Spis treści

Rozdział 1

Egzamin 2005

1.1. Część 2

1.1.1. Zadanie 5

Udowodnij, że zbiór takich par $< G_1, G_2 >$ 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 sa palidromami jest CFL.

$$A = \{< G_1, G_2 >: L(G_1) \subseteq L(G_2)\}$$

Korzystając ze wskazówki pokażę, że:

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

1. Wczytaj zbiór par: $P = \{(l_1, r_1), \cdots, (l_k, r_k)\}$. 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 palidromami.

3. Niech:
$$G_1 =< \Sigma, \{S\}, \{S \to \#\} \cup \left\{S \to l_i^R S r_i : 1 \geqslant i \leqslant k\right\}, S >$$
Zaóważmy, że:
$$L(G_1) = \left\{t^R v : \exists_{w_1, \cdot w_s \in \{1, \cdots, k\}} t = l_{w_1} \cdots l_{w_s} \wedge v = r_{w_1} \cdot r_{w_s}\right\}$$

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

Zaóważmy że powyższy program rzeczywiście rozstrzyga przynależność do PCP:

$P \in PCP$ Wtedy istnieje taki ciąg $w_1, \cdots, w_s \in \{1, \cdots, k\}$, że $t = l_{w_1} \cdots l_{w_s}$, $v = r_{w_1} \cdots r_{w_s}$ i $t = v$. Słowo $t^R \# v$ jest palidromem i $t^R \# v \notin A$, wiec $\phi_{PCP}(P) = \phi_A(< G_1, G_2 >) = 0$

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

a) jeżeli t i v są rożnej długości to symbol # nie leży po środku,

b) jeżeli t i v są równej długości to słowo nie jest palidromem 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 rozwań 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 uruchominy dla danej k się zatrzymuje, a program o numerze m uruchominy dla danej k się nie zatrzymuje, albo program o numerze m uruchomiony dla danej k się zatrzymuje, a program o numerze n uruchominy dla danej k się nie zatrzymuje.

Czy A jest rekurencyjnie przeliczalny?

$$A = \{< n, m >: \forall_k [(\phi_n(k) = \perp \wedge \phi_m(k) \neq \perp) \vee (\phi_n(k) \neq \perp \wedge \phi_m(k) = \perp)]\} \tag{2.1}$$

Pokażemy, że A nie jest rekurencyjnie przeliczalny.

Załóżmy niewprost że:

$$A - r.e. \Leftrightarrow \exists_{\phi_A} \forall_{n,m} (\phi_A(< n, m >) = 1 \Leftrightarrow < n, m > \in A) \wedge (\phi_A(< n, m >) = \perp \Leftrightarrow < n, m > \notin A) \tag{2.2}$$

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

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

Program rzeczywiście rozpoznaje przynależność do $\overline{K}$:

$n \in \overline{K}$

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

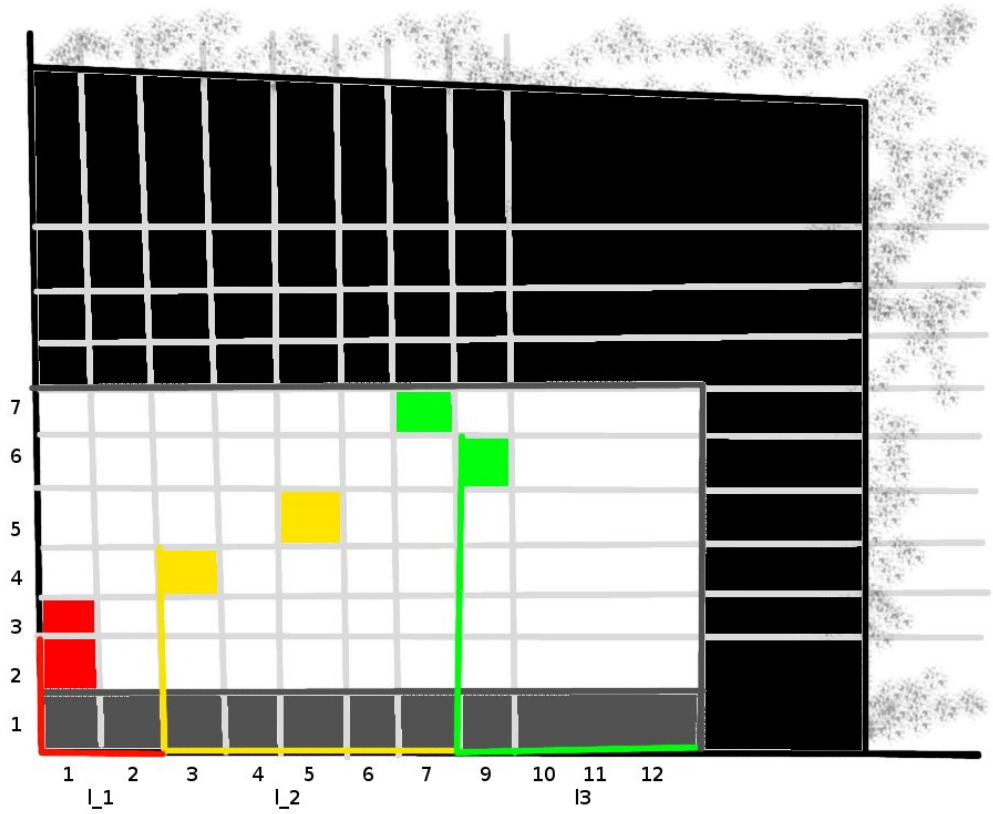
$n \notin \overline{K}$

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

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

2.1.2. zadanie 5

...



Rysunek 2.1: bla

1. Wszystkie pola poza wyszczególnionymi poniżej są białe.
2. W wierszu 1 zakodowane jest słowo w za pomoca odwrotności bijekcji f_Σ, zakończone białymi polami.
$$f_\Sigma : K_\Sigma \to \Sigma \tag{2.5}$$
Niech $n = |w|$.
3. Pola w powyższych wierszach będą kodwać numery słów, z których zbudowane jest słowo w za pomocą bijekcji:
$$f_I : K_I \to \{1, \cdots, r\} \tag{2.6}$$
Dla każdego wiersza (n pól od lewej krawędzi) $i > 1$ wszystkie pola są białe oprucz dokładnie jednego, którego kolor k należy do K_I. Niech $a_i = f_I(k)$ dla wiersza $2 \cdot i$. Niech $b_i = f_I(k)$ dla wiersza $2 \cdot i + 1$.
4. Korzystając z powyższych oznaczeń $\forall_i a_i = b_i$.
5. Jeżeli kolor z a_i znajduje się na pozycji j od lewej to $w_j \cdots w_{j+|l_{a_i}|} = l_{a_i}$.

6. Jeżeli kolor z a_i znajduje się na pozycji j od lewej to kolor z a_{i+1} znajduje się na pozycji $j + |l_{a_i}| + 1$.

7. Jeżeli w wierszu $i > 1$ kolor znajdue się na pozycji j od lewej

Automat będzie działał w następujących fazach:

1. hello

Dowód. □

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 jedneo symbulu ze słowa wejściowego, jednego symbolu ze stosu i w zależności od wczytanych danych oraz od aktualnego stanu, na zmianie stanu i odłożeni na stos ciągu (być może pustego) symboli (w szczególności taki automat nie wykonue ε-przejść). Automat akceptuje słowo, jeśli po wczytaniu tego słowa stos jest pusty (tzn. zaiwera jedynie symbol dna stosu).

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

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

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

$$L(A_1) = L(\{i_{j_1} \cdots i_{j_s} l^R_{j_s} \cdots l^R_{j_1}\}) \tag{3.2}$$

$$L(A_2) = L(\{i_{j_1} \cdots i_{j_s} r^R_{j_s} \cdots r^R_{j_1}\}) \tag{3.3}$$

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

1. automat wczytuje kolejne indeksy i_j i wrzuca na stos słowa $l^R_{i_j}$
2. gdy zobaczy literke to ściąga ją ze stosu jeżeli taka sama leży na wierzchu lub się zacina

3.1.2. zadanie 5

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

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

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

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

Mamy pokazać że:

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

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

Niech $b : N \to N \times N$ 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 krokow
                jesli fi_n(x) zatrzyma sie i zwroci wynik to zapetl sie
                w.p.p. zwroc 1
zwroc t
```

$n \in E$

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

$n \notin E$

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

3.1.3. zadanie 6

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

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

$$K \leqslant_{rek} A \ \wedge \ E \leqslant_{rek} \overline{K} \ \Rightarrow \ A \not\leqslant_{rek} E \tag{3.7}$$

Najpierw pokażemy, że:

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

W tym celu skonstrujemy następująca redukcje f:

```
wczytaj n
t <- znajdz numer nastepujacego 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:

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

W tym celu skonstrujuemy następującą redukcje f:

```
wczytaj n
t <- znajdz numer nastepujacego programu:
        wczytaj m
        dla kazdego i od 1 do ....
                dla kazdego j od 1 do ...
                        jezeli fi_n(i) zwroci wyniki po j krokach zwroc 1
        zwroc 1
zwroc t
```

$n \in E$

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

$n \notin E$

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

Bibliografia

[1] test reference