, \ldots, t_l$ částečným řešením.

Instance PKP určená tabulkou

| $t$ | 1 | 2 | 3 |
|---|---|---|---|
| $r_t$ | $a$ | $ab$ | $bba$ |
| $s_t$ | $baa$ | $aa$ | $bb$ |

má řešení $3, 2, 3, 1$, protože

$$bba \cdot ab \cdot bba \cdot a = bb \cdot aa \cdot bb \cdot baa$$

kde $\cdot$ označuje zřetězení slov. Existují také částečná řešení, která nejsou prefixem žádného řešení, například $3, 2, 1, 1$. V tomto případě je slovo

$$r_3 r_2 r_1 r_1 = bba \cdot ab \cdot a \cdot a$$

prefixem slova

$$s_3 s_2 s_1 s_1 = bb \cdot aa \cdot baa \cdot baa$$

Příklad neřešitelné instance PKP je

| $t$ | 1 | 2 |
|---|---|---|
| $r_t$ | $ba$ | $bab$ |
| $s_t$ | $bab$ | $abb$ |

Pro tuto instanci jsou částečná řešení tvořena právě všemi počátečními úseky posloupnosti $1, 2, 2, 2, \ldots$, ale žádné z nich není řešením.

Budeme uvažovat také variantu PKP, kterou budeme nazývat inicializovaný Postův korespondenční problém (IPKP). Vstup je stejný jako pro PKP a úlohou je zjistit, zda existuje řešení, které začíná indexem 1 a všechny další indexy v posloupnosti jsou různé od 1.

**Věta 1** *Inicializovaný Postův korespondenční problém je algoritmicky nerozhodnutelná úloha.*

*Důkaz.* Ukážeme redukci problému zastavení na inicializovaný PKP. Budeme uvažovat konfigurace TS popsané v sekci Automaty předchozího textu, tedy slova tvaru $uqv\# \in \Gamma^* Q \Gamma^* \#$, kde $u, v \in \Gamma^*$ a $q \in Q$. Počáteční konfigurace má tvar $\triangleright\, q_0 w \#$, kde $w$ je vstupní slovo a $q_0$ je počáteční stav TS. Pro konstrukci přepisovacích pravidel, která simulují výpočet TS, budeme symbol # považovat za zápis nekonečné posloupnosti $\square\square\square\ldots$, tedy nekonečné posloupnosti symbolů pro prázdné pole pásky. Během vlastní simulace TS se tento symbol bude pro jednoduchost posouvat pouze vpravo. Pokud TS zapíše na pásku prázdné pole, bude toto pole reprezentováno symbolem $\square$ i v případě, že na ním následují pouze další symboly $\square$ a symbol #. Výpočet TS je popsán přepisovacími pravidly, která jsou odvozena z přechodové funkce $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 0, 1\}$ následovně. Pro libovolné $q, q' \in Q$, $x, y, x' \in \Gamma$ a $s \in \{-1, 0, 1\}$ takové, že $\delta(q, x) = (q', x', s)$, zařadíme pravidla $\alpha \to \beta$ podle následující tabulky

| $\alpha$ | $\to$ | $\beta$ | pokud platí |
|---|---|---|---|
| $qx$ | $\to$ | $q'x'$ | $s = 0$ |
| $q\#$ | $\to$ | $q'x'\#$ | $s = 0, x = \square$ |
| $yqx$ | $\to$ | $q'yx'$ | $s = -1$ |
| $yq\#$ | $\to$ | $q'yx'\#$ | $s = -1, x = \square$ |
| $qx$ | $\to$ | $x'q'$ | $s = 1$ |
| $q\#$ | $\to$ | $x'q'\#$ | $s = 1, x = \square$ |

(1)

Výpočet TS nikdy nepřejde vlevo z nejlevějšího pole pásky. Žádné z uvedených pravidel proto neprodlužuje zápis konfigurace doleva. Prodlužování vpravo nastává tehdy, když je přepsán symbol $\square$, který není v konfiguraci explicitně, ale je reprezentován symbolem #.

Kromě pravidel, která reprezentují výpočet, přidáme ještě pravidla, která v případě, že výpočet skončí v přijímajícím stavu $q_+$, umožní zkrátit zápis konfigurace až na slovo $q_+\#$. Pro tento účel zařadíme následující pravidla pro všechna $x \in \Gamma$

| $\alpha$ | $\to$ | $\beta$ |
|---|---|---|
| $q_+x$ | $\to$ | $q_+$ |
| $xq_+\#$ | $\to$ | $q_+\#$ |

(2)

**Lemma 2** *Existuje jednoznačně určená posloupnost konfigurací maximální délky, která může být nekonečná, která začíná počáteční konfigurací a jejíž členy vzniknou postupnou aplikací pravidel (1) a (2). Navíc platí, že tato posloupnost je konečná a poslední konfigurace má tvar $q_+\#$ právě tehdy, když výpočet TS skončí v přijímajícím stavu $q_+$.*

*Důkaz.* Lze ověřit, že pravidla (1) a (2) zachovávají tvar zápisu konfigurace $\Gamma^* Q \Gamma^* \#$. Speciálně, v každém slově odvozeném těmito pravidly je právě jeden

výskyt symbolu pro stav řídící jednotky z $Q$, který označme $q$. Za ním následuje symbol $x \in \Gamma \cup \{\#\}$. Pokud $x \in \Gamma$, pak jednoznačnost následujícího kroku použití pravidel plyne z toho, že ke každé dvojici symbolů $q \in Q \setminus \{q_-, q_+\}$ a $x \in \Gamma$ existuje právě jedna kombinace symbolů $q', x', s$ splňující $\delta(q, x) = (q', x', s)$ a existuje právě jedno pravidlo (1), které lze použít. Toto pravidlo provede odpovídající změnu konfigurace. Pokud $x = \#$, jsou symboly $q', x', s$ určeny podmínkou $\delta(q, \square) = (q', x', s)$ a opět existuje právě jedno pravidlo, které lze použít. Toto pravidlo provede odpovídající změnu konfigurace, což v tomto případě zahrnuje posun symbolu # o jednu pozici vpravo. Pokud $q = q_+$ a konfigurace není rovna $q_+\#$, pak existuje právě jedno pravidlo (2), kterým lze zápis zkrátit. □

K libovolné instanci problému zastavení, která je vyjádřena počáteční konfigurací TS, sestrojme instanci IPKP z dvojic slov $r_t, s_t$ podle následující tabulky. Pro první dvě dvojice slov je uvedena hodnota indexu $t$, pro ostatní dvojice hodnoty indexu $t \geq 3$ nejsou uvedeny a tyto dvojice jsou pouze rozlišeny do typů označených I, II, III. Položky v tabulce, které obsahují $x$, $\alpha$ nebo $\beta$, jsou míněny pro každý symbol $x \in \Gamma$ a pro všechna pravidla $\alpha \to \beta$ popsaná v (1) a (2).

| $t$ | 1 | 2 | I | II | III |
|---|---|---|---|---|---|
| $r_t$ | # | $q_+\#\#$ | $\alpha$ | $x$ | # |
| $s_t$ | $\#q_0w\#$ | # | $\beta$ | $x$ | # |

(3)

Řešení $t_1 = 1, t_2, \ldots, t_k$ instance IPKP nazveme minimální, pokud žádný jeho prefix není řešením. Pokud existuje řešení IPKP, existuje i minimální řešení. Následující tvrzení popisuje strukturu minimálních řešení, která pro neminimální řešení obecně neplatí.

**Lemma 3** *Každé minimální řešení $t_1 = 1, t_2, \ldots, t_k$ instance IPKP (3) splňuje, že slovo $r_1 \cdot r_{t_2} \cdot \ldots \cdot r_{t_k} = s_1 \cdot s_{t_2} \cdot \ldots \cdot s_{t_k}$ má tvar*

$$\#u_1u_2 \ldots u_m\#$$

*kde $u_1, \ldots, u_m$ je posloupnost konfigurací TS, $u_i \in \Gamma^*Q\Gamma^*\#$, $u_1$ je počáteční konfigurace, $u_m = q_+\#$ a přechody mezi jednotlivými konfiguracemi jsou určené pravidly (1) a (2),*

*Důkaz.* Každé řešení musí použít dvojici $r_2, s_2$, protože $r_1$ obsahuje méně znaků # než $s_1$ a dvojice $r_2, s_2$ je jediná dvojice $r_i, s_i$, ve které je více znaků # v $r_i$ než v $s_i$. Nechť $l$ je nejmenší index takový, že $t_l = 2$.

Dvojici $r_{t_j}, s_{t_j}$ pro $1 \leq j \leq l$ nazveme uzavírající, pokud $r_{t_j}$, a tedy také $s_{t_j}$, obsahuje symbol #. Nechť $U \subseteq \{1, \ldots, l\}$ je množina indexů uzavírajících dvojic. Speciálně, $1 \in U$, protože dvojice $(r_{t_1}, s_{t_1}) = (r_1, s_1)$ je uzavírající, a $l \in U$, protože dvojice $(r_{t_l}, s_{t_l}) = (r_2, s_2)$ je uzavírající.

Nechť $m = |U| - 1$. Zřetězení dvojic $r_{t_j}, s_{t_j}$ pro $1 \leq j \leq l$ rozdělíme na $m + 1$ úseků tak, že $i$-tý úsek, kde $1 \leq i \leq m + 1$, končí $i$-tou uzavírající dvojicí. Nechť

$R_i$ je zřetězení slov $r_{t_j}$ v $i$-tém úseku a $S_i$ je zřetězení slov $s_{t_j}$ v $i$-tém úseku. Pro zřetězení všech úseků platí

$$\begin{aligned} R_1 \dots R_{m+1} &= r_{t_1} \dots r_{t_l} \\ S_1 \dots S_{m+1} &= s_{t_1} \dots s_{t_l} \end{aligned}$$

Indukcí dokažme, že pro $i = 1, \dots, m$ platí

$$R_1 \dots R_i = \#u_1 \dots u_{i-1} \quad (4)$$
$$S_1 \dots S_i = \#u_1 \dots u_{i-1}u_i \quad (5)$$

pro nějakou posloupnost konfigurací $u_1, \dots, u_i$. Pro $i = 1$ tvrzení platí, protože slova $R_1, S_1$ zahrnují pouze počáteční dvojici $r_1, s_1$, a tedy platí

$$\begin{aligned} R_1 &= \# \\ S_1 &= \#u_1 \end{aligned}$$

kde $u_1 = q_0w\#$ je počáteční konfigurace. Dokažme (4) a (5) pro $i = 2$, tedy, že platí

$$R_1R_2 = \#u_1 \quad (6)$$
$$S_1S_2 = \#u_1u_2 \quad (7)$$

Poslední z dvojic $r_t, s_t$, které tvoří $R_2, S_2$, je uzavírající, a tedy příslušné $r_t$ obsahuje jeden symbol # jako poslední znak a tento znak je tedy posledním znakem $R_2$. Protože $R_1$ je kratší než $S_1$ a $S_1$ končí symbolem #, musí symbol # na konci $R_2$ odpovídat symbolu #, který ukončuje $S_1$. To znamená, že $R_2 = u_1$ a tedy platí (6).

Slova $R_2$ a $S_2$ jsou tvořena zřetězením slov $r_t$ a $s_t$ pro $t \geq 3$, která jsou v tabulce (3) označena jako typ I, II a III. Konfiguraci $u_1$ lze vyjádřit jako zřetězení slov $r_t$ z těchto dvojic nejvýše jedním způsobem, protože jedno z těchto slov musí obsahovat stav řídící jednotky v $u_1$, tedy musí být typu I, a různá slova $r_t$ z dvojic typu I obsahují daný stav $q$ s různými kontexty, které se vylučují, a lze tedy použít nejvýše jedno z těchto slov. Ostatní slova $r_t$, která jsou obsažena ve zřetězení, které tvoří $R_2 = u_1$, musí být typu II nebo III a jsou jednoznačně určena. Protože předpokládáme, že vyjádření $R_2 = u_1$ pomocí $r_t$ z dvojic typů I, II a III existuje, je slovo $S_2$ jednoznačně určeno jako konfigurace, která vznikne z $u_1$ pomocí pravidel (1) a (2). Když tuto konfiguraci označíme $u_2$, platí (7).

Analogicky se dokáže požadované tvrzení i pro $i = 3, \dots, m$. Ve všech těchto případech končí odpovídající $R_i$ jedním symbolem #, který odpovídá symbolu # na konci $S_{i-1}$ a slovo $S_i = u_i$ je konfigurace odvozená z $R_i = u_{i-1}$ pomocí pravidel (1) a (2).

Dokázali jsme tedy, že platí

$$\begin{aligned} R_1 \dots R_m &= \#u_1 \dots u_{m-1} \\ S_1 \dots S_m &= \#u_1 \dots u_{m-1} u_m \end{aligned}$$

kde $u_1, \dots, u_m$ je posloupnost konfigurací odvozených pravidly (1) a (2). Slovo $R_{m+1}$ končí slovem $r_2$. Poslední dva symboly $R_{m+1}$ tedy jsou symboly # a první z nich musí odpovídat symbolu # na konci $S_m = u_m$. Tedy $S_{m+1}$ musí začínat symbolem #, který odpovídá druhému # na konci $R_{m+1}$. Kdyby $m+1$-ní úsek řešení, ze kterého jsou vytvořena slova $R_{m+1}, S_{m+1}$, obsahoval před dvojicí $r_2, s_2$ jinou dvojici, není tato dvojice uzavírající, a tedy by $S_{m+1}$ nezačínalo symbolem #. Platí tedy $R_{m+1} = r_2 = q_+\#\#$, $S_{m+1} = s_2 = \#$. Protože slova $R_1 \dots R_{m+1}$ a $S_1 \dots S_{m+1}$ tvoří částečné řešení, musí být slova $q_+\#\#$ a $u_m\#$ buď shodná nebo jedno je prefixem druhého. Obě tato slova končí slovem ## a jiné výskyty symbolu # neobsahují. Musí tedy platit $q_+\#\# = u_m\#$. Z toho plyne $u_m = q_+\#$ a navíc platí $R_1 \dots R_{m+1} = S_1 \dots S_{m+1}$. Protože uvažujeme minimální řešení, je $l = k$ a platí tedy současně

$$\begin{aligned} R_1 \dots R_{m+1} &= r_{t_1} \dots r_{t_k} \\ S_1 \dots S_{m+1} &= s_{t_1} \dots s_{t_k} \end{aligned}$$

a

$$\begin{aligned} R_1 \dots R_{m+1} &= \#u_1 \dots u_{m-1} u_m \# \\ S_1 \dots S_{m+1} &= \#u_1 \dots u_{m-1} u_m \# \end{aligned}$$

což implikuje dokazované tvrzení. □

**Lemma 4** *Jestliže se výpočet uvažovaného TS zastaví v přijímajícím stavu, pak existuje řešení $t_1 = 1, t_2, \dots, t_k$ instance IPKP popsané v (3).*

*Důkaz.* Postup, který je v důkazu předchozího lemmatu použit k nalezení výpočtu TS z řešení IPKP, lze použít v obráceném pořadí k nalezení řešení IPKP na základě výpočtu TS. □

Lemma 3 implikuje, že z existence řešení instance IPKP pro danou počáteční konfiguraci TS plyne, že se výpočet TS pro tuto počáteční konfiguraci zastaví ve stavu $q_+$. Opačná implikace plyne z Lemmatu 4. Popsaná konstrukce instance IPKP je tedy redukcí problému zastavení na problém existence řešení IPKP, což implikuje Větu 1. □

**Věta 5** *Postův korespondenční problém je algoritmicky nerozhodnutelná úloha.*

*Důkaz.* Ukážeme redukci IPKP na PKP. Nechť $r_1, \ldots, r_n$ a $s_1, \ldots, s_n$ je instance IPKP v abecedě $\Sigma$, která neobsahuje symboly $0, 1$. Definujme pomocné funkce $l, p : \Sigma^* \to (\Sigma \cup \{0,1\})^*$, které vkládají symbol 1 vlevo nebo vpravo od každého symbolu vstupního slova, tedy pro slovo $x_1 x_2 \ldots x_k \in \Sigma^*$ je

$$l(x_1 x_2 \ldots x_k) = 1x_1 1x_2 \ldots 1x_k$$

a

$$p(x_1 x_2 \ldots x_k) = x_1 1 x_2 1 \ldots x_k 1$$

Uvažme instanci PKP ve tvaru

| $i$ | 1 | 2 | $\ldots$ | $n$ | $n+1$ |
|---|---|---|---|---|---|
| $r_i$ | $l(r_1)$ | $l(r_2)$ | $\ldots$ | $l(r_n)$ | 10 |
| $s_i$ | $1p(s_1)$ | $p(s_2)$ | $\ldots$ | $p(s_n)$ | 0 |

Z každého řešení výchozí instance IPKP lze sestrojit řešení výsledné instance PKP a také naopak. Z toho plyne, že problém IPKP lze efektivně redukovat na problém PKP, což dokazuje tvrzení věty. □

# 2 Neprázdnost průniku bezkontextových jazyků

**Věta 6** *Úloha rozhodnout pro libovolné zadané bezkontextové gramatiky $G_1$ a $G_2$, zda jazyk $L(G_1) \cap L(G_2)$ je neprázdný, je algoritmicky nerozhodnutelná.*

*Důkaz.* Uvažme instanci PKP ve tvaru $r_1, \ldots, r_n$ a $s_1, \ldots, s_n$, kde slova $r_i$, $s_i$ jsou v abecedě $\Sigma$. Zvolme symboly $c_1, \ldots, c_n$, které nejsou v $\Sigma$ a budou kódovat indexy $1, \ldots, n$. Sestrojme gramatiky $G_1$ a $G_2$ tak, že gramatika $G_1$ obsahuje pro $t = 1, \ldots, n$ pravidla

$$S_1 \to r_t S_1 c_t$$
$$S_1 \to r_t c_t$$

a gramatika $G_2$ obsahuje pravidla

$$S_2 \to s_t S_2 c_t$$
$$S_2 \to s_t c_t \ .$$

Pokud $L(G_1) \cap L(G_2)$ je neprázdný, obsahuje slovo tvaru $u c_{t_k} \ldots c_{t_2} c_{t_1}$, kde $k \geq 1$, $u \in \Sigma^*$ a platí současně

$$u = r_{t_1} r_{t_2} \ldots r_{t_k}$$
$$u = s_{t_1} s_{t_2} \ldots s_{t_k}$$

tedy existuje řešení výchozí instance PKP. Na druhé straně, z každého řešení výchozí instance PKP lze obráceným postupem sestrojit slovo v jazyce $L(G_1) \cap L(G_2)$. □