Zadanie 31 TPI

Bartosz Michalak

4 June 2024

1 Zadanie 31

Pokaż, że problem odpowiedniości Posta pozostaje nierozstrzygalny, nawet jeśli ograniczymy sie tylko do słów długości co najwyżej 2. Czy dla słów długości dokładnie 2 dalej jest to nierozstrzygalne? (Dozwolono pokazanie, dla słów długości ≤ 3)

1.1 Słowa długości ≤ 3

W zadaniu 20 z listy 3 pokazaliśmy, że problem: ”Czy maszyna Turinga zatrzymuje sie na pustym słowie wejściowym?”jest nierozstrzygalny.

$M = (Q, \sum, \delta, q_0)$ - dowolna maszyna Turinga.

Proces tworzenia A i B:

A	B	Komentarz
$\$$	$\$q_0\$$	
a	a	$\forall a \in \sum \cup \{\_, \$\}$
qa	bp	jeśli $\delta(q, a) = (p, b, \rightarrow)$
qa	pb	jeśli $\delta(q, a) = (p, b, -)$
cqa	pcb	jeśli $\delta(q, a) = (p, b, \leftarrow) \wedge \forall c \in \sum \cup \{\_\}$
$q\$$	$bp\$$	jeśli $\delta(q, \_) = (p, b, \rightarrow)$
$q\$$	$pb\$$	jeśli $\delta(q, \_) = (p, b, -)$
$\$qa$	$\$p\_b$	jeśli $\delta(q, a) = (p, b, \leftarrow)$
tak a	tak	$\forall a \in \sum \cup \{\_\}$
a tak	tak	$\forall a \in \sum \cup \{\_\}$
nie a	nie	$\forall a \in \sum \cup \{\_\}$
a nie	nie	$\forall a \in \sum \cup \{\_\}$
tak $\$\$$	$\$$	
nie $\$\$$	$\$$	

Jest to analogiczna konstrukcja jak ta przedstawiona na wykładzie - redukujaca problem stopu do OPOP.
Tworzymy nowy alfabet: $\sum' = \sum \cup \{\overset{q}{a} : q \in Q \cup \{tak, nie\} \wedge a \in \sum \cup \{\_, \$\}\}$, gdzie $\overset{q}{a}$ jest skróconym do jednego znaku zapisem qa.

Otrzymujemy wiec:

A	B	Komentarz
$\$$	$\$\overset{q}{\$}$	
a	a	$\forall a \in \sum \cup \{\_, \$\}$
$\overset{q}{a}\mathrm{c}$	$\mathrm{b}\overset{p}{c}$	jeśli $\delta(q, a) = (p, b, \rightarrow) \wedge \forall c \in \sum \cup \{\_, \$\}$
$\overset{q}{a}$	$\overset{q}{a}$	jeśli $\delta(q, a) = (p, b, -)$
$\mathrm{c}\overset{q}{a}$	$\overset{p}{c}\mathrm{b}$	jeśli $\delta(q, a) = (p, b, \leftarrow) \wedge \forall c \in \sum \cup \{\_\}$
$\overset{q}{\$}$	$\mathrm{b}\overset{p}{\$}$	jeśli $\delta(q, \_) = (p, b, \rightarrow)$
$\overset{q}{\$}$	$\overset{p}{b}\$$	jeśli $\delta(q, \_) = (p, b, -)$
$\$\overset{q}{a}$	$\$\underline{p}\mathrm{b}$	jeśli $\delta(q, a) = (p, b, \leftarrow)$
$\overset{tak}{a}\mathrm{c}$	$\overset{tak}{c}$	$\forall a \in \sum \cup \{\_\}, \forall c \in \sum \cup \{\_, \$\}$
$\mathrm{a}\overset{tak}{c}$	$\overset{tak}{c}$	$\forall a \in \sum \cup \{\_\}, \forall c \in \sum \cup \{\_, \$\}$
$\overset{nie}{a}\mathrm{c}$	$\overset{nie}{c}$	$\forall a \in \sum \cup \{\_\}, \forall c \in \sum \cup \{\_, \$\}$
$\mathrm{a}\overset{nie}{c}$	$\overset{nie}{c}$	$\forall a \in \sum \cup \{\_\}, \forall c \in \sum \cup \{\_, \$\}$
$\overset{tak}{\$}\$$	$\$$	
$\overset{nie}{\$}\$$	$\$$	

W czterech pierwszych wierszach zawierajacych "tak"lub "nie"oraz w pierwszym wierszu po parze (a,a) musieliśmy uwzglednić dodatkowy, nastepny symbol po słowach z pierwszej tabelki, co było niezbedne do użycia nowododanych do alfabetu symboli postaci $\overset{q}{a}$. Pierwotnie ten dodatkowy symbol byłby wypisany w nastepnym kroku. Zatem działane zmodyfikowanej maszyny jest bardzo podobne do pierwotnej maszyny.

Obecnie dzieki zastosowanej konstrukcji, każde słowo należace do A i B ma długość $\leq 3 \Longrightarrow$ spełnia założenia zadania. Dowód poprawności redukcji - analogiczny do redukcji problemu stopu do OPOP.

1.2 Słowa długości dokładnie 2

$(*) := (\forall k \in \mathbb{N})(a = (a_1, a_2, ..., a_k), b = (b_1, b_2, ..., b_k)) \wedge$
$(\exists n \leq k)(\exists i = (i_1, i_2, ..., i_n))((a_{i_1}, ..., a_{i_n}) = (b_{i_1}, ..., b_{i_n}))$

$(*) \iff (\exists 1 \leq j \leq k)(a_j = b_j)$
Nie wprost:
$\neg((\exists 1 \leq j \leq k)(a_j = b_j)) \wedge (*) \iff$
$(\forall i \in i_1, ..., i_n)(a_i \neq b_i) \wedge (*) \iff$
$(a_{i_1} \neq b_{i_1}) \wedge (a_{i_2} \neq b_{i_2}) \wedge ... \wedge (a_{i_n} \neq b_{i_n}) \wedge (*) \iff$
$\neg(\exists n \leq k)(\exists i = (i_1, i_2, ..., i_n))((a_{i_1}, ..., a_{i_n}) = (b_{i_1}, ..., b_{i_n})) \wedge (*) \iff$
$\neg(\exists n \leq k)(\exists i = (i_1, i_2, ..., i_n))((a_{i_1}, ..., a_{i_n}) = (b_{i_1}, ..., b_{i_n})) \wedge$
$(\exists n \leq k)(\exists i = (i_1, i_2, ..., i_n))((a_{i_1}, ..., a_{i_n}) = (b_{i_1}, ..., b_{i_n})) \wedge$
$(\forall k \in \mathbb{N})(a = (a_1, a_2, ..., a_k), b = (b_1, b_2, ..., b_k)) \iff \perp \square$
Zatem wystarczy jedynie sprawdzić czy istnieja takie same $a_i \in A$ i $b_i \in B$, co oczywiście jest rozstrzygalne.