Zadanie [2 punkty] Język $L \subseteq \{1,2,3\}^*$ składa się z tych słów, które są inne niż $11, 22, 33$, i gdzie każde trzyliterowe podsłowo abc spełnia jeden z poniższych warunków:

$$a < b < c \qquad a > b > c \qquad (a > b < c \quad \text{i} \quad c \geq a) \qquad (a < b > c \quad \text{i} \quad c \geq a).$$

Zbuduj minimalny automat deterministyczny rozpoznający ten język.

Rozwiązanie.

Naturalny automat, który nie jest minimalny. Zacznijmy od naiwnego automatu, a potem go zminimalizujmy. W naiwnym automacie jest stan nieakceptujący $\perp$, z którego nie da się wyjść (bo słowa nieakceptowanego nie da się przedłużyć do akceptowanego). Pozostałe stany są akceptujące i pamiętają sufiks długości dwa (z wyjątkiem dla słów długości zero lub jeden, gdzie pamiętane jest całe słowo). A więc stany to

$$\perp, \epsilon, 1, 2, 3, 12, 13, 21, 23, 31, 32\ .$$

W powyższej liście nie uwzględniliśmy sufiksów $11, 22, 33$, które utożsamiamy ze stanem $\perp$. Stan początkowy to ϵ. Funkcja przejścia dopisuje nową literę do stanu, ewentualnie usuwa najstarszą literę, gdy stan miał dwie litery. Jeśli dodanie litery powoduje wypadnięcie z języka, przechodzimy do stanu $\perp$. Dokładniej rzecz biorąc, mamy przejścia:

$$\begin{array}{rl}
\epsilon \xrightarrow{a} a & \qquad \text{dla } a \in \{1,2,3\} \\
a \xrightarrow{b} ab & \qquad \text{dla } a,b \in \{1,2,3\}, a \neq b \\
ab \xrightarrow{c} bc & \qquad \text{dla } a,b,c \in \{1,2,3\} \text{ gdzie } abc \text{ spełnia warunek zadania}
\end{array}$$

Utożsamianie stanów naturalnego automatu. Które stany tego automatu są równoważne? Równoważna jest para stanów 1 i 21, bo oba są akceptujące i przechodzą do tych samych stanów po pojedynczych literach, a więc mają dokładnie te same przyszłości:

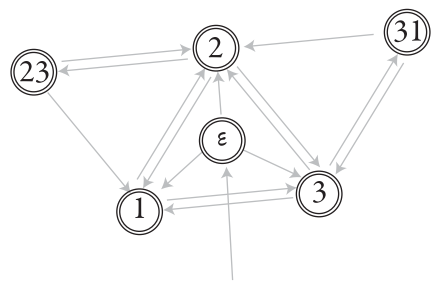
$$\begin{array}{ccc}
1 \xrightarrow{1} \perp & 1 \xrightarrow{2} 12 & 1 \xrightarrow{3} 13 \\
21 \xrightarrow{1} \perp & 21 \xrightarrow{2} 12 & 21 \xrightarrow{3} 13
\end{array}$$

Podobnie równoważna jest para stanów 3 i 13. Wreszcie równoważne są trzy stany 2, 12 i 32:

$$\begin{array}{ccc}
2 \xrightarrow{1} 21 & 2 \xrightarrow{2} \perp & 2 \xrightarrow{3} 23 \\
12 \xrightarrow{1} 21 & 12 \xrightarrow{2} \perp & 12 \xrightarrow{3} 23 \\
32 \xrightarrow{1} 21 & 32 \xrightarrow{2} \perp & 32 \xrightarrow{3} 23
\end{array}$$

Aby zminimalizować automat, zlepiamy stany $\{1, 21\}$ w stan 1, zlepiamy stany $\{2, 12, 32\}$ w stan 2 i zlepiamy stany $\{3, 13\}$ w stan 3. Po takim zlepieniu automat rozpoznaje ten sam język, bo zlepialiśmy stany o tych samych przyszłościach.

Rysunek.



Dla czytelności rysunku pominęliśmy stan $\perp$ i przejścia do niego prowadzące, oraz etykiety innych przejść. Etykieta przejścia jest taka, jak ostatnia litera stanu docelowego.

Nie może mieć mniej stanów. Pokażemy, że automat ten jest minimalny, ponieważ wszystkie stany można odróżnić przyszłościami. Okazuje się, że wystarczy rozważać przyszłości jednoliterowe. W poniższej tabelce, piszemy w wierszu q i kolumnie i symbol $\in$ jeśli przejście ze stanu q po literze i prowadzi do stanu akceptującego, w przeciwnym przypadku piszemy $\notin$.

stan	1	2	3
ϵ	$\in$	$\in$	$\in$
$\perp$	$\notin$	$\notin$	$\notin$
1	$\notin$	$\in$	$\in$
2	$\in$	$\notin$	$\in$
3	$\in$	$\in$	$\notin$
31	$\notin$	$\notin$	$\in$
23	$\notin$	$\in$	$\notin$

Zadanie [3 punkty] Kasa ma trzy operacje: “dodaj grosz”, “odejmij grosz”, “wyrównaj do złotówki”, oznaczone literami d, o, r. Język $L \subseteq \{d, o, r\}^*$ to ciągi operacji, które z pustej kasy wracają do pustej kasy, jak np.

$$dr \in L \qquad (ddo)^{34}r \in L \qquad o^{99}r \in L \qquad dro \notin L.$$

Kasa może być ujemna. Wyrównanie obcina grosze w kierunku zera, np. wyrównanie dla 1 zł 99 gr to 1 zł, a wyrówanie dla -3 zł 25 gr to -3 zł. Pokaż, że L jest bezkontekstowy, ale nie regularny.

Rozwiązanie.

Język L nie jest regularny. Tę część zadania robi się jak nieregularność języka $a^n b^n$. Gdbyby był regularny, to by spełniał założenia lematu o pompowaniu. Stosując lemat o pompowaniu dojdziemy do sprzeczności, a więc język nie jest regularny. Niech M stała z lematu o pompowaniu. Rozważmy słowo

$$d^M o^M \in L \ .$$

Skoro podsłowo d^M ma co najmniej M liter, można w nim pompować. Innymi słowy, istnieje podział

$$d^M = uvw \qquad u, v, w \in d^*, v \neq \epsilon$$

taki, że dla każdego k, słowo

$$uv^k wo^M$$

należy do języka L. Już dla $k = 2$ dochodzimy do sprzeczności, gdyż zawartość kasy po operacjach wv^2wo^M to $|v|$ groszy. A więc język L nie jest regularny.

Język L jest bezkontekstowy. Skonstruujemy automat ze stosem. Automat ma dwa symbole stosowe: $\oplus$ i $\ominus$. Zawartość stosu $\oplus^k$ oznacza, że w kasie jest k groszy, zawartość stosu $\ominus^k$ oznacza, że w kasie jest minus k groszy. Aby poradzić sobie z równaniem, automat pamięta w stanie ilość groszy ponad równą złotówkę, czyli stany automatu to liczby $0, \ldots, 99$. Stan początkowy to 0. Automat akceptuje, jeśli po przeczytaniu słowa wejściowego ma pusty stos. Oto przejścia automatu:

stary stan	czubek stosu	czytana litera	nowy stan	operacja stosowa
i	pusty lub $\oplus$	d	$(i+1) \bmod 100$	push $\oplus$
i	$\ominus$	d	$(i-1) \bmod 100$	pop
i	pusty lub $\ominus$	o	$(i+1) \bmod 100$	push $\ominus$
i	$\oplus$	o	$(i-1) \bmod 100$	pop
i	dowolny	r	0	i razy pop

Kto nie lubi wielokrotnych operacji stosowych w jednym przejściu, może dodać stany pomocnicze i zastąpić ostatnie przejście i epsilon-przejściami.

Zadanie [2 punkty] Dodajmy do kasy operację "podwój", oznaczoną literą p, która mnoży zawartość kasy przez dwa. Język $K \subseteq \{d, o, r, p\}^*$ jest zdefiniowany podobnie jak w poprzednim zadaniu.

$$dp^{10}o^{1024} \in L \qquad p^{17} \in L \qquad op^3d^8 \in L$$

Pokaż, że K nie jest bezkontekstowy.

Rozwiązanie.

Gdyby język K był bezkontekstowy, to bezkontekstowe też byłoby jego przecięcie z językiem regularnym dp^*o^*:

$$K \cap dp^*o^* \quad = \quad \{dp^m o^n : n = 2^m\} \ .$$

Do powyższego języka, nazwijmy go L, zastosujemy lemat o pompowaniu dla języków bezkontestowych i dojdziemy do sprzeczności. Niech M stała z lematu i rozważmy słowo

$$dp^M o^{2^M} \in L \ .$$

Lemat o pompowaniu stanowi, że istnieje podział

$$dp^M o^{2^M} = u_1 v_1 w v_2 u_2 \qquad v_1 v_2 \neq \epsilon \ , |v_1 w v_2| < M$$

który można pompować, czyli

$$u_1 v_1^k w v_2^k u_2 \in L \qquad \text{dla każdego } k.$$

Rozważmy pompowanie dla $k = 2$. Łatwo zauważyć, że litera d nie może być pompowana, gdyż inaczej wypadlibyśmy z języka dp^*o^*. Niech m' ilość liter p w słowie

$$w' = u_1 v_1^2 w v_2^2 u_2$$

i niech n' ilość liter o w tym słowie. Jeśli $n' = 2^M$, czyli nie przybyło liter o, to musi być $m' > M$, skoro pompowany kawałek jest niepusty. To daje sprzecznośc, gdyż $n' = 2^M < 2^{m'}$. Musi więc zachodzić $n' > 2^M$. Skoro jednak $|v_1 w v_2| < M$, to liczba n' jest mniejsza niż 2^{M+1}, a więc nie może być potęgą 2.