Lista 3. Star Height. Ta lista nie jest jeszcze dopracowana.

1. Gwiazdkową formą normalną języka regularnego L nazwiemy wyrażenie postaci $\Sigma_{i=1}^{n} R_i$, gdzie R_i jest wyrażeniemnie zawierającym operacji $+$. Pokaż, że każdy język regularny jest opisywany przez pewne wyrażenie regularne w gwiazdkowej formie normalnej.
2. Jaki Sh mają języki opisane przez poniższe wyrażenia regularne:
   (a) $(ab^*c)^*$,
   (b) $(ba + a^*ab)^*a^*$,
   odpowiedź uzasadnij.
3. W rozszerzonych wyrażeniach regularnych oprócz konkatenacji, sumy i gwiazdki Kleene'go można używać operacji dopełnienia i przecięcia (jak i również symbolu zbioru pustego $\emptyset$). Pokaż, że język opisany przez wyrażenie $(ababa)^*$ może być opisany przez wyrażenie rozszerzone bez użycia operacji gwiazdki.
4. Jaki Sh ma język opisany wyrażeniem $((a_1^*a_2^*a_3)^*(a_4^*a_5^*a_6)^*a_7)^*$. Odpowiedź uzasadnij.
5. Dwuspójny digraf G nazywamy *kulą* jeśli zawiera co najmniej jedna krawędź. *Loop complexity* grafu G definiujemy nastepująco:
   (a) $\mathsf{lc}(G) = 0$ jeśli G nie zawiera kuli jako podgrafu.
   (b) $\mathsf{lc}(G) = \max\{\mathsf{lc}(\mathcal{P}) \mid \mathcal{P} \text{ jest kulą w } G\}$ jeśli G nie jest dwuspójny.
   (c) $\mathsf{lc}(G) = 1 + \min\{\mathsf{lc}(G \setminus \{s\}) \mid s \text{ jest wierzchołkiem } G\}$ jeśli G jest dwuspójny.
   W rozdziale drugim książki Hopcrofta, Ullmana przedstawiono algorytm dokonujący translacji z DAS na wyrażenie regularne. Pochodzi on od Mcnaughtona-Yamady. Pokaż, że stosując algorytm Mcnaughtona-Yamady, Sh wyjściowego wyrażenia E jest nie mniejsze od loop complexity automatu wejściowego $\mathcal{A}$ ($\mathsf{lc}(\mathcal{A})$). W zadaniu tym zakładamy, że automat wejściowy jest *trim*, czyli że każdy jego stan jest osiągalny ze stanu początkowego i można z każdego stanu osiągnąc stan końcowy. Pokaż algorytm przekształcający zadane wyrażenie regularne E na automat A taki, że $\mathsf{lc}(\mathcal{A}) = Sh(E)$.
6. **(**)** Deterministyczny automat jest *bideterministyczny* jeśli ma jeden stan początkowy jeden stan końcowy oraz po odwróceniu strzałek przejść nadal pozostaje deterministyczny. Język regularny jest bideterministyczny jeśli jest rozpoznawany przez automat bideterministyczny. Pokaż, że problem wyznaczenia Sh jezyka bideterministycznego jest problemem NP-zupełny.
7. Pokaż przykład języka kóry nie jest bideterministyczny.
8. **(*)** Problem niepustości przęcięcia jest pytaniem czy istnieje niepuste słowo $w \in \bigcap_{i=1}^{n} R_i$, gdzie R_i to języki zadane przez wyrażenia regularne. Wiadomo, że problem ten jest PSPACE$-$zupełny. Pokaż, że jeśli wszystkie R_i, są wyrażeniami regularnymi bez wystąpień $+$ (nazwijmy takie wyrażenia *uboższymi*) oraz ich star height wynosi conajwyżej 2 to problem wciąż pozostaje PSPACE$-$trudny.
9. **(*)** Pokaż, że niepustość przecięcia języków opisanych na wejściu wyrażeniami uboższymi o star height co najwyżej 1 jest NP-zupełna.
10. Semi- rozszerzone wyrażenia regularne to takie w których dodatkowo można używać operacji przecięcia. Na wykładzie definiowaliśmy klasę języków opisanych przez wyrażenia regularne α_i takie, że $Sh(L(\alpha_i)) = i$. Czy $Sh(L(\alpha_i)) = i$ w sensie semi-rozszerzonym? Jakie jest Sh języka α_i w sensie semi-rozszerzonym?

11. Wyobraź sobie, że jest rok 2050 i pracujesz w największej korporacji programistycznej w googlaktyce. W Twojej kompetencji leży weryfikowanie poprawności programów i jego efektywności (załóżmy, że programy są napisane w jakimś języku imperatywnym). Zaproponuj prostą miarę automatycznego odrzucania programów skomplikowanych, które mogą mieć wysoką złożoność czasową - w 2050 roku nie ma komputera kwantowego ani nie masz pod ręką, klawiszem żadnego zaawansowanego narzędzia.

`http://www.literateprogramming.com/mccabe.pdf`

`http://en.wikipedia.org/wiki/Cyclomatic_complexity`