Języki, Automaty i Obliczenia

Zadania

Damian Niwiński i Wojciech Rytter

Październik 1998

1 Słowa i liczby

Do tej partii zadań może podejść Czytelnik bez żadnej znajomości teorii automatów.

1. *Słowa pierwotne.* Słowo $w \in \Sigma^*$ nazywamy *pierwotnym* (ang. *primitive*), jeśli nie da się go przedstawić $w = v^n$, inaczej niż dla $n = 1$ i $v = w$.

   (a) Dowieść, że dla każdego słowa niepustego w, istnieje *dokładnie jedno* słowo pierwotne v, takie że $w = v^n$, dla pewnego $n \geq 1$.

   Liczbę n nazywamy *wykładnikiem* słowa w.

   (b) O słowach wv i vw mówimy, że są w *relacji koniugacji*. Dowieść, że jest to relacja równoważności.

   Dowieść, że dwa słowa będące w relacji koniugacji mają ten sam wykładnik. Jaka jest moc klasy abstrakcji relacji koniugacji dla słowa o długości m i wykładniku n?

2. *Język nawiasowy.* Wykazać, że zbiór poprawnie uformowanych ciągów nawiasów może być zdefiniowany na dwa *równoważne* sposoby:

   - Jako najmniejszy zbiór L zawierający ciąg pusty oraz taki, że jeśli $w, v \in L$, to również $(w), wv \in L$.
   - Zbiór słów nad alfabetem $\{(,)\}$, w których ilość ")" jest taka sama jak ilość "(", a w każdym prefiksie ilość ")" jest niewiększa niż ilość "(" .

3. *Zbiory semi–liniowe.* Zbiór liczb naturalnych postaci $\{a + bn : n \in \mathbf{N}\}$, dla ustalonych $a, b \in \mathbf{N}$ nazywamy *liniowym*. Zbiór bedący sumą skończonej liczby zbiorów liniowych nazywamy *semiliniowym*. (Gdy sumowana rodzina jest pusta, otrzymujemy zbiór pusty.)

   (a) Dowieść, że każdy zbiór postaci $\{a + b_1 n_1 + \ldots + b_k n_k : n_1, \ldots, n_k \in \mathbf{N}\}$, dla ustalonych k i $a, b_1, \ldots, b_k \in \mathbf{N}$, jest semi–liniowy.

   Wskazówka. Wykorzystać podziały zbioru liczb naturalnych na klasy abstrakcji relacji przystawania *modulo* m, dla odpowiednio dobranych liczb m.

(b) W grafie zorientowanym ustalamy dwa wierzchołki. Interesuje nas zbiór długości wszystkich możliwych ścieżek pomiędzy tymi wierzchołkami. Dowieść, że jest to zbiór semi–liniowy.

(c) Dowieść, że rodzina zbiorów semi–liniowych jest zamknięta na skończone sumy, przecięcia oraz na uzupełnienie względem N.

4. *Kody.* Zbiór $C \subseteq \Sigma^+$ nazywamy *kodem* jeli każde słowo $w \in \Sigma^*$ dopuszcza co najwyżej jedną faktoryzację względem C (tzn., da się "odkodować").

   Niech $\Sigma = \{a, b\}$. Dowieść, że zbiór $\{aa, baa, ba\}$ jest kodem, a zbiór $\{a, ab, ba\}$ nie jest.

2 Języki regularne

2.1 Automaty skończone i wyrażenia regularne

1. Dowieść, że dla dowolnych języków L, M, $(L^*M^*)^* = (L \cup M)^*$.

2. *Dodawanie.* Rozważamy słowa nad alfabetem $\{0,1\}^3$. Powiemy, że słowo $\langle a_1, b_1, c_1\rangle \ldots \langle a_n, b_n, c_n\rangle$ długości n *reprezentuje dodawanie*, jeśli liczba reprezentowana binarnie przez słowo $c_1 \ldots c_n$ jest sumą liczb reprezentowanych przez $a_1 \ldots a_n$ i $b_1 \ldots b_n$. Na przykład, ciąg $\langle 0,0,1\rangle\langle 1,1,1\rangle\langle 0,1,0\rangle\langle 1,1,0\rangle$ reprezentuje dodawanie (5+7=12). Podać wyrażenie regularne opisujące zbiór wszystkich słów reprezentujących dodawanie.

3. *Podzielność.*

   (a) Skonstruować skończony automat deterministyczny nad alfabetem $\{0, 1, \ldots, 8, 9\}$ rozpoznający zbiór dziesiętnych reprezentacji wielokrotności liczby 7.

   (b) Jak wyżej, ale przy reprezentacji odwrotnej, tj. poczynając od cyfr najmniej znaczących.

   (c) Uogólnić niniejsze zadanie.

4. *Alfabet jednoliterowy.*

   (a) Dowieść, że język $L \subseteq \{a\}^*$ jest regularny wtedy i tylko wtedy, gdy zbiór liczb naturalnych $\{n : a^n \in L\}$ jest semi–liniowy w sensie rozdziału 1.

   (b) Dowieść, że dla dowolnego zbioru $X \subseteq \{a\}^*$, język X^* jest regularny.

5. *Zbiory semi–liniowe.* Dowieść, że jeśli $M \subseteq \mathbf{N}$ jest zbiorem semi–liniowym, to język $\{\mathrm{bin}(m) : m \in M\}$ jest regularny, gdzie $\mathrm{bin}(m)$ oznacza binarną reprezentację liczby m.

   Uwaga. Analogiczny fakt można udowodnić dla dowolnej reprezentacji, a zatem zbiór semi–liniowy jest regularny w reprezentacji o podstawie k, dla $k \geq 1$. W przeciwnym kierunku wiadomo, że jeśli zbiór jest regularny w reprezentacjach o względnie pierwszych podstawach p i q, to jest semi–liniowy (wniosek z Twierdzenia Cobhama).

6. Dowieść, że dla danych liczb $a, b, p, r \in \mathbf{N}$, język

$$L = \{x\$y : (a \cdot \text{bin}(x) + b \cdot \text{bin}(y)) \equiv r \pmod{p}\}$$

jest regularny.

7. Dla danego automatu A rozpoznającego język L, skonstruować automat rozpoznający język

$$Cycle(L) = \{vu : uv \in L\}$$

2.2 Własności języków regularnych

1. Niech L bedzie językiem regularnym nad alfabetem $\{0, 1\}$. Dowieść, że następujący język jest regularny:

$\{w : w \in L \wedge$ spośród słów o dlugości $|w|$, w jest najmniejsze w porządku leksykograficznym$\}$

2. (*) Przyjmujemy, że każde niepuste słowo binarne $w \in \{0, 1\}^*$, $w = w_1 \ldots w_k$, reprezentuje pewien ułamek w przedziale $[0, 1)$,

$$\text{bin}(w) = w_1\frac{1}{2} + w_2\frac{1}{2^2} + \ldots + w_k\frac{1}{2^k}$$

Dla liczby rzeczywistej $r \in [0, 1]$, niech

$$L_r = \{w : \text{bin}(w) \leq r\}$$

Dowieść, że język L_r jest regularny wtedy i tylko wtedy, gdy liczba r jest wymierna.

3. Niech L będzie językiem regularnym. Dowieść, że następujące języki są również regularne:

- $\frac{1}{2}L =_{df} \{w : (\exists u)\,|u| = |w| \wedge wu \in L\}$
- $\sqrt{L} =_{df} \{w : ww \in L\}$

4. (*) Niech L będzie językiem regularnym. Dowieść, że następujące języki są również regularne:

(a) $\text{Root}(L) = \{w : (\exists n \in N)\, w^n \in L\}$

(b) $\text{Sqrt}(L) = \{w : (\exists u)|u| = |w|^2 \wedge wu \in L\}$
Wskazówka. $n^2 = 1 + 3 + \ldots(2n - 1)$.

(c) $\text{Log}(L) = \{w : (\exists u)|u| = 2^{|w|} \wedge wu \in L\}$
Wskazówka. $2^n = 1 + 2 + 2^2 + 2^{n-1} + 1$.

(d) $\text{Fibb}(L) = \{w : (\exists u)|u| = F_{|w|} \wedge wu \in L\}$
gdzie F_n jest n-tą liczbą Fibonacciego tzn.

$$F_1 = F_2 = 1$$
$$F_{n+2} = F_n + F_{n+1}$$

5. Dowieść, że dla dowolnego języka regularnego L, język

$$\{w : w^{|w|} \in L\}$$

jest również regularny.

6. Niech będzie dany język regularny L i dowolne, niekoniecznie regularne, języki L_1, $\ldots$, L_m. Skonstruować skończony automat deterministyczny rozpoznający język nad alfabetem $\{1, \ldots, m\}$

$$L = \{i_1 i_2 \ldots i_k : L_{i_1} L_{i_2} \ldots L_{i_k} \subseteq L\}$$

7. Dowieść, że zbiór palindromów nad alfabetem o co najmniej dwóch elementach nie jest regularny.

8. Niech $\mathrm{Pal}_\Sigma^{\geq 1}$ oznacza zbiór palindromów o długości co najmniej 2 nad alfabetem Σ. Dowieść, że język $(\mathrm{Pal}_\Sigma^{\geq 1})^*$ jest regularny wtedy i tylko wtedy, gdy $|\Sigma| \leq 2$.

9. Dowieść, że zbiór wyrażeń regularnych nie jest regularny.

10. Dowieść, że następujące języki nie są regularne:

- $\{a^{2^n} : n \in N\}$
- $\{a^p : p \text{ jest liczbą pierwszą}\}$
- $\{a^i b^j : \mathrm{nwd}(i,j) = 1\}$

11. Niech L będzie językiem regularnym. Dowieść, że języki:

- $L_{+--} = \{w : (\exists u)\, |u| = 2|w| \wedge wu \in L\}$
- $L_{++-} = \{w : (\exists u)\, 2|u| = |w| \wedge wu \in L\}$
- $L_{-+-} = \{w : (\exists u, v)\, |u| = |v| = |w| \wedge uwv \in L\}$

są językami regularnymi, a język

- $L_{+-+} = \{uv : (\exists w)\, |u| = |v| = |w| \wedge uwv \in L\}$

może nie być regularny.

12. Czy z faktu, że język $\mathit{Cycle}(L) = \{vu : uv \in L\}$ jest regularny, można wnioskować, że język L jest regularny?

13. Przeplotem słów w i v nazwiemy dowolne słowo długości $|w| + |v|$, które można rozbić na rozłączne podciągi w i v. Na przykład, przeplotami słów ab i acb są słowa $abacb$, $aabcb$, $acabb$, $acbab$, $aacbb$. Przeplotem języków L i M jest zbiór wszystkich możliwych przeplotów słów $w \in L$, $v \in M$. Język ten oznaczamy $L \parallel M$.

Dowieść, że jeśli L i M są językami regularnymi, to $L \parallel M$ jest również regularny.

14. Określamy $L^\sharp = L \cup (L \parallel L) \cup (L \parallel L \parallel L) \cup \ldots$. Podać przykład języka regularnego L nad dwuelementowym alfabetem, dla którego $L^\sharp$ nie jest językiem regularnym.

15. Dowieść, że jeśli w Zadaniu 2 z punktu 2.1 zamienimy *dodawanie* przez *mnożenie*, to otrzymany język (nad alfabetem $\{0,1\}^3$) jest wprawdzie dobrze określony, ale nie jest regularny.

16. Zaprojektować algorytm, który dla danych automatów deterministycznych A, B, rozstrzyga, czy $L(A) = L(B)$.

17. Niech A będzie ustalonym automatem deterministycznym. Zaprojektować algorytm, który dla danej liczby n oblicza ilość słów długości n akceptowanych przez A.

2.3 Automaty minimalne.

1. Niech $w \in \Sigma^*$ będzie słowem o długości $|w| = n > 0$. Dowieść, że minimalny automat deterministyczny rozpoznający wszystkie słowa nad Σ, które zawierają w jako podsłowo, ma dokładnie $n + 1$ stanów.

2. Niech $w \in \Sigma^*$ będzie słowem o długości $|w| = n > 0$. Dowieść, że minimalny automat deterministyczny rozpoznający wszystkie podsłowa słowa w ma nie więcej niż $2n$ stanów.

3. Trzy drużyny piłkarskie A, B i C rozgrywają między sobą serię spotkań towarzyskich, przy czym umówiły się, że każdy kolejny mecz jest rozgrywany pomiędzy drużyną, która wygrała poprzedni mecz i drużynę, która nie brała udziału w tym spotkaniu (tzn. jeśli drużyna X wygrała z drużyną Y, to następny mecz rozegrają X i Z). Zakładając, że *nie ma remisów*, rozważamy zbiór słów nad alfabetem $\{A, B, C\}$, stanowiących ciągi możliwych wyników spotkań. Dowieść, że jest to język regularny. Zbudować minimalny automat deterministyczny, który go rozpoznaje.

4. Pan X zakupił pakiety trzech różnych akcji: A, B i C. Każdego dnia bada wzajemne położenie wartości swoich akcji, które może być reprezentowane jako *uporządkowanie* zbioru $\{A, B, C\}$, w kierunku od najmniej do najbardziej wartościowej. (Przyjmujemy założenie, że dwie różne akcje mają zawsze *różne* wartości.) Pan X postanowił, że jeśli w ciągu dwóch kolejnych dni któraś z akcji dwukrotnie spadnie na niższą pozycję, to sprzeda tę akcję. Np. jeśli kolejne notowania (w tym wypadku: uporządkowania) są $(A, B, C), (B, C, A), (C, B, A)$ to pan X sprzeda pakiet akcji C. Rozważamy zbiór G wszystkich uporządkowań liniowych zbioru $\{A, B, C\}$. Dowieść, że zbiór tych ciągów nad alfabetem G, które opisują takie wyniki notowań giełdowych, przy których pan X nie pozbędzie się żadnego pakietu akcji, jest językiem regularnym. Znaleźć liczbę stanów automatu minimalnego akceptującego ten język.

5. Pan X postanowił, że każdego dnia będzie pracował lub nie, przestrzegając przy tym zasady, by na żadne siedem kolejnych dni nie przypadało więcej niż cztery dni pracy. Możliwy rozkład dni pracy pana X w ciągu n kolejnych dni możemy przedstawić jako n-bitowe słowo, gdzie 1 odpowiada dniowi pracy, a 0 dniowi odpoczynku. Dowieść, że zbiór wszystkich tak otrzymanych słów jest językiem regularnym (nad alfabetem $\{0,1\}$). Znaleźć minimalny automat deterministyczny.

6. *Eksplozja stanów.* Niech $\Sigma = \{0, 1, \ldots, k\}$ i niech dla $m = 1, \ldots, k$, L_m będzie zbiorem słów v nad Σ o następującej własności:

   > m występuje w v, ale przed pierwszym i pomiędzy każdymi dwoma kolejnymi wystąpieniami m, $m-1$ występuje co najmniej dwa razy.

   Nietrudno jest skonstruować deterministyczny automat o 4 stanach rozpoznający L_m.

   Dowieść, że każdy (nawet niedeterministyczny) automat rozpoznający język $L_1 \cap \ldots \cap L_k$ musi mieć co najmniej $2^{k+1} - 1$ stanów.

   Wskazówka. Oszacować od dołu długość najkrótszego słowa w tym języku.