

1.15.11 Univerzální jazyk. *Univerzální jazyk* L_U je množina slov tvaru $\langle M \rangle \# w$, kde $\langle M \rangle$ je kód Turingova stroje a $w \in \{0, 1\}^*$ je binární slovo takové, že $w \in L(M)$.

1.15.12 Univerzální Turingův stroj. Popíšeme, velmi zhruba, Turingův stroj, který přijímá univerzální jazyk L_U . Tomuto Turingovu stroji se také říká *univerzální Turingův stroj* a značíme ho U .

Univerzální Turingův stroj U má 4 pásy. První páska obsahuje vstupní slovo $\langle M \rangle \# w$, druhá páska simuluje pásku Turingova stroje M a třetí páska obsahuje kód stavu, ve kterém se Turingův stroj M nachází. Dále má U ještě čtvrtou, pomocnou pásku.

Na začátku práce Turingova stroje U je na první pásce vstupní slovo $\langle M \rangle \# w$, ostatní pásy obsahují pouze B , blanky. Připomeňme, že kód Turingova stroje získáme takto. Předpokládejme, že Turingův stroj M se skládá z $(Q, \{0, 1\}, \{0, 1, B\}, \delta, q_1, \{q_2\})$, kde $Q = \{q_1, q_2, \dots, q_n\}$. Označme 0 jako X_1 , 1 jako X_2 , B jako X_3 , pohyb doprava R jako D_1 , pohyb doleva L jako D_2 . Pak jednotlivé přechody $\delta(q_i, X_j) = (q_k, X_l, D_m)$ kódujeme

$$t = 0^i 10^j 10^k 10^l 10^m, \text{ kde } 1 \leq i, k \leq n, 1 \leq j, l \leq 3, 1 \leq m \leq 2.$$

Turingův stroj M má kód

$$111 t_1 11 t_2 11 \dots 11 t_r 111.$$

Turingův stroj U neprve zkontroluje, že vstup je opravdu kódem Turingova stroje M následovaný binárním slovem. Jestliže není, U se neúspěšně zastaví.

V případě, že vstupní slovo je tvaru kód Turingova stroje M následovaný binárním slovem w , U přepíše slovo w na druhou pásku a na třetí pásku napíše 0. To je proto, že Turingův stroj je na začátku práce ve stavu q_1 kódovaném jako 0.

Nyní Turingův stroj U simuluje kroky Turingova stroje M s tím, že kdykoli se stroj M dostane do stavu q_2 (koncový „přijímací“ stav M), U se úspěšně zastaví. Toto poznáme tak, že na třetí pásce se objeví 00 předcházené a následované B , blanky.)

Poznamenejme, že je třeba ještě dalších technických detailů. Např. při přepisování slova w na druhou pásku to děláme tak, že za 0 ve vstupním slově w na pásku napíšeme 10, za 1 ve w na druhou pásku zapíšeme 100. Je-li na druhou pásku potřeba (vzhledem k přechodové funkci Turingova stroje M na druhou pásku napsat B , napíšeme 1000. Čtvrtá páska slouží k tomu, abychom na druhou pásku byli schopni vždy napsat stav pásy TM M .

1.15.13 Důsledek. Univerzální jazyk L_U je rekursivně spočetný.

1.15.14 Tvzení. Univerzální jazyk L_U není rekursivní.

Kdyby totiž L_U byl rekursivní, existoval by Turingův stroj M , který rozhodne L_U . Tj. M se vždy zastaví a na slovech z jazyka L_U se úspěšně zastaví, na slovech neležících v L_U se neúspěšně zastaví. Na základě tohoto Turingova stroje M bychom byli schopni rozhodnout diagonální jazyk L_d , o kterém víme, že není ani rekursivně spočetný, viz 1.15.10.

1.15.15 Redukce. Připomeňme definici redukce z 1.9.1.

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 úlohu \mathcal{V} značíme

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

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

1.15.16 Tvzení. Jsou dány dvě úlohy \mathcal{U} a \mathcal{V} takové, že $\mathcal{U} \triangleleft \mathcal{V}$. 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.

1.15.17 Tvzení. 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. Jazyk L_e není ani rekursivně spočetný.

1.15.18 Poznámka. Uvědomme si, že jazyk L_e je doplnkem 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 .

Univerzální Turingův stroj U se dá využít i k tomu abychom ukázali, že jazyk L_{ne} je rekursivně spočetný. Z redukce $L_U \triangleleft L_{ne}$ a 1.15.16 dostáváme, že L_{ne} není rekursivní. Fakt, že L_e není ani rekursivně spočetný pak vyplývá z 1.15.5.

1.15.19 Věta (Rice). Jakákoli netriviální 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.

1.16 Další nerozhodnutelné úlohy

1.16.1 V minulé přednášce 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} úplné úlohy.

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

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

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

$$w_{i_1} w_{i_2} \dots w_{i_r} = x_{i_1} x_{i_2} \dots x_{i_r}.$$

Otázka: Existuje řešení dané instance?

1.16.3 Příklady.

- 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

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

- Jsou dány seznamy

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

Tato instance nemá řešení.

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

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

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

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

Otázka: Existuje řešení dané instance?

1.16.5 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.

1.16.6 Věta. Platí

$$\text{MPCP} \triangleleft \text{PCP}.$$

1.16.7 Věta. Platí

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

1.16.8 Poznámka. Druhá redukce vyplývá z věty 1.16.6.

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 . Bude tedy slovo vytvořené podle seznamu A prefixem slova vytvořeného podle seznamu B . Slova se stanou stejnými teprve v okamžiku, kdy se TM dostaneme do koncového stavu; tj. kdy se ve slově podle seznamu B objeví koncový stav.

1.16.9 Důsledek. Postův korespondenční problém je nerozhodnutelný.

1.16.10 Poznámka. Kdybychom omezili možnou délku hledané posloupnosti i_1, i_2, \dots, 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.