Języki, automaty i obliczenia
kolokwium – przykładowe rozwiązania zadań
24 maja 2017

Rozwiązanie zadania 1. Gdyby język był rozpoznawany przez deterministyczny automat ze stosem, to jego dopełnienie $L = \{ww \mid w \in \{a,b\}^*\}$ też byłoby rozpoznawane przez deterministyczny automat ze stosem. Ale L nie jest bezkontekstowy – sprzeczność.

To że L nie jest bezkontekstowy pokazujemy za pomocą pompowania. Rozważmy słowo postaci

$$w_n \; = \; a^n b^n a^n b^n \in L$$

dla dowolnie dużego n, i wszystkie możliwe wybory dwóch infiksów u, v, które będą pompowane, pamiętając, że u i v mieszczą w pewnym infiksie słowa w_n o długości co najwyżej n. Czyli mamy 2 przypadki:

1. u i v należą do tego samego bloku (przez *blok* rozumiemy maksymalny ciąg jednakowych liter),
2. u i v należą do dwóch sąsiednich bloków (przy czym albo każde z nich całkowicie mieści się w jednym z bloków, albo jedno z nich mieści się całkowicie w bloku a drugie przecina się z dwoma blokami).

W każdym z dwóch przypadków wycięcie ze słowa infiksów u i v (czyli zastąpienie u przez u^i oraz v przez v^i dla $i = 0$) zmniejsza długość jednego lub dwóch sąsiednich bloków i tym samym wyprowadza słowo poza język L. Zatem L nie jest bezkontekstowy.

Rozwiązanie zadania 2. Gdy $L = \{a,b\}^*$, język L' zawiera słowa nad $\{a, b, \#\}$ zawierające dokładnie jeden symbol $\#$, dokładnie w środku. Język L' oczywiście nie jest regularny, co łatwo pokazać za pomocą pompowania: wybieramy do pompowania dowolne słowo z L, np.

$$a^n \# a^n \in L',$$

i jakikolwiek wybór infiksu do pompowania, w pierwszej połówce słowa, wyprowadzi nas poza L'.

Z kolei niezależnie od wyboru L, język L' jest bezkontekstowy. Żeby to pokazać, rozważmy dowolny (być może niedeterministyczny) automat skończony $\mathcal{A}$ rozpoznający L o jednym stanie początkowym q_0, i skonstruujmy gramatykę (liniową) generującą L'. Symbole nieterminalne gramatyki to stany automatu $\mathcal{A}$ (zakładamy tutaj, bez utraty ogólności, że zbiór stanów $\mathcal{A}$ jest rozłączny z alfabetem wejściowym). Symbol początkowy to q_0. Dla dowolnych dwóch symboli nieterminalnych q i p, i dla dowolnej litery alfabetu $a \in \Sigma$, gramatyka zawiera produkcję

$$q \longrightarrow apa$$

o ile automat $\mathcal{A}$ ma przejście $q \xrightarrow{a} p$. Ponadto, dla każdego stanu akceptującego q, gramatyka ma produkcję

$$q \longrightarrow \#.$$

Łatwo widać, że gramatyka generuje język L': mamy odpowiedniość jeden-do-jeden pomiędzy biegami akceptującymi automatu $\mathcal{A}$ (na słowie w), a wyprowadzeniami gramatyki (dla słowa $w\#w^R$).

Rozwiązanie zadania 3. Zaczniemy od pokazania, że język generowany przez gramatykę nieciekawą jest regularny. Niech G będzie nieciekawą gramatyką nad alfabetem Σ. Rozważmy trzy skończone zbiory słów nad Σ:

$$\begin{aligned}
L &= \{w \in \Sigma^* \mid G \text{ ma produkcję } X \longrightarrow wX\} &&= \{l_1, \dots, l_k\} \\
P &= \{w \in \Sigma^* \mid G \text{ ma produkcję } X \longrightarrow Xw\} &&= \{p_1, \dots, p_m\} \\
S &= \{w \in \Sigma^* \mid G \text{ ma produkcję } X \longrightarrow w\} &&= \{s_1, \dots, s_n\}.
\end{aligned}$$

Język gramatyki G jest generowany przez następujące wyrażenie regularne:

$$(l_1 + \ldots + l_k)^*(s_1 + \ldots + s_n)(p_1 + \ldots + p_m)^*.$$

Z jednej strony każde słowo generowane przez to wyrażenie ma wyprowadzenie w G. Z drugiej strony, każde wyprowadzenie w G generuje słowo powyższej postaci.

Zatem język gramatyki nieciekawej jest regularny, a ponadto umiemy algorytmicznie obliczyć wyrażenie regularne (a więc i automat skończony) dla tego języka. Tym samym problem równości języków dwóch gramatyk nieciekawych redukuje się do problemu równości języków dwóch automatów skończonych. Jeden z możliwych algorytmów dla tego ostatniego problemu wykonuje minimalizację obydwu automatów i sprawdza, czy po minimalizacji są one izomorficzne.

Rozwiązanie zadania 4. Każde słowo nad $\{a, b\}$ jest konkatenacją bloków liter a i bloków liter b, naprzemiennie; bloki takie nazwijmy *a-blokami* i *b-blokami*. Zadane wyrażenie regularne

$$e \;=\; \underbrace{((b+aa)^* + \varepsilon)}_{(e1)}\underbrace{b(bb)^*}_{(e2)}\underbrace{(aa(b+aa)^* + \varepsilon)}_{(e3)}$$

opisuje zbiór wszystkich słów spełniających koniunkcję dwóch poniższych warunków:

($\forall a$) każdy a-blok ma długość parzystą,

($\exists b$) pewien b-blok ma długość nieparzystą.

Faktycznie, podwyrażenie (e2) wymusza warunek ($\exists b$), ponieważ każde słowo generowane przez (e1) kończy się na a albo jest puste, i symetrycznie dla (e3). Ponadto powyrażenia (e1) i (e3) zapewniają ($\forall a$). Z drugiej strony, każde słowo spełniające ($\forall a$) i ($\exists b$) jest generowane przez wyrażenie e: wystarczy wybrać dowolny b-blok długości nieparzystej i wygenerować go podwyrażeniem (e2), a wtedy odpowiedni prefiks i sufiks są generowane odpowiednio przez (e1) i (e3).

Zatem dopełnienie języka generowanego przez wyrażenie e opisane jest przez *alternatywę* poniższych warunków:

($\exists a$) pewien a-blok ma długość nieparzystą,

($\forall b$) każdy b-blok ma długoś parzystą,

czyli jest generowane przez wyrażenie:

$$((a+b)^*b+\varepsilon)a(aa)^*(b(a+b)^*+\varepsilon) \quad + \quad (a+bb)^*.$$