

**Dwieście dwadzieścia dwa zadania
z języków formalnych i złożoności obliczeniowej*
(w tym jedno czy dwa trudne)**

Jerzy Marcinkowski

luty 2025

*Jest to kolejna edycja zbioru zadań, stanowiącego podstawę ćwiczeń z przedmiotu *Języki formalne i złożoność obliczeniowa*, który prowadzę corocznie w Instytucie Informatyki Uniwersytetu Wrocławskiego.

Część Pierwsza Kursu

1 Deterministyczne Automaty Skończone

Zadanie 1. Rozważmy język $L = \{w0s : |s| = 9\}$, złożony z tych słów nad alfabetem $\{0, 1\}$ których dziesiąty symbol od końca to 0. Udowodnij, że DFA rozpoznający ten język ma co najmniej 1024 stany.

Zadanie 2. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający zbiór tych wszystkich słów nad alfabetem $\{a, b, c\}$, które wśród ostatnich trzech znaków mają po jednym wystąpieniu każdej z liter alfabetu?

Zadanie 3. Udowodnij, że język $L = \{a^n b^{2n} : n \in \mathbb{N}\}$ nie jest regularny.

Zadanie 4. (za 2 punkty) Dla danego języka $L \subseteq \Sigma^*$ przez L^* rozumiemy najmniejszy język spełniający następujące warunki:

- (i) $\varepsilon \in L^*$;
- (ii) $\forall x, y [x \in L^* \wedge y \in L] \Rightarrow xy \in L^*$.

Gdzie ε oznacza, jak zawsze, słowo puste.

Niech L będzie dowolnym podzbiorem $\{0, 1\}^*$. Udowodnij, że L^* jest językiem regularnym.

Zadanie 5. Udowodnij, że język L tych słów nad alfabetem $\{0, 1\}$, które są zapisem binarnym liczby pierwszej, nie jest regularny.

Definicja. Dla danego słowa w , nad pewnym ustalonym alfabetem, niech w^R oznacza "w czytane od końca", tzn. $\varepsilon^R = \varepsilon$ i $(aw)^R = w^R a$ jeśli a należy do alfabetu, zaś w jest dowolnym słowem.

Zadanie 6. Czy język $L = \{ww^R x : w, x \in \{0, 1\}^* \text{ i } w, x \neq \varepsilon\}$ jest regularny? Czy język $L = \{xwx : w, x \in \{0, 1\}^* \text{ i } x \neq \varepsilon\}$ jest regularny?

2 Twierdzenie o indeksie

Zadanie 7. (Twierdzenie o indeksie, za 2 punkty) Niech $L \subseteq \mathcal{A}^*$. Relację $\sim_L \subseteq \mathcal{A}^* \times \mathcal{A}^*$ definiujemy w następujący sposób: $w \sim_L w'$ w.t.w gdy $\forall v \in \mathcal{A}^* (wv \in L \Leftrightarrow w'v \in L)$. Udowodnij następujące twierdzenie o indeksie: L jest regularny wtedy i tylko wtedy gdy liczba klas abstrakcji relacji \sim_L jest skończona. Minimalna liczba stanów DFA rozpoznającego L jest wtedy równa liczbie tych klas abstrakcji.

Niech Σ będzie skończonym alfabetem i niech $L \subseteq \Sigma^*$. Jak pamiętamy, relacja \sim_L z Twierdzenia o indeksie zdefiniowana jest, na zbiorze Σ^* jako: $w \sim_L v$ wtedy i tylko wtedy gdy $\forall x \in \Sigma^* (wx \in L \Leftrightarrow vx \in L)$. Podobnie możemy zdefiniować relację równoważności \sim_L^{inf} . Mianowicie $w \sim_L^{inf} v$ zachodzi wtedy i tylko wtedy gdy $\forall x, y \in \Sigma^* (xwy \in L \Leftrightarrow xvy \in L)$.

Niech i_L (od słowa *indeks*) będzie równe $|\Sigma^* / \sim_L|$ (czyli i_L to liczba klas abstrakcji na jakie \sim_L dzieli Σ^*). Podobnie, niech $i_L^{inf} = |\Sigma^* / \sim_L^{inf}|$.

Kolejne trzy zadania dotyczą wzajemnych relacji między liczbami i_L i i_L^{inf} .

Zadanie 8. Udowodnij, że jeśli jedna z liczb i_L, i_L^{inf} jest skończona, to obie są skończone (z Twierdzenia o Indeksie wiemy, że ma to miejsce wtedy i tylko wtedy gdy L jest regularny). Dokładniej mówiąc:

- a. udowodnij, że $i_L \leq i_L^{inf}$;
- b. udowodnij, że $i_L^{inf} \leq i_L$.

Zadanie 9. (za 2 punkty).

W zadaniu tym należy pokazać, że szacowanie z punktu **b** poprzedniego zadania nie może być poprawione. Dokładniej mówiąc:

a. Udowodnij, że jeśli $\Sigma = \{a, b, c\}$ to dla każdego skończonego zbioru Q istnieje minimalny DFA A , o zbiorze stanów Q i funkcji przejścia δ , taki że dla każdej funkcji $f : Q \rightarrow Q$ istnieje słowo w dla którego dla każdego $q \in Q$ zachodzi: $\delta(q, w) = f(q)$. Przez automat minimalny rozumiemy tu taki, w którym każdy stan jest osiągalny ze stanu początkowego, i w którym dla każdych dwóch stanów q, q' istnieje słowo w takie że dokładnie jeden ze stanów $\delta(q, w)$, $\delta(q', w)$ jest akceptujący.

b. Korzystając z tezy punktu **a.** udowodnij, że dla każdej liczby naturalnej n istnieje język L taki, że $i_L \leq n$ zaś $n^n \leq i_L^{inf}$.

Zadanie 10. Pokaż, że jeśli $|\Sigma| = 1$ to $i_L^{inf} = i_L$.

Zadanie 11. Pokaż, że dla każdego n istnieje język $L_n \subseteq \{a, b, \#\}^*$, którego wszystkie słowa mają długość $2n$, i taki, że najmniejszy rozstrzygający go deterministyczny automat skończony ma przynajmniej 2^n stanów.

Zadanie 12. Pokaż, że stała z lematu o pompowaniu dla języka regularnego może być wykładniczo mniejsza od indeksu tego języka. Dokładniej mówiąc, pokaż, że istnieje wielomian p oraz ciąg języków $\{L_i\}_{i \in \mathbb{N}}$ taki, że dla każdego n najmniejszy automat rozstrzygający L_n ma przynajmniej 2^n stanów ale liczba $p(n)$ może być przyjęta jako stała z lematu o pompowaniu dla języka L_n . *Wskazówka.* Nie bez powodu najpierw jest poprzednie zadanie a teraz jest to.

3 Wyrażenia regularne

Zadanie 13. Skonstruuj automat skończony rozpoznający i wyrażenie regularne definiujące, nad alfabetem $\{a, b\}$, język słów, które nie zawierają wzorca *baba*.

Zadanie 14. Skonstruuj automat skończony rozpoznający i wyrażenie regularne definiujące, nad alfabetem $\{a, b, c, d\}$, język słów, które zawierają tyle samo symboli a co b , tyle samo symboli c co d i w których każdym prefiksie liczba symboli a różni się co najwyżej o jeden od liczby symboli b , zaś liczba symboli c różni się co najwyżej o jeden od liczby symboli d .

Zadanie 15. Dodanie do definicji wyrażeń regularnych pozwolenia na użycie symbolu \cap , oznaczającego przekrój języków nie umożliwia reprezentowania nowych zbiorów, wyrażenia jednak stają się krótsze. Udowodnij, że użycie \cap może wykładniczo skrócić wyrażenie.

Wskazówka: rozważyć język składający się z jednego słowa $(\dots((a_0 a_1)^2 a_2)^2 \dots)^2$.

Zadanie 16. Czy istnieje wyrażenie regularne ϕ , oznaczające jakiś niepusty język regularny, takie że $L_{a\phi} = L_{\phi b}$? Czy istnieje wyrażenie regularne ϕ , oznaczające jakiś niepusty język regularny, takie że $L_{a^*\phi} = L_{\phi b^*}$?

4 Zadania o deterministycznych wyrażeniach regularnych.

Deterministic regular expressions, znane również jako *unambiguous regular expressions* pojawiły się kiedyś, jak się wydaje niechcący, w definicji standardu XML

Definicja. Niech ϕ będzie wyrażeniem regularnym nad alfabetem \mathcal{A} , a w słowem nad tym alfabetem. Niech f będzie funkcją, której argumentami są wystąpienia liter alfabetu w słowie w (czyli "kolejne litery słowa w "), a wartościami są wystąpienia liter w wyrażeniu ϕ . Powiemy, że f jest **poprawnym mapowaniem** w na ϕ , jeśli zachodzi któryś z warunków:

1. ϕ jest słowem nad \mathcal{A} , $\phi = w$ i f jest identycznością lub $\phi = \varepsilon$ i w jest puste;
2. $\phi = \phi_1 + \phi_2$ i f jest poprawnym mapowaniem w na ϕ_1 lub f jest poprawnym mapowaniem w na ϕ_2 ;
3. $\phi = \phi_1\phi_2$, $w = w_1w_2$ i f ograniczona do w_1 jest poprawnym mapowaniem tego słowa na ϕ_1 , zaś f ograniczona do w_2 jest poprawnym mapowaniem tego słowa na ϕ_2 ;
4. $\phi = \psi^*$, $w = w_1w_2 \dots w_k$, dla jakiegoś $k \geq 0$ i dla każdego $1 \leq i \leq k$ funkcja f ograniczona do w_i jest poprawnym mapowaniem w_i na ψ .

Intuicja jest taka, że poprawne mapowanie słowa przyporządkowuje każdej jego literze, literę wyrażenia z której ta litera słowa "się wzięła". Wyrażenie ϕ jest **deterministycznym wyrażeniem regularnym**, jeśli dla każdego $w \in L_\phi$ istnieje dokładnie jedno poprawne mapowanie w na ϕ . Deterministyczne wyrażenie regularne pozwala odczytać, które litery w słowie biorą się z których liter w wyrażeniu, ale to odczytanie następuje dopiero, gdy znamy całe słowo. Inaczej jest dla deterministycznych on-line wyrażen regularnych. Wyrażenie regularne ϕ jest **deterministyczne on-line**, jeśli dla każdego słów ww_1 , $ww_2 \in L_\phi$ i każdej funkcji f_1, f_2 , będących poprawnymi mapowaniami słów (odpowiednio) ww_1 i ww_2 na ϕ , funkcje f_1 i f_2 zgadzają się na prefiksie w .

Zadanie 17. a. Które z poniższych wyrażen są deterministyczne, a które są deterministyczne on-line?

- i. $0^*10^* + 0^*$
- ii. $(0 + 1)^*1(0 + 1)$
- iii. $(0 + 1)(0 + 2)^* + (1 + 2)(0 + 1)^* + (0 + 2)(1 + 2)^*$

b. Znajdź deterministyczne wyrażenie regularne oznaczające język tych wszystkich słów nad alfabetem zerojedynekowym, które zawierają wzorzec 101.

Zadanie 18. Czy dla każdego języka regularnego istnieje deterministyczne on-line wyrażenie regularne, które go definiuje?

Zadanie 19. Znajdź deterministyczne on-line wyrażenie regularne oznaczające język tych wszystkich słów nad alfabetem zerojedynekowym, które zawierają jedną lub dwie jedyne.

5 Niedeterministyczne Automaty Skończone

Zadanie 20. Skonstruuj niedeterministyczny automat skończony rozpoznający język tych słów nad $\{0, 1\}^*$ które, jako liczba w systemie dwójkowym, dzielą się przez 5, przy czym liczba jest wczytywana

- a) począwszy od najbardziej znaczącego bitu,
- b) począwszy od najmniej znaczącego bitu.

Zadanie 21. Udowodnij, że jeśli dla pewnego języka L istnieje rozpoznający go $NDFA$, to istnieje również $NDFA$ rozpoznający język $L^R = \{w : w^R \in L\}$

Zadanie 22. Wiadomo, że L jest językiem regularnym. Pokaż, że w takim razie język $\{w : \exists n \in \mathbb{N} w^n \in L\}$ jest też językiem regularnym. Przez w^n rozumiemy tu słowo w skonkatenowane ze sobą n razy.

Zadanie 23. Załóżmy, że L jest pewnym językiem regularnym. Czy język $L/2 = \{w : \exists v vw \in L \wedge |v| = |w|\}$ jest regularny?

Zadanie 24. Załóżmy, że $L \subseteq \{0, 1\}^*$ jest regularny. Czy wynika z tego, że język

$$\sqrt{L} = \{w \in \{0, 1\}^* : \exists x \in \{0, 1\}^* \exists y \in L \quad wx = y \wedge |y| = |w|^2\}$$

jest regularny?

Zadanie 25. Minimalny DFA rozpoznający język L ma zawsze tyle samo stanów co minimalny DFA rozpoznający dopełnienie L . Stwierdzenie to przestaje być prawdziwe, jeśli rozważamy automaty niedeterministyczne. Udowodnij, że istnieje język L , który daje się rozpoznać za pomocą N DFA o mniej niż 20 stanach, ale którego dopełnienie nie daje się rozpoznać żadnym N DFA o mniej niż 200 stanach. *Wskazówka: wystarczy rozważyć alfabet jednoelementowy.*

Zadanie 26. Niech $L_k = \{0^n : k \text{ nie dzieli } n\}$. Dla języka regularnego L , niech $d(L)$ oznacza minimalną liczbę stanów automatu deterministycznego rozpoznającego L , zaś $n(L)$ niech oznacza minimalną liczbę stanów automatu niedeterministycznego rozpoznającego L . Podaj nieskończenie wiele liczb naturalnych k , dla których zachodzi $d(L_k) = n(L_k)$ i nieskończenie wiele k naturalnych, dla których ta równość nie zachodzi.

W kolejnych dwóch zadaniach niech $p \geq 5$ będzie pewną liczbą pierwszą, a L_p językiem tych słów nad $\{0, 1\}$ które czytane jako liczba w zapisie binarnym dają, jako resztę z dzielenia przez p , jedną z liczb $\{1, 2, \dots, (p-1)/2\}$, przy czym liczby czytamy „od prawej”, czyli od najmniej znaczącego bitu (to znaczy pierwszy znak słowa jest ostatnią cyfrą liczby).

Zadanie 27. Czy istnieje niedeterministyczny automat skończony o mniej niż $p+3$ stanach rozpoznający język L_p ?

Zadanie 28. Czy istnieje deterministyczny automat skończony o mniej niż $2p$ stanach rozpoznający język L_p ?

W kolejnych dwóch zadaniach niech M_n będzie językiem tych słów nad alfabetem $\{1, 2, \dots, n\}$ (gdzie n jest pewną liczbą parzystą) w których występują wszystkie litery alfabetu oprócz być może jednej. Przez $\overline{M_n}$ rozumiemy dopełnienie języka M_n do zbioru $\{1, 2, \dots, n\}^*$.

Zadanie 29. a. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający $\overline{M_n}$?

b. Jaką minimalną liczbę stanów musi mieć niedeterministyczny automat skończony rozpoznający $\overline{M_n}$?

Zadanie 30. Udowodnij, że każdy niedeterministyczny automat skończony rozpoznający język M_n musi mieć więcej niż $2^{\frac{n}{2}-1}$ stanów.

Wskazówka. Dla liczby naturalnej k , takiej, że $1 \leq k \leq n/2$ nazwijmy parę liczb $\{2k-1, 2k\}$ rodziną. Powiemy że słowo $x \in \{1, 2, \dots, n\}^$ nie rozdziela rodzin, jeśli zawsze wtedy, gdy jedna z liter z jakiejś rodziny występuje w słowie x , w słowie tym występuje również druga z tych liczb. Powiemy że słowo x jest rosnące, gdy każda jego kolejna litera jest liczbą większą niż poprzednia. Ile jest słów nie należących do M_n , które są rosnące i nie rozdzielają rodzin?*

Zadanie 31. Zadanie 15 ilustrowało fakt, że rozszerzenie definicji wyrażeń regularnych o operację przecięcia języków nie umożliwi wprowadzenia nowych języków, ale dla niektórych ciągów $\{L_i\}_{i \in \mathbb{N}}$ języków pozwoli wykładniczo skrócić wyrażenia.

Pokaż, że jest tak również gdy przyjmiemy $L_i = \{w \in \{1, \dots, i\}^* : \forall j \in \{1, \dots, i\} |w|_j \neq 0\}$.

Wskazówka. Nie bez powodu to zadanie jest w rozdziale o niedeterministycznych automatach skończonych.

6 Zadania o hipotezie Černego

Kolejne zadania mają związek z – otwartą od ponad pół wieku – hipotezą Černego. Mówi ona, że jeśli zbiór $\text{sync}(Q)$ jest niepusty, to zawiera on jakieś słowo o długości nie większej niż $(|Q| - 1)^2$ (znany jest automat, z niepustym $\text{sync}(Q)$, dla którego najkrótsze słowo w $\text{sync}(Q)$ ma długość dokładnie $(|Q| - 1)^2$).

Definicja. Dla danego deterministycznego automatu skończonego $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ i zbioru $S \subseteq Q$, przez $\text{sync}(S)$ oznaczmy zbiór $\{w \in \Sigma^* : \forall q, q' \in S \ \hat{\delta}(q, w) = \hat{\delta}(q', w)\}$. Zauważ, że definicja nie zależy od wyboru stanów q_0 i F a tylko od zbioru stanów Q , od alfabetu Σ i od funkcji przejścia δ .

Zadanie 32. Język $L \subseteq \Sigma^*$ nazywany jest regularnym ideałem jeśli jest regularny i jeśli dla każdego słowa $w \in L$ i każdego słów $v, v' \in \Sigma^*$ zachodzi $vvw' \in L$.

a. Czy dla każdego automatu \mathcal{A} i zbioru S zawartego w zbiorze stanów automatu \mathcal{A} język $\text{sync}(S)$ jest regularny?

b. Czy dla każdego automatu \mathcal{A} i zbioru S zawartego w zbiorze stanów automatu \mathcal{A} język $\text{sync}(S)$ jest regularnym ideałem?

c. Czy dla każdego automatu \mathcal{A} język $\text{sync}(Q)$ jest regularnym ideałem? (Q jest ponownie zbiorem stanów automatu \mathcal{A}).

Zadanie 33. a. Udowodnij, że jeśli S jest dwuelementowy i zbiór $\text{sync}(S)$ jest niepusty, to zawiera on jakieś słowo o długości nie większej niż $|Q|^2$.

b. Udowodnij, że jeśli zbiór $\text{sync}(Q)$ jest niepusty, to zawiera on jakieś słowo o długości nie większej niż $|Q|^3$. *Wskazówka: skorzystaj z a.*

Zadanie 34. Udowodnij, że dla każdego dostatecznie dużego n naturalnego istnieje automat $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$, gdzie $\Sigma = \{a, b\}$, $n = |Q|$, i dwuelementowy zbiór $S \subseteq Q$, takie że zbiór $\text{sync}(S)$ jest niepusty, ale nie zawiera słowa o długości mniejszej niż $n^2/4$.

W kolejnych zadaniach rozważamy Częściowe Deterministyczne Automaty Skończone (PDFA). PDFA różni się od DFA tym, że funkcja przejścia δ może być w nim funkcją częściową, to znaczy $\delta(q, a)$ może nie być określona dla niektórych par $\langle q, a \rangle$, gdzie $q \in Q$ i $a \in \Sigma$. W rezultacie, dla niektórych słów $w \in \Sigma^*$ i stanów $q \in Q$, wartość $\hat{\delta}(q, w)$ może być nieokreślona.

Dla danego PDFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ i zbioru $S \subseteq Q$, przez $\text{csync}(S)$ („zbiór słów ostrożnie synchronizujących S ”) oznaczmy zbiór takich słów $w \in \Sigma^*$ że dla każdego $q \in S$ wartość $\hat{\delta}(q, w)$ jest określona, oraz dla każdych dwóch stanów $q, q' \in S$ zachodzi $\hat{\delta}(q, w) = \hat{\delta}(q', w)$. Zauważ, że definicja nie zależy od wyboru stanów q_0 i F a tylko od zbioru stanów Q , od alfabetu Σ i od funkcji przejścia δ .

Zadanie 35. Załóżmy, że dla każdego dwuelementowego $S \subseteq Q$ zbiór $\text{csync}(S)$ jest niepusty. Czy wynika z tego, że $\text{csync}(Q)$ jest niepusty?

Zadanie 36. Niech $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ będzie PDFA.

a. Załóżmy że dla pewnego trzelementowego $S \subseteq Q$ zbiór $\text{csync}(S)$ jest niepusty. Pokaż, że w takim razie istnieje $w \in \text{csync}(S)$ o długości nie większej niż $2|Q|^3$.

b. Udowodnij, że jeśli zbiór $\text{csync}(Q)$ jest niepusty, to zawiera on jakieś słowo o długości nie większej niż $2^{|Q|}$.

Zadanie 37. (za 3 punkty - każda wersja za punkt) Udowodnij, że dla każdego (dostatecznie dużego) n istnieje PDFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$, taki że $|Q| = n$ i że...

Wersja M. ...istnieje trzelementowy $S \subseteq Q$ taki że zbiór $csync(S)$ jest niepusty ale nie zawiera słowa krótszego niż $n^3/10000$.

Wersja L. ... $csync(Q)$ jest niepusty ale nie zawiera słowa krótszego niż $p(n)$, gdzie p jest dowolnym, ustalonym wcześniej, wielomianem. Zakładamy, że $\Sigma = \{0, 1, 2\}$.

Wersja XL. ... $csync(Q)$ jest niepusty ale nie zawiera słowa krótszego niż $p(n)$, gdzie p jest dowolnym, ustalonym wcześniej, wielomianem. Zakładamy, że $\Sigma = \{0, 1\}$.

Wskazówka. Rozwiązując wersję M warto może pamiętać, że między każdym naturalnym k a $2k$ znajduje się liczba pierwsza. Rozwiązując wersje L i XL warto być może wiedzieć, że suma pierwszych n liczb pierwszych jest zawsze mniejsza niż $n^2 \log n$, zaś ich iloczyn zawsze jest większy od $2^{n \log n}$.

7 Relacje automatyczne

Zdefiniujmy funkcję $l : \{0, 1\}^* \rightarrow \mathbb{N}$ jako: $l(\varepsilon) = 0$, $l(0w) = 2l(w)$, $l(1w) = 2l(w) + 1$.

Dla liczby naturalnej k zdefiniujmy $\Sigma_k = \{0, 1\}^k$.

Dla liczb naturalnych $j \leq k$ zdefiniujmy funkcję $\Pi_k^j : \Sigma_k^* \rightarrow \{0, 1\}^*$ jako: $\Pi_k^j(\varepsilon) = \varepsilon$,

$\Pi_k^j(\langle a_1, a_2, \dots, a_j, \dots, a_k \rangle w) = a_j \Pi_k^j(w)$, gdzie $\langle a_1, a_2, \dots, a_j, \dots, a_k \rangle \in \Sigma_k$.

Relację $R \subseteq \mathbb{N}^k$ nazwiemy na tej liście zadań *automatyczną*, jeśli język L_R złożony z tych słów $w \in \Sigma_k^*$, dla których zachodzi $R(l(\Pi_k^1(w)), l(\Pi_k^2(w)), \dots, l(\Pi_k^k(w)))$, jest regularny.

Zadanie 38. Czy relacja dodawania jest automatyczna? Przez relację dodawania rozumiemy tu $\{\langle a, b, c \rangle \in \mathbb{N}^3 : a + b = c\}$.

Zadanie 39. Czy relacja mnożenia jest automatyczna? Przez relację mnożenia rozumiemy tu $\{\langle a, b, c \rangle \in \mathbb{N}^3 : ab = c\}$.

Zadanie 40. Udowodnij, że rzut relacji automatycznej jest relacją automatyczną. Innymi słowy, jeśli $R \subseteq \mathbb{N}^k$ jest relacją automatyczną, to również relacja $R' = \{r \in \mathbb{N}^{k-1} : \exists m \in \mathbb{N} \langle r, m \rangle \in R\}$ jest relacją automatyczną (dla uproszczenia możesz przyjąć, że $k = 2$).

8 Gramatyki bezkontekstowe i automaty ze stosem

Zadanie 41. Zbuduj automat ze stosem rozpoznający język *dobrze rozstawionych nawiasów dwóch rodzajów* generowany przez gramatykę:

$$S \rightarrow SS|(S)|[S]|\varepsilon$$

która ma jeden symbol nieterminalny S i cztery symbole terminalne $(,), [,]$.

Zadanie 42. Zbuduj gramatykę bezkontekstową generującą język:

$$L = \{w \in \{0, 1\}^* : |w|_0 = 2|w|_1 \wedge |w|_1 \text{ jest liczbą parzystą}\}.$$

Zadanie 43. Czy język $L = \{w \in \{0, 1\}^* : |w|_0 \leq |w|_1 \leq 2|w|_0\}$ jest bezkontekstowy?

Zadanie 44. Czy język $L = \{w \in \{0, 1\}^* : \exists n \in \mathbb{N} \ 2n|w|_0 \leq |w|_1 \leq (2n+1)|w|_0\}$ jest bezkontekstowy?

Zadanie 45. Pokaż, że $L \subseteq \{0\}^*$ jest bezkontekstowy wtedy i tylko wtedy gdy jest regularny.

Zadanie 46. Niech G będzie gramatyką generującą poprawnie zbudowane formuły rachunku zdań ze zmiennymi zdaniowymi p i q . Symbolami terminalnymi w G są $p, q, (,), \neg, \Rightarrow$, zaś produkcjami $S \rightarrow \neg S | (S \Rightarrow S) | p | q$

Znajdź gramatykę w postaci normalnej Chomsky'ego generującą ten sam język.

Zadanie 47. Czy język $L_3 = \{w \in \{0, 1, 2\}^* : \neg \exists x \in \{0, 1, 2\}^* w = xx\}$ jest bezkontekstowy?

Zadanie 48. Czy dopełnienie języka L_3 z poprzedniego zadania, język $L_4 = \{w \in \{0, 1, 2\}^* : \exists x \in \{0, 1, 2\}^* w = xx\}$ jest bezkontekstowy?

Wskazówka: (1) rozważ język $L_4 \cap L$ gdzie $L = L_0^* 10^* 10^* 10^* 1$.

(2) Skorzystaj z lematu o pompowaniu, pamiętaj że podział $w = sztyx$, którego istnienie postuluje lemat jest taki, że $|zty| \leq c$, gdzie c jest stałą z lematu.

Zadanie 49. Zbuduj *NDPDA* i gramatykę bezkontekstową G dla języka $\{0, 1\}^* - \{www : w \in \{0, 1\}^*\}$.

Zadanie 50. (Za 3 punkty, chyba bardzo trudne, bo nie umiem go rozwiązać) Czy istnieje gramatyka bezkontekstowa generująca zbiór tych słów nad alfabetem $\{0, 1\}$, które nie są postaci vww dla żadnych słów w, v , takich że $|v| = |w|$?

Zadanie 51. Czy język $L = \{0^n 1^{n^2} : n \in \mathbb{N}\}$ jest bezkontekstowy?

Zadanie 52. Czy zbiór takich słów nad alfabetem $\{0, 1\}$, które mają parzystą długość, i w których pierwszej połowie jest przynajmniej tyle samo jedynek, co w drugiej połowie, jest bezkontekstowy?

Zadanie 53. Wiadomo z jednego z poprzednich zadań, że jeśli język $L \subseteq \Sigma^*$ jest regularny, to również język $L/2 = \{w \in \Sigma^* : \exists v \in \Sigma^* |w| = |v| \wedge wv \in L\}$ jest regularny.

Pokaż, że podobna implikacja nie zachodzi dla języków bezkontekstowych. To znaczy istnieje taki CFL L , dla którego $L/2$ nie jest CFL.

9 Transducery

- Transducer Moore'a to krotka $\langle \Sigma, \Sigma_1, Q, q_0, \delta, \sigma \rangle$ gdzie $\langle \Sigma, Q, q_0, \emptyset, \delta \rangle$ jest DFA (z pustym zbiorem stanów akceptujących) i gdzie $\sigma : Q \rightarrow \Sigma_1^*$ dla pewnego alfabetu Σ_1 . Jeśli $T = \langle \Sigma, \Sigma_1, Q, q_0, \delta, \sigma \rangle$ jest transducerem Moore'a to $f_T : \Sigma^* \rightarrow \Sigma_1^*$ jest zdefiniowana jako $f_T(\varepsilon) = \varepsilon$ oraz $f_T(wa) = (f_T(w))\sigma(\delta(wa, q_0))$.

- Transducer Mealy'ego zdefiniowany jest analogicznie, z tą różnicą że $\sigma : Q \times \Sigma \rightarrow \Sigma_1^*$ oraz $f_T(wa) = (f_T(w))\sigma(\delta(w, q_0), a)$.

- Transducery T i T' są *równoważne* jeśli funkcje f_T i $f_{T'}$ są równe.

- Dla języków $A \subseteq \Sigma^*$ i $B \subseteq \Sigma_1^*$ definiujemy $A \leq_{reg} B$ jeśli istnieje transducer T (Moore'a lub Mealy'ego) taki że dla każdego $w \in \Sigma^*$ zachodzi $w \in A$ w.t.w. gdy $f_T(w) \in B$.

Zadanie 54. Pokaż że dla każdego transducera Moore'a istnieje równoważny transducer Mealy'ego. Pokaż że dla każdego transducera Mealy'ego istnieje równoważny transducer Moore'a.

Zadanie 55. Pokaż że jeśli $A \leq_{reg} B$ i B jest regularny to A też.

Zadanie 56. Pokaż że dla każdego n istnieje transducer Mealy'ego $T = \langle \Sigma, \Sigma_1, Q, q_0, \delta, \sigma \rangle$ taki że $|Q| = |\Sigma| = n$ i że każdy transducer Moore'a równoważny T ma przynajmniej n^2 stanów.

Zadanie 57. Niech $A \subseteq \{ (,), [,], \langle, \rangle \}^*$ będzie językiem poprawnie rozstawionych nawiasów trzech rodzajów zaś $B \subseteq \{ (,), [,] \}^*$ językiem poprawnie rozstawionych nawiasów dwóch rodzajów. Pokaż że $A \leq_{reg} B$. *Wskazówka:* każde słowo produkowane przez σ ma się składać z dwóch symboli.

10 Inne zadania o językach regularnych i bezkontekstowych

10.1 Separowalność języków

Dla rozłącznych języków A i B mówimy że język C *separuje* A od B jeśli $A \subseteq C$ i $C \cap B = \emptyset$.

Scenariusz zwykle jest taki, że A i B są skomplikowane (w jakimś sensie) i pytamy czy istnieje prosty język który je separuje. Zadania w tym rozdziale są wariacjami na temat tego scenariusza.

Zadanie 58. Czy istnieje stała $c > 0$, taka że dla każdego $n \in \mathbb{N}$ istnieją rozłączne języki A i B , z których każdy daje się rozstrzygać NFA o n stanach, takie że każdy język regularny C który separuje A od B ma indeks równy przynajmniej $c2^n$?

Zadanie 59. Niech $\Sigma = \{a_1, a_2, \dots, a_n\}$ będzie alfabetem. Przez L_i oznaczmy język tych słów nad Σ w których a_i występuje parzyście wiele razy.

(Mnie wygodnie jest tu myśleć o literkach z Σ jak o pstryczkach-elektryczkach. Mamy n pstryczków, każdy z nich ma swoją lampę. Pstrykniesz zgaszoną lampę to się zapali, pstrykniesz zapaloną to się zgasi. Słowa z L_i to ciągi pstryknięć po których i -ta lampa jest zgaszona.).

Niech $PZ_i = \bar{L}_i \cap \bigcap_{j < i} L_j$. Czyli zbiór tych słów nad Σ po wczytaniu których lampa i jest Pierwszą Zapaloną (stąd PZ).

Niech wreszcie $A = \bigcup_m PZ_{3m+1}$ zaś $B = \bigcup_m PZ_{3m+2}$, gdzie m przebiega te liczby naturalne dla których odpowiednie indeksy mają sens.

Czy istnieje język regularny C , o indeksie wielomianowym względem n , który separuje A od B ?

Zadanie 60. Niech $A = \{a^i b^j c^k : i \neq j \wedge i \neq k \wedge j \neq k\}$ i niech $B = \{a^i b^j c^k : i = j \vee j = k\} \cup L_{\Sigma^*(ba+ca+cb)\Sigma^*}$. Czy istnieje język bezkontekstowy C który separuje A od B ? Wskazówka: skorzystaj z następnego zadania

Zadanie 61. Udowodnij, że prawdziwy jest następujący wariant Lematu o Pompowaniu, który słusznie będzie nazwać Uproszczonym Lematem Ogdena:

Jeśli $L \subseteq \Sigma^*$ jest CFL, to istnieje stała k_L taka, że dla każdego słowa $w \in L$ i każdego $d \in \Sigma$, jeśli w słowie w jest przynajmniej k_L wystąpień d , to istnieje podział $sztyx$ słowa w , taki że zy zawiera przynajmniej jedno wystąpienie d , zty zawiera najwyżej k_L wystąpień d , i że dla każdego $m \in \mathbb{N}$ słowo $sz^m ty^m x$ należy do L .

10.2 Automaty z restartem

Przez **słaby automat skończony z restartem** będziemy w tym rozdziale rozumieć automat, podobny do DFA, ale mający zbiór stanów początkowych q_0, \dots, q_{k-1} , oraz (oprócz normalnych dla DFA instrukcji) instrukcje typu „jeśli w stanie q widzisz a to restart”. Automat napotkawszy taką instrukcję uruchomi się ponownie (wczytując słowo wejściowe od początku) w stanie początkowym $q_{i \bmod k}$ gdzie i jest liczbą dotychczasowych restartów (wliczając aktualny). Pierwsze uruchomienie automatu następuje w stanie q_0 .

Przez **słaby automat ze stosem z restartem** będziemy rozumieć analogicznie zdefiniowany deterministyczny automat ze stosem (przy każdym restarcie stos zostaje opróżniony).

Zauważ, że automat restartując traci całą wiedzę jaką uzyskał w trakcie swojego działania, oprócz wiedzy o liczbie dotychczasowych restartów. Zauważmy też, że jeśli liczba restartów przekroczy k to automat się zapętli i na pewno nie zaakceptuje słowa wejściowego. Umawiamy się, że takie słowo jest odrzucane.

Przez **mocny automat skończony z restartem** będziemy rozumieli automat podobny do słabego automatu skończonego z restartem, którego funkcja przejścia ma dodatkowy argument, mówiący czy aktualnie wczytywany symbol jest ostatnim symbolem słowa (dzięki czemu koniec słowa automatu nie zaskoczy – zawsze będzie wiedział, że nadeszła ostatnia chwila, żeby się zrestartować).

Zadanie 62. Pokaż, że istnieje słaby automat ze stosem z restartem rozstrzygający język spoza klasy CFL.

Zadanie 63. Pokaż, że każdy język rozpoznawany przez jakiś mocny automat skończony z restartem jest regularny.

Zadanie 64. Pokaż, że dla każdego $n \in \mathbb{N}$ istnieje język L_n dający się rozstrzygać niedeterministycznym automatem skończonym o n stanach, ale wymagający słabego automatu z restartem o wykładniczej względem n liczbie stanów. *Wskazówka: nie będzie dla nikogo zaskoczeniem że kandydatem na L_n jest $\{w1v \in \{0,1\}^* : |v| = n-1\}$.*

Zadanie 65. Czy język L_n ze wskazówki do Zadania 3 daje się rozstrzygać mocnym automatem skończonym z restartem o liczbie stanów wielomianowej względem n ?

Zadanie 66. Pokaż, że dla każdego n istnieje język S_n dający się rozstrzygać słabym automatem z restartem o n stanach, ale wymagający DFA o wykładniczej względem n liczbie stanów.

Uwaga: Autorzy zadań nie umieli znaleźć odpowiedzi na pytanie, czy każdy język dający się rozstrzygać NFA o n stanach daje się również rozstrzygać mocnym automatem skończonym z restartem o wielomianowej względem n liczbie stanów.

10.3 Splecenie i podobne pojęcia

Zadanie 67. (za 2 punkty) *Splecenie* definiujemy w tym zadaniu jako najmniejszą relację ternarną na słowach nad pewnym ustalonym alfabetem \mathcal{A} spełniające warunki:

- spleceniem słowa pustego ze słowem pustym jest słowo puste;
- jeśli w jest spleceniem słowa s ze słowem t , to jest również spleceniem słowa t ze słowem s ;
- jeśli $v = at$, $a \in \mathcal{A}$ i w jest spleceniem słowa s ze słowem t to aw jest spleceniem słowa s ze słowem v .

Dla danych dwóch języków $L_1, L_2 \subseteq \mathcal{A}^*$ zdefiniujemy ich splecenie jako zbiór wszystkich w , które są spleceniami pewnego $s \in L_1$ z pewnym $t \in L_2$.

Czy splecenie dwóch języków regularnych zawsze jest językiem regularnym?

Czy splecenie dwóch języków bezkontekstowych zawsze jest językiem bezkontekstowym?

Niech \mathcal{A} będzie skończonym alfabetem i niech $L \subseteq \mathcal{A}^*$. Przez $\text{Lustro}(L)$ będziemy w kolejnych trzech zadaniach (i ani chwili dłużej) rozumieli język $\{wv^R \in \mathcal{A}^* : wv \in L\}$

Zadanie 68. Pokaż, że jeśli L jest regularny, to $\text{Lustro}(L)$ również jest regularny.

Zadanie 69. Pokaż, że deterministyczny automat skończony rozpoznający język $\text{Lustro}(L)$ może potrzebować liczby stanów wykładniczo większej niż deterministyczny automat skończony rozpoznający język L .

Zadanie 70. Czy teza Zadania 68. pozostanie prawdziwa, jeśli oba wystąpienia słowa „regularny” zmienimy w nim na „bezkontekstowy”?

Przez *język rodzinkowy* będziemy przez chwilę (to znaczy w kolejnych trzech zadaniach i ani chwili dłużej) rozumieć język będący podzbiorem $L_{a^*ba^*}$.

Dla danego języka regularnego L napis $i(L)$ będzie oznaczał na tej liście indeks języka L , czyli minimalną liczbę stanów deterministycznego automatu rozpoznającego L .

Zadanie 71. Czy istnieje język rodzinkowy L taki, że L^* jest bezkontekstowy ale nie jest regularny?

Zadanie 72. Dla ustalonego n naturalnego niech L_n będzie językiem składającym się ze wszystkich słów postaci $a^i b a^j$, gdzie $0 \leq i, j \leq 2n$ oraz $|i - j| \leq 1$. Udowodnij, że $i(L_n^*)$ szacuje się (z dokładnością do stałej multiplikatywnej) przez n^2 .

Wskazówka: Warto rozważyć słowa postaci $a^k (b a^{2n})^l$, dla odpowiednich liczb k i l .

Zadanie 73. Udowodnij, że dla każdego naturalnego n istnieje język rodzinkowy L_n , taki że $i(L_n L_n) \geq c 2^i(L_n)$, gdzie c jest pewną stałą niezależną od n . Jeśli nie potrafisz pokazać takiego wykładniczego dolnego ograniczenia na wzrost $i(L_n L_n)$, to dostaniesz punkty również za inne ograniczenie, jeśli nie będzie mniejsze niż $c i(L_n)^3$.

10.4 Jeszcze inne zadania

Zadanie 74. Na wykładzie udowodniliśmy, że rozszerzenie definicji automatu skończonego o możliwość poruszania się po słowie wejściowym w obie strony nie zmieni klasy rozpoznawanych języków. Czy podobnie jest w przypadku automatów ze stosem? Mówiąc dokładniej, rozważamy automaty, których relacja przejścia zawiera się w

$$(Q \times T \times U) \times (Q \times U^* \times \{L, R\})$$

gdzie Q jest skończonym zbiorem stanów („w jakim stanie jestem”), T jest zbiorem symboli taśmowych („co widzę na taśmie”), U zbiorem symboli stosowych (z analogicznymi jak dla zwykłych automatów ze stosem założeniami dotyczącymi symbolu dna stosu), zaś L i R należy rozumieć jako instrukcje „idź w lewo” i „idź w prawo”. Automaty takie są uruchamiane dla słów, których koniec i początek zaznaczone są dodatkowym symbolem taśmowym, nie występującym wewnątrz słowa. Czy każdy język jaki można rozpoznać przy pomocy takiego automatu jest bezkontekstowy? *Wskazówka:* wystarczy rozważać takie deterministyczne automaty akceptujące po osiągnięciu jakiegoś końcowego stanu akceptującego.

O języku $A \subseteq \Sigma^*$ powiemy, w kolejnych trzech zadaniach, że jest *konfluentny*, jeśli:

$$\forall w, v \in \Sigma^* \exists x \in \Sigma^* \forall y \in \Sigma^* (wxy \in A \Leftrightarrow vxy \in A).$$

O języku $A \subseteq \Sigma^*$ powiemy, że jest *jednostajnie konfluentny*, jeśli istnieje takie $c \in \mathbb{N}$, że:

$$\forall w, v \in \Sigma^* \exists x \in \Sigma^* (|x| \leq c \wedge \forall y \in \Sigma^* (wxy \in A \Leftrightarrow vxy \in A)).$$

Zadanie 75. Czy każdy język regularny jest konfluentny? Czy każdy język konfluentny jest regularny?

Zadanie 76. Pokaż, że jeśli język regularny jest konfluentny, to jest jednostajnie konfluentny.

Zadanie 77. Pokaż, że istnieje konfluentny język bezkontekstowy który nie jest jednostajnie konfluentny.

W kolejnych trzech zadaniach przyjmujemy że $\Sigma = \{a, b, c, d\}$. Niech $P \subseteq \Sigma^* \times \Sigma^*$ będzie określona – również w kolejnych trzech zadaniach – jako najmniejsza symetryczna relacja taka, że:

- dla każdego $w \in \Sigma^*$ zachodzi $P(w, \varepsilon)$;
- dla każdego $a \in \Sigma$ i każdych $w, v \in \Sigma^*$ jeśli $P(w, v)$ to $P(aw, av)$;

Przez $L_{p/q}$ gdzie $L \subseteq \Sigma^*$, oznaczać będziemy język:

$$\{w \in \Sigma^* : \exists v \ v \in L \wedge P(w, v) \wedge |w|/|v| = p/q\}$$

Zadanie 78 (łatwe). Niech $L \subseteq \Sigma^*$ będzie regularny. Czy wynika z tego, że:

- a. język $L_{3/2}$ jest regularny?
- b. język $\bigcup_{i=1}^{\infty} L_{1/i}$ jest regularny?

Zadanie 79 (trudne, za 2 punkty). Pokaż że istnieje takie $c > 0$, że dla każdego $m \in \mathbb{N}$ istnieją $n > m$ i $L \subseteq \Sigma^*$ takie, że minimalny DFA rozstrzygający L ma n stanów, zaś każdy DFA rozstrzygający $L_{1/2}$ ma przynajmniej cn^2 stanów.

Komentarz: kto uważał na poprzednich ćwiczeniach wie, że jeśli L jest regularny to $L_{1/2}$ jest również regularny. Konstrukcja deterministycznego automatu rozstrzygającego $L_{1/2}$ którą znam wymaga wykładniczej, względem n , liczby stanów i nie wiem czy jest optymalna. W zadaniu masz pokazać że optymalna konstrukcja wymaga przynajmniej kwadratowej liczby stanów.

Zadanie 80. Niech $L \subseteq \Sigma^*$ będzie CFL. Czy wynika z tego, że $L_{3/4}$ jest CFL?

Jak każdy pamięta, w Zadaniu 1 należało udowodnić, że język $\mathcal{L} = \{vawc : v, w \in \{a, b\}^*, |w| = 9\}$ daje się rozstrzygać niedeterministycznym automatem skończonym o 11 stanach, ale nie daje się rozstrzygać deterministycznym automatem o mniej niż 1024 stanach. To zadanie, wraz z jego rozwiązaniem, warto mieć w głowie przy okazji kolejnych trzech zadań.

Zadanie 81. Jak każdy świetnie pamięta, język $L_{v \neq w} = \{vw : v, w \in \{0, 1\}^*, |v| = |w|, v \neq w\}$ jest bezkontekstowy. Daje się go rozstrzygać niedeterministycznym automatem z jednym licznikiem (to znaczy automatem ze stosem, którego alfabet symboli stosowych jest jednoelementowy, jeśli nie liczyć symbolu dna stosu). Pokaż, że języka $L_{v \neq w}$ nie da się rozstrzygać deterministycznym automatem z jednym licznikiem. *Uwaga: To zadanie ma dwa rozwiązania, nie wiadomo które jest bardziej “kanoniczne”.*

W kolejnych dwóch zadaniach rozważamy Odrobinę Niedeterministyczne Automaty Skończone (ONFA). Taki automat definiujemy sobie podobnie jak DFA albo NFA, jako krotkę $\langle \Sigma, Q, Q_0, F, \delta \rangle$, gdzie Σ, Q, F i δ oznaczają to co zawsze ($\delta : Q \times \Sigma \rightarrow Q$ jest funkcją przejścia) zaś $Q_0 \subseteq Q$ jest zbiorem dopuszczalnych stanów początkowych. Słowo w jest akceptowane przez taki automat gdy istnieje $q_0 \in Q_0$ takie, że $\hat{\delta}(q_0, w) \in F$.

Zadanie 82. Pokaż, że istnieje ciąg języków $\{L_i\}_{i \in \mathbb{N}}$ i stałe $c, k > 0$ takie, że każdy L_i daje się rozstrzygać przy pomocy ONFA o nie więcej niż ki stanach, ale żaden L_i nie daje się rozstrzygać przy pomocy DFA o mniej niż 2^{ci} stanach.

Zadanie 83. Pokaż, że istnieje ciąg języków $\{L_i\}_{i \in \mathbb{N}}$ i stałe $c, k > 0$ takie, że każdy L_i daje się rozstrzygać przy pomocy NFA o nie więcej niż ki stanach, ale żaden L_i nie daje się rozstrzygać przy pomocy ONFA o mniej niż 2^{ci} stanach.

W kolejnych dwóch zadaniach umawiamy się, że skończony alfabet Σ nie zawiera symbolu $\#$. Dla danego języka $L \subseteq \Sigma^*$ definiujemy $L_{\#} = \{w\#v : vw \in L\}$.

Zadanie 84. Niech L będzie językiem dobrze rozstawionych nawiasów. Pokaż że język $L_\#$ jest bezkontekstowy.

Zadanie 85. (chyba trudne; za 2 punkty). Pokaż że, dla każdego języka bezkontekstowego L , język $L_\#$ jest również bezkontekstowy.

Część Druga Kursu

11 Zbiory i funkcje rekurencyjne

Zadanie 86. Rozszerz definicję zbioru rekurencyjnego tak, aby można było rozważać rekurencyjne zbiory par liczb naturalnych i udowodnij, że jeśli zbiór $\mathcal{A} \subseteq \mathbb{N}^2$ jest rekurencyjny, to zbiór $\{n : \exists m [n, m] \in \mathcal{A}\}$, czyli rzut \mathcal{A} na pierwszą oś, jest zbiorem rekurencyjnie przeliczalnym.

Zadanie 87. Pokaż, że każdy zbiór rekurencyjnie przeliczalny jest rzutem pewnego zbioru rekurencyjnego, to znaczy jeśli B jest r.e., to istnieje taki rekurencyjny $A \subseteq \mathbb{N}^2$, że $B = \{n : \exists m [n, m] \in A\}$.

Zadanie 88. Powtórz, podany na wykładzie, dowód nierozstrzygalności problemu stopu, to znaczy faktu, że zbiór $K = \{n : \phi_n(n) \in \mathbb{N}\}$ nie jest rekurencyjny.

Zadanie 89. Pokaż, że $\{n : |Dom(\phi_n)| \geq 7\}$ jest rekurencyjnie przeliczalny.

Zadanie 90. Niech A, B, C, D będą zbiorami rekurencyjnie przeliczalnymi, takimi że każda liczba naturalna należy do dokładnie dwóch z nich. Udowodnij, że w takim razie wszystkie cztery zbiory są rekurencyjne.

Zadanie 91. Udowodnij, że jeśli ϕ jest niemalejącą całkowitą funkcją rekurencyjną, to zbiór $\phi(\mathbb{N})$ jej wartości jest rekurencyjny. Czy pozostaje to prawdą bez założenia o całkowitości ϕ ?

Zadanie 92. Udowodnij, że każdy niepusty zbiór rekurencyjnie przeliczalny jest postaci $\phi(\mathbb{N})$ dla pewnej całkowitej funkcji rekurencyjnej ϕ .

Zadanie 93. Udowodnij, że każdy nieskończony zbiór rekurencyjnie przeliczalny jest postaci $\phi(\mathbb{N})$ dla pewnej całkowitej, różnowartościowej funkcji rekurencyjnej ϕ .

Zadanie 94. Udowodnij, że zbiór $\{n : Dom(\phi_n) = \mathbb{N}\}$ nie jest rekurencyjnie przeliczalny.

Zadanie 95. (Długie, więc za 2 punkty) Załóżmy, że f jest funkcją rekurencyjną, całkowitą. Które z poniższych implikacji są prawdziwe?

- jeśli A jest rekurencyjny, to $f(A)$ też;
- jeśli A jest rekurencyjny, to $f^{-1}(A)$ też;
- jeśli A jest r.e., to $f(A)$ też;
- jeśli A jest r.e., to $f^{-1}(A)$ też.

Co zmieni się, jeśli założymy, że f jest funkcją częściową?

Zadanie 96. Nie korzystając z tw. Rice'a udowodnij, że zbiór $B = \{n : Dom(\phi_n) \text{ i } \mathbb{N} - Dom(\phi_n) \text{ są nieskończone}\}$ nie jest rekurencyjny.

Zadanie 97. Udowodnij, że zbiór $B = \{n : Dom(\phi_n) \text{ i } \mathbb{N} - Dom(\phi_n) \text{ są nieskończone}\}$ nie jest nawet rekurencyjnie przeliczalny.

Zadanie 98. Udowodnij, że zbiór numerów tych programów, które zatrzymują się dla wszystkich argumentów oprócz co najwyżej skończonej liczby, nie jest rekurencyjnie przeliczalny.

Zadanie 99. Niech $A, B \subseteq \mathbb{N}$. Mówimy, że $A \leq_{rek} B$ jeśli istnieje całkowita funkcja rekurencyjna f (zwana redukcją), taka że $f(x) \in B$ wtedy i tylko wtedy gdy $x \in A$. Pokaż, że dla każdych dwóch zbiorów $A, B \subseteq \mathbb{N}$ istnieje ich najmniejsze ograniczenie górne w sensie \leq_{rek} , to znaczy taki zbiór C , że:

- i) $A \leq_{rek} C$ i $B \leq_{rek} C$,
- ii) jeśli D jest taki, że $A \leq_{rek} D$ i $B \leq_{rek} D$ to $C \leq_{rek} D$.

Zadanie 100. Czy $K \leq_{rek} \overline{K}$? Czy $\overline{K} \leq_{rek} K$?

Zadanie 101. Niech T będzie zbiorem tych par liczb $\langle n, m \rangle$ dla których ϕ_n i ϕ_m to ta sama funkcja częściowa.

- i) Pokaż, że T nie jest rekurencyjnie przeliczalny.
- ii) Czy dopełnienie zbioru T jest rekurencyjnie przeliczalne?

Zadanie 102 (Hierarchia arytmetyczna). Niech $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ będzie pewną ustaloną obliczalną bijekcją. Oznaczmy klasę zbiorów rekurencyjnych jako Σ_0 . Dla danego Σ_i niech $\Pi_i = \{A \subseteq \mathbb{N} : \mathbb{N} \setminus A \in \Sigma_i\}$, zaś $A \in \Sigma_{i+1}$, jeśli istnieje $B \in \Pi_i$, takie że $A = \{n \in \mathbb{N} : \exists m f(n, m) \in B\}$.

Niech L będzie zbiorem numerów tych niepustych funkcji rekurencyjnych których dziedzina jest skończona. Jakie jest najmniejsze i dla którego zachodzi $L \in \Sigma_i$?

Zadanie 103. Niech A, B będą podzbiorami zbioru liczb naturalnych. Załóżmy, że f jest redukcją świadczącą o tym, że $A \leq_{rek} B$. Załóżmy, że f jest „na” (tzn. jej obrazem jest cały zbiór liczb naturalnych). Pokaż, że w takim razie zachodzi również $B \leq_{rek} A$.

Kolejne dwa zadania będą miały związek z twierdzeniem Rice’a. Mówi ono, jak pamiętamy, że zbiór $A \subseteq \mathbb{N}$ nie może być rekurencyjny jeśli:

(*) jest ekstensjonalny, niepusty i nie zawiera numerów funkcji pustej.

W tekście poniżej litery f i g będą oznaczać funkcje rekurencyjne (być może częściowe). Napis $f \in A$ (lub $g \in A$) będziemy czytać jako „numery funkcji f (odp. g) należą do A ”.

Zadanie 104. Załóżmy, że $A \subseteq \mathbb{N}$ spełnia warunek (*) i jest r.e. Niech $f \in A$. Pokaż, że w takim razie istnieje skończony podzbiór $g \subseteq f$ taki, że $g \in A$.

Zadanie 105. Załóżmy, że $A \subseteq \mathbb{N}$ spełnia warunek (*) i jest r.e. Niech $g \in A$ będzie jakąś funkcją o skończonej dziedzinie i niech $g \subseteq f$. Pokaż, że wtedy również $f \in A$.

Zadanie 106. Podzbiór zbioru liczb naturalnych nazywamy co-r.e. jeśli jego dopełnienie jest rekurencyjnie przeliczalne. Odcinkami początkowymi zbioru liczb naturalnych nazywamy zbiory postaci $\{1, 2, \dots, n\}$ dla $n \in \mathbb{N}$. Oznaczmy przez B zbiór numerów tych programów, których dziedziny są odcinkami początkowymi zbioru liczb naturalnych.

a. Czy zbiór B jest co-r.e.?

b. Udowodnij, że istnieje zbiór trójek liczb naturalnych A , który jest co-r.e. i którego rzut na pierwszą oś jest zbiorem B . *Uwaga. Wymaga to oczywiście milczącego rozszerzenia definicji zbiorów co-r.e. na zbiory trójek liczb.*

Zadanie 107. Niech f będzie pewną całkowitą funkcją rekurencyjną. O każdym z następujących warunków rozstrzygnij, czy implikuje on rekurencyjność zbioru $f(\mathbb{N})$, to znaczy obrazu zbioru wszystkich liczb naturalnych przez funkcję f .

- a. Istnieje skończony podzbiór A zbioru liczb naturalnych, taki że jeśli $f(i) > f(i+1)$ to $i+1 \in A$.
- b. Istnieje skończony podzbiór A zbioru liczb naturalnych, taki że jeśli $f(i) > f(i+1)$ to $f(i+1) \in A$.

Zadanie 108. Niech f będzie całkowitą funkcją rekurencyjną o takiej własności, że dla każdej liczby naturalnej m istnieje n takie że $f(n) = m$. Czy w takim razie funkcja $g(m) = \min\{n \in \mathbb{N} : f(n) = m\}$ też jest całkowita rekurencyjna?

A jeśli nie zakładamy całkowitości funkcji f ?

Zadanie 109. Niech $\mathcal{D} \subseteq \mathcal{P}(\mathbb{N})$. Udowodnij, że następujące warunki są równoważne:

- \mathcal{D} jest przeliczalny;
- istnieje $B \subseteq \mathbb{N}$ taki, że dla każdego $A \in \mathcal{D}$ zachodzi $A \leq_{rek} B$.

Wskazówka: Dla ustalonej całkowitej funkcji rekurencyjnej f i zbioru $B \subseteq \mathbb{N}$, ile może być takich zbiorów $A \subseteq \mathbb{N}$, że f jest redukcją, świadczącą o tym, że $A \leq_{rek} B$?

Zadanie 110. Oznaczmy przez Tot zbiór $\{n \in \mathbb{N} : Dom(M_n) = \mathbb{N}\}$, zaś przez $Nemp$ zbiór $\{n \in \mathbb{N} : Dom(M_n) \neq \emptyset\}$.

- (a) Czy prawdą jest, że $Nemp \leq_{rek} Tot$?
- (b) Czy prawdą jest, że $Tot \leq_{rek} Nemp$?

Zadanie 111. Czy każda częściowa funkcja rekurencyjna jest podzbiorem jakiejś całkowitej funkcji rekurencyjnej?

Zadanie 112. Czy każdy nieskończony podzbiór \mathbb{N} zawiera jako swój podzbiór jakiś nieskończony zbiór rekurencyjnie przeliczalny? *Wskazówka: Inteligentna diagonalizacja.*

Zadanie 113. Czy istnieje zbiór rekurencyjnie przeliczalny, o nieskończonym dopełnieniu (do \mathbb{N}), przecinający się niepusto z każdym nieskończonym zbiorem rekurencyjnie przeliczalnym? *Wskazówka. W moim rozwiązaniu, występuje zbiór $\{[n, m] \in \mathbb{N}^2 : m > 2n\}$ par leżących w pewnej tabelce powyżej czegoś w rodzaju przekątnej.*

Ostatnie dwa zadania w tej sekcji dotyczą *zbiorów produktywnych* (nie ja wymyśliłem taką głupią nazwę). Zbiór $A \subseteq \mathbb{N}$ jest produktywny, jeśli istnieje częściowa funkcja rekurencyjna f taka że dla każdego $n \in \mathbb{N}$ jeśli $Dom(\phi_n) \subseteq A$ to $f(n)$ zwraca wynik, i $f(n) \in A \setminus Dom(\phi_n)$ (czyli, mówiąc po ludzku, dla każdego rekurencyjnie przeliczalnego podzbioru A funkcja f pokazuje liczbę która jest w A ale nie w tym podzbiorze).

Zadanie 114. Pokaż że istnieje zbiór jednocześnie produktywny i co-r.e.

Zadanie 115. Pokaż że zbiór wszystkich numerów programów wyliczających funkcje rekurencyjne całkowite jest produktywny.

12 Maszyny Turinga

Rozwiązując zadania z tego rozdziału należy dość dokładnie podać ideę konstrukcji, ale nie wymaga się wypisywania listy instrukcji konstruowanej maszyny.

Zadanie 116. Udowodnij, że zastąpienie w definicji maszyny Turinga taśmy nieskończoną płaszczyzną nie zmienia klasy funkcji obliczalnych.

Zadanie 117. Skonstruuj maszynę Turinga rozpoznającą język $A = \{ww^R : w \in \{0,1\}^*\}$

Zadanie 118. *Skanująca maszyna Turinga* będzie dana przez piątkę $\langle \Sigma, Q, q_0, q_F, \delta \rangle$, gdzie Σ jest skończonym alfabetem taśmowym, Q skończonym zbiorem stanów, $q_0, q_F \in Q$ to odpowiednio stany początkowy i końcowy, zaś $\delta : (Q \setminus \{q_F\}) \times \Sigma \rightarrow Q \times (\Sigma \setminus \{B\})$ jest funkcją przejścia. Maszyna działa tak, że na początku głowica ustawiona jest w stanie q_0 na pierwszym symbolu słowa wejściowego. Gdy w stanie q głowica widzi symbol $a \in \Sigma$, to przechodzi do stanu q' , zamiast a wpisuje a' , gdzie $\delta(q, a) = (q', a')$. Następnie jest przesuwana o jedną komórkę w prawo, chyba że a było blankiem, wtedy wiadomo że przeskanowano cały dotychczas używany fragment taśmy i głowica jest przesuwana (w aktualnym stanie) ponownie nad pierwszy symbol na taśmie, skąd ponownie wędruje w prawo itd. Obliczenie kończy się, gdy głowica znajduje się w stanie q_F .

Czy problem ustalenia, dla danej skanującej maszyny Turinga M i słowa wejściowego w , czy M uruchomiona na w się zatrzyma, jest rozstrzygalny?

Zadanie 119. Tak samo jak w poprzednim zadaniu, tylko odpowiedni fragment brzmi: "Maszyna działa tak, że na początku głowica ustawiona jest w stanie q_0 na pierwszym symbolu słowa wejściowego. Gdy w stanie q głowica widzi symbol $a \in \Sigma$, to przechodzi do stanu q' , zamiast a wpisuje a' , gdzie $\delta(q, a) = (q', a')$. Następnie jest przesuwana o jedną komórkę w prawo, chyba że a było blankiem, wtedy wiadomo że przeskanowano cały dotychczas używany fragment taśmy i głowica zawraca, to znaczy po wykonaniu każdej kolejnej instrukcji przesuwana jest o jedną komórkę w lewo, aż w końcu ponownie znajdzie się nad pierwszym symbolem na taśmie. Wtedy ponownie zawraca w prawo itd."

Każdy wie, że Maszyna Turinga umie policzyć wszystko co pomyśli głowa. To w takim razie pokaż, że umie obliczyć coś zupełnie prostego, mianowicie funkcję identycznościową:

Zadanie 120. Przedstaw (pokrótce acz czytelnie) konstrukcję Maszyny Turinga, z alfabetem taśmowym zawierającym jedynie symbole $0, 1, \alpha, \omega$ oraz blank B , która otrzyma jako wejście słowo $\alpha w \omega$, gdzie $w \in \{0,1\}^*$ (na prawo od ω są już blanki) i ma się zatrzymać po doprowadzeniu do tego, że na taśmie będzie napisane słowo $\alpha v \omega w$, dla pewnego $v \in \{0,1\}^*$ (a potem blanki). Nie wolno jej przy tym nadpisywać symboli α ani ω , ani pisać nowych symboli α i ω (to znaczy instrukcje pozwalają jedynie na zastąpienie każdego z symboli $0,1,B$ przez 0 albo 1 oraz na zastępowanie α przez α i ω przez ω).

Uwaga: Autorzy nie umieją rozwiązać analogicznego zadania, w którym wynikiem działania maszyny miałyby być, zamiast słowa $\alpha v \omega w$, słowo $\alpha w \omega w$.

13 Nierozstrzygalność. Kanoniczne zadania.

Zadanie 121 (Maszyna Minsky'ego). (za 2 punkty).

a. Zauważ, że problem stopu dla maszyn podobnych do automatu ze stosem, lecz posiadających dwa stosy, jest nierozstrzygalny. Dokładniej mówiąc, instancją problemu jest teraz lista instrukcji dla automatu o dwóch stosach, ale bez taśmy wejściowej. W jednym kroku obliczenia automat, w zależności od tego co widzi na stosach, modyfikuje stan i stosy. Pytamy o to, czy automat uruchomiony w stanie q_0 i przy dwóch pustych stosach, kiedykolwiek się zatrzyma.

b. Wywnioskuj z **a**, że analogiczny problem pozostaje nierozstrzygalny, jeżeli dwa stosy zastąpimy czterema licznikami (tzn. stosami o jednym symbolu stosowym, nie licząc symbolu dna stosu).

c. Wywnioskuj z **b**, że analogiczny problem pozostaje nierozstrzygalny, jeżeli cztery liczniki zastąpimy dwoma (taki automat, z dwoma licznikami, nazywa się Maszyną Minsky'ego).

Zadanie 122. Dla gramatyki bezkontekstowej G niech L_G oznacza generowany przez nią język. Skorzystaj z nierozstrzygalności problemu odpowiedniości Posta aby pokazać, że zbiór tych par gramatyk G, H dla których zachodzi $L_G \cap L_H \neq \emptyset$ nie jest rekurencyjny. Czy jest on rekurencyjnie przeliczalny?

Zadanie 123. Udowodnij, że nie istnieje algorytm rozstrzygający, dla danej gramatyki bezkontekstowej G i alfabetu A , czy $A^* = L(G)$

Zadanie 124. Czy istnieje algorytm rozstrzygający, dla danych dwóch gramatyk bezkontekstowych G i H , czy $L(G) = L(H)$?

Zadanie 125. Udowodnij nierozstrzygalność problemu sprawdzenia dla danego procesu Thuego Π i słowa w czy zbiór $A_w = \{v : w \xrightarrow{\Pi}^* v\}$ jest skończony.

Wskazówka (nieobowiązkowa, jak wszystkie wskazówki): Rozważ maszyny Turinga z dodanym gdzieś na taśmie licznikiem, który jest zwiększany o jeden przy każdym ruchu wykonywanym przez maszynę. Naśladuj dowód nierozstrzygalności problemu słów.

Zadanie 126. Rozpatrzmy skończony zbiór par słów P i binarną relację \rightarrow_P na słowach zdefiniowaną jak następuje: $w \rightarrow_P v$ wtedy i tylko wtedy, gdy istnieje para $\langle a, b \rangle \in P$ taka, że $w = ax$ i $v = xb$ gdzie x jest pewnym słowem. Niech \xrightarrow_P^* będzie przechodnim domknięciem \rightarrow_P (to znaczy najmniejszą relacją przechodnią zawierającą \rightarrow_P).

Czy problem: dane P, x, y , czy $x \xrightarrow_P^* y$? jest rozstrzygalny?

Zadanie 127. (trudne, za 2 punkty) Funkcję $f : \mathbb{N} \rightarrow \mathbb{N}$ nazywamy funkcją Conway'a jeśli istnieją liczby naturalne $p, a_1, b_1, a_2, b_2, \dots, a_p, b_p$ takie, że dla każdego n jeśli $n = k \bmod p$ to $f(n) = na_k/b_k$. Pokaż, że nie ma algorytmu, który dla danych $p, a_1, b_1, a_2, b_2, \dots, a_p, b_p$ odpowiedziałby, czy dla definiowanej przez te współczynniki funkcji Conway'a istnieje m takie, że $f^m(2) = 1$ gdzie f^m oznacza funkcję f złożoną m razy ze sobą.

Zadanie 128. Dla danej liczby naturalnej n przez $kcomp(n)$ oznaczamy liczbę znaków najkrótszego programu w MUJP który (nie wczytując żadnych danych) wypisze na wyjściu n . Czy funkcja $kcomp$ jest rekurencyjna?

Wskazówka: Dowód jest oczywiście nie wprost. Trzeba się zastanowić jaka jest najmniejsza liczba naturalna, której się nie da zdefiniować przy pomocy mniej niż dwudziestu słów.

Uwaga: wybrane w sformułowaniu zadania oznaczenie $kcomp$ pochodzi oczywiście od słów „Kolmogorov complexity”.

13.1 Kafelkowanie

Zadanie 129. (za 3 punkty) Udowodnij nierozstrzygalność następującego problemu: dany jest skończony zbiór kolorów C , zawierający co najmniej kolory: *czerwony* i *biały*, oraz zbiór $N \subseteq C^4$ czwórek kolorów, uznanych za *nieestetyczne*. Mamy nieskończenie wiele kwadratowych kafelków każdego koloru o boku długości 1. Czy istnieje kwadrat (o całkowitych wymiarach i boku nie mniejszym niż 2), który można wypełnić kafelkami w taki sposób by w lewym dolnym i w lewym górnym narożniku znalazł się czerwony kafelek, pozostałe kafelki dolnego i górnego brzegu były białe, oraz by w całym kwadracie nie pojawiła się nieestetyczna sekwencja kafelków, tj. cztery sąsiadujące kafelki:

c_1	c_2
c_3	c_4

takie że $(c_1, c_2, c_3, c_4) \in N$.

Zadanie 130. (za 2 punkty). Instancją problemu Kolorowania Wszystkimi Kolorami, który rozważamy w tym zadaniu, jest skończony zbiór \mathcal{K} kolorów oraz zbiór $\mathcal{N} \subseteq \mathcal{K}^4$ nieestetycznych czwórek.

Dla pewnej liczby naturalnej n funkcja $kolor : \{1, 2, \dots, n\}^2 \rightarrow \mathcal{K}$ (czyli kolorowanie szachownicy $n \times n$) jest rozwiązaniem problemu Kolorowania Wszystkimi Kolorami jeśli zachodzą dwa warunki:

- kolorowanie nie prowadzi do pojawienia się nieestetycznej czwórki, to znaczy nie ma takich $1 \leq i, j < n$, że: $\langle kolor[i, j], kolor[i, j + 1], kolor[i + 1, j], kolor[i + 1, j + 1] \rangle \in \mathcal{N}$;
- funkcja $kolor$ jest „na”, czyli w kolorowaniu użyte są wszystkie dostępne kolory.

Pokaż że problem, czy dana instancja problemu Kolorowania Wszystkimi Kolorami ma rozwiązanie, jest nierozstrzygalny.

Uwaga: najbardziej by nam się podobało rozwiązanie opierające się na rozwiązaniu Zadania 127 i skupiające się na pokonaniu trudności biorących się z różnic między tymi zadaniami.

13.2 Zadania o dziesiątym problemie Hilberta

Na wykładzie mówiliśmy o nierozstrzygalności **dziesiątego problemu Hilberta** (nazwijmy go H10). Problem ten polega na tym, aby dla danego układu równań diofantycznych, to znaczy równań między wielomianami wielu zmiennych o współczynnikach całkowitych, odpowiedzieć, czy układ ten ma rozwiązanie w liczbach naturalnych.

Zadanie 131. Niech H10prim będzie problemem ustalenia, dla danego układu równań diofantycznych, czy układ ten ma rozwiązanie w liczbach całkowitych.

a. Pokaż, że $H10_{\text{prim}} \leq_{\text{rek}} H10$.

b. Pokaż, że $H10 \leq_{\text{rek}} H10_{\text{prim}}$.

Zadanie 132. Niech H10bis będzie problemem H10 w którym ograniczamy się jedynie do równań diofantycznych, w którym każdy wielomian jest stopnia co najwyżej dwa. Czy H10bis pozostaje nierozstrzygalny?

Wskazówka (do tego zadania i poprzedniego): W jednym z zadań możesz zechcieć odwołać się do faktu, że każda liczba naturalna daje się przedstawić jako suma czterech kwadratów liczb naturalnych.

Zadanie 133. Udowodnij, że problem z dziesiątego problemu Hilberta (H10), to znaczy problem czy dany układ równań diofantycznych (czyli równań między wielomianami wielu zmiennych o współczynnikach całkowitych) ma rozwiązanie w liczbach całkowitych, pozostaje nierozstrzygalny jeśli, zamiast o rozwiązanie w liczbach całkowitych, będziemy pytać o rozwiązanie w liczbach całkowitych nieparzystych. Wolno oczywiście skorzystać z nierozstrzygalności H10.

Zadanie 134. Czy X Problem Hilberta, to znaczy problem istnienia rozwiązania, w liczbach naturalnych, dla danego jednego (równania) diofantycznego pozostaje nierozstrzygalny gdy ograniczymy się do instancji w których wielomiany po obu stronach tego równania mają stopień 4 (to znaczy, dla każdego jednomianu w tych wielomianach, łączny stopień wszystkich zmiennych w tym jednomianie jest nie większy od 4).

14 Nierozstrzygalność. Różne inne zadania.

Zadanie 135. Dla danych funkcji $f, g, h : \{0, 1, \dots, p - 1\} \rightarrow \{0, 1, \dots, p - 1\}$ i danego nieskończonego ciągu liczb naturalnych (a_1, a_2, a_3, \dots) , niech $F_{f,g,h}(a_1, a_2, \dots)$ będzie ciągiem liczb naturalnych, którego i -ty element jest równy $f(a_{i-1}) +_p g(a_i) +_p h(a_{i+1})$, gdzie $+_p$ oznacza dodawanie modulo p (przyjmujemy, że $a_0 = 0$). Udowodnij, że problem:

Dane funkcje f, g i h oraz skończone ciągi (b_1, b_2, \dots, b_k) i (c_1, c_2, \dots, c_k) . Czy istnieje liczba iteracji n taka, że $F_{f,g,h}^n(b_1, b_2, \dots, b_k, 0, 0, 0, \dots) = (c_1, c_2, \dots, c_k, 0, 0, 0, \dots)$? jest nierozstrzygalny.

Zadanie 136. Jak każdy pamięta, deterministyczny automat ze stosem, to urządzenie zadane przez skończony zbiór instrukcji w formacie: *jeśli widzisz na taśmie wejściowej a , jesteś w stanie q , a z czubka stosu zdjąłeś b , to przejdź do stanu q' , a na czubek stosu włóż słowo w .* Taki automat czyta słowo wejściowe litera po literze, zmieniając przy tym stan jak zwykły automat skończony, a do tego jeszcze buduje sobie stos. Czy istnieje algorytm odpowiadający, dla danych dwóch deterministycznych automatów ze stosem, czy istnieje niepuste słowo wejściowe, po przeczytaniu którego, oba te automaty będą miały na swoich stosach takie same ciągi symboli?

Zadanie 137. Przez *gramatykę bezkontekstową z kontekstowym znikaniem* będziemy w tym zadaniu rozumieć obiekt różniący się od gramatyki bezkontekstowej jedynie obecnością – w zbiorze produkcji – dodatkowych reguł postaci $w \rightarrow \varepsilon$, gdzie w jest słowem złożonym z nieterminali, zaś ε jest jak zwykle słowem pustym.

Przez *problem znikania* rozumiemy w tym zadaniu problem w którym dana jest gramatyka ze znikaniem, mająca symbol początkowy S i zbiór produkcji Π , i w którym pytamy czy $S \xrightarrow{*}_{\Pi} \varepsilon$.

Udowodnij, że problem znikania jest nierozstrzygalny.

Zadanie 138. Powiemy, że semiprocess Thuego Π jest bezkontekstowy, jeśli dla każdej pary $[w, v] \in \Pi$ słowo w składa się z jednej litery. Czy problem słów dla bezkontekstowych semiprocessów Thuego jest rozstrzygalny?

Zadanie 139. Powiemy, że semiprocess Thuego Π jest prawie bezkontekstowy, jeśli dla każdej pary $[w, v] \in \Pi$ jedno ze słów w i v składa się tylko z jednej litery, drugie zaś z dwóch liter. Czy problem słów dla prawie bezkontekstowych semiprocessów Thuego jest rozstrzygalny?

Uwaga: Użyta w tym i poprzednim zadaniu nomenklatura (pojęcia procesów bezkontekstowych i prawie bezkontekstowych) została wymyślona tylko by po to, by wygodniej było sformułować te zadania i nie ma wiele wspólnego z jakimkolwiek standardem.

Zadanie 140. Semiprocess Thuego Π nad alfabetem $\{0, 1\}$ nazwiemy, na potrzeby tego zadania, fajnym, jeśli każda produkcja $\langle l, r \rangle \in \Pi$ ma własność $|l|_1 = |r|_1$ (to znaczy ma po lewej stronie tyle samo jedynek co po prawej). Udowodnij, że problem czy dla danego słowa w i danego fajnego semiprocessu Thuego Π zachodzi $1111 \xrightarrow{*}_{\Pi} w$, jest nierozstrzygalny. *Wskazówka: Typowe i mało skomplikowane.*

Zadanie 141. Niech $\phi(x, y)$ będzie pewną formułą arytmetyki liczb naturalnych z dodawaniem i mnożeniem, z dwiema zmiennymi wolnymi.

Napisz zdanie ψ arytmetyki liczb naturalnych z dodawaniem i mnożeniem, które będzie prawdziwe wtedy i tylko wtedy, gdy istnieją liczba $l \geq 1$ i skończony ciąg liczb naturalnych a_1, a_2, \dots, a_l , taki, że $a_1 = 1$, $a_l = 2$, oraz taki, że dla każdego $1 \leq i \leq l-1$ zachodzi $\phi(a_i, a_{i+1})$.

Wskazówka: Możesz na przykład użyć chińskiego twierdzenia o resztach (choć są również inne rozwiązania). Posłuż się makrami, podobnymi do tych, których używaliśmy na wykładzie — na przykład Pierwsza(z), Kolejne-Pierwsze(z, t).

W kolejnych dwóch zadaniach, po napisanym na skończonej taśmie słowie poruszają się żuczki. Dwa albo trzy. Każdy z żuczków jest rodzajem dwukierunkowego deterministycznego automatu skończonego, to znaczy ma skończony zbiór stanów i funkcję przejścia (inną dla każdego żuczka), która w zależności od tego w jakim jest stanie, jaki symbol widzi w aktualnej komórce taśmy, i które z pozostałych żuczków znajdują się wraz z nim w aktualnej komórce

taśmy każe mu odpowiednio zmienić stan i poruszyć się w lewo lub w prawo (dla porządku zakładamy, że końce słowa oznaczone są unikalnymi symbolami, dzięki czemu żuczek nigdy nie opuści taśmy i że na początku wszystkie żuczki stoją na początku słowa, w pewnym ustalonym stanie początkowym). Kolejny ruch żuczka, czyli wykonanie funkcji przejścia, następuje zawsze wtedy, gdy usłyszy on tyknięcie zegara.

Przez *problem niepustości* rozumiemy pytanie, czy dla danych funkcji przejścia żuczków istnieje słowo, które żuczki zaakceptują, to znaczy takie, na którym któryś z nich osiągnie, po skończonej liczbie kroków, ustalony stan akceptujący.

Zadanie 142. Dwa synchroniczne żuczki. Pokaż, że problem niepustości jest nierozstrzygalny jeśli rozważamy pary (funkcji dla) żuczków i zakładamy, że są one synchroniczne, to znaczy oba słyszą tykanie tego samego zegara.

Zadanie 143. Trzy asynchroniczne żuczki. Pokaż, że z tezy Zadania 139 wynika, że nierozstrzygalny jest również problem niepustości dla trójek (funkcji dla) żuczków jeśli zakładamy, że są one asynchroniczne, to znaczy w każdej komórce taśmy słychać osobny zegarek, który tyka jak chce (np. czasem wolniej czasem szybciej).

Uwaga. Różne zachowania zegarków mogą tu skutkować różnymi obliczeniami, czyli różnymi zachowaniami żuczków. Myślimy o tym jak o obliczeniu niedeterministycznym: słowo zostaje zaakceptowane, gdy istnieje zachowanie zegarków, które prowadzi do obliczenia akceptującego.

Komentarz. Nie znam rozwiązania zadania o dwóch asynchronicznych żuczках.

Zadanie 144. Czy problem niepustości języka $L_G \cap L_G L_G$, dla danej gramatyki bezkontekstowej G , jest rozstrzygalny?

Zadanie 145. Automat skończony z dwoma stoperami czyta słowo wejściowe jak zwykły **niedeterministyczny** automat skończony, ale oprócz skończonego zbioru stanów ma dwa stopery. Automat może, gdy uzna to za stosowne, uruchomić¹ lub zatrzymać, każdy ze stoperów, z tym że raz zatrzymanego stopera nie da się już ponownie uruchomić. Uruchomiony stoper działa jak licznik, zwiększający się o 1 z każdym krokiem automatu. Po zatrzymaniu obu stoperów automat umie porównać ich wskazania i uzależnić swój kolejny stan od tego czy te wskazania są równe czy różne (zwróć uwagę, że to jest jedyny sposób w jaki automat może dowiedzieć się czegośkolwiek o wskazaniach stoperów). Pokaż, że problem totalności dla automatów skończonych z dwoma stoperami jest nierozstrzygalny.

Żuczek Kleksiorek ($\dot{Z}K$) jest jak deterministyczny dwukierunkowy automat skończony, tylko że ma dodatkowo pewną, niewielką, zdolność pisania. Może mianowicie pozostawić symbol w aktualnie odwiedzanej komórce taśmy bez zmiany, a może go też zastąpić wielkim czarnym kleksem. Dokładniej mówiąc, $\dot{Z}K$ zadany jest przez skończony alfabet Σ , skończony zbiór stanów Q , stany początkowy q_0 i końcowy q_F , należące do Q , oraz funkcję przejścia $\delta(\Sigma \cup \{\alpha, \beta, \blacksquare\}) \times Q \rightarrow Q \times \{\perp, \blacksquare\} \times \{L, R\}$, gdzie α i ω to specjalne znaki mówiące dwukierunkowemu automatu że jest w początku/końcu słowa, L i R mówią w którą stronę taśmy żuczek ma się ruszyć, \perp to instrukcja mówiąca że aktualna komórka taśmy ma być pozostawiona bez zmiany zaś \blacksquare to instrukcja mówiąca, że w aktualnej komórce ma być umieszczony kleks. Żuczek zaczyna obliczenia stojąc na znaku α na początku słowa, rusza się w naturalny sposób opisany przez δ i akceptuje gdy osiągnie stan q_F .

Zadanie 146. Czy niepustość jest dla Żuczków Kleksiorów rozstrzygalna? To znaczy czy rozstrzygalne jest, czy dla danego żuczka istnieje jakieś słowo które ten żuczek zaakceptuje?
Wskazówka: PCP.

¹Z wartością równą zero.

Żuczek Końcojadek (ŻK) jest prawie jak **niedeterministyczny** dwukierunkowy automat skończony. To znaczy chodzi sobie po słowie wejściowym pamiętając jeden ze skończenie wielu stanów ze zbioru Q . Stojąc na symbolu $a \in \Sigma'$ (gdzie $\Sigma' = \Sigma \cup \{\alpha, \omega\}$ i gdzie Σ jest wejściowym alfabetem) podejmuje, zgodnie z tym na co pozwala mu relacja przejścia δ i w zależności od a i od tego w jakim stanie $q \in Q$ akurat jest, decyzję do jakiego stanu $q' \in Q$ ma przejść i czy skierować się, w kolejnym kroku, o jedną komórkę taśmy w lewo czy w prawo. Chyba że stoi na znaku α , oznaczającym lewy koniec słowa – wtedy wolno mu iść tylko w prawo, albo na znaku ω – wtedy wolno mu iść tylko w lewo. Zaczyna stojąc (w wyróżnionym stanie $q_0 \in Q$) na znaku α znajdującym się na lewym końcu pewnego słowa postaci $\alpha w \omega$, dla $w \in \Sigma^*$ i akceptuje to słowo jeśli, robiąc kroki zgodnie z opisanymi wyżej regułami, może w końcu dojść do wyróżnionego stanu akceptującego $q_F \in Q$.

Od niedeterministycznego dwukierunkowego automatu skończonego różni się Żuczek Końcojadek tym, że stojąc na jakimś symbolu $a \in \Sigma$ może zamienić ten symbol (jeśli akurat pozwala mu na to jego relacja przejścia) na α albo na ω . Oczywiście, jeśli zamieni symbol na α to potem musi iść w prawo, a jeśli na ω to musi iść w lewo – miejsce gdzie został napisany symbol końca słowa staje się w ten sposób, z punktu widzenia żuczka, nowym końcem słowa.

Zadanie 147. Czy totalność jest dla Żuczków Końcojadków rozstrzygalna? To znaczy czy rozstrzygalne jest, czy dany żuczek (zadany przez $\langle \Sigma, Q, q_0, q_F, \delta \rangle$) zaakceptuje wszystkie słowa nad swoim alfabetem?

Zadanie 148. A co w sprawie problemu niepustości dla Żuczków Lewykońcojadków?

Zadanie 149. Scenariusz w tym zadaniu jest taki, że Chrząszczycza z Chrząszczątkiem wędrują po pewnym nieskończonym słowie nad jakimś ustalonym skończonym alfabetem Σ . Zaczynają wędrówkę razem, stojąc na pierwszym symbolu słowa wejściowego, w stanie q_0 . A potem, w każdym kolejnym kroku, Chrząszczątko oddala się o jeden symbol od początku słowa. Chrząszczycza zaś w każdym nieparzystym kroku oddala się o jeden symbol od początku słowa, w każdym zaś parzystym kroku odpoczywa sobie w bezruchu. Chyba że osiągną stan q_F , wtedy dalej nie idą.

Chrząszczycza z Chrząszczątkiem porozumiewają się (telefonicznie), zatem oboje wiedzą nie tylko jaki symbol z Σ sami widzą, ale też jaki symbol widzi to drugie. Dlatego myślimy, że mają oni wspólny skończony zbiór stanów Q (taki stan opisuje ich dwoje łącznie), gdzie $q_0, q_F \in Q$, i wspólną funkcję przejścia $\delta : Q \times \Sigma \times \Sigma \rightarrow Q$.

No i zadanie w końcu jest takie: czy rozstrzygalny jest problem „dana funkcja przejścia δ dla Chrząszczyczy z Chrząszczątkiem; czy istnieje nieskończone słowo po którym będą szli w nieskończoność, nigdy nie osiągając stanu q_F ”?

Wskazówka/Uwaga. Rozwiązując to zadanie, będzie prawdopodobnie trzeba jakoś reprezentować kolejne konfiguracje maszyny Turinga. Autor rozwiązania powinien wyjaśnić, jakie problemy są tu do pokonania i jak zostały pokonane.

Część Trzecia Kursu

15 Niedeterministyczne maszyny Turinga i klasa NP

Zadanie 150. Pokaż, że wielomianową maszynę niedeterministyczną można przerobić tak, aby zgadywała rozwiązanie wcześniej niż pozna dane. Dokładniej rzecz ujmując, udowodnij że jeśli zbiór A należy do klasy NP, to istnieją wielomiany p, q oraz niedeterministyczna maszyna Turinga M rozpoznająca A , działająca dla danego n w następujący sposób: M wyznacza na taśmie blok klatek o długości $p(|n|)$ - zatem interesuje ją wielkość n , ale nie jego dokładna wartość - po czym niedeterministycznie i nie czytając n zapełnia ten blok ciągiem zer i jedynek. Dopiero następnie czyta n i przechodzi do fazy, w której obliczenie jest już deterministyczne i nie zabiera więcej niż $q(|n|)$ kroków.

Zadanie 151. Pokaż, że jeśli zbiór $A \subseteq \mathbb{N}^2$ jest w P i p jest wielomianem, to zbiór $\{n : \exists m |m| \leq p(|n|) \text{ i } [n, m] \in A\}$, czyli rodzaj rzutu A na pierwszą oś, jest w NP.

Zadanie 152. Pokaż, że każdy zbiór w NP jest rzutem pewnego zbioru z P. To znaczy jeśli B jest w NP, to istnieje wielomian p i zbiór $A \subseteq \mathbb{N}^2$ należący do P i taki, że $B = \{n : \exists m |m| \leq p(|n|) \text{ i } [n, m] \in A\}$.

16 Redukcje wielomianowe i NP-trudność

Zadanie 153. Pokaż, że $5SAT \leq_p 3SAT$.

Zadanie 154. Problem STASI² jest taki: mamy dany graf nieskierowany i liczbę k . Czy da się rozstawić k agentów w wierzchołkach grafu tak, aby każdy wierzchołek w którym nie stoi agent miał (co najmniej jednego) agenta za sąsiada? Pokaż, że $3SAT \leq_p STASI$.

Wskazówka: To nie jest trudne. Idea jest podobna jak przy dowodzie faktu, że $3SAT \leq_p 3COL$, który był na wykładzie. Tylko łatwiej.

Zadanie 155. Niech H oznacza problem cyklu Hamiltona dla grafów nieskierowanych (tzn. język tych wszystkich grafów nieskierowanych, w których istnieje ścieżka zamknięta przechodząca dokładnie raz przez każdy wierzchołek).

Niech H_d oznacza problem cyklu Hamiltona dla grafów skierowanych. Pokaż, że $H \leq_p H_d$ i $H_d \leq_p H$.

Wskazówka: W trudniejszą stronę trzeba każdy wierzchołek zastąpić trzema.

Zadanie 156. Klauzula nazywa się *hornowską* jeśli co najwyżej jeden z jej literalów jest niezanegowany. Pokaż, że problem HORNSAT spełnialności formuł, w postaci CNF, których każda klauzula jest hornowska, jest w P.

Zadanie 157. (za 2 punkty) Pokaż, że problem spełnialności formuł w koniunkcyjnej postaci normalnej, w których każda klauzula jest alternatywą co najwyżej dwóch literalów jest w klasie \mathcal{P} . (Patrz definicja na stronie 375 polskiego wydania książki Hopcrofta i Ullmana. Tłumaczka z bożej łaski tłumaczy CNF jako PNK).

Zadanie 158. Pokaż, że $3SAT \leq_p 3SAT_3$. To ostatnie to $3SAT$ ograniczony tylko do formuł, w których żadna zmienna nie występuje więcej niż 3 razy.

²To się naprawdę nazywa "Problem zbioru dominującego". Zadanie sformułowałem, tak jak jest teraz sformułowane, w latach 90, kiedy było modne śmiać się z NRD (wiecie co to było NRD?), bo wydawało się (wtedy), że u nas było inaczej.

Zadanie 159. (za 2 punkty) Udowodnij, że problem cyklu Hamiltona jest NP-zupełny.

Zadanie 160. Problem komiwojażera jest taki: dany jest graf nieskierowany pełny, którego krawędzie etykietowane są liczbami całkowitymi. Waga drogi w grafie jest definiowana jako suma wag krawędzi należących do tej drogi. Dana liczba k . Czy istnieje w grafie cykl Hamiltona o wadze mniejszej niż k ?

Pokaż, że problem komiwojażera jest NP-zupełny. Wolno skorzystać z NP-zupełności problemu Hamiltona.

Zadanie 161. Pokaż, że jeśli istnieje wielomianowy algorytm wskazujący, dla danego przykładu problemu komiwojażera, cykl nie więcej niż dwa razy dłuższy od optymalnego, to $\mathcal{P} = \text{NP}$.

Wskazówka: Podobnie jak w poprzednim zadaniu trzeba się odwołać do NP-zupełności problemu Hamiltona.

Zadanie 162. Pokaż, że jeśli ograniczymy się do przykładów problemu komiwojażera, w których wagi krawędzi spełniają nierówność trójkąta, to znaczy dla każdych wierzchołków v_1, v_2, v_3 zachodzi $d(v_1, v_2) + d(v_2, v_3) \geq d(v_1, v_3)$, to istnieje wielomianowy algorytm znajdujący cykl Hamiltona o wadze nie więcej niż dwa razy większej od optymalnej.

Zadanie 163. Udowodnij, że problem komiwojażera pozostaje NP-zupełny, gdy ograniczymy się do przykładów, w których funkcja wagi krawędzi d spełnia *mocny warunek trójkąta*: $d(x, y) < d(x, z) + d(z, y)$.

Zadanie 164. Jaka jest złożoność problemu SAT_2 , tzn. problemu spełnialności formuł w postaci CNF, w których żadna zmienna nie występuje więcej niż 2 razy?

Zadanie 165. Udowodnij, że problem istnienia w danym grafie o n wierzchołkach kliku mającej $n/2$ wierzchołków jest NP-zupełny.

Zadanie 166. (za 2 punkty) Udowodnij, że jeśli istnieje wielomianowy algorytm znajdujący w danym grafie klikę wielkości co najmniej połowy kliku maksymalnej³, to istnieje również wielomianowy algorytm znajdujący w danym grafie klikę wielkości co najmniej $1/\sqrt{2}$ kliku maksymalnej.

Zadanie 167. Rozważmy następujący *problem spełnialności dla zdecydowanej większości klauzul*: Dane różne klauzule rachunku zdań $\phi_1, \phi_2, \dots, \phi_n$. Czy można podstawić wartości 0 i 1 za zmienne zdaniowe tak aby więcej niż $9/10$ spośród klauzul $\phi_1, \phi_2, \dots, \phi_n$ przyjęła wartość logiczną 1? Udowodnij, że *problem spełnialności dla zdecydowanej większości klauzul* jest NP-zupełny. Przypominam, że klauzulą nazywamy formułę postaci $l_1 \vee l_2 \vee \dots \vee l_k$, gdzie l_i są literałami, to znaczy zmiennymi zdaniowymi lub ich negacjami.

Wskazówka: Skorzystaj z NP-zupełności SAT.

Zadanie 168. Niech KLIKA_c będzie problemem istnienia w danym grafie o n wierzchołkach kliku zawierającej nie mniej niż n/c wierzchołków. Pokaż, że dla każdych c, c' zachodzi $\text{KLIKA}_c \leq_p \text{KLIKA}_{c'}$.

Zadanie 169. Udowodnij, że problem istnienia dla danego grafu nieskierowanego, takiego kolorowania wierzchołków tego grafu trzema kolorami, aby każdy wierzchołek sąsiadował z co najwyżej jednym wierzchołkiem tego samego koloru, jest NP-zupełny.

³Nie wiemy czy istnieje taki algorytm.

Zadanie 170. Przykładem problemu pokrycia zbioru podzbiorami rozłącznymi (PZPR) jest skończona rodzina $A = \{A_1, A_2, \dots, A_N\}$ skończonych zbiorów. $A \in PZPR$ jeśli istnieje rodzina $B \subseteq A$ zbiorów rozłącznych, taka że suma wszystkich zbiorów z B jest równa sumie wszystkich zbiorów z A . Udowodnij, że $3SAT \leq_p PZPR$.

Wskazówka: pokaż, że $3SAT_3 \leq_p PZPR$ gdzie $3SAT_3$ to problem spełnialności dla formuł w postaci 3CNF w których każda zmienna występuje co najwyżej 3 razy.

Zadanie 171. Jaka jest złożoność problemu istnienia, dla danej formuły boolowskiej w postaci CNF, wartościowania przy którym w każdej klauzuli wszystkie literały przyjmują wartość 1 albo wszystkie literały przyjmują wartość 0?

Zadanie 172. Jaka jest złożoność następującego problemu: Dany graf nieskierowany. Czy istnieje taki sposób pokolorowania jego krawędzi dwoma kolorami, czerwonym i niebieskim, aby każda krawędź była pokolorowana i aby nie pojawił się żaden niebieski cykl nieparzystej długości ani żaden czerwony cykl nieparzystej długości?

W pewnej Wzorowej Demokracji odbyły się wybory, w wyniku których do parlamentu weszła pewna liczba partii. Żadna z nich nie uzyskała większości, konieczne stało się zatem wyłonienie koalicji rządowej dysponującej więcej niż połową głosów. Każda z partii złożyła w związku z tym oświadczenie o następującej formie: *wejdziemy do koalicji wtedy i tylko wtedy gdy otrzymamy następujące stanowiska: lista stanowisk*. Listy stanowisk żądanych przez partie przecinają się czasem niepusto, a partie w swych żądaniach są nieugięte.

Oznaczmy przez $KOAL$ problem istnienia większościowej koalicji, przy której można zaspokoić oczekiwania tworzących ją partii. Dane stanowi tu lista partii, wraz z ilością mandatów jakimi każda partia rozporządza i listą stanowisk jakich się domaga. Przez $KOAL_j^i$ oznaczmy wariant problemu $KOAL$, w którym każda partia może żądać co najwyżej i stanowisk, a każdego stanowiska żąda co najwyżej j partii (brak któregoś z indeksów oznacza, że nie ograniczamy tego parametru).

Zadanie 173. Jaka jest złożoność problemu $KOAL_2$?

Wskazówka: Wolno skorzystać z NP-zupełności problemu istnienia, w grafie nieskierowanym o n wierzchołkach, zbioru wierzchołków niezależnych o mocy $n/4$.

Zadanie 174. Jaka jest złożoność problemu $KOAL_3^3$?

Zadanie 175. Jaka jest złożoność problemu $KOAL_2^2$?

Komentarz: Nie potrafiłem niestety rozstrzygnąć jaka jest złożoność problemu $KOAL^2$. Dlatego pomyślałem, że może lepiej będzie, jeśli nie umieszczę tu takiego zadania.

Zadanie 176. Jaka jest złożoność problemu 3-kolorowania grafów, jeśli ograniczymy się do grafów o stopniu wierzchołków równym co najwyżej 4?

Zadanie 177. Przez liczbę chromatyczną grafu nieskierowanego $G = \langle V, E \rangle$ rozumiemy najmniejszą liczbę naturalną n dla której istnieje funkcja $l : V \rightarrow \{1, 2, \dots, n\}$ taka, że zachodzi formuła $\forall x, y \in V \quad E(x, y) \Rightarrow l(x) \neq l(y)$. Liczbę chromatyczną grafu G oznaczamy przez $\chi(G)$. Udowodnij, że jeżeli istnieje wielomianowy algorytm który, dla danego grafu G , zwróci zawsze jedną z liczb $\{\chi(G), \chi(G) + 1\}$, to $P = NP$.

Przez *pokrycie cyklowe* nieskierowanego grafu $G = \langle V, E \rangle$ rozumieć będziemy zbiór rozłącznych (wierzchołkowo) cykli o wierzchołkach w V i krawędziach w E , taki, że każdy wierzchołek ze zbioru V należy do któregoś z tych cykli. *Moc pokrycia cyklowego* to liczba cykli z

których to pokrycie się składa. Dla grafu G przez $\sigma(G)$ oznaczmy, w kolejnych trzech zadaniach, minimalną moc pokrycia cyklowego G (ponieważ wierzchołek jest sam w sobie cyklem, więc liczba $\sigma(G)$ jest zawsze określona i nie większa od liczby wierzchołków G).

Zadanie 178. Niech $c > 3$. Jaka jest złożoność problemu stwierdzenia, dla danego grafu G , czy $\sigma(G) \leq n/c$, gdzie n to liczba wierzchołków G ?

Zadanie 179. Załóżmy, że istnieje wielomianowy algorytm, który dla każdego grafu G takiego że $\sigma(G) = 1$ zwraca pokrycie cyklowe G składające się z nie więcej niż dwóch cykli. Pokaż, że w takim razie $P=NP$.

Zadanie 180. Pokaż, że jeśli P jest różne od NP to funkcja $\sigma(G)$ nie może być, przez żaden wielomianowy algorytm, aproksymowana z dokładnością do stałej multiplikatywnej. Mówiąc dokładniej, pokaż że nie istnieje wtedy wielomianowy algorytm M i liczba $c > 0$ taka, że dla każdego grafu G algorytm M uruchomiony dla G zwróci jego pokrycie cyklowe o mocy nie większej niż $c\sigma(G)$.

Zadanie 181. Instancją problemu NAE-SAT (Not All Equal SAT) jest formuła boolowska w postaci 3CNF. Formuła należy do NAE-SAT jeśli istnieje wartościowanie zmiennych, przy którym każda z klauzul jest spełniona, ale w każdej jest przynajmniej jeden fałszywy literal. Pokaż, że $3COL \leq_P$ NAE-SAT.

Zadanie 182. Jaka jest złożoność problemu istnienia takiego kolorowania wierzchołków danego nieskierowanego grafu G dwoma kolorami, przy którym nie powstaje żaden trójkąt o wszystkich trzech wierzchołkach tego samego koloru? Przez trójkąt rozumiemy tu 3-klikę w grafie G .

Zadanie 183. Czy odpowiedź na pytanie z poprzedniego zadania zmienia się, jeśli ograniczymy się do instancji problemu będącymi grafami 4-kolorowalnymi?

Zadanie 184. Problem wesołego dwukolorowania zdefiniujemy w tym zadaniu następująco. Instancją problemu jest graf nieskierowany. Pytamy, czy da się pokolorować wierzchołki tego grafu dwoma kolorami w taki sposób, aby każdy wierzchołek miał wśród swoich sąsiadów przynajmniej po jednym wierzchołku każdego koloru. Pokaż, że problem wesołego dwukolorowania jest NP-zupełny.

Wskazówka: Wolno skorzystać z NP-zupełności problemu not-all-equal-3-SAT. Instancjami tego problemu są formuły w postaci 3-CNF, a pytamy, czy istnieje takie wartościowanie zmiennych, że w każdej klauzuli jest przynajmniej jeden literal zwartościowany jako prawdziwy i przynajmniej jeden literal zwartościowany jako fałszywy.

Zadanie 185. Instancją problemu *Deadlock Recovery* (w skrócie DR) jest trójka $\langle V, E, n \rangle$, gdzie $\langle V, E \rangle$ to graf skierowany, zaś n to liczba naturalna.

Instancja $\langle V, E, n \rangle$ należy do DR gdy z grafu $\langle V, E \rangle$ da się usunąć k krawędzi tak, że powstały w ten sposób graf nie będzie miał cykli (skierowanych).

Pokaż że DR jest NP-zupełny.

Na planecie Nijak używa się logiki o trzech wartościach: T (prawda), F (fałsz) i M (mhm). Spójniki logiczne \vee, \wedge i \neg są dla wartości T i F określone tak jak na Ziemi, $\neg(M) = M$, zaś \vee i \wedge są symetryczne i $M \vee M = M \wedge M = M$, $T \vee M = T \wedge M = T$ i $F \vee M = F \wedge M = F$. Definicje postaci CNF, 2CNF itd. są na Nijak takie same jak na Ziemi. Podobnie, formuła jest na Nijak *spełnialna* jeśli istnieje wartościowanie zmiennych przy którym ma ona wartość logiczną T .

Zadanie 186. Jaka jest na Nijak złożoność problemu spełnialności formuł w postaci 2CNF?

Zadanie 187. Jaka jest na Nijak złożoność problemu spełnialności formuł w postaci 3CNF? W zadaniu tym zakładamy, że literałami są zmienne, negacje zmiennych, oraz stałe T,F i M.

Zadanie 188. Jaka jest na Nijak złożoność problemu spełnialności formuł w postaci 3CNF? W zadaniu tym zakładamy, że literałami są zmienne i negacje zmiennych, ale nie są nimi stałe T,F i M.

Zadanie 189. Jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły w postaci 2CNF, czy istnieje wartościowanie spełniające przynajmniej $3/4$ wszystkich klauzul w tej formule?

Zadanie 190. Jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły w postaci 2CNF, czy istnieje wartościowanie spełniające przynajmniej $3/4$ spośród pierwszych stu klauzul w tej formule, oraz wszystkie pozostałe?

Zadanie 191. Melmażelon z planety Melmak umie odpowiedzieć, zawsze zgodnie z prawdą, na jedno pytanie o spełnialność formuły boolowskiej. Dokładniej mówiąc, melmażelon pożera formułę, po czym, jeśli formuła jest spełnialna to robi się cały seledynowy, zaś jeśli jest niespełnialna robi się cały pomarańczowy. Po czym, w obu przypadkach, rusza tak śmiesznie łapkami i zaraz zdycha.

Oznaczmy przez $PTIME^M$ klasę problemów, które można rozwiązać w deterministycznym czasie wielomianowym kosztem jednego melmażelona. To znaczy takich problemów, dla których istnieje wielomianowy algorytm, działający w czasie wielomianowym, zadający, w trakcie swojego działania, co najwyżej jedno pytanie do melmależona o spełnialność jakiejś formuły, i uzależniający dalsze działanie od odpowiedzi na to pytanie.

Czy $PTIME^M = NP \cup co-NP$? Zakładamy, że $co-NP \neq NP \neq PTIME$.

Zadanie 192. Czy istnieje język $L \in PTIME$, który nie daje się rozpoznawać, na maszynie Turinga, w czasie kwadratowym? (Przez *czas kwadratowy* rozumiemy czas ograniczony przez $c(n^2 + 1)$ gdzie n jest wielkością wejścia a c pewną stałą niezależną od wielkości wejścia).

Kolejne cztery zadania są o układaniu puzzli. Każdy kiedyś układał puzzle, więc poniższe definicje nie będą dla nikogo niespodzianką.

Częścią każdej instancji problemu układania puzzli jest zbiór \mathbb{W} wycięć. \mathbb{W} jest sumą rozłączną $\{b\} \cup W \cup W'$, gdzie $W' = \{w' : w \in W\}$. Każdy rozumie, że wycięcie b nazywamy *brzegowym*, wycięcia z W są *wklęsłe* a wycięcia z W' są *wypukłe*. I że w pasuje do w' .

Puzzle jest prostokątem $1 \times k$ gdzie k jest jakąś liczbą naturalną. Przez *krawędź puzzle* rozumiemy każdy odcinek boku puzzle, o długości 1, którego końce znajdują się w całkowitych odległościach od odpowiednich rogów puzzle (zatem puzzle $1 \times k$ ma $2k + 2$ krawędzie)

Każdej krawędzi każdego puzzle jest przypisane wycięcie ze zbioru \mathbb{W} .

Puzzle układa się w prostokątnej ramce (o brzegach, których długości są liczbami naturalnymi). Puzzle są dobrze ułożone jeśli wypełniają całą ramkę, nie zachodzą na siebie, i jeśli:

- krawędź puzzle ma wycięcie b wtedy i tylko wtedy gdy ta krawędź dotyka brzegu ramki;
- jeśli krawędzie dwóch puzzli się stykają więcej niż w jednym punkcie (zauważ, że wtedy stykają się całymi swoimi długościami) to muszą do siebie pasować: jedna z tych krawędzi ma wycięcie $w \in W$ a druga $w' \in W'$.

Zadanie 193. Jaka jest złożoność następującego problemu:

Dają nam (skończony) worek puzzli (worek, czyli multiset; to znaczy dają nam zbiór puzzli, oraz informację, o każdym puzzlu z tego zbioru, ile kopii tego puzzla jest w worku) i liczbę naturalną n . Pytają, czy używając puzzli z danego worka da się dobrze wypełnić ramkę 2 na n , tak żeby żaden puzzle nie został w worku. Wielkością instancji jest liczba n .

Wskazówka: może warto zadać sobie najpierw pytanie o złożoność problemu w którym dana jest ternarna relacja $E \subseteq V^3$, pytają nas zaś, czy istnieje kolorowanie $kol : V \rightarrow \{r, g, b\}$ takie, że zawsze jeśli $E(x, y, z)$ to $\{kol(x), kol(y), kol(z)\} = \{r, g, b\}$.

W następnych trzech zamiast worka z puzzlami (jak zadaniu poprzednim), dostajemy zbiór puzzli i **maszynkę do ich kopiowania**: jeśli jakiś puzzle jest w danym zbiorze to możemy, układając, użyć tyłu jego kopii ile chcemy (w szczególności możemy jakiegoś puzzle wcale nie wykorzystywać). W każdym z kolejnych trzech zadań wielkością instancji jest suma n i liczby puzzli w danym zbiorze.

Zadanie 194. Jaka jest złożoność następującego problemu:

Dają nam zbiór puzzli i liczbę naturalną n . Pytają, czy używając puzzli z danego zbioru da się dobrze wypełnić ramkę 2 na n .

Zadanie 195. Jaka jest złożoność następującego problemu:

Dają nam zbiór puzzli o wymiarach 1 na 1 i liczbę naturalną n . Pytają, czy używając puzzli z danego worka da się dobrze wypełnić ramkę o n kolumnach i n wierszach.

Zadanie 196. Jaka jest złożoność następującego problemu:

Dają nam zbiór puzzli i liczbę naturalną n . Pytają, czy używając puzzli z danego worka da się dobrze wypełnić jakąś ramkę o n kolumnach.

17 O teoretycznych kłopotach kryptografii

Zadanie 197. (za 2 punkty) Funkcja różnowartościowa $f : \mathbb{N} \rightarrow \mathbb{N}$ i taka, że dla każdego n zachodzi $|n| = |f(n)|$ jest *jednostronna* jeśli istnieje wielomianowy algorytm obliczający f , ale nie ma wielomianowego algorytmu obliczającego f^{-1} . Pokaż, że jeśli istnieje jakaś funkcja jednostronna to $\text{NP} \cap \text{co-NP} \neq \text{PTIME}$.

Wskazówka: Rozważ zbiór $\{\langle x, y \rangle : f^{-1}(x) < y\}$.

Zadanie 198. Niech $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ będzie bijekcją obliczalną w czasie wielomianowym. Czy wynika z tego, że f^{-1} też jest bijekcją obliczalną w czasie wielomianowym?

18 Problemy być może nie należące do klasy NP

Zadanie 199. Patrokles, mając daną formułę boolowską ϕ , taką że $\text{Var}(\phi) = \{p_1, p_2, \dots, p_n\}$ próbuje ją spełnić. W tym celu wartościuje zmienną p_1 , potem zmienną p_2 itd. Ale gdy rozważa z wartościowanie kolejnej zmiennej, niech to będzie p_k , przerwać mu może Mojra, i rzec: *pozwól kolego, że p_k to ja z wartościuję nie ty*. I tak jak rzecze, uczynić. Po czym wszystko wraca do normalnego biegu rzeczy, to znaczy Patrokles bierze się za zmienną p_{k+1} .

Nie trzeba dodawać, że Mojra dąży do tego żeby formuła pozostała niespełniona. Jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły ϕ , czy Patrokles może spełnić ją bez względu na wybory Mojry, gdy:

- Mojra może mu przerwać (i z wartościować kolejną zmienną) w sumie najwyżej trzy razy?
- Mojra może mu przerwać ile razy chce?

Zadanie 200. (za 2 punkty) Rozważmy ponownie sytuację opisaną w Zadaniu 196. Jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły ϕ , czy Patrokles może spełnić ją bez względu na wybory Mojry jeśli Mojra może mu przerwać w sumie nie więcej niż $n/2$ razy?
Wskazówka: sowy tym razem są tym czym się wydają. Tylko jak to udowodnić?

W kolejnych trzech zadaniach rozważamy dziwną grę między dwoma graczami, Bolkiem i Olkiem. Dana jest formuła zdaniowa ϕ , której ze zmiennymi zdaniowymi $p_1, q_1, p_2, q_2 \dots p_n, q_n$. Gracze na przemian wartościują zmienne – w i -tym ruchu gry zmienną p_i wartościuje Bolek, a następnie zmienną q_i wartościuje Olek. Powstaje w ten sposób pewne wartościowanie $\sigma : \{p_1, q_1, p_2, q_2 \dots p_n, q_n\} \rightarrow \{0, 1\}$ i Bolek wygrywa grę jeśli $\bar{\sigma}(\phi) = 1$. W każdym z zadań pytamy jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły ϕ , czy Bolek ma strategię wygrywającą w tej dziwacznej grze, jeśli dodatkowo założymy, że:

Zadanie 201. W ciągu całej gry każdemu z graczy wolno tylko co najwyżej trzy zmienne zwartościować jako zero.

Zadanie 202. Liczba jedynek w ciągu $(\sigma(q_i))_{i=1\dots n}$ może się różnić co najwyżej o 1 od liczby jedynek w ciągu $(\sigma(p_i))_{i=1\dots n}$. W przeciwnym razie Olek przegrywa bez względu na to czy $\bar{\sigma}(\phi) = 1$.

Zadanie 203. Wartość jaką Olek nadaje zmiennej q_i nigdy nie może być większa niż wartość jaką Bolek nadał właśnie zmiennej p_i , zaś wartość jaką Bolek nadaje zmiennej p_i (dla $p \geq 2$) nigdy nie może być mniejsza niż wartość jaką Bolek nadał właśnie zmiennej q_{i-1} .

Zadanie 204. Rozważmy następującą *jeszcze bardziej dziwną grę* między dwoma graczami, Bolkiem i Olkiem. Dana jest formuła zdaniowa ϕ , której zbiór zmiennych zdaniowych to $p_1, p_2, \dots p_n$. Pierwszy ruch należy do Bolka. Gracz który wykonuje i -ty ruch wartościuje zmienną p_i . Jeśli zwartościował ją jako 0, to $i+1$ -szy ruch również należy do niego. Jeśli natomiast zwartościował ją jako 1, to $i+1$ -szy ruch należy do przeciwnika.

Bolek wygrywa grę, jeśli po n ruchach formuła ϕ (już teraz bez zmiennych, bo wszystkie zostały zwartościowane) jest prawdziwa. W przeciwnym razie wygrywa Olek.

Jaka jest złożoność problemu rozstrzygnięcia, dla danej formuły ϕ , czy Bolek ma strategię wygrywającą w *jeszcze bardziej dziwacznej grze*?

W Zadaniach 202-204 Ewa gra z Adamem w Szanowanie Relacji. W Zadaniach 202 i 203 instancja gry zadana jest przez:

- zbiór X z relacją $E \subseteq X^4$;
- porządek liniowy \leq_X , na zbiorze X ;
- zbiór K kolorów z relacją $R \subseteq K^4$.

W Zadaniu 204 częścią instancji jest również:

- funkcja $f : X \rightarrow K$ (czyli kolorowanie wierzchołków hipergrafu $[X, E]$)

Dla danej funkcji $f : X \rightarrow K$ powiemy że f *szanuje relację* R gdy dla każdych $x_1, x_2, x_3, x_4 \in X$, jeśli zachodzi $E(x_1, x_2, x_3, x_4)$ to zachodzi również $R(f(x_1), f(x_2), f(x_3), f(x_4))$.

W każdym zadaniu gracze będą wspólnie konstruować funkcję f . Celem Ewy będzie zapewnienie, że po zakończeniu gry (w Zadaniach 202 i 203) lub w każdym momencie gry (w Zadaniu 204) funkcja f będzie szanowała relację R . Celem Adama będzie doprowadzenie do tego, by na końcu gry (w Zadaniach 202 i 203) lub w pewnym momencie gry (w Zadaniu 204) funkcja f nie szanowała relacji R . W każdym z zadań należy rozstrzygnąć jaka jest złożoność problemu istnienia strategii wygrywającej dla Ewy.

Zadanie 205. W tym zadaniu, zakładamy że elementy zbioru X kolorowane są po kolei, tak jak każdy porządek \leq_X , i kolorowane są w zasadzie przez Ewę, z tym że dwa wybrane przez siebie wierzchołki koloruje Adam, przy czym informuje on Ewę o swoim zamiarze pokolorowania wierzchołka dopiero gdy kolej kolorowania przypada na ten wierzchołek. Gdy wszystkie wierzchołki są pokolorowane, gra się kończy.

Zadanie 206. Teraz zakładamy że elementy zbioru X kolorowane są po kolei, tak jak każdy porządek \leq_X , nieparzyste przez Ewę, parzyste przez Adama. Gdy wszystkie wierzchołki są pokolorowane gra się kończy.

Zadanie 207. A w tym Zadaniu zakładamy, że elementy zbioru X są po kolei przemalowywane (na nowy, lub ten sam kolor), tak jak każdy porządek \leq_X , nieparzyste przez Ewę, parzyste przez Adama. Gdy dojdą do ostatniego elementu zbioru X wracają do początku, przemalowując wierzchołki. I tak do końca świata. *Uwaga: kompletne rozwiązanie tego zadania może być długie. Należy się ograniczyć do pokazania głównych pomysłów rozwiązania.*

Zadanie 208. Grę w kompromis definiujemy, na potrzeby tego zadania, następująco. Planszę do gry stanowi skierowany graf dwudzielny $\langle V, E \rangle$ (gdzie $V = L \cup P$ jest podziałem V wynikającym z dwudzielności grafu; zakładamy ponadto, że minimalny stopień wyjściowy wierzchołka jest ≥ 2), z wyróżnionym wierzchołkiem $c_0 \in L$ zwanym początkowym, i ze zbiorem $W \subseteq L$, zwanym zbiorem pozycji wygrywających. W kompromis grają dwaj gracze, \mathcal{L} i \mathcal{P} , wykonujący ruchy na przemian.

Protokół ruchu gracza \mathcal{L} jest następujący. Zastaje on planszę z kamieniem umieszczonym w jakimś wierzchołku $s \in L$. Następnie wybiera dwa różne wierzchołki $t_1, t_2 \in P$ takie, że zachodzi $E(s, t_1)$ i $E(s, t_2)$ i mówi: *wybierz sobie bracie, gdzie chcesz bym się ruszył*. Jego przeciwnik wybiera jeden spośród wierzchołków t_1, t_2 , a gracz \mathcal{L} przesuwa tam kamień. Protokół ruchu gracza \mathcal{P} jest analogiczny, z tym że zamienione są role zbiorów P i L .

Na początku gry kamień leży w c_0 , zatem pierwszy ruch wykonuje gracz \mathcal{L} . Gra kończy się zwycięstwem gracza \mathcal{P} gdy uda mu się postawić kamień w wierzchołku należącym do W . Gra kończy się zwycięstwem gracza \mathcal{L} jeśli gracz \mathcal{P} nie wygra w ciągu $2|V|$ ruchów.

Jaka jest złożoność problemu rozstrzygnięcia, dla danej planszy do gry w kompromis, który z graczy ma w niej strategię wygrywającą?

Zadanie 209. Udowodnij, że problem, czy dane wyrażenie regularne opisuje wszystkie słowa nad danym alfabetem, jest w PSPACE.

Zadanie 210. Udowodnij, że problem rozstrzygania prawdziwości formuł o postaci

$$\exists! x_1 \exists! x_2 \dots \exists! x_n \phi$$

jest w PSPACE. Zmienne x_1, x_2, \dots, x_n przebiegają tu zbiór $\{0, 1, 2\}$. Kwantyfikator $\exists!$ oznacza *istnieje dokładnie jeden*, zaś ϕ jest formułą bez kwantyfikatorów ze zmiennymi x_1, x_2, \dots, x_n , zbudowaną przy pomocy spójników boolowskich i symboli arytmetyki modulo 3, to znaczy symboli dodawania i mnożenia modulo 3, symbolu równości i stałych $\{0, 1, 2\}$. Jak zmieniłoby się rozwiązanie gdyby zmienne przebiegały zbiór $\{0, 1\}$ a arytmetyka była modulo 2?

Wskazówka: uważaj!

Zadanie 211. Jaka jest złożoność problemu istnienia, dla danej formuły boolowskiej ϕ w postaci 2CNF wartościowania spełniającego ϕ i przyporządkowującego wartość logiczną 1 przynajmniej trzem spośród zmiennych występujących w ϕ ?

Zadanie 212. (za 2 punkty) Udowodnij, że są języki rekurencyjne, które nie są w PSPACE.

Zadanie 213. Napisz formułę rachunku predykatów mówiącą, że w danym grafie $\langle V, R \rangle$ istnieje ścieżka prowadząca z danego wierzchołka c do danego wierzchołka k złożona z dokładnie 16 krawędzi. Formuła ta ma mieć przy tym nie więcej niż 10 wystąpień kwantyfikatorów.

Przez formułę rachunku predykatów rozumiemy tu formułę, w której używa się kwantyfikatorów wiążących zmienne przebiegające zbiór V , symbolu relacji R , symbolu równości, nawiasów i spójników logicznych.

Wyjaśnienie: formuła: $\exists x_1 \exists x_2 \dots \exists x_{15} R(c, x_1) \wedge R(x_1, x_2) \wedge \dots \wedge R(x_{15}, k)$ spełnia wszelkie wymogi zadania, oprócz tego, że ma 15 kwantyfikatorów.

Zadanie 214. Jaka jest złożoność problemu rozstrzygnięcia, dla danych deterministycznych automatów skończonych M_1, M_2, \dots, M_k czy język $L_{M_1} \cap L_{M_2} \cap \dots \cap L_{M_k}$ jest niepusty? (wielkością zadania jest tu łączna liczba stanów tych automatów).

Instancją Gry w Zwiedzanie jest (w kolejnych trzech zadaniach) graf skierowany $G = \langle V, E \rangle$, zbiór $S \subseteq V$ „naszych pozycji”, zbiór $T \subseteq V$ „pozycji docelowych” i „wierzchołek początkowy” $v_0 \in V$.

Dla danej instancji Gry w Zwiedzanie rozgrywka przebiega następująco. Na początku gry znajdujemy się w punkcie v_0 . W każdym kolejnym ruchu:

- jeśli aktualnie znajdujemy się w jakimś wierzchołku $w \in S$ to możemy się przesunąć do wybranego przez nas wierzchołka w' , takiego że $E(w, w')$, ale tylko jeśli nie byliśmy jeszcze w tym w' ;
- jeśli aktualnie znajdujemy się w jakimś wierzchołku $w \in V \setminus S$ to Zły Przewodnik może nas przesunąć do wybranego przez siebie wierzchołka w' , takiego że $E(w, w')$, ale tylko jeśli nie byliśmy jeszcze w tym w' .

Gra kończy się gdy osoba mająca ruch nie może go, zgodnie z powyższą regułą, zrobić. Jeśli wcześniej odwiedziliśmy wszystkie wierzchołki ze zbioru T to gra kończy się naszym zwycięstwem. W przeciwnym razie kończy się naszą porażką.

Przez GwZ oznaczamy problem rozstrzygnięcia, dla danej instancji Gry w Zwiedzanie, który z graczy ma w tej instancji strategię wygrywającą. Przez GwZ_k oznaczamy problem GwZ ograniczony do instancji w których $|S|, |T| \leq k$.

Zadanie 215. Pokaż, że problem GwZ jest PSPACE zupełny.

Zadanie 216. Pokaż, że dla żadnej liczby naturalnej k (dla ustalenia uwagi możesz myśleć, że $k=7$), problem GwZ_k nie jest PSPACE-zupełny.

Uwaga: Zakładamy, że $NP \neq PSPACE$.

Zadanie 217. Pokaż, że dla żadnej liczby naturalnej k , problem GwZ_k nie jest w PTIME.

Uwaga: Zakładamy, że $NP \neq PTIME$.

Zadanie 218. W niedziele, które są słusznie wolne od zakupów, mieszkańcy Melmak zajmują się złożonością obliczeniową.

Nieuchronnie jednak, pewne konwencje notacyjne, które przyjęli, są inne od ziemskich. Tam, gdzie Ziemianie piszą ψ^* (gdzie ψ jest wyrażeniem regularnym) na Melmak pisze się $\psi^{[**...]}$, gdzie liczba gwiazdek jest równa $2^{|\psi|}$, gdzie $|\psi|$ jest długością wyrażenia ψ . Tam zaś, gdzie Ziemianie piszą $\bar{\psi}$ (gdzie ψ jest wyrażeniem regularnym) na Melmak pisze się $\psi^{[dd...d]}$, gdzie liczba literek d jest równa $2^{|\psi|}$, gdzie $|\psi|$ jest długością wyrażenia ψ .

a. Jaka jest na Melmak złożoność problemu totalności wyrażeń regularnych?

b. Jaka jest na Melmak złożoność problemu totalności wyrażeń regularnych z dopełnieniem?

W kolejnych dwóch zadaniach używamy następujących oznaczeń.

Dla formuły rachunku zdań $\phi(p_1, p_2 \dots p_n, q_1, q_2, \dots q_n)$ o $2n$ zmiennych przez G_ϕ oznaczamy następujący graf skierowany $\langle V, E \rangle$:

- $V = \{0, 1\}^n$;
- $E([x_1, x_2, \dots x_n], [y_1, y_2, \dots y_n])$ wtedy i tylko wtedy gdy $\phi(x_1, x_2, \dots x_n, y_1, y_2, \dots y_n)$ jest prawdziwa.

Przez 0^n (i 1^n) oznaczamy n elementowe ciągi składające się z samych zer (z samych jedynek).

Przez *problem skompresowanej osiągalności* będziemy rozumieli problem rozstrzygnięcia, dla danej formuły ϕ o $2n$ zmiennych, czy w grafie G_ϕ jest skierowana ścieżka z 0^n do 1^n .

Zadanie 219. Czy problem skompresowanej osiągalności jest w PSPACE?

Zadanie 220. Czy problem skompresowanej osiągalności jest PSPACE-trudny?

19 Tematem ostatnich zadań są Klasy Alternujące.

Definicja. Powiemy, że język A należy do klasy altPTIME jeśli istnieją wielomian p i język $B \in PTIME$ takie, że zachodzi równoważność:

$w \in A$ wtedy i tylko wtedy, gdy gracz pierwszy ma strategię wygrywającą w opisanej poniżej grze $Gra(w, p, B)$

$Gra(w, p, B)$ ma następujące reguły. Zaczyna się od napisania na taśmie słowa $s_1 = w\#$. Następnie, w rundzie i -tej, najpierw gracz pierwszy dopisuje do aktualnie zapisanego słowa s_i wybrany przez siebie sufix $w_i\#$ a następnie gracz drugi dopisuje do $s_i w_i\#$ pewien wybrany przez siebie sufix $v_i\#$, tworząc w ten sposób słowo s_{i+1} . Żąda się przy tym aby długości w_i i v_i były obie równe $p(n)$, gdzie n jest długością słowa w . Gracz pierwszy wygrywa gdy $s_{p(n)} \in B$.

Zadanie 221. (za 2 punkty) Udowodnij, że altPTIME=PSPACE.

Definicja. Powiemy, że język A należy do klasy altPSPACE jeśli istnieją wielomian p oraz języki $B, C \in PTIME$ takie, że zachodzi równoważność:

$w \in A$ w.t.w. gdy gracz pierwszy ma strategię wygrywającą w opisanej poniżej grze $Gra2(w, p, B, C)$.

$Gra2(w, p, B, C)$ ma następujące reguły. Zaczyna się od napisania na taśmie słowa $w\#$. Następnie, w pierwszej rundzie, najpierw gracz pierwszy dopisuje do $w\#$ wybrany przez siebie sufix $w_1\#$ tworząc w ten sposób $t_1 = w\#w_1\#$, a następnie gracz drugi dopisuje do t_1 pewien wybrany przez siebie sufix $v_1\#$, tworząc w ten sposób słowo s_1 . W i -tej rundzie najpierw gracz pierwszy wymazuje z taśmy słowo w_{i-1} zastępując je wybranym przez siebie w_i (powstałe w ten sposób słowo nazywamy t_i), a następnie drugi gracz wymazuje z taśmy słowo v_{i-1} zastępując je przez v_i . Powstałe słowo (równe $w\#w_i\#v_i$) oznaczamy przez s_i . Żąda się przy tym aby długości w_i i v_i były obie równe $p(n)$, gdzie n jest długością słowa w . Gra kończy się porażką pierwszego gracza, gdy w którejś rundzie pojawi się słowo t_i takie, że $t_i \notin B$. Gra kończy się porażką drugiego gracza, gdy w którejś rundzie pojawi się słowo s_i takie że $s_i \notin C$.

Zadanie 222. Udowodnij, że jeśli któryś z uczestników ma strategię wygrywającą w grze $Gra2(w, p, B, C)$, to może doprowadzić do zwycięstwa nie dalej, niż po wykładniczej względem długości w liczbie rund.

Zadanie 223. (za 2 punkty) Udowodnij, że $altPSPACE \subseteq EXPTIME$

Zadanie 224. (za 3 punkty) Udowodnij, że $EXPTIME \subseteq altPSPACE$

Zadanie 225. (za 3 punkty) Instancja Gry w Kamienie to: skończony zbiór X (zwany zbiorem pól), relacja $R \subseteq X^3$, zbiór $Y \subseteq X$ i element $f \in X$.

Grę toczą dwaj gracze wykonujący na przemian ruchy. Przed każdym ruchem na niektórych polach ze zbioru X znajdują się kamienie: przed pierwszym ruchem pierwszego gracza mamy po jednym kamieniu w każdym polu zbioru Y , który to zbiór wyznacza w ten sposób pozycję początkową w grze. W swoim ruchu każdy z graczy przesuwa jeden kamień zgodnie z regułą, że jeśli zachodzi $R(x, y, z)$ oraz w x i y są kamienie, a w z nie ma kamienia, to wolno przesunąć kamień z x do z . Wygrywa ten, kto pierwszy postawi kamień w f .

Jaka jest złożoność problemu: dana instancja gry w kamienie, czy gracz pierwszy ma strategię wygrywającą?