Zadanie 1. (6 punktów) Rozważmy język słów nad alfabetem $\{1, 2, 3\}$, w których podciąg z pozycji parzystych i podciąg z pozycji nieparzystych są oba niemalejące. Na przykład 121333 należy do języka, a 2111 nie. Podać minimalny automat deterministyczny dla tego języka.

Rozwiązanie.

Pierwszy pomysł na automat jest taki, że ma specjalne stany $\epsilon, 1, 2$ i 3 dla słów długości < 2, a dla słów długości co $\geqslant 2$ ma stany postaci (i, j), gdzie i to przedostatnia litera, a j to ostatnia litera. Ponadto, jest stan śmietnikowy $\perp$, dla słów które nie należą do języka.

Można jednak zaobserwować, że stan ϵ jest równoważny stanowi $(1, 1)$, a każdy stan $n \in \{1, 2, 3\}$ jest równoważny stanowi $(1, n)$. A więc wystarczy 10 stanów

$$\{1, 2, 3\} \times \{1, 2, 3\} \quad \cup \quad \{\perp\}.$$

Stanem początkowym jest $(1, 1)$. Wszystkie stany są akceptujące, oprócz $\perp$. Funkcja przejścia jest określona tak:

$$\delta((i, j), k) = \begin{cases} (j, k) & \text{jeśli } k \geqslant j \\ \perp & \text{w przeciwnym przypadku.} \end{cases}$$

Dla śmietnika oczywiście mamy $\delta(\perp, k) = \perp$ dla $k \in \{1, 2, 3\}$.

Aby pokazać, że automat jest minimalny, wskażemy dla dowolnych stanów $p \neq q$ odróżniającą przyszłość. Jeśli jednym ze stanów jest $\perp$, to przyszłością jest ϵ, bo $\perp$ jest jedynym stanem nieakceptującym. W przeciwnym przypadku, stany te są postaci $p = (i, j)$ oraz $q = (i', j')$. Bez straty ogólności zachodzi $i < i'$ lub $j < j'$. Jeśli $i < i'$, przyszłością odróżniającą jest i. Jeśli $j < j'$, przyszłością odróżniającą jest dwuliterowe słowo $3i$.

Zadanie 2. (6 punktów) Rozważmy alfabet $A = \{1, 2, 3, 4\}$. Dla $w \in A^*$, oznaczamy przez $sort(w)$ słowo otrzymane z w poprzez posortowanie liter. Na przykład $sort(321442123) = 112223344$. Wskazać język bezkontekstowy $L \subseteq A^*$ taki, że

$$\tilde{L} = \{w : sort(w) \in L\}$$

nie jest językiem bezkontekstowym.

Rozwiązanie. Rozważmy język

$$L = \{1^n 2^n 3^m 4^m : n, m \in \mathbb{N}\},$$

który jest generowany przez następującą gramatykę

$$S \to XY \qquad X \to 1X2|\epsilon \qquad Y \to 3Y4|\epsilon.$$

Pokażemy, że język $\tilde{L}$ nie jest bezkontekstowy. Gdyby był, to bezkontekstowe byłoby jego przecięcie z językiem regularnym

$$K = \tilde{L} \cap (1^* 3^* 2^* 4^*) = \{1^n 3^m 2^n 4^m : n, m \in \mathbb{N}\}.$$

Pokażemy poniżej, że język K nie jest bezkontekstowy, z czego wynika, że $\tilde{L}$ nie może być bezkontekstowy.

Zastosujmy do języka K lemat o pompowaniu dla języków bezkontekstowych. Niech M stała z lematu. Rozważmy słowo $w = 1^M 3^M 2^M 4^M$. Zgodnie z lematem, istnieje podział

$$w = w_1 v_1 u v_2 w_2 \qquad 0 < |v_1 v_2| \qquad |v_1 u v_2| \leqslant M$$

taki, że dla każdego k, słowo $w_1 v_1^k u v_2^k w_2$ też należy do języka K. Łatwo zauważyć, że v_1 nie może zawierać dwóch różnych liter, bo inaczej przy $k = 2$ byśmy dostali słowo spoza $1^* 3^* 2^* 4^* \supseteq K$. Podobnie v_2. Ponieważ $|v_1 u v_2| \leqslant M$, objęty pompowaniem fragment $v_1 u v_2$ należy do jednego z trzech zbiorów

$$1^* 3^* \qquad 3^* 2^* \qquad 2^* 4^*.$$

W każdym z tych przypadków dostaniemy, przy pompowaniu dla $k = 2$, słowo spoza K.

Zadanie 3. (6 punktów) Niedeterministyczną maszynę Turinga z jedną taśmą nazwiemy niewnikliwą, jeśli dla każdego słowa w, które maszyna akceptuje, pewien bieg akceptujący liczy mniej niż $|w|$ kroków. Czy rozstrzygalny jest następujący problem? Dany jest kod maszyny Turinga M. Pytanie: czy maszyna M jest niewnikliwa?

Rozwiązanie. Problem jest nierozstrzygalny.

Zadanie to można rozwiązać na dwa sposoby. Pierwszy sposób opiera się na obserwacji, że maszyna jest niewnikliwa wtedy i tylko wtedy gdy nie akceptuje żadnego słowa. Obserwacji tej dowodzi się poprzez analizę najkrótszego biegu akceptującego maszyny niewnikliwej. Jak już dokonaliśmy tej obserwacji, to problem z zadania okazuje się być dokładnie tym samym problemem co problem niepustości dla maszyn Turinga, o którym wiadomo, że jest nierozstrzygalny.

Opiszemy teraz drugi sposób, który nie wymaga powyższej obserwacji. Korzystamy jedynie z prostszej implikacji tej obserwacji, mianowicie z faktu, że maszyna, która nie akceptuje żadnych słów jest niewnikliwa.

Aby pokazać nierozstrzygalność problemu z zadania, zredukujemy do niego problem pustości dla maszyn Turinga. Rozważmy kod maszyny Turinga M. Aby dokonać redukcji, musimy na podstawie kodu maszyny M obliczyć kod maszyny M', w taki sposób, że maszyna M jest pusta wtedy i tylko wtedy gdy maszyna M' jest niewnikliwa. (Powtórzmy: jak ktoś dokonał wyżej opisanej obserwacji, to zrozumie, że wystarczy wziąć $M = M'$.) Maszyna M' działa tak. Najpierw przechodzi głowicą całe słowo do końca, wraca na początek, a potem zachowuje się tak samo jak maszyna M. Nietrudno jest obliczyć kod maszyny M' na podstawie kodu maszyny M, wystarczy dodać trzy stany i cztery przejścia. Maszyna M' w każdym biegu na słowie długości n wykonuje $2n+1$ kroków, więc jedyna sposób, żeby była niewnikliwa polega na tym, że nie akceptuje żadnego słowa.

Zadanie 4. (6 punktów *) Niech $\tilde{L}$ jak w zadaniu 2. Jeśli $L \subseteq \{1,2\}^*$ jest bezkontekstowy, to czy $\tilde{L}$ też?

Rozwiązanie. Rozważmy funkcję $\pi : \{a,b\}^* \to \mathbb{N}^2$, która słowu w przyporządkowuje parę $\pi(w) = (i,j)$ gdzie na pierwszej współrzędnej jest liczba liter 1, a na drugiej współrzędnej jest liczba liter 2. Taką funkcję nazywa się czasami obrazem Parikha słowa.

Przypomnijmy teraz definicję semiliniowego zbioru wektorów, w przypadku dwuwymiarowym. Podzbiór $X \subseteq \mathbb{N}^2$ nazwiemy liniowym, jeśli dla pewnych wektorów $(i_0,j_0),\dots,(i_n,j_n) \in \mathbb{N}^2$ zachodzi

$$X = \{(i_0,j_0) + x_1 \cdot (i_1,j_1) + \cdots + x_n \cdot (i_n,j_n) : x_1,\dots,x_n \in \mathbb{N}\}$$

Zbiór semiliniowy to skończona suma $X_1 \cup \cdots \cup X_n$ zbiorów liniowych.

W szczególnym przypadku alfabetu dwuliterowego, twierdzenie Parikha mówi, że dla każdego języka bezkontekstowego $L \subseteq \{1,2\}^*$, obraz $\pi(L) \subseteq \mathbb{N}^2$ jest zbiorem semiliniowym. W przypadku, gdy język L jest podzbiorem 1^*2^* można tego faktu dowieść też bezpośrednio na palcach.

Łatwo zobaczyć, że

$$\tilde{L} = \{w : \pi(w) \in \pi(L)\}.$$

Pokażemy teraz, że dla dowolnego zbioru semiliniowego X, język

$$L_X = \{w : \pi(w) \in X\}$$

jest bezkontekstowy. Skoro klasa języków bezkontekstowych jest zamknięta na sumę, oraz zachodzi równość

$$L_{X_1 \cup \cdots \cup X_n} = L_{X_1} \cup \cdots \cup L_{X_n},$$

to wystarczy pokazać, że L_X jest bezkontekstowy dla dowolnego zbioru liniowego X. Rozważmy więc dowolny zbiór liniowy X, opisany wzorem

$$X = \{(i_0,j_0) + x_1 \cdot (i_1,j_1) + \cdots + x_n \cdot (i_n,j_n) : x_1,\dots,x_n \in \mathbb{N}\}.$$

Opiszemy teraz automat ze stosem który rozpoznaje język L_X. Dla ułatwienia opisu, nasz automat będzie korzystał nie ze stosu, ale z $n+2$ liczników o nazwach $A_0,\dots,A_n,B$. Dla $m \in \{0,\dots,n\}$, licznik A_m będzie przechowywał liczbę z zakresu $\{0,\dots,i_m\}$. Licznik B będzie przechowywał liczbę całkowitą (być może ujemną). Liczniki takie łatwo zaimplementować w automacie stosowym: zawartość liczników $A_0,\dots,A_n$ trzymamy w stanie (bo mają ustaloną z góry pojemność), a licznik B kodujemy za pomocą stosu.

Początkowo, wszystkie liczniki są puste, czyli zawierają 0.

Gdy automat czyta literę 2, zmniejsza licznik B.

Gdy automat czyta literę 1, wykonuje następujące czynności. Niedeterministycznie zgaduje liczbę $m \in \{0,\dots,n\}$, taką, że licznik A_m nie jest pełen (licznik

A_m jest pełen jeśli zawiera swoją maksymalną wartość, czyli i_m). Zwiększa licznik A_m o jeden. Jeśli po tym zwiększeniu, licznik A_m jest pełen, to zwiększa licznik B o j_m. Ponadto, jeśli $m \neq 0$ i licznik A_m jest pełen (po zwiększeniu), to zeruje licznik A_m.

Automat akceptuje, jeśli na końcu licznik A_0 jest pełen, a liczniki $A_1, \ldots, A_n, B$ są puste.

(9 punktów) Wybierz 9 z poniższych pytań i odpowiedz na nie. Za prawidłowe odpowiedzi dajemy +1 punkt, za złe -1 punkt. Punkty policzymy za 9 najgorszych odpowiedzi, czyli nie opłaca się odpowiadać na więcej niż 9 pytań.

1. Rozważmy nową semantykę dla automatu skończonego: automat akceptuje, w nowym sensie, słowo $a_1 \cdots a_n$, jeśli dla każdego $i \in \{1, \dots, n\}$ akceptuje, w klasycznym sensie, słowo $a_i \cdots a_n \$ a_1 \cdots a_{i-1}$. Czy automaty z nową semantyką są równoważne klasycznym?

   Odpowiedź: Tak. 6/6 (poprawne/próby)
   Krótkie uzasadnienie: Dla słowa $a_1 \dots a_n$ wprowadzamy oznaczenie

   $$\phi(a_1 \cdots a_n) = \{a_i \cdots a_n \$ a_1 \cdots a_{i-1} : i \in \{1, \dots, n\}\}.$$

   Trzeba pokazać dwie implikacje.

   Pierwsza implikacja mówi, że nowe automaty nie są silniejsze od klasycznych. Czyli: jeśli $L \subseteq (A \cup \{\$\})^*$ jest regularny, to $K = \{w : \phi(w) \subseteq L\} \subseteq A^*$ też. Łatwiej jest pokazać, że regularne jest dopełnienie K, czyli zbiór słów w takich, że pewne $v \in \phi(w)$ nie należy do L. To pokazuje się tak samo jak zadanie, że $\mathit{cycle}(M) = \{vw : wv \in M\}$ jest regularny, o ile M jest regularny.

   Druga implikacja mówi, że nowe automaty nie są słabsze od klasycznych. Czyli: każdy język regularny $K \subseteq A^*$ jest postaci $K = \{w : \phi(w) \subseteq L\}$ dla pewnego regularnego języka $L \subseteq (A \cup \{\$\})^*$. Wystarczy wziąć za $L = \bigcup_{w \in L} \phi(w)$. Tutaj ważna jest obserwacja, że zbiory $\phi(w), \phi(v)$ są rozłączne dla różnych w, v.

2. Czy każdy język bezkontekstowy można opisać gramatyką z jednym nieterminalem?

   Odpowiedź: Nie. 14/14 (poprawne/próby)
   Krótkie uzasadnienie: Języka $a^*b^*c^*$ nie da się opisać gramatyką z jednym nieterminalem. Przypuśćmy, że jest taka gramatyka, z jednym nieterminalem X. Dla każdej litery $\sigma \in \{a, b, c\}$ musi istnieć w gramatyce reguła postaci $X \to \alpha_\sigma X \beta_\sigma$, gdzie $\alpha_\sigma, \beta_\sigma \in \{X, a, b, c\}^*$, taka, że σ występuje w α_σ lub β_σ. A więc muszą być dwie różne litery $\sigma, \tau \in \{a, b, c\}$ takie, że σ występuje w α_σ i τ występuje w α_τ, lub σ występuje w β_σ i τ występuje w β_τ. W obu przypadkach X wygeneruje słowo, którego podciągiem, niekoniecznie spójnym, jest $\sigma\tau\sigma$.

3. Czy istnieje wielomian $p(n)$, taki że dla każdych dwóch deterministycznych automatów skończonych o n stanach, najkrótsze słowo akceptowane przez pierwszy ale nie drugi ma długość co najwyżej $p(n)$?

   Odpowiedź: Tak. 2/5 (poprawne/próby)
   Krótkie uzasadnienie: Załóżmy, że stany tych automatów to Q_1 i Q_2, a zbiory akceptujące to $F_1 \subseteq Q_1$ i $F_2 \subseteq Q_2$. Takie najkrótsze słowo jest akceptowane przez deterministyczny automat produktowy o stanach $Q_1 \times Q_2$,

gdzie akceptujące stany to $Q_1 \times (Q_2 - F_2)$. Ponieważ automat produktowy ma n^2 stanów, akceptuje słowo długości co najwyżej n^2. A więc wielomian ten to $p(n) = n^2$.

4. Rozważmy model automatu deterministycznego, który może wykonać instrukcję "wróć do początku słowa w stanie q". Czy istnieje wielomian $p(n)$, taki że każdy niepusty automat tego rodzaju o n stanach akceptuje słowo długości co najwyżej $p(n)$?

   Odpowiedź: Nie. 6/6 (poprawne/próby)
   Krótkie uzasadnienie: Można napisać automat tego rodzaju, który ma $O(n)$ stanów i akceptuje dokładnie jedno słowo długości $(n+1) \cdot 2^n$ nad alfabetem $\{0, 1, \#\}$, mianowicie słowo

   $$bin(0)\#bin(1)\#bin(2)\# \cdots \#bin(2^n - 1)\#$$

   gdzie $bin(i)$ to binarny zapis liczby $i \in \{0, \ldots, 2^n - 1\}$, dopełniony wiodącymi 0 tak, żeby napis miał długość n.

5. Czy istnieje wielomian $p(n)$, taki że dopełnienie każdej gramatyki bezkontekstowej o n regułach jest albo puste albo zawiera słowo długości co najwyżej $p(n)$?

   Odpowiedź: Nie. 8/8 (poprawne/próby)
   Krótkie uzasadnienie: Rozważmy alfabet $\{a\}$. Napiszemy gramatykę rozmiwaru $O(n)$, która nie akceptuje żadnego słowa dłuższego niż 2^n, ale akceptuje wszystkie słowa długości co najwyżej 2^n. Gramatyka ma nieterminale $X_n, \ldots, X_0$ i reguły $X_{i+1} \to X_i X_i$ dla $i \in \{n, \ldots, 1\}$ oraz $X_0 \to a|\epsilon$. Startowy jest X_n.

6. Rozważmy wyrażenia regularne nad alfabetem jednoliterowym $\{a\}$. Czy istnieje wielomian $p(n)$, taki że jest co najwyżej $p(n)$ różnych języków $L \subseteq a^*$ generowanych przez wyrażenie rozmiaru n?

   Odpowiedź: Nie. 4/7 (poprawne/próby)
   Krótkie uzasadnienie: Dla dowolnego skończonego zbioru pierwszych $P = \{p_1, \ldots, p_k\}$ napiszmy wyrażenie regularne

   $$(a^{p_1})^* + \cdots + (a^{p_k})^*.$$

   Dla różnych zbiorów P wyjdą różne języki. Liczba zbiorów P takich, że wyrażenie dla P jest mniejszego rozmiaru niż n nie jest wielomianowa względem n. (Jeśli chcemy, żeby wyrażenie miało rozmiar dokładnie n, można wiele razy napisać gwiazdkę przy którymś wyrażeniu, np a^{**} zamiast a^*.)

7. Czy następujący problem jest rozstrzygalny? Dane dwie gramatyki bezkontekstowe, generujące języki L i K. Pytanie: czy niepusta jest różnica symetryczna $(L - K) \cup (K - L)$?

Odpowiedź: Nie. 8/9 (poprawne/próby)
Krótkie uzasadnienie: To jest to samo co równość języków bezkontekstowych L i K, a równość języków bezkontekstowych jest nierozstrzygalna. Nawet dla tych instancji, gdzie L generuje wszystkie słowa.

8. Czy następujący problem jest rozstrzygalny? Dany automat skończony. Pytanie: czy akceptacja nie zależy od ostatniej litery? (Inaczej mówiąc, czy dla każdego słowa w i każdych liter a i b, automat akceptuje wa wtedy i tylko wtedy gdy akceptuje wb.)

   Odpowiedź: Tak. 5/8 (poprawne/próby)
   Krótkie uzasadnienie: Niech L język obliczony przez ten automat. Musimy stwierdzić, czy języki La^{-1} i Lb^{-1} są równe dla każdych a, b. Automat dla ilorazu La^{-1} można obliczyć na podstawie automatu dla L, a równość języków regularnych jest rozstrzygalna.

9. Rozważmy alfabet jednoliterowy $\{a\}$. Dla n stanów losujemy automat niedeterministyczny w taki sposób, że wszystkie następujące wydarzenia są niezależne i mają prawdopodobieństwo $\frac{1}{2}$: stan p jest początkowy, stan p jest akceptujący, jest przejście po literze a ze stanu p do stanu q. Czy prawdą jest, że gdy $n \to \infty$, to prawdopodobieństwo otrzymania automatu rozpoznającego a^* dąży do 1?

   Odpowiedź: Tak. 4/5 (poprawne/próby)
   Krótkie uzasadnienie: Prawdopodobieństwo, że ustalony stan jest naraz początkowy, końcowy i ma samopętlę po literze a wynosi $\frac{1}{8}$. Wystarczy, że przynajmniej jeden stan w automacie ma tę własność, a prawdopodobieństwo tego dąży do 1.

10. Gramatykę nazwiemy półliniową, jeśli w każdej regule prawa strona zawiera co najwyżej jedno wystąpienie każdego nieterminala, ale mogą być różne nieterminale. Czy każdy język bezkontekstowy można opisać gramatyką półliniową?

    Odpowiedź: Tak. 9/11 (poprawne/próby)
    Krótkie uzasadnienie: Jeśli mamy na przykład regułę $X \to YabYX$ to wprowadzamy nowy nieterminal Y' i zamieniamy regułę na dwie: $X \to YabY'X$ oraz $Y' \to Y$.

11. Niech $L \subseteq A^*$ język bezkontekstowy. Czy bezkontekstowy jest również język słów różniących się o dokładnie 2 litery od pewnego słowa z L, czyli język

    $$\{w_0 a_1 w_1 a_2 w_3 : w_0 b_1 w_1 b_2 w_3 \in L \text{ dla pewnych liter } b_1 \neq a_1, b_2 \neq a_2\}.$$

    Odpowiedź: Tak. 11/11 (poprawne/próby)
    Krótkie uzasadnienie: Można tak przerobić automat ze stosem, żeby dokładnie dwa razy podmienił literę. W stanie pamięta ile razy już podmienił literę.

12. Niech $L \subseteq A^*$ język rozstrzygalny. Czy rozstrzygalny jest również język słów różniących się o dokładnie 2 litery od pewnego słowa z L?

    Odpowiedź: Tak. 7/8 (poprawne/próby)
    Krótkie uzasadnienie: Maszyna Turinga najpierw niedeterministycznie podmienia dwie litery na swoim wejściu.

13. Ustalmy dwie litery a, b. Czy istnieje algorytm wielomianowy dla następującego problemu. Dana gramatyka bezkontekstowa. Pytanie: czy gramatyka generuje słowo, które zawiera infiks ab?

    Odpowiedź: Tak. 7/8 (poprawne/próby)
    Krótkie uzasadnienie: To jest problem o niepuste przecięcie gramatyki z językiem regularnym A^*abA^*. Jeśli gramatyka ma n nieterminali, to gramatyka dla przecięcia z tym językiem regularnym ma $O(n)$ nieterminali.

14. Czy dla każdego nieskończonego języka bezkontekstowego L, zbiór długości $\{|w| : w \in L\} \subseteq \mathbb{N}$ zawiera nieskończony podciąg będący ciągiem arytmetycznym?

    Odpowiedź: Tak. 11/11 (poprawne/próby)
    Krótkie uzasadnienie: Z lematu o pompowaniu dla gramatyk bezkontekstowych, L zawiera słowo postaci $w_1v_1uv_2w_2$ takie, że dla każdego k, słowo $w_1v_1^kuv_2^kw_2$ też należy do L. Ciągiem arytmetycznym jest więc $\{|w_1uw_2| + k \cdot |v_1v_2| : k \in \mathbb{N}\}$. Ciąg ten jest nieskończony, bo lemat o pompowaniu dowodzi, że jedno ze słów v_1, v_2 jest niepuste.

15. Gramatyka G to
    $S \to PQ$
    $Q \to PR$
    $R \to PQ|\varepsilon$
    $P \to aUa|bUb|a|b$
    $U \to aUa|bUb|a|b|\varepsilon$
    Czy G generuje wszystkie słowa długości większej niż 3 nad $\{a, b\}$ (i być może niektóre krótsze słowa)?

    Odpowiedź: Nie. 2/10 (poprawne/próby)
    Krótkie uzasadnienie: U generuje palindromy. P generuje niepuste palindromy. Q generuje konkatenacje nieparzystej liczby niepustych palindromów. R generuje konkatenacje parzystej liczby niepustych palindromów. S generuje konkatenacje parzystej (i większej od zera)liczby niepustych palindromów.

    ababa nie jest konkatenacją parzystej liczby niepustych palindromów, bo nie mogą to być wszystko palindromy jednoliterowe, dwuliterowych w tym słowie nie ma, a jak się w podziale użyje palindromu trzyliterowego, to zostają tylko dwie litery, z których nie ma jak zrobić drugiego (i ostatniego) palindromu.