JFiZO

Michał Kucharczyk

March 2020

1 Zadanie nr 50

1.1 Treść

Czy język $L_3 = \{w \in \{0,1,2\}^* \ : \ \neg\exists x \in \{0,1,2\}^* \ w = xx\}$ jest bezkontekstowy?

1.2 Rozwiązanie

Tak. Skonstruuję niedeterministyczny automat skończony ze stosem T rozpoznający L_3.
Dodatkowo założę, że każde słowo na wejściu jest parzystej długości. Wszystkie słowa nieparzystej długości należą do języka, więc mając automat poprawnie rozpoznający słowa parzystej długości należące do języka i automat sprawdzający czy słowo jest nieparzystej długości (zadanie dla czytelnika jak taki skonstruować) łatwo można zbudować automat, który rozpoznaje L_3. Obserwacja:
w $\in L_3 \Leftrightarrow \exists x,y,x',y' \in \{0,1,2\}^* \ \exists a,b \in \{0,1,2\} \ w = xayx'by' \wedge a \neq b \wedge |x| = |x'| \wedge |y| = |y'|$
$\Leftrightarrow \exists x,y,x',y' \in \{0,1,2\}^* \ \exists a,b \in \{0,1,2\} \ w = xax'yby' \wedge a \neq b \wedge |x| = |x'| \wedge |y| = |y'|$ $(*)$
W języku naturalnym: jeśli pierwsza połowa w jest różna od drugiej to świadkiem tego jest pozycja i taka, że $w(i)$ ($:= i$-ta literka w, nazwijmy ją dla wygody a) jest różna od $w(\frac{|w|}{2} + i)$ (nazwijmy ją b). Dodatkowo, pomiędzy literkami a i b jest tyle samo literek, co w sumie literek przed a i po b. Szukaną pozycję i (o ile takowa istnieje) podpowie nam wiadomo kto (wierząc, że Takowy istnieje).
$T = \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$, gdzie:
$Q = \{`push_0`, `pop_0`, `push_1`, `pop_1`, `accept`, `fail`\} \times \{`null`, `0`, `1`, `2`\}$
$q_0 = \langle `push_0`, `null` \rangle$
$\Sigma = \Gamma = \{0,1,2\}$
$F = \{`accept`\} \times \{`0`, `1`, `2`\}$
Stan automatu T to krotka dwuelementowa. Wartość jej pierwszej współrzędnej będę nazywał skrótowo stanem, natomiast wartość drugiej "wczytaną literką". Definiując funkcję przejścia, mówiąc o zmianie stanu, będę miał na myśli zmianę pierwszej współrzędnej, drugą pozostawiając taką, jaka była (o ile wyraźnie nie zaznaczę, że jest inaczej). Słowna definicja δ:
Mówiąc o kroku mam na myśli sekwencję zmian stanów po wczytaniu 1 literki. Na początku, tj. będąc w stanie $`push_0`$ automat każdą wczytaną literkę wrzuca na stos, nie zmieniając stanu. Dopiero, gdy usłyszy głos, że wczytał literkę a (na pozycji i, świadka należenia do L_3) zmieni krotkę stanu z $\langle `push_0`, `null` \rangle$ na $\langle `pop_0`, `a` \rangle$ i nie wrzuci literki a na stos. Będąc w

stanie ‘pop_0‘ automat wyrzuca w każdym kroku jedną literkę, aż opróżni stos (jeśli przed tym momentem usłyszy drugą podpowiedź, zmienia stan na ‘$fail$‘). Wtedy zmienia stan na ‘$push_1$‘. Będąc w tym stanie znów po wczytaniu literki wrzuca ją na stos, aż do momentu, gdy po raz drugi usłyszy głos. Będzie to oznaczało, że wczytał literkę b. Jej również (tak jak a) nie wrzuci na stos, natomiast sprawdzi, czy wczytana literka (druga współrzędna stanu) jest równa ‘b‘. Jeśli tak, to automat przechodzi do stanu ‘$fail$‘, z którego już nigdy nie wyjdzie (pierwsza połowa słowa nie różni się w tym miejscu od drugiej). W przeciwnym przypadku automat przechodzi do stanu ‘pop_1‘. W tym stanie w pojedynczym kroku automat wyrzuca jedną literkę ze stosu. Jeśli w pewnym momencie stos się opróżni, to zmieniamy stan na ‘$accept$‘. Jeśli jesteśmy w stanie ‘$accept$‘ i wczytamy jeszcze jakąś literkę, to zmieniamy stan na ‘$fail$‘.

W tym rozwiązaniu, jak ognia, wystrzegałem się symbolicznego zapisania funkcji przejścia, aby nie zaciemnić rozwiązania i przekazać pewną intuicję, z której powinna wynikać poprawność działania automatu T.
”Dowód”poprawności:
Jeśli wczytywane w należy do L_3, to dostaniemy 2 podpowiedzi i zweryfikujemy, czy warunek (*) jest spełniony, czyli czy zasłyszane pozycje (i, j) spełniają $j = \frac{|w|}{2} + i$ (**) oraz czy odpowiadające im literki są różne. Jeśli natomiast słowo nie należy do alfabetu, to znaczy, że jest postaci $w = xx$, dla pewnego x, więc po usłyszaniu dowolnej pary pozycji (i, j) jeśli $j \neq |x| + i$ to trafimy do stanu ‘$fail$‘, w p.p. $w(i) = w(j)$, więc automat również znajdzie się w stanie ‘$fail$‘. Gdybyśmy natomiast usłyszeli mniej niż 2 podpowiedzi, być może nie znajdziemy się w stanie ‘$fail$‘, ale również nie trafimy do stanu (stanów) ‘$accept$‘, będącymi jedynymi,które należą do F, więc T słowa nie zaakceptuje.
Wyjaśnienie (**):
Niech $w = xaybz$, gdzie a, b są literkami na pozycjach usłyszanych w podpowiedziach. Nasz automat sprawdza czy $|x| + |z| = |y|$ ”dzieląc”w w ten sposób: $w = xax'yby'$. W stanie ‘$push_0$‘ tworzy x, w stanie ‘pop_0‘ x' (jeśli b należy do x' przechodzi do stanu ‘$fail$‘). Będąc w stanie ‘$push_1$‘ automat stwierdził równość $|x| = |x'|$, więc musi sprawdzić, czy $|y| = |y'|$, a więc czy literek na stosie przed wczytaniem literki b jest tyle, co po wczytaniu. Automat to robi.