Hubert Michalski

30 maja 2023

1 Podpunkt a)

Najpierw wykażemy, że dopełnienie języka L nie jest obliczalne, a z tego w szczególności wynika, że język L też nie jest obliczalny. Załóżmy nie wprost, że język L^C jest obliczalny. Wykażemy, za pomocą odpowiedniej redukcji, że wówczas język $HALT$ byłby obliczalny, co jest nieprawdą.

Weźmy zatem dowolną instancję $(u_{\mathcal{M}}, w)$ problemu $HALT$ i zredukujmy ją do instancji problemu $coSIMILAR$, czyli dopełnienia problemu $SIMILAR$. Oznaczmy przez # symbol, który nie występuje w alfabecie taśmowym maszyny $\mathcal{M}$. Dodatkowo niech $\mathcal{M}_{w\to\#}$ oznacza maszynę która działa identycznie jak $\mathcal{M}$ poza przypadkiem gdzie na wejście otrzymuje w, w takim wypadku nowa maszyna wypisywałaby na wyjście symbol # i terminowała. Można pokazać, że taką maszynę da się w prosty sposób skonstruować dodając jednego "$if'a$" przed wywołaniem $\mathcal{M}$. Rozważmy więc następującą funkcję:

$$(u_{\mathcal{M}}, w) \mapsto (u_{\mathcal{M}}, u_{\mathcal{M}_{w\to\#}})$$

Widać z definicji, że taka funkcja jest obliczalna. Zbadajmy zatem jak zachowują się obie maszyny na słowie w (na reszcie słów zachowują się identycznie). Zauważmy, że jeśli $\mathcal{M}$ terminuje na słowie w to maszyny $\mathcal{M}$ i $\mathcal{M}_{w\to\#}$ nie uznamy za *podobne*, ponieważ pierwotna maszyna nie może wypisać znaku # na wyjście. W przeciwnym przypadku jeśli $\mathcal{M}$ nie terminuje na w to widzimy, że predykat o terminowaniu obu maszyn nie jest spełniony, czyli maszyny uznamy za *podobne*. Łatwo więc zauważyć, że zachodzi równoważność:

$$(u_{\mathcal{M}}, w) \in HALT \iff (u_{\mathcal{M}}, u_{\mathcal{M}_{w\to\#}}) \in coSIMILAR$$

Zatem gdyby zachodziło $coSIMILAR = L(\mathcal{K})$ dla pewnej maszyny $\mathcal{K}$, to skonstruowalibyśmy maszynę $\mathcal{N}$ dla języka $HALT$, która dla słów postaci $(u_{\mathcal{M}}, w)$ oblicza słowo $(u_{\mathcal{M}}, u_{\mathcal{M}_{w\to\#}})$ i uruchamia na nim maszynę $\mathcal{K}$. Maszyna $\mathcal{N}$ akceptowałaby język $HALT$, co jest niemożliwe. ■

2 Podpunkty b) i c)

Udowodnimy, że dopełnienie języka L jest częściowo obliczalne, czyli skonstruujemy maszynę, która dla wszystkich słów postaci $(u_{\mathcal{M}}, u_{\mathcal{N}})$ z języka L^C będzie terminowała i mówiła, że słowo należy do języka a dla reszty będzie się wykonywać w nieskończoność. Rozważmy maszynę $\mathcal{K}$ (nazwijmy ją nadzorującą) która będzie uruchamiać nowe instancje maszyn $\mathcal{M}$ i $\mathcal{N}$ na słowach $w \in \{0,1\}^*$. Żeby w konsekwentny sposób wybierać kolejne słowa do sprawdzenia można skonstruować pod-procedurę która liczy poniższą funkcję $f : \mathbb{N} \to \{0,1\}^*$:

$$f(n) = \begin{cases} \varepsilon & n = 0 \\ bin(n-1) & n \in \mathbb{N} - \{0\} \end{cases}$$

Schemat działania maszyny nadzorującej będzie następujący: dla indeksu pętli $i = 0, 1, 2 \ldots$ będzie ona uruchamiać nowe instancje maszyn $\mathcal{M}$ i $\mathcal{N}$ na $f(i)$ oraz wykonywać jeden ruch na wszystkich dotychczasowo uruchomionych maszynach, które jeszcze nie terminowały. Jeśli jakieś dwie maszyny które zostały uruchomione na tym samym słowie zakończą działanie i wypiszą na wyjście różne słowa, to znaczy, że maszyny $\mathcal{M}$ i $\mathcal{N}$ nie są *podobne*, ponieważ znaleźliśmy świadka który na to wskazuje. Zauważmy, że postępując w ten sposób uruchomimy przeliczalnie wiele maszyn oraz dla danego słowa w zostaną kiedyś uruchomione na nim obie maszyny z wejścia, o ile wcześniej $\mathcal{K}$ się nie zatrzyma. Zatem pod warunkiem, że słowo $(u_{\mathcal{M}}, u_{\mathcal{N}})$ należy do języka L^C to uruchamiając maszynę nadzorującą $\mathcal{K}$ kiedyś się o tym dowiemy, czyli język ten jest częściowo obliczalny.

Otrzymujemy więc, że język L nie jest częściowo obliczalny, ponieważ gdyby L był częściowo obliczalny to by oznaczało, że jest on także obliczalny co prowadzi do sprzeczności. ■