1.13.15 Redukce. Připomeňme definici redukce.

Jsou dány dvě rozhodovací úlohy $\mathcal{U}$ a $\mathcal{V}$. Řekneme, že úloha $\mathcal{U}$ se *redukuje* na úlohu $\mathcal{V}$, jestliže existuje algoritmus (program pro RAM, Turingův stroj) $\mathcal{A}$, který pro každou instanci I úlohy $\mathcal{U}$ zkonstruuje instanci I' úlohy $\mathcal{V}$ a to tak, že

$$I \text{ je ANO instance } \mathcal{U} \text{ iff } I' \text{ je ANO instance } \mathcal{V}.$$

Fakt, že úloha $\mathcal{U}$ se redukuje na úlohy $\mathcal{V}$ značíme

$$\mathcal{U} \lhd \mathcal{V}.$$

Tato definice má význam i pro jazyky. Rozhodovací úlohu chápeme jako jazyk obsahující ta slova, která odpovídají ANO instancím.

1.13.16 Tvrzení. Jsou dány dvě úlohy $\mathcal{U}$ a $\mathcal{V}$ takové, že $\mathcal{U} \lhd \mathcal{V}$. Pak platí:

1. Jestliže $\mathcal{U}$ je nerozhodnutelná, pak i $\mathcal{V}$ je nerozhodnutelná.

2. Jestliže $\mathcal{U}$ není rekursivně spočetná, pak i $\mathcal{V}$ není rekursivně spočetná.

1.13.17 Tvrzení. Jsou dány jazyky

$$L_e = \{M \mid L(M) = \emptyset\}, \quad L_{ne} = \{M \mid L(M) \neq \emptyset\}.$$

Pak jazyk L_{ne} je rekursivně spočetný, ale ne rekursivní. Jakyk L_e není ani rekursivně spočetný.

1.13.18 Poznámka. Uvědomme si, že jazyk L_e je dopňkem jazyka L_{ne}. Ano, jestliže slovo w není kódem nějakého Turingova stroje, pak ho považujeme za kód stroje, který nepřijímá žádné slovo, tj. patří do jazyka L_e.

Universální Turingův stroj U se dá využít k tomu abychom ukázali, že jazyk L_{ne} je rekursivně spočetný. Z redukce $L_u \lhd L_{ne}$ a 1.13.16 dostáváme, že L_{ne} není rekursivní. Fakt, že L_e není ani rekursivně spočetný pak vyplývá z 1.13.10.

1.13.19 Věta (Rice). Jakákoli netriviálni vlastnost rekursivně spočetných jazyků (jazyků přijímaných Turingovým strojem) je nerozhodnutelná.

Netriviální vlastností rozumíme každou vlastnost, kterou má aspoň jeden rekursivně spočetný jazyk a nemají ho všechny rekursivně spočetné jazyky.

1.14 Další nerozhodnutelné úlohy

1.14.1 V předchozí části jsme uvedli několik nerozhodnutelných jazyků — úloh. Věta (Rice) dokonce říká, že každá netriviální vlastnost rekursivních jazyků je nerozhodnutelná. Na druhou stranu úlohy týkající se rekursivních jazyků se mohou zdát jako značně umělé. V této části ukážeme další úlohy, které jsou nerozhodnutelné. Poznamenejme ještě, že univerzální jazyk L_u hraje pro nerozhodnutelné jazyky/úlohy obdobnou roli jako hrál problém splnitelnosti booleovských formulí pro $\mathcal{NP}$ úlné úlohy.

1.14.2 Postův korespondenční problém (PCP). Jsou dány dva seznamy slov A, B nad danou abecedou Σ.

$$A = (w_1, w_2, \ldots, w_k), \quad B = (x_1, x_2, \ldots, x_k),$$

kde $w_i, x_i \in \Sigma^\star$, $i = 1, 2, \ldots, k$. Řekneme, že dvojice A, B *má řešení*, jestliže existuje posloupnost $i_1, i_2, \ldots, i_r$ indexů, tj $i_j \in \{1, 2, \ldots, k\}$, taková, že

$$w_{i_1}\, w_{i_2}\, \ldots\, w_{i_r} = x_{i_1}\, x_{i_2}\, \ldots\, x_{i_r}.$$

Otázka: Existuje řešení dané instance?

1.14.3 Příklady.

1. Jsou dány seznamy

	1	2	3	4	5
A	011	0	101	1010	010
B	1101	00	01	00	0

Tato instance má řešení, např. $2, 1, 1, 4, 1, 5$ je

$$w_2\, w_1\, w_1\, w_4\, w_1\, w_5 = 00110111010011010 = x_2\, x_1\, x_1\, x_4\, x_1\, x_5.$$

2. Jsou dány seznamy

	1	2	3	4	5
A	11	0	101	1010	010
B	101	00	01	00	0

Tato instance nemá řešení.

1.14.4 Modifikovaný Postův korespondenční problém (MPCP). Jsou dány dva seznamy slov A, B nad danou abecedou Σ.

$$A = (w_1, w_2, \ldots, w_k), \quad B = (x_1, x_2, \ldots, x_k),$$

kde $w_i, x_i \in \Sigma^\star$, $i = 1, 2, \ldots, k$. Řekneme, že dvojice A, B *má řešení*, jestliže existuje posloupnost $1, i_1, i_2, \ldots, i_r$ indexů, tj $i_j \in \{1, 2, \ldots, k\}$, taková, že

$$w_1\, w_{i_1}\, w_{i_2}\, \ldots\, w_{i_r} = x_1\, x_{i_1}\, x_{i_2}\, \ldots\, x_{i_r}.$$

Otázka: Existuje řešení dané instance?

1.14.5 Poznámka. Modifikovaný Postův korespondenční problém se od Postova korespondenčního problému liší tím, že v MPCP vyžadujeme, aby hledaná posloupnost indexů vždy začínala jedničkou. Význam MPCP spočívá v tom, že se dá dokázat následující věta.

1.14.6 Věta. Platí

$$L_u \lhd \text{ MPCP } \lhd \text{ PCP}.$$

1.14.7 Poznámka. Druhá redukce z věty 1.14.6 je jednodušší. Pomocí rozšíření abecedy jsme schopni zkonstruovat instanci PCP tak, abychom měli zajištěno, že řešení, posloupnost indexů, **musí** začínat 1.

První redukce je obtížnější. Jedná se o popis práce Turingova stroje pomocí slov nad vhodnou abecedou. Trik spočívá v tom, že posloupnost pro MPCP musí začínat prvním slovem (to zajistí, že Turingův stroj začne pracovat v počátečním stavu s daným obsahem pásky). Pro seznam A bude slovo vždy „dohánět výpočet podle přechodové funkce Turingova stroje“, který bude odpovídat seznamu B.

1.14.8 Důsledek. Postův korespondenční problém je nerozhodnutelný.

1.14.9 Poznámka. Kdybychom omezili možnou délku hledané posloupnosti $i_1, i_2, \ldots, i_r$, (tj. omezili r), problém by se stal algoritmicky řešitelným — existoval by algoritmus hrubé síly. Také, kdybychom místo seznamů A, B uvažovali množiny slov, problém by byl dokonce polynomiálně řešitelný.

1.14.10 Víceznačnost bezkontextových gramatik. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$, kde N je množina neterminálních symbolů, Σ je množina terminálních symbolů, S je startovací symbol a P je množina pravidel typu $X \to \alpha$ pro $X \in N$, $\alpha \in (N \cup \Sigma)^\star$.

Otázka: Rozhodněte, zda existuje slovo w, které má dva různé derivační stromy.

1.14.11 Věta. Platí

$$\text{PCP} \ \lhd \ \text{víceznačnost bezkontextových gramatik.}$$

1.14.12 Nástin redukce pro důkaz věty 1.14.11. Je dána instance PCP, tj. seznamy slov $A = (w_1, w_2, \ldots, w_k)$ a $B = (x_1, x_2, \ldots, x_k)$. Sestrojíme bezkontextovou gramatiku $\mathcal{G} = (\{S, A, B\}, \Sigma \cup \{a_1, a_2, \ldots, a_k\}, S, P)$, kde P obsahuje tato pravidla

$$S \to A \mid B,$$

$$A \to w_1\, A\, a_1 \mid w_2\, A\, a_2 \mid \ldots \mid w_k\, A\, a_k,$$

$$A \to w_1\, a_1 \mid w_2\, a_2 \mid \ldots \mid w_k\, a_k,$$

$$B \to x_1\, B\, a_1 \mid x_2\, B\, a_2 \mid \ldots \mid x_k\, B\, a_k,$$

$$B \to x_1\, a_1 \mid x_2\, a_2 \mid \ldots \mid x_k\, a_k,$$

Pak gramatika $\mathcal{G}$ je víceznačná právě tehdy, když nějaké slovo $wa_{i_1}a_{i_2}\ldots a_{i_r}$, $w \in \Sigma^\star$, má dvě různá odvození. Tato situace nastává právě tehdy, když instance PCP má řešení. (Uvědomte si, že dvě různá odvození jsou možná jen, můžeme-li stejné slovo $wa_{i_1}a_{i_2}\ldots a_{i_r}$ odvodit při použití pravidla $S \to A$ i $S \to B$, tedy w vytvořit ze seznamu A i ze seznamu B při použití slov se stejným indexem.)

1.14.13 Věta. Jsou dány bezkontextové gramatiky $\mathcal{G}_1$ a $\mathcal{G}_2$. Označme $L(\mathcal{G}_1)$ a $L(\mathcal{G}_2)$ jazyky generované gramatikami $\mathcal{G}_1$ a $\mathcal{G}_2$. Následující úlohy jsou nerozhodnutelné.

1. $L(\mathcal{G}_1) \cap L(\mathcal{G}_2) = \emptyset$.
2. $L(\mathcal{G}_1) = L(\mathcal{G}_2)$.
3. $L(\mathcal{G}_1) \subseteq L(\mathcal{G}_2)$.
4. $L(\mathcal{G}_1) = \Sigma^\star$.

1.14.14 Tiling problém. Jsou dány čtvercové dlaždičky velikosti 1 cm^2 několika typů. Každá dlaždička má barevné okraje. Máme neomezený počet dlaždiček každého typu.

Otázka: Je možné dlaždičkami vydláždit každou plochu daného typu tak, aby se dlaždičky dotýkaly hranami stejné barvy, za předpokladu, že dlaždičky nesmíme rotovat?

1.14.15 Věta. Tiling problém je nerozhodnutelný.