Dominik Wawszczak
numer indeksu: 440014
numer grupy: 6

Zadanie 3

Rozpocznijmy od udowodnienia, że język L nie jest obliczalny.

Przypuśćmy nie wprost, że $L = L(\mathcal{H})$, dla pewnej maszyny Turinga $\mathcal{H}$, która zawsze terminuje i zostawia na taśmie 1, jeżeli wejściowy napis jest postaci $u_1\$u_2$, gdzie u_1 i u_2 są kodami podobnych maszyn Turinga, lub 0 w przeciwnym wypadku. Wskażemy redukcję

$$\mathrm{HALT}_\varepsilon \ \leqslant_f \ L,$$

co da oczywistą sprzeczność, gdyż problem $\mathrm{HALT}_\varepsilon$ jest nierozstrzygalny.

Weźmy funkcję f o następującej definicji:

$$f(\mathrm{kod}(\mathcal{M})) \ = \ \langle \mathrm{kod}(\mathcal{M}), \mathrm{kod}(\mathcal{M}') \rangle ,$$

gdzie $\mathcal{M}$ jest dowolną maszyną Turinga, a $\mathcal{M}'$ maszyną Turinga działającą w sposób opisany poniżej:

- jeśli wejściowym napisem jest ε, to maszyna $\mathcal{M}'$ najpierw symuluje działanie maszyny $\mathcal{M}$ na tym napisie, a następnie w miejsce wystąpienia pierwszego blanka wpisuje 1 i kończy działanie;
- w przeciwnym wypadku symuluje działanie maszyny $\mathcal{M}$ na napisie wejściowym, po czym kończy działanie.

Funkcja f jest oczywiście obliczalna.

Zauważmy, że

$$\mathcal{M} \text{ terminuje na } \varepsilon \quad \Longleftrightarrow \quad \text{maszyny } \mathcal{M} \text{ oraz } \mathcal{M}' \text{ nie są podobne,}$$

czyli równoważnie

$$\mathrm{kod}(\mathcal{M}) \ \in \ \mathrm{HALT}_\varepsilon \quad \Longleftrightarrow \quad \langle \mathrm{kod}(\mathcal{M}), \mathrm{kod}(\mathcal{M}') \rangle \ \notin \ L.$$

Wynika to z tego, że jeżeli $\mathcal{M}$ terminuje na ε, to $\mathcal{M}'$ również terminuje na ε, jednak daje inny wynik. Z drugiej strony, jeśli maszyny $\mathcal{M}$ oraz $\mathcal{M}'$ nie są podobne, to $\mathcal{M}$ terminuje na ε, ponieważ w przeciwnym wypadku, obie maszyny $\mathcal{M}$ oraz $\mathcal{M}'$ nie terminowałaby na ε, dając jednak inny wynik. Z implikacji w obie strony dostajemy więc równoważność.

Z powyższego wynika, że język L nie jest obliczalny.

Udowodnimy teraz, że dopełnienie języka L jest częściowo obliczalne.

Stworzymy maszynę Turinga $\mathcal{M}$, która na wejściu będzie przyjmować słowo postaci $u_1\$u_2$ i terminować, jeżeli u_1 i u_2 są kodami maszyn Turinga, które nie są podobne. Maszyna $\mathcal{M}$ będzie symulować działanie maszyn $\mathcal{M}_{u_1}$ i $\mathcal{M}_{u_2}$ na wszystkich możliwych wejściach $w \in \{0,1\}^*$, aż znajdzie wejście na którym dają one różne wyniki, stwierdzając przy tym, że nie są podobne.

Biorąc dowolną obliczalną bijekcję $g : \mathbb{Z}^+ \cup \{0\} \to \{0,1\}^*$ możemy łatwo symulować działanie maszyn $\mathcal{M}_{u_1}$ i $\mathcal{M}_{u_2}$ na wszystkich możliwych wejściach, obliczając wartości funkcji g dla kolejnych liczb całkowitych nieujemnych. Przykładem takiej bijekcji jest funkcja:

$$g(n) \ = \ \text{zapis binarny liczby } n+1 \text{ bez pierwszego znaku.}$$

Wówczas kilka pierwszych wartości funkcji g to: $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100 \ldots$.

Problem napotkamy w momencie, gdy co najmniej jedna z maszyn $\mathcal{M}_{u_1}$, $\mathcal{M}_{u_2}$ nie terminuje na pewnym wejściu, a obie terminują i dają inne wyniki na innym, jeszcze nie rozpatrzonym wejściu. Aby rozwiązać ten problem nie będziemy symulować maszyn na wszystkich możliwych wejściach po kolei, lecz sumulację będziemy przeprowadzać równolegle, na coraz większej liczbie wejść.

Na ćwiczeniach dowodzone było, że dowolna maszyna Turinga, zamiast działać na taśmie jednowymiarowej, może działać na taśmie dwuwymiarowej. Główna idea opierała się na tym, żeby na taśmie trzymać wszystkie trójki $(\text{wiersz}, \text{kolumna}, \text{wartość})$ wykorzystywanych pól, jednak nie będę zagłębiał się w szczegóły.

Maszyna $\mathcal{M}$ będzie działać w fazach, trzymając numer aktualnej fazy na taśmie. W n-tej fazie będzie przeprowadzać po jednym kroku symulacji obu maszyn $\mathcal{M}_{u_1}$ i $\mathcal{M}_{u_2}$ na pierwszych n wejściach, tj. $\{g(0), g(1), g(2), \ldots, g(n-1)\}$.

Kilka początkowych wierszy (taśm) naszej dwuwymiarowej taśmy jest potrzebne na istotne rzeczy takie jak np.: kody u_1 i u_2, numer aktualnej fazy i inne informacje niezbędne do pełnego sformalizowania działania maszyny $\mathcal{M}$. Niech więc k bedzie numerem pierwszego nieużywanego wiersza. Wykorzystamy go do symulowania działania maszyny $\mathcal{M}_{u_1}$ na $g(0)$. Wiersz $k+1$ natomiast zostanie wykorzystany do symulacji maszyny $\mathcal{M}_{u_2}$ na $g(0)$. Kolejne wiersze będą wykorzystywane do symulowania $\mathcal{M}_{u_1}$ albo $\mathcal{M}_{u_2}$ w zależności od parzystości numeru wiersza, na odpowiednich wartościach funkcji g.

Jeżeli maszyny $\mathcal{M}_{u_1}$ i $\mathcal{M}_{u_2}$ nie są podobne, to maszyna $\mathcal{M}$ w skończonym czasie znajdzie wejście, na którym obie maszyny $\mathcal{M}_{u_1}$, $\mathcal{M}_{u_2}$ terminują, dając różne wyniki, po czym sama również sterminuje.

Aby dokończyć rozwiązanie zadania, należy zauważyć, że język L nie jest częściowo obliczalny, gdyż w przeciwnym wypadku język L byłby obliczalny, co było udowodnione na wykładzie.

Z powyższego wynika, że odpowiedzi na pytania zawarte w podpunktach a), b) oraz c) to kolejno „Nie.", „Nie." oraz „Tak.", co kończy rozwiązanie zadania.

$\square$