Problemy trudne obliczeniowo — Zadania

Część pierwsza

Damian Niwiński

Marzec 2001

Zadania punktowane

1. *Funkcja Ackermana* (2 pt). Klasę funkcji *pierwotnie rekurencyjnych* określamy jako najmniejszy zbiór PR funkcji $\mathbf{N}^k \to \mathbf{N}$ zawierający funkcje $\lambda x.x+1$, $\lambda x_1 \dots x_n.0$, $\lambda x_1 \dots x_n.x_i$ (dla każdych $1 \le i \le n$) oraz zamknięty na operacje

   — *złożenia:* jeśli $f : \mathbf{N}^k \to \mathbf{N}$ oraz $g_i : \mathbf{N}^m \to \mathbf{N}$, dla $i = 1, \dots, k$, należą do PR, to funkcja $f(g_1, \dots, g_k)$ należy do PR[1],

   — *rekursji pierwotnej:* jeśli g i h są funkcjami o arnościach k i $k+2$ odpowiednio, to funkcja $f : \mathbf{N}^{k+1} \to \mathbf{N}$ określona równaniami

$$\begin{array}{rcl} f(0, x_1, \dots, x_k) & = & g(x_1, \dots, x_k) \\ f(n+1, x_1, \dots, x_k) & = & h(n, f(n, x_1, \dots, x_k), x_1, \dots, x_k) \end{array}$$

   należy do PR.

   Funkcja Ackermana $A : \mathbf{N}^3 \to \mathbf{N}$, określona jest równaniami rekurencyjnymi

$$\begin{array}{rcl} A(0,0,x) & = & 0 \\ A(0,y+1,x) & = & A(0,y,x)+1 \\ A(1,0,x) & = & 0 \\ A(z+2,0,x) & = & 1 \\ A(z+1,y+1,x) & = & A(z, A(z+1,y,x), x) \end{array}$$

   Nie trudno jest sprawdzić, że $A(0,y,x) = x+y$, $A(1,y,x) = x \cdot y$, $A(2,y,x) = x^y$, oraz, że, dla każdego n, funkcja $\lambda xy.A(n,y,x)$ jest pierwotnie rekurencyjna. Dowieść, że funkcja A nie jest pierwotnie rekurencyjna.

   Wskazówka. Dowieść, że dla każdej funkcji pierwotnie rekurencyjnej $f : \mathbf{N} \to \mathbf{N}$, istnieje m, takie że $f(n) \le A(m,n,n)$, dla prawie wszystkich n.

2. (1 pt, zamknięte) Dowieść, że problem, czy dana gramatyka bezkontekstowa generuje język regularny jest algorytmicznie nierozstrzygalny.

[1] Dokładniej, mamy na myśli funkcję $\lambda x_1 \dots x_m.f(g_1(x_1 \dots x_m), \dots, g_k(x_1 \dots x_m))$.

3. (3 pt) Powiemy, że rodzina zbiorów $\mathcal{A} \subseteq P(\Sigma^*)$ ($\Sigma = \{0,1\}$) jest *rekurencyjnie prezentowalna*, jeśli istnieje funkcja rekurencyjna $f : \Sigma^* \to \Sigma^*$, taka że $L \in \mathcal{A}$ wtedy i tylko wtedy, gdy istnieje $x \in \Sigma^*$ takie, że $L(M_{f(x)}) = L$. Czy istnieje rekurencyjna prezentacja f rodziny wszystkich rekurencyjnie przeliczalnych podzbiorów Σ^*, w której każdy zbiór jest prezentowany *dokładnie raz* (tzn. dla każdego x istnieje *dokładnie jedno* z, takie że $W_x = L(M_{f(z)})$)?

   Powyższe pytanie można sformułować w sposób przypominający pewnik wyboru:

   > Dla $x \in \Sigma^*$, niech $A_x = \{z : W_z = W_x\}$. Zauważmy, że rodzina $\{A_x : x \in \Sigma^*\}$ tworzy podział zbioru Σ^*. Czy istnieje rekurencyjnie przeliczalny zbiór B mający dokładnie jeden element wspólny z każdym A_x?

4. (1pt) Rozważamy maszynę Turinga z *jedną* taśmą nieskończoną w prawo. Dodatkowo nakładamy ograniczenie (ang. *write-once*), że maszyna może pisać tylko w pustych komórkach taśmy, a symbol raz napisany nie może być zastąpiony żadnym innym symbolem (w szczególności nie może być "zmazany" tj. zastąpiony przez "blank"). Czy maszyna taka może rozpoznać język nieregularny?

   Uwaga. Jeśli przy ograniczeniu *write-once* pozwolimy używać dwóch taśm, to dowolny język rekurencyjnie przeliczalny może być akceptowany w ten sposób.

5. (2 pt) Dowieść, że jeśli M jest deterministyczną maszyną Turinga i $L(M)$ nie jest językiem regularnym, to istnieje stała c taka, że dla nieskończenie wielu słów w, maszyna M wykorzystuje co najmniej $c \cdot \log\log|w|$ komórek pamięci.

   (Wnika stąd, że na przykład klasa DSPACE $(\log\log\log n)$ zawiera jedynie języki regularne.)

Zadania przygotowawcze

Maszyny Turinga

1. Skonstruować maszynę Turinga obliczającą funkcję 2^n reprezentowaną *unarnie.* Bardziej dokładnie, zakładamy, że alfabet wejściowy składa się z jednego symbolu, powiedzmy $\{1\}$. Jeśli na wejściu dany jest ciąg n jedynek, (tzn. konfiguracja wejściowa jest $q_0 1^n$), to po wykonaniu obliczenia konfiguracja powinna być $q_f 1^{2^n}$.

2. Skonstruować maszynę Turinga obliczającą funkcję $\lceil \log n \rceil$ reprezentowaną *unarnie.*

3. Przyjmijmy $\Sigma = \{0,1\}$. Skonstruować maszyny Turinga rozpoznające następujące języki:

   (a) zbiór palindromów

   (b) $\{w\$w : w \in \Sigma^*\}$

   (c) $\{ww : w \in \Sigma^*\}$

   (d) zbiór ciągów reprezentujących binarnie liczby pierwsze

4. Graf zorientowany o n wierzchołkach ponumerowanych $0, 1, \ldots, n-1$, reprezentujemy ciągiem zer i jedynek długości n^2, takim, że k-ty bit jest 1 o ile istnieje krawędź z wierzchołka o numerze i do wierzchołka o numerze j, gdzie $k = i \cdot n + j + 1$.
   (a) Skonstruować *niedeterministyczną* maszynę Turinga rozpoznającą zbiór słów zero-jedynkowych, które w powyższy sposób reprezentują te grafy, w których istnieje ścieżka z wierzchołka o numerze 0 do wierzchołka o numerze $n-1$.
   (b) Skonstruować maszynę *deterministyczną* realizującą to samo zadanie.
5. Udowodnić, że dla każdej maszyny Turinga nad alfabetem wejściowym $\{0, 1\}$, istnieje równoważna jej maszyna, której alfabet *wszystkich* symboli roboczych obejmuje jedynie symbole 0,1,B.
6. (*) Udowodnić, że dla każdej maszyny Turinga (nad dowolnym alfabetem) istnieje równoważna jej maszyna o *trzech* stanach (oczywiście, maszyna ta ma na ogół większy alfabet symboli roboczych).
7. Pojęcie automatu ze stosem można rozszerzyć do automatu z k stosami, $k \geq 2$. Dowieść, że dla każdej maszyny Turinga istnieje równoważny jej deterministyczny automat z dwoma stosami. Wywnioskować stąd, że automat z k stosami, $k > 2$ może być symulowany przez automat z 2 stosami (symulację tą można również opisać bezpośrednio).
8. (*) Automat z k licznikami $c_1, \ldots, c_k$ jest zadany przez skończony ciąg instrukcji postaci $m : c_i := c_i + 1$, $m : c_i \dot{-} 1$, $m :$ if $c_i = 0$ then m_1 else m_2, STOP, gdzie m, m_1, m_2 są etykietami instrukcji, $x \dot{-} 1 = x - 1$, gdy $x > 0$ i $0 \dot{-} 1 = 0$. Zakładamy, że w chwili początkowej licznik c_1 zawiera daną wejściową, a pozostałe liczniki mają wartość 0. Dowieść najpierw, że dla każdej maszyny Turinga nad alfabetem $\{1\}$ istnieje równoważny jej automat z 4 licznikami. Następnie wykazać, że liczba liczników może być dalej zredukowana do dwóch. (Ograniczenie alfabetu maszyny Turinga nie jest istotne, bo słowa nad alfabetem n–literowym można również łatwo "przerobić" na liczby naturalne w systemie unarnym.)

Obliczalność i nierozstrzygalność

1. Udowodnić równoważność następujących warunków
   - język L jest częściowo obliczalny,
   - język L jest dziedziną pewnej funkcji częściowo obliczalnej,
   - język L jest zbiorem wartości pewnej funkcji częściowo obliczalnej,
   - język L jest zbiorem wartości pewnej funkcji obliczalnej.
2. Udowodnić, że język L jest obliczalny, wtedy i tylko wtedy, gdy jest skończony lub jest zbiorem wartości pewnej funkcji obliczalnej rosnącej.
3. Udowodnić następujące twierdzenie Turinga–Posta : jeśli zarówno język jak i jego uzupełnienie są częściowo obliczalne, to są również obliczalne.

4. Udowodnić, że istnieje zbiór częściowo obliczalny, którego uzupełnienie nie zawiera żadnego nieskończonego zbioru częściowo obliczalnego.

5. Dowieść, że następujące problemy są nierozstrzygalne :

   - Dane dwie gramatyki bezkontekstowe G_1, G_2. Pytanie : czy $L(G_1) \cap L(G_2) \neq 0$?
   - Dana gramatyka bezkontekstowa G nad alfabetem Σ. Pytanie : czy $L(G) = \Sigma^*$? Wskazówka : rozważać język słów $w\$v^R$, gdzie w i v są konfiguracjami maszyny Turinga M i $w \vdash_M v$.

6. Czy istnieje funkcja rekurencyjna $f : N \to N$, taka że, dla każdego n, jeżeli zbiór $L(M_n)$ jest rekurencyjny, to $L(M_{f(n)}) = L(M_n)$, przy czym maszyna $M_{f(n)}$ zatrzymuje się dla każdych danych? (Intuicyjnie, taka funkcja poprawiałaby, "tam gdzie się da", maszyny Turinga do maszyn, które zawsze się zatrzymują.)

7. *Drugie twierdzenie Rice'a.* Dowieść, że dla każdej nietrywialnej własności rekurencyjnie przeliczalnych podzbiorów N, tj. dla każdej rodziny zbiorów $\emptyset \subset \mathcal{S} \subset \{W_n : n \in N\}$, zbiór

$$L_{\mathcal{S}} \quad = \quad \{n : W_n \in \mathcal{S}\}$$

   jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy spełnione są następujące warunki:

   - jeśli $L \in \mathcal{S}$ i $L \subseteq L'$, to $L' \in \mathcal{S}$,
   - jeśli $L \in \mathcal{S}$, to pewien skończony podzbiór L też należy do $\mathcal{S}$,
   - rodzina zbiorów skończonych należących do $\mathcal{S}$ jest rekurencyjnie prezentowalna.

   Wywnioskować stąd, że następujące własności zbiorów rekurencyjnie przeliczalnych nie są rekurencyjnie przeliczalne:

   - $L = \emptyset$
   - $L = N$
   - L jest skończony
   - L jest rekurencyjny
   - L nie jest rekurencyjny
   - L jest semiliniowy
   - $L \cap \bar{K} \neq \emptyset$

   Natomiast następujące własności zbiorów rekurencyjnie przeliczalnych są rekurencyjnie przeliczalne:

- $L \neq \emptyset$
- L zawiera co najmniej 10 elementów
- $1994 \in L$
- $L \cap K \neq \emptyset$

8. *Zbiory proste w sensie Posta.* Dowieść, że istnieje rekurencyjnie przeliczalny zbiór liczb naturalnych, taki że jego uzupełnienie nie zawiera żadnego nieskończonego zbioru rekurencyjnie przeliczalnego.

9. Powiemy, że zbiór $A \subseteq N$ jest *m–sprowadzalny* do zbioru $B \subseteq N$ (notacja: $A \leq_m B$), jeśli istnieje funkcja rekurencyjna $f : N \to N$, taka że

$$(\forall n)\, n \in A \;\Leftrightarrow\; f(n) \in B$$

Zbiory A i B są *m–równoważne* (notacja: $A \equiv_m B$), jeśli $A \leq_m B$ i $B \leq_m A$. Dowieść, że następujące zbiory (por. def. poniżej) wyznaczają rózne klasy abstrakcji:

$$K,\; \bar{K},\; K \times \bar{K},\; K \oplus \bar{K}$$

Jakie relacje $\leq_m$ zachodzą pomiędzy tymi zbiorami?

Uwaga. Dla $A, B \subseteq N$, definiujemy

$$A \oplus B \;=\; \{2x \,:\, x \in A\} \cup \{2x+1 \,:\, x \in B\}$$

Natomiast

$$A \times B \;=\; \{\langle x, y\rangle \,:\, x \in A,\, y \in B\}$$

gdzie $\langle\, ,\, \rangle$ jest dowolną rekurencyjną funkcją pary.

10. Liczbę rzeczywistą nazywamy *rekurencyjną*, jeśli jej ciąg cyfr w zapisie dziesiętnym jest rekurencyjny. Innymi słowy, liczba $r \in R$ jest rekurencyjna, o ile istnieje liczba całkowita a i funkcja rekurencyjna $f : N \to N$, o wartościach w $\{0, 1, \ldots, 9\}$, taka że

$$r \;=\; a + \sum_{n=1}^{\infty} \frac{f(n)}{10^n}$$

Oczywiście, własność "być liczbą rekurencyjną" nie zależy od wyboru podstawy systemu pozycyjnego.

(a) Dowieść, że liczby rekurencyjne tworzą ciało rzeczywiście domknięte (tzn. każdy wielomian stopnia nieparzystego ma pierwiastek).

(b) Podać przykład funkcji rekurencyjnych f i g, takich że ciąg $f(n)/g(n)$ jest zbieżny do pewnej liczby rzeczywistej, ale liczba ta **nie** jest rekurencyjna.

11. *Uwaga.* W tym i w następnych zadaniach należy zastosować *twierdzenie o rekursji*:

Dla dowolnej funkcji rekurencyjnej f, istnieje x takie, że $W_x = W_{f(x)}$.

Wykazać, że istnieje $x \in \Sigma^*$, takie że $W_x = \{x\}$.

12. Wykazać, że istnieje $x \in \Sigma^*$, takie że φ_x (tj. funkcja częściowa obliczana przez maszynę o kodzie x) jest wszędzie określoną funkcją stałą, stale równą x.

13. Napisać program komputerowy P (np. w języku Pascal) nie przyjmujący danych wejściowych ani nie komunikujący się z otoczeniem, taki że w wyniku wykonania programu P na ekranie komputera wyświetla się tekst t, przy czym jest to właśnie dokładnie tekst programu P.