

Spis treści

1	Egzamin 2005	2
1.1	Część 2	2
1.1.1	Zadanie 5	2
2	Egzamin 2007 poprawka	4
2.1	Część 2	4
2.1.1	Zadanie 4	4
3	Egzamin 2008	5
3.1	Część 2	5
3.1.1	Zadanie 4	5
3.1.2	zadanie 5	5
3.1.3	zadanie 6	6
4	Egzamin 2011	8
4.1	Zadanie 5	8
4.2	Zadanie 5	8

Rozdział 1

Egzamin 2005

1.1. Część 2

1.1.1. Zadanie 5

Udowodnij, że zbiór takich par $\langle G_1, G_2 \rangle$ gramatyk bezkontekstowych, dla których $L(G_1) \subseteq L(G_2)$ nie jest rekurencyjny. Wskazówka: skorzystaj z nierozstrzygalności PCP, i z tego że język słów które nie są palindromami jest CFL.

$$A = \{ \langle G_1, G_2 \rangle : L(G_1) \subseteq L(G_2) \}$$

Korzystając ze wskazówki pokażę, że:

Dowód. NIEWPROST Załóżmy że $A - \text{rek} \Leftrightarrow \exists \phi_A \forall x [(\phi_A(x) = 1 \Leftrightarrow x \in A) \wedge (\phi_A(x) = 0 \Leftrightarrow x \notin A)]$. Korzystając z powyższego założenia skonstruujemy program phi_{PCP} rozpoznający przynależność do PCP. Program będzie działać następująco:

1. Wczytaj zbiór par: $P = \{(l_1, r_1), \dots, (l_k, r_k)\}$. Niech Σ będzie alfabetem symboli, z których zbudowane są słowa l_i oraz r_i z jednym dodatkowym symbolem '#' nie występującym w żadnym l_i oraz r_i .
2. Korzystając ze wskazówki niech G_2 będzie gramatyka generująca język słów nad alfabetem Σ , które nie są palindromami.
3. Niech:

$$G_1 = \langle \Sigma, \{S\}, \{S \rightarrow \#\} \cup \{S \rightarrow l_i^R S r_i : 1 \leq i \leq k\}, S \rangle$$

Zaóważmy, że:

$$L(G_1) = \{t^R \# v : \exists_{w_1, \dots, w_s \in \{1, \dots, k\}} t = l_{w_1} \dots l_{w_s} \wedge v = r_{w_1} \dots r_{w_s}\}$$

4. zwróć 1 jeżeli $\phi_A(\langle G_1, G_2 \rangle) = 0$, 0 wpp.

Zaóważmy że powyższy program rzeczywiście rozstrzyga przynależność do PCP:

$P \in PCP$ Wtedy istnieje taki ciąg $w_1, \dots, w_s \in \{1, \dots, k\}$, że $t = l_{w_1} \dots l_{w_s}$, $v = r_{w_1} \dots r_{w_s}$ i $t = v$. Słowo $t^R \# v$ jest palindromem i $t^R \# v \notin A$, więc $\phi_{PCP}(P) = \phi_A(\langle G_1, G_2 \rangle) = 0$

$P \notin PCP$ Wtedy nie istnieje taki ciąg by $t = v$. Słowo $t^R \# v$ nie jest palindromem, gdyż:

- a) jeżeli t i v są różnej długości to symbol $\#$ nie leży po środku,
- b) jeżeli t i v są równej długości to słowo nie jest palindromem bo $t \neq v$.

Zaóważmy, że wtedy $\forall_{t,v} t^R \# v \in A$ czyli żadne słowo generowane przez G_1 nie jest palidromem. Z powyższych rozważań wynika że: $\phi_{PCP}(P) = \phi_A(< G_1, G_2 >) = 1$

Z powyższych rozważań wynika sprzeczność gdyż pokazaliśmy że PCP jest rek, a wiemy że tak nie jest.

□

Rozdział 2

Egzamin 2007 poprawka

2.1. Część 2

2.1.1. Zadanie 4

Niech A będzie zbiorem takich par numerów programów $\langle n, m \rangle$, że dla każdej naturalnej k albo program o numerze n uruchomiony dla danej k się zatrzymuje, a program o numerze m uruchomiony dla danej k się nie zatrzymuje, albo program o numerze m uruchomiony dla danej k się zatrzymuje, a program o numerze n uruchomiony dla danej k się nie zatrzymuje.

Czy A jest rekurencyjnie przeliczalny?

$$A = \{ \langle n, m \rangle : \forall k [(\phi_n(k) = \perp \wedge \phi_m(k) \neq \perp) \vee (\phi_n(k) \neq \perp \wedge \phi_m(k) = \perp)] \} \quad (2.1)$$

Pokażemy, że A nie jest rekurencyjnie przeliczalny.

Założmy niewprost że:

$$A - r.e. \Leftrightarrow \exists \phi_A \forall n, m (\phi_A(\langle n, m \rangle) = 1 \Leftrightarrow \langle n, m \rangle \in A) \wedge (\phi_A(\langle n, m \rangle) = \perp \Leftrightarrow \langle n, m \rangle \notin A) \quad (2.2)$$

Skonstruujemy następujący program, rozpoznający przynależność do \overline{K} :

```
wczytaj n
t <- numer nastepujacego programu:
    wczutaj m
    return 1
t' <- numer nastepujacego programu:
    wczutaj n
    return phi_n(n)
return phi_A(t, t')
```

Program rzeczywiście rozpoznaje przynależność do \overline{K} :

$$n \in \overline{K} \Rightarrow \forall x \phi_{t'}(x) = \perp \Rightarrow \phi_A(t, t') = 1 \quad (2.3)$$

$$n \notin \overline{K} \Rightarrow \forall x \phi_{t'}(x) \neq \perp \Rightarrow \phi_A(t, t') = \perp \quad (2.4)$$

Tutaj mamy sprzeczność gdyż wiemy że \overline{K} nie jest r.e.

Rozdział 3

Egzamin 2008

3.1. Część 2

3.1.1. 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.

$$f(\langle (l_1, r_1), \dots, (l_k, r_k) \rangle) = (A_1, A_2) \text{ gdzie } A_1 \text{ i } A_2 \text{ to automaty opisane w zadaniu} \quad (3.1)$$

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

$$L(A_1) = L(\{i_{j_1} \dots i_{j_s} l_{j_s}^R \dots l_{j_1}^R\}) \quad (3.2)$$

$$L(A_2) = L(\{i_{j_1} \dots i_{j_s} r_{j_s}^R \dots r_{j_1}^R\}) \quad (3.3)$$

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

3.1.2. 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, którą masz zbudować i jakiego typu mają być wartości. Nie gap się w pustą kartkę, napisz sobie równoważność, która ma zachodzić.

$$A = \{i : \forall_x \varphi_i(x) \neq \perp\} \quad (3.4)$$

$$E = \{j : \forall_x \varphi_j(x) = \perp\} \quad (3.5)$$

Mamy pokazać że:

$$E \leq_{rek} A \Leftrightarrow (\exists_f \forall_{x \in N} x \in E \Leftrightarrow f(x) \in A) \quad (3.6)$$

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
    jeśli fi_n(x) zatrzyma się i zwróci wynik to zapętl się
    w.p.p. zwróć 1
zwroc t
```

$n \in E$

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

$n \notin E$

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

3.1.3. 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 $K \leq_{rek} \bar{K}$?

Korzystając ze wskazówki wiemy że $K \not\leq_{rek} \bar{K}$. Pokażemy, że:

$$K \leq_{rek} A \wedge E \leq_{rek} \bar{K} \Rightarrow A \not\leq_{rek} E \quad (3.7)$$

Najpierw pokażemy, że:

$$K \leq_{rek} A \Leftrightarrow \exists_f \forall_x x \in K \Leftrightarrow f(x) \in A \quad (3.8)$$

W tym celu skonstruujemy następującą redukcję f :

```
wczytaj n
t ← znajdź numer następującego programu:
    wczytaj m
    zwroc fi_n(n)
zwroc t
```

$n \in K$

$$n \in K \Rightarrow \forall_x \varphi_t(x) = \varphi_n(n) \neq \perp \Rightarrow n \in A$$

$n \notin K$

$$n \notin K \Rightarrow \forall_x \varphi_t(x) = \varphi_n(n) = \perp \Rightarrow n \notin A$$

Teraz pokażemy, że:

$$E \leq_{rek} \overline{K} \Leftrightarrow \exists_f \forall_x x \in K \Leftrightarrow f(x) \in \overline{K} \quad (3.9)$$

W tym celu skonstruujemy następującą redukcję f:

```

wczytaj n
t ← znajdz numer następnego programu:
    wczytaj m
    dla każdego i od 1 do ....
        dla każdego j od 1 do ...
            jeżeli fi_n(i) zwróci wyniki po j krokach zwróć 1
    zwróć 1
zwróć t

```

$n \in E$

$$n \in E \Rightarrow \forall_x \varphi_n(x) = \perp \Rightarrow \exists_{i,j} \varphi_n(i) \text{ zatrzyma się po } j \text{ krokach} \Rightarrow \forall \varphi_t(x) = \perp \Rightarrow \varphi_t(t) = \perp \Rightarrow t = f(n) \in \overline{K}$$

$n \notin E$

$$n \notin E \Rightarrow \exists_{i,j} \varphi_n(i) \text{ zatrzyma się po } j \text{ krokach} \Rightarrow \forall_x \varphi_t(x) = 1 \Rightarrow \varphi_t(t) = 1 \Rightarrow t = f(n) \notin \overline{K}$$

Rozdział 4

Egzamin 2011

4.1. Zadanie 5

$$A_0 = \{\phi_n : Dom(\phi_n) = \emptyset\}$$
$$A_1 = \{\phi_n : |Dom(\phi_n)| = 1\}$$

Czy $A_0 \leq_{REK} A_1$?

$$A_0 \leq_{REK} A_1 \Leftrightarrow \exists f \forall n n \in A_0 \Leftrightarrow f(n) \in A_1$$

REDUKCJA f:

```
wczytaj n
t <- znajdz numer programu:
    wczytaj m
    if m == 1 then zwroc 1
    else zwroc phi_n(m-1)
return t
```

4.2. Zadanie 5

Czy $A_1 \leq A_0$?

Pokażemy, że:

$$K \leq A_1 \wedge A_0 \overline{K} \Rightarrow A_1 \not\leq A_0 \quad (4.1)$$

1) Pokażemy, że:

$$K \leq A_1 \Leftrightarrow (\exists f_1 \forall n n \in K \Leftrightarrow f_1(n) \in A_1)$$

REDUKCA f_1 :

```
wczytaj n
t <- znajdz numer programu:
    wczytaj m
    if m == 1 then
        uruchom $phi_n(n)$
    else zapetl sie
return t
```


2) Pokażemy, że:

$$A_0 \leq \overline{K} \Leftrightarrow (\exists_{f_2} \forall_n n \in A_0 \Leftrightarrow f(n) \in \overline{K})$$

REDUKCJA f_2 :

```
wczytaj n
t <- znajdz numer programu:
    wczytaj m
    for i=1 to +inf
        for j=1 to i
            uruchom phi_n(m) na i krokow
            jezeli phi_n(m) zatrzyma sie i zwroci wynik to zwroc 1
return t
```

Bibliografia

[1] test reference