

Contents

1 Problem 3.1	2
2 Problem 3.2	2
3 Problem 3.3	2
4 Problem 3.4	3
5 Problem 3.6	3
6 Problem 3.7	4
7 Problem 3.8	5
8 Problem 3.9	5
9 Problem 3.11	5

1 Problem 3.1

1. Виміряти 45 хвилин:

- (a) Запалюємо першу мотузку з двох сторін, так виміряємо 30 хвилин.
- (b) Одночасно з тим, як запалюємо першу запалимо другу, але тільки з однієї сторони.
- (c) Як догорить перша мотузка, запалюємо кінець другої мотузки. На той момент вона матиме лише 30 хвилин горіння, а з обома кінцями, що будуть горіти, вона згорить за 15 хвилин.
- (d) Таким чином ми виміряємо $30 + 15 = 45$ хвилин.

2 Problem 3.2

Для зручності вважатимемо $00i \equiv i$ для $i = \overline{1, 9}$ та $0j \equiv j$ для $j = \overline{10, 99}$.

Порахуємо кількість цифр, де маємо трійку на першому місці:

$$|A_1| = |\{3 * * \mid * - \text{будь яка цифра від } 0 \text{ до } 9\}| = 10 \cdot 10 = 100.$$

Порахуємо кількість цифр, де маємо трійку на другому місці:

$$|A_2| = |\{* 3 * \mid * - \text{будь яка цифра від } 0 \text{ до } 9\}| = 10 \cdot 10 = 100.$$

Порахуємо кількість цифр, де маємо трійку на третьому місці:

$$|A_3| = |\{* * 3 \mid * - \text{будь яка цифра від } 0 \text{ до } 9\}| = 10 \cdot 10 = 100.$$

$$\begin{aligned} |A_1 \cup A_2 \cup A_3| &= |A_1 \cup A_2 \cup A_3| - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_1 \cap A_3| + |A_1 \cap A_2 \cap A_3| = \\ &= 300 - 10 - 10 - 10 + 1 \\ &= 271 \end{aligned}$$

3 Problem 3.3

Докажемо узагальнену теорему про нерухому точку використовуючи звичайну теорему про нерухому точку.

sdasdasd

1. Маємо таку функцію в заданій нумерації Геделя:

$$\forall k \in \mathbb{N} : \varphi_c = \varphi_i : 0 \leq i \leq k.$$

2. Побудуємо таку функцію $g(x)$, котра буде всюди визначеною та обчислюваною:

$$g(x) = \begin{cases} c & x \leq k \\ f(x) & x > k \end{cases}.$$

$f(x)$ - довільна всюди визначена функція.

3. $g(x)$ задовільняє умовам теореми про нерухому точку.

$$\exists n \in \mathbb{N} : \varphi_n \simeq \varphi_{g(n)}.$$

4. $n \leq k \Rightarrow g(n) = c \Rightarrow \varphi_c \simeq \varphi_n$ - суперечить визначенню φ_c , що нас влаштовує.

5. $n > k \Rightarrow g(n) = f(n) \Rightarrow \varphi_{f(n)} = \varphi_n$

4 Problem 3.4

Простіше, потрібно довести:

$$\text{Range}(\varphi_n) = \text{Range}(\varphi_{f(n)}).$$

З теореми про нерухому точку, маючи $f(n)$ - унарна обчислювана функція:

$$\exists n \in \mathbb{N} : \varphi_n \simeq \varphi_{f(n)}.$$

Оскільки $\varphi_n \simeq \varphi_{f(n)}$ то і $\text{Range}(\varphi_n) = \text{Range}(\varphi_{f(n)})$ (легко доводиться від супротивного).

$$\exists n \in \mathbb{N} : \varphi_n \simeq \varphi_{f(n)} \Rightarrow \exists n \in \mathbb{N} : \text{Range}(\varphi_n) = \text{Range}(\varphi_{f(n)}).$$

Інакше кажучи:

$$\text{Range}_n = \text{Range}_{f(n)}.$$

5 Problem 3.6

Назвемо цю задачу *ACCEPT*.

Proof. Нехай існує вирішувач для задачі *ACCEPT*: $\exists M : M(\langle M_i, x \rangle) = \begin{cases} 1 & \text{if } M_i(x) = 1 \\ 0 & \text{if } M_i(x) = 0 \end{cases}$

$$\text{Побудуємо } M_r(M_i) = \begin{cases} 0 & M_i(\langle M_i \rangle) = 1 \\ 1 & M_i(\langle M_i \rangle) = 0 \end{cases}$$

$$M(\langle M_r, M_r \rangle) = ?$$

$$\text{if } M_r(\langle M_r \rangle) = 1 \xRightarrow{M_r \text{ definition}} M_r(\langle M_r \rangle) = 0$$

$$\text{if } M_r(\langle M_r \rangle) = 0 \xRightarrow{M_r \text{ definition}} M_r(\langle M_r \rangle) = 1$$

$$\Rightarrow \nexists M(\langle M_i, x \rangle)$$

□

6 Problem 3.7

Union Будуємо машину Тюрінга з $k_1 + k_2 + 1$ стрічками, де k_1 кількість стрічок на МТ M_1 , яка вирішує мову L_1 , M_2 машина Тюрінга з k_2 стрічками, яка вирішує мову L_2 . Та запускаємо ці машини в M паралельно на слові x . Якщо обидві не розпізнають слово x , то і машина M не розпізнає це слово. Інакше $M(x) = 1$.

Intersection Аналогічна побудова машини M на основі машин M_1, M_2 . Тепер машина M повертає 1 лише у випадку, коли M_1 та M_2 обидві приймають вхідне слово, інакше повертає 0.

Concatenation Будуємо машину Тюрінга M з $k_1 + k_2 + 2$ стрічками. Маємо M_1, M_2 з k_1, k_2 стрічками відповідно, які є вирішувачами для мов L_1, L_2 . Далі машина M на вхідному слові $x = x_1x_2 \dots x_n$ розглядає для $i = \overline{0, n}$ кожен префікс $x_1x_2 \dots x_i$ та суфікс $x_{i+1}x_{i+2} \dots x_n$. Передає цей префікс на вхід на машину M_1 , а суфікс на вхід машині M_2 . Якщо якась із машин M_1, M_2 повертає 0, то M переходить до наступного значення i , інакше очікує, що обидві вбудовані машини повернуть 1 та тоді сама M поверне 1. Якщо M пройшла всі значення i та не змогла отримати від M_1 та M_2 відповідь 1, то вона (машина M) повертає 0, тобто це слово не розпізнається.

Kleene closure Оскільки довели існування машини Тюрінга, існування якої доводить, що конкатенація вирішуваних за Тюрінгом мов не виводить за межі множини вирішуваних за Тюрінгом мов, то і замикання Кліні також можливо довести. Для вхідного слова $x = x_1x_2 \dots x_n : x_i \in L$ тобто фактично $x = x_{00}x_{01} \dots x_{10}x_{11} \dots x_{nm}$ потрібно перевірити усі можливі розбиття слова на $i \in \mathbb{N}$ частин. Побудова такої машини буде простою. Нехай маємо універсальну МТ M_u , яка буде проводити розбиття слова $x \in L^*$ на частини та послідовно запускати у собі МТ M , яка вирішує мову L , на цих розбиттях. Якщо M повертає 0, то M_u переходить до наступного розбиття. Якщо на всіх індексах розбиття M повертає 1, M поверне 1 також. Розбиття, де два сусідні індекси є порожнім словом вважаємо недопустимими. Якщо закінчилися усі можливі розбиття, та на них знаходився такий індекс, на якому M повертала 0, то повертає M 0. Таке МТ послідовно за скінчений час перебере усі розбиття скінченого слова.

Complement Маємо M яка вирішує мову L . Вирішувач мови \bar{L} :

$$\bar{M}(x) = \begin{cases} 1 & M(x) = 0 \\ 0 & M(x) = 1 \end{cases}.$$

7 Problem 3.8

Побудова МТ, що доводять замкненість множини мов, які розпізнаються за Тюрінгом, відносно операцій об'єднання, перетину, конкатенації та замикання Кліні є аналогічними рішенню 3.7, окрім випадку, коли вбудовані МТ не повертають нічого. Тоді сама МТ, яка моделює їх, не може повернути нічого та зациклюється також. Це не впливає на доведення, оскільки машина залишається розпізнавачем.

Але також ми можемо обмежити кількість тактів для машин, що моделюються, це допоможе для розпізнавання замикання, наприклад.

!CompI Якщо матимемо такий розпізнавач \overline{M} , який для мови L , яка розпізнається за Тюрінгом, буде розпізнавати її доповнення, то матимемо вирішувальність мови L , оскільки розпізнаємо L та \overline{L} . Але ж ми припустили, що мову L можемо лише розпізнати \Rightarrow *протиріччя* \Rightarrow **такого розпізнавача не існує, а отже множина мов, що розпізнаються за Тюрінгом не є замкненою.**

8 Problem 3.9

\Rightarrow Нехай маємо M , що вирішує мову L . Очевидно, що ця МТ також є і розпізнавачем мови L . Чи розпізнається доповнення цієї мови? Так, це доводить наступна побудована МТ:

$$\overline{M}(x) = \begin{cases} 1 & M(x) = 0 \\ 0 & M(x) = 1 \end{cases}.$$

Отже вирішувач довільної мови дозволяє розпізнавати її та доповнення до цієї мови.

\Leftarrow Нехай маємо мову L , яку розпізнає машина Тюрінга M_1 та також маємо машину Тюрінга M_2 , яка розпізнає доповнення цієї мови. Побудуємо наступну МТ:

$$M(x) = \begin{cases} 1 & M_1(x) = 1 \\ 0 & M_2(x) = 1 \end{cases}.$$

Машина Тюрінга M є вирішувачем мови L , а отже мова L є вирішуваною.

9 Problem 3.11

a) $L_{11} = \{\langle M \rangle : M \text{ accepts input word } 11\} \subseteq \{0, 1\}^*$

Використаймо теорему Райса: властивість саме мови; МТ M_1 приймає всі слова, МТ M_2 відхиляє всі слова $\Rightarrow 11 \notin L(M_1), 11 \in L(M_2) \Rightarrow$ властивість нетривіальна, а отже L_{11} не є вирішуваною.