

Spis treści

| | | |
|----------|---------------------|----------|
| 1 | Egzamin 2008 | 2 |
| 1.1 | Część 2 | 2 |
| 1.1.1 | Zadanie 4 | 2 |
| 1.1.2 | zadanie 5 | 2 |
| 1.1.3 | zadanie 6 | 3 |

Rozdział 1

Egzamin 2008

1.1. Część 2

1.1.1. Zadanie 4

Czy istnieje algorytm rostrzygający dla danych dwóch deterministycznych automatów ze stosem, czy istnieje niepuste słowo, które zostanie zaakceptowane przez oba te automaty?

Przez deterministyczny automat ze stosem rozumiemy tu automat, którego każdy krok polega na wczytaniu jednego symbolu ze słowa wejściowego, jednego symbolu ze stosu i w zależności od wczytanych danych oraz od aktualnego stanu, na zmianie stanu i odłożeniu na stos ciągu (być może pustego) symboli (w szczególności taki automat nie wykonuje ε -przejść). Automat akceptuje słowo, jeśli po wczytaniu tego słowa stos jest pusty (tzn. zawięra jedynie symbol dna stosu).

Taki algorytm nieistnieje. W celu pokazania tego, stworzymy redukcję, przedstawioną poniżej, problemu PCP (nierozstrzygalnego) do naszego problemu z automatami.

$$f(\langle\langle l_1, r_1 \rangle, \dots, \langle l_k, r_k \rangle \rangle) = (A_1, A_2) \text{ gdzie } A_1 \text{ i } A_2 \text{ to automaty opisane w zadaniu} \quad (1.1)$$

Automaty A_1 i A_2 zbudujemy tak by akceptowały języki opisane poniżej:

$$L(A_1) = L(\{i_{j_1} \dots i_{j_s} l_{j_s}^R \dots l_{j_1}^R\}) \quad (1.2)$$

$$L(A_2) = L(\{i_{j_1} \dots i_{j_s} r_{j_s}^R \dots r_{j_1}^R\}) \quad (1.3)$$

Idea automaty A_1 : Automat wczytuje tylko słowa postaci ciąg indeksów, po którym występuje ciąg literek, w przeciwnym przypadku zatrzymuje się z niepustym stosem.

1. automat wczytuje kolejne indeksy i_j i wrzuca na stos słowa $l_{i_j}^R$
2. gdy zobaczy literkę to ściąga ją ze stosu jeżeli taka sama leży na wierzchu lub się zaczyna

1.1.2. zadanie 5

Niech $A \subset N$ (nie jest r.e.) będzie zbiorem tych liczb naturalnych, które są numerami programów zatrzymujących się dla wszystkich danych. Niech $E \subset N$ (nie jest r.e.) będzie zbiorem tych liczb naturalnych, które są numerami programów nie zatrzymujących się dla żadnych danych. Udowodnij, że $E \leq_{rek} A$.

Wskazówka: To jest łatwe zadanie. Przypomnij sobie co to jest redukcja. Pomyśl jakiego typu muszą być argumenty redukcji, która masz zbudować i jakiego typu mają być wartości. Nie gap się w pustą kartkę, napisz sobie równoważność, która ma zachodzić.

$$A = \{i : \forall_x \varphi_i(x) \neq \perp\} \quad (1.4)$$

$$E = \{j : \forall_x \varphi_j(x) = \perp\} \quad (1.5)$$

Mamy pokazać że:

$$E \leq_{rek} A \Leftrightarrow (\exists f \forall_{x \in N} x \in E \Leftrightarrow f(x) \in A) \quad (1.6)$$

Pokażemy że istnieje takie f . W tym celu stworzymy odpowiednią redukcję, której kod przedstawiony jest poniżej:

Niech $b : N \rightarrow N \times N$ będzie bijekcją (np taką przekątniową $b(1) = (1, 1)$, $b(2) = (1, 2)$, $b(3) = (2, 1)$, ...).

```
wczytaj n
t = numer takiego programu:
    wczytaj m
    x, k = b(m)
    uruchom fi_n(x) na k kroków
    jesli fi_n(x) zatrzyma sie i zwroci wynik to zapetl sie
    w.p.p. zwroc 1
zwroc t
```

$n \in E$

$$n \in E \rightarrow \forall_x \varphi_n(x) = \perp \rightarrow \forall_x \varphi_t(x) = 1 \neq \perp \rightarrow t = f(n) \in A$$

$n \notin E$

$$n \notin E \rightarrow \exists_x \varphi_n(x) \neq \perp \rightarrow \exists_m b(m) = (x, k) \wedge \varphi_n(x) \text{ uruchomione na } k \text{ kroków} \neq \perp \rightarrow \exists_m \varphi_t(m) = \perp \rightarrow n \notin A$$

1.1.3. zadanie 6

Niech A i E będą takie, jak w poprzednim zadaniu. Czy zachodzi nierówność $A \leq_{rek} E$?
Wskazówka: Co pamiętasz z ćwiczeń na temat nierówności $K \leq_{rek} \bar{K}$?

Bibliografia

[1] test reference