Około dwustu łatwych zadań z języków formalnych i złożoności obliczeniowej*
(i jedno chyba trudne)

Jerzy Marcinkowski

luty 2017

1 Deterministyczne Automaty Skończone

Zadanie 1. Let $L = \{w0s : |s| = 9\}$ be the language of all the words over the alphabet $\{0, 1\}$ whose 10th symbol from the end equals 0. Prove that every DFA deciding L has at least 1024 states.

Zadanie 2. What is the minimal possible number of states of a Deterministic Finite Automaton deciding the language consisting of all the words over $\{a, b, c\}$ which have all the symbols from alphabet occurring among their last three symbols.

Zadanie 3. What is the minimal possible number of states of a Deterministic Finite Automaton deciding the language consisting of all the words over $\{a, b, c\}$ which have at least 4 symbols, and whose last 4 symbols are all identical?

Zadanie 4. Udowodnij, że język $L = \{a^n b^{2n} : n \in \mathbb{N}\}$ nie jest regularny.

Zadanie 5. (worth 2 points) Let L be any subset of $L(0^*)$. Show that L^* is regular.

Zadanie 6. Prove that language L consisting of those words over $\{0, 1\}$ which – when read as numbers written in binary – are primes, is not regular.

Definicja. For a word w over some alphabet let w^R mean "w read backwards", that is $\varepsilon^R = \varepsilon$ and $(aw)^R = w^R a$ for each a from the alphabet and for any word w.

Zadanie 7. Is the language $L = \{ww^R x : w, x \in \{0, 1\}^* \text{ i } w, x \neq \varepsilon\}$ regular? Is the language $L = \{xwx : w, x \in \{0, 1\}^* \text{ i } x \neq \varepsilon\}$ regular?

Zadanie 8. Rozważmy alfabet A_n składający się z liter $a, b_1, b_2, \ldots, b_n$. Niech język L_n^1 składa się z tych słów nad A_n, które mają parzystą liczbę wystąpień wzorca $b_1 b_2$. Niech język L_n^2 składa się z tych słów nad A_n, które mają parzystą liczbę wystąpień wzorca $b_2 b_3$, itd. Niech wreszcie język L_n^n składa się z tych słów nad A_n, które mają parzystą liczbę wystąpień wzorca $b_n b_1$. Zdefiniujmy język L_n jako przecięcie $L_n^1 \cap L_n^2 \cap \ldots \cap L_n^n$. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający L_n?

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

2 Twierdzenie o indeksie

Zadanie 9. (**Myhill-Nerode theorem**) Let $L \subseteq \mathcal{A}^*$. Relation $\sim_L \subseteq \mathcal{A}^* \times \mathcal{A}^*$ is defined as follows: $w \sim_L w'$ if and only if $\forall v \in \mathcal{A}^*$ $(wv \in L \Leftrightarrow w'v \in L)$. Prove the following *Myhill-Nerode theorem*: L is regular if and only if the number k of equivalence classes with respect to $\sim_L$ is finite. The minimal possible number of states of a DFA deciding L is then equal to k.

Let Σ be a finite alphabet and let $L \subseteq \Sigma^*$. Recall that $\sim_L$ from *Myhill-Nerode Theorem* is defined, on Σ^*, as: $w \sim_L v$ if and only if $\forall x \in \Sigma^*$ $(wx \in L \Leftrightarrow vx \in L)$. In a similar manner we can define another equivalence, deonoted as $\sim_L^{inf}$: let $w \sim_L^{inf} v$ if and only if $\forall x, y \in \Sigma^*$ $(xwy \in L \Leftrightarrow xvy \in L)$.

Let i_L be equal to $|\Sigma^*/\sim_L|$ (so that i_L is the number of equivalence classes of $\sim_L$ in Σ^*). Analogously, let $i_L^{inf} = |\Sigma^*/\sim_L^{inf}|$.

Next three problems concern the relation between numbers i_L and i_L^{inf}.

Zadanie 10. Show that if any of the numbers i_L, i_L^{inf} is finite then both are finite. More precisely:

a. prove that $i_L \leqslant i_L^{inf}$;

b. prove that $i_L^{inf} \leqslant i_L^{i_L}$.

Zadanie 11. Show that the estimations from the previous problem cannot be improved. More precisely:

a. Let $\Sigma = \{a, b, c\}$. Show that for each finite set Q there exists a minimal DFA A, whose set of states is Q and with δ as its transition function, such that for every function $f : Q \longrightarrow Q$ there is $w \in \Sigma^*$ such that for each $q \in Q$ it holds that $\delta(q, w) = f(q)$. By a minimal automaton we mean an automaton whose each state is reachable from the initial state, and such that for each pair $q \neq q'$ of its states there exists word w such that exactly one of the states $\delta(q, w)$, $\delta(q', w)$ is accepting.

b. Assuming **a.** show that for every natural n there exists language L such that $i_L \leqslant n$ but $n^n \leqslant i_L^{inf}$.

Zadanie 12. Prove that if $|\Sigma| = 1$ then $i_L^{inf} = i_L$.

3 Regular expressions

Zadanie 13. Skonstruuj automat skończony rozpoznający i wyrażenie regularne definiujące, nad alfabetem $\{0, 1\}$, język słów, które zawierają tyle samo jedynek co zer i w których każdym prefiksie liczba zer różni się co najwyżej o dwa od liczby jedynek.

Zadanie 14. Construct a DFA which recognizes, and a regular expression that denotes, the language of all words (over the alphabet $\{a, b\}$) which do not contain the pattern *baba*.

Zadanie 15. A possibility to use $\cap$ as one more operator allowed in regular expressions (with the obvious semantics) would not lead to greater *expressive power*: still only regular sets would be definable. But the expressions would be more succinct. Prove that using $\cap$ can make a regular expression exponentially shorter. *Hint: consider the language consisting of a single word* $(\ldots((a_0a_1)^2a_2)^2\ldots)^2$.

Zadanie 16. Construct a DFA which recognizes, and a regular expression that denotes, the language of all words (over the alphabet $\{a, b, c, d\}$) which have the same number of a's and b's and the same number of c's and d's and such that in every prefix the numbers of a's and b's differ by at most one and the numbers of c's and d's differ by at most one.

Zadanie 17. Does there exist a regular expression ϕ, such that $L_{a\phi} = L_{\phi b}$? Does there exist a regular expression ϕ, such that $L_{a^*\phi} = L_{\phi b^*}$?

4 Deterministic regular expressions

Deterministic regular expressions a.k.a. *unambiguous regular expressions* appear, most probably by mistake, in the definition of the XML standard.

Definition. Let ϕ be a regular expression over $\mathcal{A}$ and let w be a word over $\mathcal{A}$. Let f be a function whose arguments are occurrences of symbols from $\mathcal{A}$ in w (that is – "subsequent symbols of w") and whose values are occurrences of symbols from $\mathcal{A}$ in ϕ. We say that$f$ is a **correct mapping** from w to ϕ if one of the following conditions holds:

1. ϕ is a word over $\mathcal{A}$, $\phi = w$ and f is the identity mapping or $\phi = w = \varepsilon$;
2. $\phi = \phi_1 + \phi_2$ and f is a correct mapping from w to ϕ_1 or f is a correct mapping from w to ϕ_2;
3. $\phi = \phi_1\phi_2,\ w = w_1w_2$ and f restricted to w_1 is a correct mapping from w_1 to ϕ_1 and f restricted to w_2 is a correct mapping from w_2 to ϕ_2;
4. $\phi = \psi^*,\ w = w_1w_2 \ldots w_k$ for some $k \geqslant 0$ and for each $1 \leqslant i \leqslant k$ function f restricted to w_i is a correct mapping from w_i to ψ.

Intuitively, a correct mapping assigns, to each symbol a in w the symbol of ϕ which "produced" a.

Regular expression ϕ is a **deterministic regular expression**, if for each $w \in L_\phi$ there exists exactly one correct mapping from w to ϕ.

Having a deterministic regular expression we can tell which symbols in the word were produced by which symbols in the expression but only **after** the whole word is known.

This leads to the definition of a deterministic on-line regular expression: regular expression ϕ is **deterministic on-line** if for each $ww_1,\ ww_2 \in L_\phi$ and each f_1, f_2 being correct mappings of (respectively) ww_1 and ww_2 to ϕ, functions f_1 i f_2 are equal on w.

Zadanie 18. A. Which of the following expressions are deterministic and which are deterministic 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. Write a deterministic regular expression denoting the set of all the words over $\{0, 1\}$ which contain the pattern 101.

Zadanie 19. Can every regular language be defined by a deterministic on-line regular expression?

Zadanie 20. Write a deterministic on-line regular expression denoting the set of all the words over $\{0, 1\}$ which contain exactly one or two symbols 1.

5 Nondeterministic Finite Automata

Zadanie 21. Construct a Nondeterministic Finite Automaton (NFA) deciding the language of words over $\{0,1\}^*$ which, as a natural number written in binary, are divisible by 5. The number is being read
a) beginning from the most significant bit,
b) beginning from the least significant bit.

Zadanie 22. Prove that if for some language L there exists a NFA deciding L then there also exists a NFA deciding $L^R = \{w : w^R \in L\}$

Zadanie 23. Suppose L is regular. Show that $\{w : \exists n \in \mathbb{N}\ \exists v \in L\ w^n = v\}$ is also regular. By w^n we mean w repeated n times.

Zadanie 24. Show that if L_1 i L_2 are regular languaes over some $\mathcal{A}$ then also $L_1 \cup L_2$, $L_1 \cap L_2$, i $\mathcal{A}^* - L_1$ are regular.

Zadanie 25. (worth 2 points) Suppose L is regular. Does it imply that $L/2 = \{w : \exists v\ vw \in L\ \wedge |v| = |w|\}$ is also regular?

Zadanie 26. Podaj algorytm rozstrzygający, dla danych dwóch niedtereministycznych automatów skończonych czy języki rozpoznawane przez te automaty są równe.

Zadanie 27. The number of states of a minimal DFA deciding a language $L \subset \Sigma^*$ is always equal to the number of states of a minimal DFA deciding $\Sigma^* \setminus L$. This is no longer true for Nondeterministic automata. Prove that there exists language L which can be decided by a NFA having less than 20 states, but whose complement cannot be decided by any NFA with less than 200 states. *Hint: consider* Σ *being a singleton.*

Zadanie 28. 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 nideterministycznego 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 $p \geqslant 5$ 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 $\{1, 2, \ldots (p-1)/2\}$, przy czym liczby czytamy „od prawej", czyli od najmniej znaczącego bitu (to znaczy pierwszy znak słowa jest ostatnią cyfrą liczby).

Zadanie 29. Czy istnieje niedeterministyczny automat skończony o mniej niż $p+3$ stanach rozpoznający język L_p?

Zadanie 30. Czy istnieje deterministyczny automat skończony o mniej niż $2p$ stanach rozpoznający język L_p?

Zadanie 31. Suppose $L \in \{0,1\}^*$ is regular. Does it imply that

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

is also regular?

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 32. 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 33. 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 $1 \leqslant k \leqslant n/2$ 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?

Let us define, by induction, the following ternary relation $\ominus$, on words over the alphabet Σ, which will be needed in the next 3 problems:

- $\ominus(\varepsilon, \varepsilon, \varepsilon)$
- $\ominus(aw, v, ax)$ if $\ominus(w, v, x)$
- $\ominus(w, av, ax)$ if $\ominus(w, v, x)$ where $a \in \Sigma$ and $w, v, x \in \Sigma^*$

For a language $L \subseteq \Sigma^*$ and word $w \in \Sigma^*$ define:
$L_{\exists w} = \{x \in \Sigma^* : \exists y \in \Sigma^* \ (\ominus(w, x, y) \wedge y \in L)\}$
$L_{\forall w} = \{x \in \Sigma^* : \forall y \in \Sigma^* \ (\ominus(w, x, y) \Rightarrow y \in L)\}$

Zadanie 34. Suppose L is regular and w is a fixed word from Σ^*. Does it follow that $L_{\forall w}$ is regular?

Zadanie 35. Suppose L is regular and w is a fixed word from Σ^*. Does it follow that $L_{\exists w}$ is regular?

Zadanie 36. Show that for each natural k there exists language $L \subseteq \{a, b, c\}^*$ which can be recognized by a deterministic finite automaton with less than kK^2 states and such that $L_{\exists c^k}$ cannot be recognized by any DFA with less than 2^{kK} states, where$K = 2^k$.

Hint: It may me useful to remember that the sum of n initial primes is always smaller than $n^2 \log n$ while their product is always greater than $2^{n \log n}$.

6 Around Černy Conjecture

Next three problems concern a conjecture by Černy, which has been an open problem for about 50 years now. The conjecture states that if the set $sync(Q)$ is non-empty then it contains some word whose length is not greater than $(|Q|-1)^2$ (an automaton is known, with non-empty $sync(Q)$, such that the shortest word in $sync(Q)$ has length exactly $(|Q|-1)^2)$.

Definition. For a given DFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ and set $S \subseteq Q$ let $sync(S)$be the set $\{w \in \Sigma^* : \forall q, q' \in S \;\; \hat{\delta}(q,w) = \hat{\delta}(q',w)\}$. Notice that this definition does not depend on q_0 and F.

Zadanie 37. $L \subseteq \Sigma^*$ is a *regular ideal* if it is regular and if for each $w \in L$ and each $v, v' \in \Sigma^*$ it holds that $vwv' \in L$.

a. Is it true that for every DFA $\mathcal{A}\langle \pm, \mathcal{Q}, \amalg_{\prime}, \mathcal{F}, \delta \rangle$ and every $S \subseteq Q$ the set $sync(S)$ is regular?

b. Is it true that for every DFA $\mathcal{A}\langle \pm, \mathcal{Q}, \amalg_{\prime}, \mathcal{F}, \delta \rangle$ and every $S \subseteq Q$ the set $sync(S)$ is a regular ideal?

c. Is it true that for every DFA $\mathcal{A}\langle \pm, \mathcal{Q}, \amalg_{\prime}, \mathcal{F}, \delta \rangle$ the set $sync(Q)$ is a regular ideal?

Zadanie 38. **a.** Show that if S consists of two elements and $sync(S)$ is non-empty then there is some word not longer than $|Q|^2$ in $sync(S)$.

b. Show that if $sync(Q)$ is non-empty then it contains some word which is not longer than $|Q|^3$. *Hint: use* **a.**

Zadanie 39. Show that for every natural n which is large enough there exists a DFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$, with $\Sigma = \{a, b\}$, $n = |Q|$, and a set $S \subseteq Q$ consisting of two elements, such that $sync(S)$ is non-empty, but it does not contain a word shorter than $n^2/4$.

W kolejnych trzech 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 $csync(S)$ („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 40. Załóżmy, że dla każdego dwuelementowego $S \subseteq Q$ zbiór $csync(S)$ jest niepusty. Czy wynika z tego, że $csync(Q)$ jest niepusty?

Zadanie 41. Niech $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ będzie PDFA.

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

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

Zadanie 42. 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 trzylelementowy $S \subseteq Q$ 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 $2k$ znajdzie 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 Automatic relations

Let us define the function $l : \{0, 1\}^* \to \mathbb{N}$ as: $l(\varepsilon) = 0$, $l(0w) = 2l(w)$, $l(1w) = 2l(w) + 1$.
For a natural k define $\Sigma_k = \{0, 1\}^k$.
For natural numbers $j \leqslant k$ let us define the function $\Pi_k^j : \Sigma_k^* \to \{0, 1\}^*$ as: $\Pi_k^j(\varepsilon) = \varepsilon$, $\Pi_k^j(\langle a_1, a_2, \ldots a_j, \ldots a_k\rangle w) = a_j \Pi_k^j(w)$, where $\langle a_1, a_2, \ldots a_j, \ldots a_k\rangle \in \Sigma_k$.

Relation $R \subseteq \mathbb{N}^k$ will be called *automatic*, if the language L_R consisting of such words $w \in \Sigma_k^*$ for which $R(l(\Pi_k^1(w)), l(\Pi_k^2(w)), \ldots l(\Pi_k^k(w)))$ is regular.

Zadanie 43. Is addition an automatic relation? By addition we mean the relation $\{\langle a, b, c\rangle \in \mathbb{N}^3 : a + b = c\}$.

Zadanie 44. Is multiplication an automatic relation automatic? By multiplication we mean the relation $\{\langle a, b, c\rangle \in \mathbb{N}^3 : ab = c\}$.

Zadanie 45. Show that a projection of an automatic relation is also automatic. In other words, if $R \subseteq \mathbb{N}^k$ is automatic then also the relation $R' = \{r \in \mathbb{N}^{k-1} : \exists m \in \mathbb{N}\ \langle r, m\rangle \in R\}$ is an automatic relation (for simplicity you can assume that $k = 2$).

8 Gramatyki bezkontekstowe i automaty ze stosem

Zadanie 46. Zbuduj automat ze stosem rozpoznający język *dobrze rozstawionych nawiasów dwóch rodzajów* generowany przez gramatykę:
$S \to SS|(S)|[S]|\varepsilon$
która ma jeden symbol nieterminalny S i cztery symbole terminalne $(,), [,]$.

Zadanie 47. Construct a context free grammar that generates the language:
$L = \{w \in \{0, 1\}^* : |w|_0 = 2|w|_1 \wedge |w|_1 \text{ is even}\}$.

Zadanie 48. Is the language $L = \{w \in \{0, 1\}^* : |w|_0 \leqslant |w|_1 \leqslant 2|w|_0\}$ context-free?

Zadanie 49. Is the language $L = \{w \in \{0, 1\}^* : \exists n \in \mathbb{N}\ 2n|w|_0 \leqslant |w|_1 \leqslant (2n + 1)|w|_0\}$ context-free?

Zadanie 50. Podaj algorytm rozstrzygający dla danej gramatyki bezkontekstowej G, czy $L(G)$ jest niepusty.

Zadanie 51. Prove that $L \subseteq \{0\}^*$ is context-free if and only if it is regular.

Zadanie 52. Niech G będzie gramatyką generująca 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 53. Is the language $L_3 = \{w \in \{0,1,2\}^* : \neg\exists x \in \{0,1,2\}^*\ w = xx\}$ context-free?

Zadanie 54. Is the complement of the language L_3 from the previous problem, which is $L_4 = \{w \in \{0,1,2\}^* : \exists x \in \{0,1,2\}^*\ w = xx\}$ a context free language?
Hint: (1) Consider the language $L_4 \cap L$ where $L = L_{0^*10^*10^*10^*1}$.
(2) Use the pumping lemma, and remember that the partition $w = sztyx$, whose existence follows from the lemma, is such that $|zty| \leqslant c$, where c is the constant from the lemma.

Zadanie 55. Zbuduj $NDPDA$ i gramatykę bezkontekstową G dla języka $\{0,1\}^* - \{www : w \in \{0,1\}^*\}$.

Zadanie 56. (Za 3 punkty, bardzo trudne) 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 57. Czy język L złożony z tych wszystkich słów nad alfabetem $\{0,1\}$, które są postaci wvw, dla pewnych słów w, v, takich że $|w| = |v|$ jest bezkontekstowy?

Zadanie 58. Czy język L będący dopełnieniem języka L z poprzedniego zadania do $\{0,1\}^*$ jest bezkontekstowy?

Zadanie 59. Czy język $L = \{vww : v, w \in \{a,b,c\}^*, w \neq \varepsilon\}$ jest bezkontekstowy?

Zadanie 60. Niech $L_=$ będzie językiem tych słów nad alfabetem $\{0,1\}$, które mają tyle samo zer co jedynek, a L_R niech będzie językiem wszystkich palindromów. Czy przekrój $L_=$ z dopełnieniem L_R jest językiem bezkontekstowym? (mamy tu na myśli dopełnienie do $\{0,1\}^*$)

Zadanie 61. Niech $L_=$ będzie językiem tych słów nad alfabetem $\{0,1\}$, które mają tyle samo zer co jedynek, a L_R niech będzie językiem wszystkich palindromów. Czy przekrój $L_=$ z L_R jest językiem bezkontekstowym?

Zadanie 62. Czy język $L = \{0^n 1^{n^2} : n \in \mathbb{N}\}$ jest bezkontekstowy?

Zadanie 63. Let L be the language consisting of the words over $\{0,1\}$ which are of even length and which have at least as many symbols 1 in the first half as in the second half. Is L context free?

9 Więcej zadań o językach regularnych i bezkontekstowych

Zadanie 64. 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 automatow 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.*

Zadanie 65. (2 points) *Shuffle* is defined in this problem as the smallest ternary relation over $\mathcal{A}^*$, where $\mathcal{A}$ is a finite alphabet, which satisfies the following conditions:

- Shuffle of the empty word with the empty word is the empty word;
- If w is a shuffle of a word s with a word t, then it is also a shuffle of t with s;
- If $v = at$, $a \in \mathcal{A}$ and w is a shuffle of a word s with a word t then aw is a shuffle of s with v.

For two languages $L_1, L_2 \subseteq \mathcal{A}^*$ define their shuffle as the set of all w which are a shuffle of some $s \in L_1$ with some $t \in L_2$.
Is shuffle of two regular languages always regular?
Is shuffle of two context-free languages always context-free?
?

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

Zadanie 66. Pokaż, że jeśli L jest regularny, to Lustro(L) również jest regularny.

Zadanie 67. Pokaż, że deterministyczny automat skończony rozpoznający język Lustro(L) może potrzebować liczby stanów wykładniczo większej niż deterministyczny automat skończony rozpoznający język L.

Zadanie 68. Czy teza Zadania 66. pozostanie prawdziwa, jeśli oba wystąpienia słowa „regularny” zmienimy w nim na „bezkontekstowy”?

Let $L_{\forall w}$ i $L_{\forall w}$ be defined as in Problems 35 – 36.

Zadanie 69. Suppose L is CFL, and w is a fixed word from Σ^*. Does it imply that $L_{\forall w}$ is CFL?

Zadanie 70. Suppose L is CFL, and w is a fixed word from Σ^*. Does it imply that $L_{\exists w}$ is CFL?

Przez *język rodzynkowy* 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 rodzynkowy 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^iba^j, gdzie $0 \leqslant i, j \leqslant 2n$ oraz $|i - j| \leqslant 1$. Udowodnij, że $i(L_n^*)$ szacuje się (z dokładnością do stałej multiplikatywnej) przez n^2 .

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

Zadanie 73. Udowodnij, że dla każdego naturalnego n istnieje język rodzynkowy L_n, taki że $i(L_nL_n) \geqslant c2^i(L_n)$, gdzie c jest pewną stałą niezależną od n. Jeśli nie potrafisz pokazać takiego wykładniczego dolnego ograniczenia na wzrost $i(L_nL_n)$, to dostaniesz punkty również za inne ograniczenie, jeśli nie będzie mniejsze niż $ci(L_n)^3$.

In the following 3 problems a language $A \subseteq \Sigma^*$ will be called *confluent* if:

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

Language $A \subseteq \Sigma^*$ will be called *uniformly confluent*, if there exists constant $c \in \mathbb{N}$ such that:

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

Zadanie 74. Is every regular language confluent?
Is every confluent language regular?

Zadanie 75. Prove that if a regular language is confluent then it is uniformly confluent.

Zadanie 76. Show that there exists a confluent context-free language which is not uniformly confluent.

9.1 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 $f_T(wa) = (f_T(w))\sigma(\hat{\delta}(wa, q_0))$.

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

• 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 $A \leqslant_{reg} B$ jeśli istnieje transducer T (Moore'a lub Mealy'ego) taki że dla kazdego $w \in \Sigma^*$ zachodzi $w \in A$ w.t.w. gdy $f_T(w) \in B$.

Zadanie 77. 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 78. Pokaż że jeśli $A \leqslant_{reg} B$ i B jest regularny to A też.

Zadanie 79. 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 80. Niech $A \subseteq \{(,), [,], \langle, \rangle\}^*$ będzie językiem poprawnie rozstawionych nawiasów trzech rodzajów zaś $B \subseteq \{(,), [,]\}^*$ językiem poprawnie rozstawionych nawiasów dwóch rodzajów. Pokaż że $A \leqslant_{reg} B$. *Wskazówka: każde słowo produkowane przez σ ma się składać z dwóch symboli.*

10 Recursive sets and recursive functions

Zadanie 81. Modify the definition of a recursive set so that one can also talk about recursive sets of pairs of natural numbers. Show that if a set $\mathcal{A} \subseteq \mathbb{N}^2$ is recursive then the set $\{n : \exists m\ [n, m] \in A\}$, which is the projection of A on the first axis, is recursively enumerable.

Zadanie 82. Show that each recursively enumerable set is a projection of some recursive set. In other words, if B is r.e. then there exists such a recursive $A \subseteq \mathbb{N}^2$ that $B = \{n : \exists m\ [n, m] \in A\}$.

Zadanie 83. Repeat the proof (presented during the lecture) that the Halting Problem is undecidable, that is that the set $K = \{n : \phi_n(n) \in \mathbb{N}\}$ is not recursive.

Zadanie 84. Show that $\{n : |Dom(\phi_n)| \geqslant 7\}$ is recursively enumerable.

Zadanie 85. Suppose A, B, C, D are recursive enumerable sets, such that each natural number is in exactly two of them. Show that they are all recursive.

Zadanie 86. Suppose ϕ is a nondecreasing total recursive function, Show that the set $\phi(\mathbb{N})$ of the values of ϕ is recursive. Is the totality assumption necessary?

Zadanie 87. Show that each non-empty recursively enumerable set is equal to $\phi(\mathbb{N})$ for some total recursive function ϕ.

Zadanie 88. Show that each infinite set which is recursively enumerable is equal to $\phi(\mathbb{N})$ for some total recursive function ϕ, which is also 1-1.

Zadanie 89. Show that the set $\{n : Dom(\phi_n) = \mathbb{N}\}$ is not r.e.

Zadanie 90. (Long, worth 2 points) Suppose f is a total recursive function. Which of the following implications always hold true?

- if A is recursive then $f(A)$ also is;
- if A is recursive then $f^{-1}(A)$ also is;
- if A is r.e., then $f(A)$ also is;
- if A is r.e., then $f^{-1}(A)$ also is.

How would the answers change if f was allowed to be a partial function?

Zadanie 91. Nie korzystając z tw. Rice'a udowodnij, że zbiór $B = \{n : Dom(\phi_n)$ i $\mathbb{N} - Dom(\phi_n)$ są nieskończone$\}$ nie jest rekurencyjny.

Zadanie 92. Show that the set $B = \{n : Dom(\phi_n)$ i $\mathbb{N} - Dom(\phi_n)$ są nieskończone$\}$ is not r.e.

Zadanie 93. Show that the set of all the numbers of the programs which terminate for all arguments except possibly a finite number, is not r.e.

Zadanie 94. Udowodnij podane na wykładzie twierdzenie Rice'a.

Zadanie 95. Poniższe zbiory nie są rozstrzygalne:

1. zbiór numerów tych maszyn Turinga, które obliczają funkcje o dziedzinie różnej od $\mathbb{N}$;
2. zbiór numerów tych maszyn Turinga, które obliczają funkcje całkowitą i których czas działania jest rosnący względem rozmiaru danych;

3. zbiór numerów tych maszyn Turinga, których czas działania dla żadnych danych nie jest liczbą pierwszą;

4. zbiór numerów tych maszyn Turinga, które obliczają funkcje częściowe, których wartościami są wyłącznie liczby pierwsze.

Nierozstrzygalność których z nich daje się udowodnić wprost z twierdzenia Rice'a?

Zadanie 96. Udowodnij nierozstrzygalność zbioru z punktu 2. poprzedniego zadania.

Zadanie 97. Let $A, B \subseteq \mathbb{N}$. By $A \leqslant_{rek} B$ we mean that there exists a total recursive function f (called *a reduction*), such that $f(x) \in B$ if and only if $x \in A$. Prove that for each two sets $A, B \subseteq \mathbb{N}$ there exists their least upper bound with respect to $\leqslant_{rek}$. By this we mean $C \subseteq \mathbb{N}$ such that:
i) $A \leqslant_{rek} C$ and $B \leqslant_{rek} C$,
ii) if D is such that $A \leqslant_{rek} D$ and $B \leqslant_{rek} D$ then $C \leqslant_{rek} D$.

Zadanie 98. Is it true that $K \leqslant_{rek} \overline{K}$? Is it true that $\overline{K} \leqslant_{rek} K$?

Zadanie 99. Let T be the set of all pairs $\langle n, m\rangle$ for which ϕ_n and ϕ_m are the same partial function.
i) Show that T is not r.e..
ii) Is the complement of T r.e.?

Zadanie 100 (Arithmetic hierarchy)**.** Let $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ be some fixed computable bijection. Denote the class of all recursive sets as Σ_0. For each Σ_i let $\Pi_i = \{A \subseteq \mathbb{N} : \mathbb{N}\backslash A \in \Sigma_i\}$, and define $A \in \Sigma_{i+1}$ if there exists $B \in \Pi_i$ such that $A = \{n \in \mathbb{N} : \exists m\ f(n, m) \in B\}$.

Let L be the set of numbers of all the programs whose domain is nonempty and finite. What is the smallest i for which $L \in \Sigma_i$?

Zadanie 101. Let $A, B \subseteq \mathbb{N}$. Let f be a reduction being a witness for $A \leqslant_{rek} B$. Suppose f is „onto" (i.e. $f(\mathbb{N}) = \mathbb{N}$). Prove that $B \leqslant_{rek} A$.

Zadanie 102. A subset A of $\mathbb{N}$ is co-r.e. if $\mathbb{N} \setminus A$ is r.e. by initial segment of $\mathbb{N}$ we mean any set of the form $\{1, 2, \ldots n\}$ for $n \in \mathbb{N}$. Let B be the set of all numbers of the programs whose domains are initial segments of $\mathbb{N}$.
a. Is B co-r.e.?
b. Show that there exists a set of triples of natural numbers which is co-r.e. and whose projection on the first axis equals B. *Remark. First one of course needs to silently extend the definition of co-r.e to sets of triples.*

Zadanie 103. Let function f be total recursive. Which of the following conditions implies that $f(\mathbb{N})$ is recursive?
a. There exists a finite $A \subseteq \mathbb{N}$ such that if $f(i) > f(i + 1)$ then $i + 1 \in A$.
b. There exists a finite $A \subseteq \mathbb{N}$ such that if $f(i) > f(i + 1)$ then $f(i + 1) \in A$.

Zadanie 104. Let $\mathcal{D} \subseteq \mathcal{P}(\mathbb{N})$. Prove that the following two conditions are equivalent:

- $\mathcal{D}$ is countable;
- there exists $B \subseteq \mathbb{N}$ such that for each $A \in \mathcal{D}$ it holds that $A \leqslant_{rek} B$.

Hint: For a fixed total recursive function f and set $B \subseteq \mathbb{N}$, how many different sets $A \subseteq \mathbb{N}$ can exist, such that f is the reduction witnessing that $A \leqslant_{rek} B$?

Zadanie 105. Let Tot be $\{n \in \mathbb{N} : Dom(M_n) = \mathbb{N}\}$, and let $Nemp$ be $\{n \in \mathbb{N} : Dom(M_n) \neq \emptyset\}$.

(a) Is it true that $Nemp \leqslant_{rek} Tot$?

(b) Is it true that $Tot \leqslant_{rek} Nemp$?

Zadanie 106. Is every partial recursive function a subset of some total recursive function?

Zadanie 107. Does every infinite subset of $\mathbb{N}$ contain, as its subset, some infinite recursively enumerable set? *Hint: clever diagonalization.*

11 Maszyny Turinga

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

Zadanie 108. Explain why replacing the infinite tape in the definition of Turing Machine by infinite plane grid would not change the set of computable functions.

Zadanie 109. Skonstruuj maszynę Turinga rozpoznającą język $A = \{ww^R : w \in \{0,1\}^*\}$

Zadanie 110 (Minsky machine)**.** (2 points) **a.** Observe that the halting problem for machines looking like a two-way two-stack automata is undecidable. More precisely, the instance of the problem is a set of instructions for an automaton with two stacks (no input tape) and we ask whether the computation of this automaton, starting from two empty stacks and state q_0, will terminate.

b. Using **a.** show that analogous problem remains undecidable if we consider automata with 4 counters instead of 2 stacks (a counter is a stack whose stack alphabet consists of one element).

c. Using **b.** show that analogous problem remains undecidable for automata with just 2 counters.

Zadanie 111. A *scanning Turing Machine* will be defined (in this problem) as a tuple $\langle \Sigma, Q, q_0, q_F, \delta \rangle$, where Σ is a finite tape alphabet, Q is a finite set of states, $q_0, q_F \in Q$ are (respectively) the initial and final states, and $\delta : (Q n \{q_F\}) \times \Sigma \to Q \times (\Sigma n \{B\})$ is the transition function.

In the initial configuration the head of the machine is in the state q_0 over the first symbol of the input word.

If, in some state q the head is over some symbol $a \in \Sigma$, then it changes its state to q' and replaces a with a', where $\delta(q, a) = (q', a')$. Then the head is moved by one cell to the right. Unless a was a blank, then the head is moved (in the current state) to the first symbol on the tape, so that it can move to the right again and so on. The computation halts once the state q_F is reached.

Is it decidable whether a given scanning Turing Machine M will halt, on a given input word w?

Zadanie 112. 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."

12 Nierozstrzygalność. Kanoniczne zadania.

Zadanie 113. Powtórz, podany na wykładzie, dowód nierozstrzygalności Problemu Odpowiedniości Posta.

Zadanie 114. For a CFG G let L_G be the language generated by G.

Use undecidability of the Post Correspondence Problem to show that the set of pairs G, H of grammars, such that $L_G \cap L_H \neq \emptyset$ is not recursive. Is it r.e.?

Zadanie 115. (for 2 points) Show that there is no algorithm deciding, for a given CFG G and alphabet A, whether $A^* = L(G)$

Zadanie 116. Does there exist an algorithm deciding, for grammars G i H, whether $L(G) = L(H)$?

Zadanie 117. Show that it is an undecidable problem to tell, for a given Thue process Π and word w whether the set $A_w = \{v : w \overset{*\Pi}{\leftrightarrow} v\}$ is finite.

Zadanie 118. (for 2 points) Consider a finite set P of pairs of words and a binary relation $\rightarrow$ defined as follows: $w \rightarrow_P\ v$ if and only if there exists a pair $\langle l, r\rangle \in P$ such that $w = lx$ and $v = xr$ where x is some word. Let $\overset{*}{\rightarrow}_P$ be the transitive closure of $\rightarrow_P$.

Is the problem: *given* P, x, y, *does it hold that* $x \overset{*}{\rightarrow}_P y$*?* decidable?

Zadanie 119. (hard, for 2 points) Function $f : \mathbb{N} \rightarrow \mathbb{N}$ is called a Conway function if there exist natural numbers $p, a_1, b_1, a_2, b_2, \ldots, a_p, b_p$ such that for each n if $n = k \ mod\ p$ then $f(n) = na_k/b_k$. Show that there is no algorithm which for given $p, a_1, b_1, a_2, b_2, \ldots, a_p, b_p$ would answer, whether for the Conway function f, defined by these numbers, there exists m such that $f^m(2) = 1$ where f^m is f composed m times with itself.

Zadanie 120. (very important, for 2 points) Prove that the following problem is undecidable. The instance of the problem consists of a finite set C of colors, containing at least colors: *red* i *white*, and a set $N \subseteq C^4$ of *ugly quadruples* . We have infinitely many square tiles of each color (each tile is 1×1). Does there exist a square (at least 2×2) which can be tiled in such a way that both the tiles in the left upper corner and the left lower corner are red, the remaining tiles of the top row and the bottom row are white and that nowhere in the square an ugly quadruple of tiles occurs, which means four tiles:

c_1	c_2
c_3	c_4

such that $(c_1, c_2, c_3, c_4) \in N$.

During the lecture we mentioned undecidability of **Hilbert's 10th Problem** (call it H10). It is the problem of deciding, for a given system of diophantine equations (this means equations between polynomials, with many variables and with coefficients being integers) whether there is a solution in natural numbers.

Zadanie 121. Let H10prim be the problem of deciding whether a given system of diophantine equations has a solution in the integers.

Show that H10prim$\leqslant_{rek}$H10. Show that H10$\leqslant_{rek}$H10prim.

Zadanie 122. Let H10bis be H10 restricted to equations in which each polynomial has degree at most two. Is H10bis still undecidable?

Hint (for this problem and the previous one): In one of the problems you may want to use the fact that each natural number is a sum of for squares of natural numbers.

13 Undecidability. More problems.

Zadanie 123. Show that the problem whether a given set of diophantine equations, like in H10, has a solution in odd integers is undecidable. You can of course assume undecidability of H10.

Zadanie 124. Czy istnieje algorytm rozstrzygający, dla danej gramatyki bezkontekstowej G, czy istnieje słowo w, takie że $ww^Rw \in L(G)$?

Zadanie 125. Dla danych funkcji $f, g, h : \{0, 1, \ldots, p-1\} \to \{0, 1, \ldots, p-1\}$ i danego nieskończonego ciągu liczb naturalnych $(a_1, a_2, a_3, \ldots)$, niech $F_{f,g,h}(a_1, a_2, \ldots)$ 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 $(b_1, b_2, \ldots, b_k)$ i $(c_1, c_2, \ldots, c_k)$. Czy istnieje liczba iteracji n taka, że $F^n_{f,g,h}(b_1, b_2, \ldots, b_k, 0, 0, 0, \ldots) = (c_1, c_2, \ldots, c_k, 0, 0, 0, \ldots)$? jest nierozstrzygalny.

Zadanie 126. 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 127. By a *context-free grammar with context-sensitive erasing* we will (in this problem) mean a "grammar", like a CFG, but having additional productions of the form $w \to \varepsilon$, where w is a word consisting of non-terminals and ε is – as always – the empty word.

By the *vanishing problem* we will mean (in this problem) the problem of deciding, for given context-free grammar with context-sensitive erasing with the initial symbol S and the set of productions Π, whether $S \xrightarrow{*}_{\Pi} \varepsilon$.

Show tha the vanishing problem is undecidable.

Zadanie 128. A Thue semi-process Π will be called context-free, if for each pair $[w, v] \in \Pi$ the word w consists of exactly one letter. Is the word problem for context-free semi-processes decidable?

Zadanie 129. A Thue semi-process Π will be called almost context-free, if for each pair $[w, v] \in \Pi$ one of the words w consists of exactly one letter and another consists of exactly two letters. Is the word problem for almost context-free semi-processes decidable?
Remark: the terminology we use in this problem, the previous one and the next, has nothing to do with any standard.

Zadanie 130. A Thue semi-process Π over $\{0, 1\}$ will be called nice if for each pair $\langle l, r\rangle \in \Pi$ there is $|l|_1 = |r|_1$ (which means that number of 1's on the left hand side of each production equals to the number of 1's on the right hand side of the same production). Prove that teh problem whether for given word w and given nice semi-Thue process Π there is $1111 \xrightarrow{*}_{\Pi} w$ is undecidable. *Hint: typical and simple.*

Zadanie 131. In this problem we will consider what we call *nondeterministic bipedal automata*. Such an automaton is defined as a tuple $\langle \Sigma, Q, q_0, F, \delta \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $q_o \in Q$ is the initial state, $F \subseteq Q$ is the set of accepting states, and $\delta \subseteq Q \times \Sigma \times \Sigma \times Q \times \{0,1\} \times \{0,1\}$ is the transition relation.

This is how the transition relation is understood: if $\delta(q, a_1, a_2, q', b_1, b_2)$, then the automaton, being in the state q, with its left leg over a_1, and right leg over a_2, ca change the state into q', move his left leg b_1 cells right and move his right leg b_2 cells right.

This automaton takes as an input a pair w_l, w_p of words, each of them of the form $a(\Sigma \setminus \{a, z\})^* z$, for some fixed symbols $a, z \in \Sigma$.

At the initial configuration the automaton is in state q_0 with its left leg on the first symbol (i.e. a) of word w_l, and with its right leg on the first symbol (a) of w_p. Automaton accepts a pair of words, if it is possible for it to reach a configuration with both legs on symbols s and state $q \in F$.

Is totality of nondeterministic bipedal automata decidable? Automaton is total if it accepts each pair of words.

Zadanie 132. Przez *funkcję liniową* będziemy w tym zadaniu rozumieć funkcję postaci $h(x) = ax + b$, gdzie a, b są liczbami naturalnymi.

Instancją problemu skaczącej pchły będzie w tym zadaniu skończony ciąg funkcji liniowych $= \langle f_1, g_1, \dots f_k, g_k \rangle$. Powiemy, że *instancja* $\mathbf{c} = \langle f_1, g_1, \dots f_k, g_k \rangle$ *problemu skaczącej pchły ma rozwiązanie* jeśli istnieje niepusty ciąg $s_1, s_2, \dots s_l$ liczb ze zbioru $\{1, 2, \dots k\}$ taki, że punkt $\langle f_{s_l} f_{s_{l-1}} \dots f_{s_1}(0), g_{s_l} g_{s_{l-1}} \dots g_{s_1}(0) \rangle$ leży na prostej $y = x$ (zatem wyobrażamy sobie, że pchła zaczyna skakać w punkcie $\langle 0, 0 \rangle$, w każdym kolejnym skoku umie przemieścić się z punktu $\langle x, y \rangle$ do dowolnego spośród $\langle f_i(x), g_i(y) \rangle$; pytamy o to, czy potrafi kiedykolwiek znów znaleźć się w punkcie o obu współrzędnych równych).

Pokaż, że problem skaczącej pchły, to znaczy problem czy dla danej instancji istnieje rozwiązanie, jest nierozstrzygalny.

Zadanie 133. Let $\phi(x, y)$ be a formula – with two free variables – of arithmetics of natural numbers, with addition and multiplication.

Write a sentence ψ of of arithmetics of natural numbers, with addition and multiplication which is true if and only if there exists number $l \geqslant 1$ and a finite sequence $a_1, a_2, \dots a_l$ of natural numbers such that $a_1 = 1$, $a_l = 2$ and for each $1 \leqslant i \leqslant l - 1$ it holds that $\phi(a_i, a_{i+1})$.

Hint: Chinese Remainder Theorem.

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

Zadanie 135. Trzy asynchroniczne żuczki. Pokaż, że z tezy Zadania 134 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 żuczkach.

W kolejnych dwóch zadaniach rozważamy płaszczyznę, po której biega k psów. Startują one jednocześnie z początku układu współrzędnych, a następnie każdy z nich, w każdej jednostce czasu, przesuwa się o jednostkę odległości na północ, południe, wschód lub zachód (zatem po każdym takim kroku psy znajdują się w punktach kratowych płaszczyzny). O kierunku kolejnego ruchu psa decyduje jego funkcja przejścia (psy są różne, i mogą mieć różne funkcje przejścia). Argumentami funkcji przejścia są: aktualny stan psa (każdy pies ma skończoną liczbę stanów) oraz zapach punktu kratowego aktualnie zajmowanego przez psa. Przez zapach pola rozumiemy tu informację o tym, które psy odwiedziły już to pole.

Wśród stanów każdego psa wyróżniamy jeden stan szczekający.

Mówiąc bardziej formalnie, funkcja przejścia i-tego psa δ_i ma typ $\delta_i : Q_i \times Z \to Q_i \times \{N, S, E, W\}$, gdzie Q_i to skończony zbiór stanów i-tego psa, zaś $Z = \mathcal{P}(\{1, \ldots k\})$ jest zbiorem zapachów (czyli rodziną wszystkich pozdbiorów zbioru wszystkich biegających psów). Funkcja ruchu i-tego psa $\hat{\delta}_i$ definiowana jest przy pomocy δ_i w naturalny dla teorii automatów skończonych sposób. Wreszcie zapach $z(p, t)$ punktu kratowego p w chwili t jest równy $z(p, t-1) \cup S(p, t-1)$, gdzie $S(p, t)$ to zbiór numerów psów znajdujących się w punkcie p w chwili t. W chwili zerowej zapachem wszystkich punktów jest zapach pusty.

Problem szczeku dla układu k psów jest następujący: dane funkcje przejścia k psów, oraz informacja o tym które stany są szczekające. Czy któryś z biegających zgodnie z tymi funkcjami psów osiągnie kiedyś stan szczekający?

Zadanie 136. Czy problem szczeku dla układu 3 psów jest rozstrzygalny?

Zadanie 137. Czy problem szczeku dla układu z jednym psem jest rozstrzygalny?

Zadanie 138. Czy problem niepustości języka $L_G \cap L_G L_G$, dla danej gramatyki bezkontekstowej G, jest rozstrzygalny?

Zadanie 139. 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, uruchomić[1] 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 czegokolwiek o wskazaniach stoperów). Pokaż, że problem totalności dla automatów skończonych z dwoma stoperami jest nierozstrzygalny.

14 Niedeterministyczne maszyny Turinga i klasa NP

Zadanie 140. Pokaż, że wielomianową maszynę niedeterministyczną można przerobić tak, aby zgadywała rozwiązanie wcześniej niż pozna dane. Dokładniej rzecz ujmując, udowodnij

[1] Z wartością równą zero.

że jeśli zbiór A należy do klasy NP, to istnieją wielomiany p, q oraz nideterministyczna 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 141. Pokaż, że jeśli zbiór $A \subseteq \mathbb{N}^2$ jest w P i p jest wielomianem, to zbiór $\{n : \exists m \, |m| \leqslant p(|n|) \text{ i } [n, m] \in A\}$, czyli rodzaj rzutu A na pierwszą oś, jest w NP.

Zadanie 142. 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 $B = \{n : \exists m \, |m| \leqslant p(|n|) \text{ i } [n, m] \in A\}$.

15 Polynomial reductions and NP-hardness

Zadanie 143. Show that 5SAT$\leqslant_p$3SAT.

Zadanie 144. (for 2 points) An instance of the STASI[2] problem is an undirected graph and natural number k. We are asked whether it is possible to place k sniches in some of the vertices of this graph in such a way that every vertex without a snitch has at least one vertex with a snitch as its neighbour. Show that 3SAT$\leqslant_p$STASI.

Hint: It is not a hard problem. The idea is similar to the one behind the proof of the fact that 3SAT$\leqslant_p$3COL.

Zadanie 145. (for 2 points) Let H denote the Hamiltonicity problem for undirected graphs (i.e the set of all the undirected graphs which contain a cycle visiting each node exactly once)

Let H_d denote the same problem for directed graphs. Prove that $H \leqslant_p H_d$ and that $H_d \leqslant_p H$.

Hint: For the harder direction replace each vertex with three new.

Zadanie 146. (for 2 points) A clause is called a *Horn clause* if it has at most one non-negated literal. Show that HORNSAT, which is the satisfiability problem for formulas in the CNF form which are conjunctions of Horn clauses, is in PTIME.

Zadanie 147. (for 2 points, not so easy) Prove that 2SAT, which is the satisfiability problem for 2CNF formulae, is in PTIME.

Zadanie 148. Prove that 3SAT$\leqslant_p$3SAT$_3$. By 3SAT$_3$ we mean 3SAT restricted to formulae in which no variable occurs more than 3 times.

Zadanie 149. (for 2 points) Prove that Hamiltonicity is NP-complete.

Zadanie 150. The Travelling Salesman Problem (TSP) is defined as follows. The instance is a complete undirected graph $G = \langle V, E \rangle$, a cost function d from E to $\mathbb{N}$ and a number $k \in \mathbb{N}$. The question is whether there exists a Hamiltonian cycle $C \subset E$ in G such that the "total cost of this cycle", that is the sum of all $d(e)$ for $e \in C$, is not greater than k.

Prove that TSP is NP-complete. You can assume that Hamiltonicity is NP-complete.

Comment: TSP wikipedia article contains a section: "TSP in popular culture".

[2]This problem is usually called "dominating set problem". The formulation I use here dates back to the 1990s when it was fashionable to poke fun at GDR (do you know what GDR was?). We believed at that time that Poland had been different. No longer we do.

Zadanie 151. Assume that there exists a polynomial algorithm finding, for a given instance of TSP, a cycle at most twice as expensive as the optimal one. Prove that, under this assumption, PTIME=NP.

Hint: NP-completeness of Hamiltonicity will be useful.

Zadanie 152. Show that if we consider only instances of TSP satisfying the triangle inequality (that is such that for all verticies v_1, v_2, v_3 there is $d(v_1, v_2) + d(v_2, v_3) \geqslant d(v_1, v_3)$, then there exists a polynomial algorithm finding a Hamiltonian cycle at most twice as expensive as the optimal one.

Zadanie 153. What is the complexity of SAT_2, by which we mean satisfiability of CNF formulae with each variable occuring at most twice?

Zadanie 154. Show that the problem whether there exists, in a given graph with n verticies, a clique of at least $n/2$ verticies, is NP-complete.

Zadanie 155. (for 2 points) For a given graph G let $c(G)$ be the maximal size of a clique in G. Assume there exists a polynomial algorithm finding, in a given graph G, a clique consisting of at least $c(G)/2$ verticies[3]. Show that, under this assumption, there also exists a polynomial algorithm finding, in a given graph G, a clique consisting of at least $c(G)/\sqrt{2}$ verticies.

Zadanie 156. Rozważmy następujący *problem spełnialności dla zdecydowanej większości klauzul*: Dane różne klauzule rachunku zdań $\phi_1, \phi_2, \ldots, \phi_n$. Czy można podstawić wartości 0 i 1 za zmienne zdaniowe tak aby więcej niż 9/10 spośród klauzul $\phi_1, \phi_2, \ldots, \phi_n$ przyjęła wartość logiczną 1? Udowodnij, że *problem spełnialności dla zdecydowanej większości klauzul* jest NP-zupełny. Przypominam, że klauzulą nazywamy formułę postaci $l_1 \vee l_2 \vee \ldots \vee l_k$, gdzie l_i są literałami, to znaczy zmiennymi zdaniowymi lub ich negacjami.

Wskazówka: Skorzystaj z NP-zupełności SAT.

Zadanie 157. Let CLIQUE_c be the problem whether there exists, in a given graph with n verticies, a clique containing at least n/c verticies. Show that for each $c, c' > 0$ there is $\mathrm{CLIQUE}_c \leqslant_p \mathrm{CLIQUE}_{c'}$.

Zadanie 158. Rozważmy następujący *problem smutnych strażników.* Dany jest pewien zbiór strażników $s_1, s_2, \ldots, s_l$. Strzegą oni obiektów $a_1, a_2, \ldots, a_k$. Odbywa się to tak, że każdy strażnik s_i ma w swoim kantorku dwa ekrany telewizyjne E_i i F_i i na każdym z tych ekranów widzi, za pośrednictwem nieruchomej kamery, pewien niezmienny zbiór obiektów (odpowiednio Z_{E_i} i Z_{F_i}, zbiory te oczywiście niekoniecznie muszą być parami rozłączne). Powiemy, że *strażników można rozweselić* jeśli da się przestawić u każdego z nich jeden telewizor na kanał gdzie akurat transmitują mecz, ale w taki sposób, że każdy obiekt ze zbioru $\{a_1, a_2, \ldots, a_k\}$ pozostaje pod obserwacją na jakimś nieprzestawionym telewizorze.

Pokaż, że problem rozstrzygnięcia, dla danego przykładu problemu smutnych strażników, czy strażników da się rozweselić, jest NP-zupełny.

Zadanie 159. Udowodnij, że następujący *problem podziału n harcerzy na 4 drużyny* jest NP-zupełny.

Dany jest hufiec H harcerzy i lista $E \subseteq H^2$ par harcerzy, którzy się nie lubią. Czy da się podzielić H na cztery drużyny tak, aby spełnione były warunki:
drużyny mają być z grubsza równej wielkości: do każdej z nich musi należeć przynajmniej jedna piąta wszystskich harcerzy z H. W żadnej drużynie nie mogą jednocześnie znaleźć się dwaj harcerze, którzy się nie lubią.

[3]We do not know whether such algorithm exists.

Zadanie 160. W Pewnej Wschodniej Krainie wszystkim rządzą trzej gangsterzy, pan K, pan G i pan B. Każda firma, która chce mieć spokój i dostawać koncesje i kontrakty, musi mieć wśród członków swojej rady nadzorczej przyjaciół przynajmniej dwóch spośród tych trzech gangsterów. Kłopot polega jednak na tym, że: gangsterzy za sobą nawzajem nie przepadają, więc każdy członek rady może przyjaźnić się co najwyżej z jednym gangsterem. Rady nadzorcze różnych firm niekoniecznie muszą być rozłączne.

Udowodnij, że problem: *Dane listy członków rad nadzorczych pewnej liczby firm, czy da się osoby figurujące na tych listach pozaprzyjaźniać z panami K, G i B w taki sposób, aby wszystkie z tych firm miały spokój?*

Jest NP-zupełny.

Zadanie 161. Jaka jest złożoność następującego *problemu klasy udającej się na wycieczkę.* Klasa ma udać się na wycieczkę do Świeradowa. Jednak ze względu na trawiący ją wewnętrzny konflikt, niektórzy z młodych ludzi obwarowują kwestię swojego wyjazdu pewnymi warunkami. Warunki te mają następującą postać:

Ja (jadę — nie jadę) jeśli X (jedzie — nie jedzie)

gdzie X przebiega zbiór uczniów klasy. Każdy uczeń może przedstawić Pani Wychowawczyni dowolną liczbę takich warunków. Jaka jest złożoność problemu sprawdzenia, czy da się zorganizować wycieczkę w sposób uwzględniający wszystkie postawione warunki? Wielkością zadania jest tu łączna liczba warunków.

Zadanie 162. 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.

Zadanie 163. An instance of the Disjoint Set Cover Problem (DSCP) is a finite family $A = \{A_1, A_2, \ldots, A_N\}$ of finite sets. $A \in DSCP$ if there exists a family $B \subseteq A$ of pairwise disjoint sets, such that the union of all sets from B equals the union of all sets from A. Show that 3SAT$\leqslant_p$DSCP.

Hint: show that $3SAT_3 \leqslant_p PZPR$ gdzie $3SAT_3$.

Zadanie 164. What is the complexity of the problem whether ther exists, for a given CNF formula, a valuation under which in every clause of the formula all literals have the same value?

Zadanie 165. Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomially computable bijection. Does this imply that f^{-1} is also a polynomially computable bijection?

Zadanie 166. Prove that TSP remains NP-complete even if we restrict attention only to instances that satisfy Strong Triangle Inequality: $d(x,y) < d(x,z) + d(z,y)$ for each verticies x, y, z.

Zadanie 167. Rozważmy następujący problem P_{z8}. Dane: graf nieskierowany $G = (V, E)$ i $A \subseteq V$. Czy można wybrać zbiór $B \subseteq A$ tak aby:

1. zbiór B był dominujący w G, tzn. dla każdego $x \in V \setminus B$ istniał $y \in B$ taki, że $E(x,y)$;
2. zbiór B był niezależny w G, tzn. dla żadnych $x, y \in B$ nie zachodziło $E(x,y)$;
3. dla każdego $x \in V$ istniał co najwyżej jeden $y \in B$ taki, że $E(x,y)$

Pokaż, że problem P_{z8} jest NP-zupełny.

Zadanie 168. 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 (Wschodniej) Krainie 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 żadanych 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 $KOAL^i_j$ oznaczmy wariant problemu $KOAL$, w którym każda partia może żądać co najwyżej i stanowisk, a każdego stanowiska żąda co najwyżej j partii (brak któregość z indeksów oznacza, że nie ograniczamy tego parametru).

Zadanie 169. 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 170. Jaka jest złożoność problemu $KOAL^3_3$?

Zadanie 171. 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 172. What is the complexity of 3COL if erstricted to graphs of degree at most 4?

Zadanie 173. Rozważmy następujący Problem Trójek Klasowych (PTK).

Rodzice uczniów każdej klasy w szkole wybierają spośród siebie (do różnych niezmiernie ważnych zadań) trójkę – zwaną trójką klasową. Ponieważ można być rodzicem więcej niż jednego dziecka w szkole, można być członkiem więcej niż jednej takiej trójki.

Spośród członków tych trójek dyrektor szkoły chce powołać szkolny komitet rodzicielski, do którego należy **dokładnie jeden** członek każdej trójki klasowej. PTK jest problemem rozstrzygnięcia czy, dla danej listy trójek klasowych, powołanie takiego komitetu jest możliwe.

Udowodnij, konstruując odpowiednią redukcję, że 3COL$\leqslant_P$PTK.

Zadanie 174. Jaka będzie złożoność problemu z poprzedniego zadania, jeśli każde wystąpienie słowa *trójka* zamienimy w nim słowem *dwójka*?

Zadanie 175. Przez *liczbę chromatyczną grafu nieskierowanego* $G = \langle V, E\rangle$ rozumiemy najmniejszą liczbę naturalną n dla której istnieje funkcja $l : V \to \{1, 2, \ldots n\}$ 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 PTIME=NP

By a *cycle cover* of a graph G we mean a set of pairwise vertex-disjoint cycles which are subgraphs of G and jointly contain all vertices of G. *Cardinality of a cycle cover* is the number of cycles this cover consists of. For a graph G by $\sigma(G)$ we denote (in next 3 problems) the minimal cardinality of a cycle cover of G (a vertex is always a cycle itself, so $\sigma(G)$ is always defined and never is greater than the number of verticies of G).

Zadanie 176. Let $c > 3$. What is the complexity of deciding, for given G, whether $\sigma(G) \leqslant n/c$, where n is the number of verticies of G?

Zadanie 177. Suppose there exists a polynomial algorithm which, for a given graph G such that $\sigma(G) = 1$, returns a cycle cover of cardinality at most 2. Show that in such case P=NP.

Zadanie 178. Prove that if P does not equal NP then no polynomial algorithm can approximate $\sigma(G)$ up to a multiplicative constant. More precisely, show that in such case no polynomial algorithm and $c > 0$ can exist, such that for each graph G this algorithm run on G would return a cycle cover of cardinality at most $c\sigma(G)$.

Zadanie 179. 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).

Niech $G = \langle V, E\rangle$ będzie grafem nieskierowanym. Przypominam, że kliką w G nazywamy zbiór $U \subseteq V$ taki, że $\forall x, y \in U\ \{x, y\} \in E$, zaś zbiorem niezależnym w G nazywamy zbiór $U \subseteq V$ taki, że $\forall x, y \in U\ \{x, y\} \notin E$.

W poniższych dwóch zadaniach wolno korzystać ze wszystkiego, czego w czasie wykładu i ćwiczeń dowiedzieliście się o złożoności różnych wersji problemu spełnialności formuł boolowskich. Nie wolno natomiast korzystać z zadań o złożoności problemu kliki, ani innych problemów grafowych.

Zadanie 180. a. Jaka jest złożoność problemu rozstrzygnięcia, dla danego grafu $G = \langle V, E\rangle$, czy istnieje klika $U \subseteq V$ taka, że $V \setminus U$ jest zbiorem niezależnym?

b. Jak zmieni się złożoność problemu z punktu **a.**, jeżeli dodatkowo zażądamy, aby moc U była dokładnie połową mocy V?

Zadanie 181. Jaka jest złożoność problemu rozstrzygnięcia, dla danego grafu $G = \langle V, E\rangle$, czy istnieją klika $U \subseteq V$ i zbiór niezależny $Y \subseteq V$, oba o mocy nie mniejszej niż jedna czwarta mocy V.

Zadanie 182. The *Funny 2-COL* problem will be defined as follows. Its instance is a undirected graph. The question is whether its vertices can be colored, using two colors, in such a way that each vertex will have at least one neighbor of each color. Show that Funny 2-COL is NP-complete.

Hint: Use NP completeness of not-all-equal-3-SAT.

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

Zadanie 184. 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 185. 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 186. What is the complexity of deciding, for a given 2CNF formula, whether there exista a valuation satisfying ar least 3/4 of the clauses of this formula?

Zadanie 187. 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 188. Melmażelon can answer, always correctly, one question concerning satisfiability of a boolean formula. More precisely, melmażelon swallows a formula and, if this formula is satisfiable it becomes green and if it is unsatisfiable it becomes orange. Then, in both cases it dies.

Denote by PTIME^M the class of problems which can be solved in deterministic polynomial time using (at most) one melmażelon (for each instance), which means problems for which there exists a polynomial algorithm asking at most one query to a melmażelon, and then using the information provided by the creature.

Is PTIME^M = NP$\cup$ co-NP? We assume that co-NP$\neq$ NP $\neq$ PTIME.

Zadanie 189. An instance of the NAE-SAT (Not All Equal SAT) problem is a boolean formula ϕ in CNF. We say that ϕ $\in$NAE-SAT if there exists a valuation of variables satisfying every clause, but such that at least one literal remains false in each clause.

Prove that 3COL$\leqslant_P$ NAE-SAT.

Zadanie 190. What is the complexity of the problem of existence of a coloring of the vertices of given graph, with two colors, such that no three vertices forming a 3-clique in this graph get the same color?

Zadanie 191. Czy odpowiedź na pytanie z poprzedniego zadania zmieni się, jeśli ograniczymy się do instancji problemu będącymi grafami 4-kolorowalnymi ?

16 On theoretical problems of cryptography

Zadanie 192. (for 2 points) A bijection $f : \mathbb{N} \to \mathbb{N}$ such that for each n there is $|n| = |f(n)|$ is *one-way* if there exists a polynomial algorithm computing f, but there is no polynomial algorithm computing f^{-1}. Show that is any one-way function exists then NP $\cap$ co-NP $\neq$ PTIME.

Hint: consider the set $\{\langle x, y\rangle : f^{-1}(x) < y\}$

17 Problemy być może nie należące do klasy NP

Zadanie 193. 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 $\geqslant 2$), 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?

Wskazówka: The owls are not what they seem.

Zadanie 194. Show that the problem whether a regular expression denotes all the words over a given alphabet is in PSPACE.

Zadanie 195. Prove that the problem of deciding whether a formula of the form

$$\exists! x_1 \exists! x_2 \ldots \exists! x_n \; \phi$$

is true is in PSPACE. Variables $x_1, x_2, \ldots, x_l$ range over the set $\{0, 1, 2\}$. The quantifier $\exists!$ means *there exists exactly one* and ϕ is a quantifier-free formula with n variables x_1, $x_2 \ldots$,x_n, built with boolean connectives, symbols of addition and multiplication modulo 3, equality (modulo 3) and constants 0,1,2. How would your reasoning change if the veriables ranged over $\{0, 1\}$ and the arithmetic was modulo 2?

Zadanie 196. Patrokles, mając daną formułę boolowską ϕ, taką że $Var(\phi) = \{p_1, p_2, \ldots p_n\}$ próbuje ją spełnić. W tym celu wartościuje zmienną p_1, potem zmienną p_2 itd. Ale gdy rozważa zwartościowanie kolejnej zmiennej, niech to będzie p_k, przerwać mu może Mojra, i rzec: *pozwól kolego że p_k to ja zwartościuję 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 zwartościować kolejną zmienną) w sumie najwyżej trzy razy?
- Mojra może mu przerwać ile razy chce?

Zadanie 197. (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: Naturalna hipoteza jest prawdziwa. Tylko jak to udowodnić?*

Zadanie 198. 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 199. (za 2 punkty) Udowodnij, że są języki rekurencyjne, które nie są w PSPACE.

Zadanie 200. Udowodnij, że problem sprawdzenia prawdziwości formuł o postaci

$$Q_1x_1Q_2x_2 \dots Q_nx_n \ \phi(x_1, x_2, \dots, x_n)$$

gdzie każdy Q_i to albo $\exists!$, oznaczający *istnieje dokładnie jeden*, albo $\exists$ (czyli *istnieje*), albo $\forall$ (czyli *dla każdego*), zaś ϕ jest fromułą boolowską, jest PSPACE-zupełny.

The topic of the next four beautiful problems are **Alternating classes.**

Definicja. Language A is defined to be in altPTIME if there exist a polynomial p and language $B \in PTIME$ such that the following equivalence holds true:

$w \in A$ if and only if Player One has a winning strategy in $Game(w, p, B)$, as described below.

$Game(w, p, B)$ is played according to the following rules. It begins from the word $s_1 = w\#$ being written on the tape. Then, in move i, first Player One adds, to the current word s_i a suffix $w_i\#$ and then Player Two adds to $s_iw_i\#$ a suffix $v_i\#$, creating in this way the word s_{i+1}. Lenghts of w_i and v_i must both be equal to $p(n)$, where n is the length of w. Player One wins if and only if $s_{p(n)} \in B$.

Zadanie 201. (for 2 points) Prove that altPTIME=PSPACE.

Definicja. Language A is defined to be in altPSPACE if there exist a polynomial p and languages $B, C \in PTIME$ such that the following equivalence holds true:

$w \in A$ if and only if Player One has a winning strategy in $Game2(w, p, B, C)$, as described below.

$Game2(w, p, B, C)$ is played according to the following rules. It begins from the word $w\#$ being written on the tape. Then, in move one, first Player One adds to $w\#$ a suffix $w_1\#$, creating in this way $t_1 = w\#w_1\#$, and then Player Two adds to t_1 some suffix $v_1\#$, creating in this way a new word s_1. In move i first Player One erases from the tape the word w_{i-1} and replaces it with some w_i (the word that emerges in this way is called t_i), and then Player Two erases from the tape the word v_{i-1} and replaces it with some v_i. This leads to a new word $s_i = w\#w_i\#v_i$). The lenghts of each w_i and v_i must be equal to $p(n)$, where n is the length of w. Game ends, and Player One loses when $t_i \notin B$ at some point. Game ends, and Player Two loses when $s_i \notin C$ at some point.

Zadanie 202. Show that if one of the players has a winning strategy in $Gra2(w, p, B, C)$, then she can win after a number of moves at most exponential with respect to the length of w

Zadanie 203. (for 2 points) Show that $altPSPACE \subseteq EXPTIME$

Zadanie 204. (for 3 points) Show that $EXPTIME \subseteq altPSPACE$

Zadanie 205. (for 3 points) An instance of the Stones Game is a finite set X (set of *fields*), relation $R \subseteq X^3$, set $Y \subseteq X$ and an element $f \in X$.

There are two players, making moves alternately. Before every move there are stones on some fields from X: before move one there is one stone on each field from Y, so that Y defines the initial position of the game. In a single move, each player can move a stone from x to z if and only if $R(x, y, z)$, there are stones in x and y and there is no stone in z. są kamienie, a w z nie ma kamienia, to wolno przesunąć kamień z x do z. One who moves a stone to f wins.

What is the complexity of deciding, for a given instance of the Stones Game, whether player one has a winning strategy?

Zadanie 206. Imagine a rectangular table, with 17 rows and 5 columns, each field of which can be filled with 0 or 1. Imagine also a boolean formula *Step* with 10 variables. We will say that the table is filled correctly, if for each two consecutive rows $a_1a_2a_3a_4a_5$ and $b_1b_2b_3b_4b_5$ it holds that

$$Step(a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5).$$

Let $k_1k_2k_3k_4k_5$ i $m_1m_2m_3m_4m_5$ be two fixed sequences of zeroes and ones. Write a QBF stating that our table can be filled correctly, in such a way that the first row is $k_1k_2k_3k_4k_5$ and the last row is $m_1m_2m_3m_4m_5$. Your formula cannot have more than 45 variables (not counting $m_1 \dots m_5$ i $k_1 \dots k_5$ which are not variables). Of course you can use *step* as a subformula.

By a QBF we mean here a boolean formula with k variables preceeded by k quantifiers binding the variables. For example $\forall p \exists q\ (p \vee q) \wedge (\neg p \vee q)$ is QBF and $\forall p\ [(\exists q\ (p \vee q)) \wedge (\exists q\ (\neg p \vee q))]$ is not.

Zadanie 207. Write a formula of FOL saying that, in a given graph $\langle V, R\rangle$ there exists a path from given vertex c to given vertex k, consisting of exactly 16 edges. Your formula must not have more than 10 (occurrences of) quantifiers.

Remark: Formula: $\exists x_1\ \exists x_2 \dots \exists x_{15}\ R(c, x_1) \wedge R(x_1, x_2) \wedge \dots \wedge R(x_{15}, k)$ is almost what we expect, except that it has 15 quantifiers.

Zadanie 208. Dla danej formuły ϕ zbudowanej, zgodnie ze zwykłymi regułami, ze zmiennych $x_1, x_2, \dots, x_n$, $y_1, y_2, \dots, y_n$, z liczb naturalnych 0, 1 i 2, ze spójników boolowskich oraz z symboli $=_3$, $+_3$ i $\times_3$ (rozumianych jako równość, dodawanie i mnożenie modulo 3), ale bez kwantyfikatorów, rozważmy następującą grę $G(\phi)$ rozgrywaną między uczestnikami X i Y.

Rozgrywka składa się z n ruchów, w trakcie których zastępuje się zmienne w formule ϕ stałymi. W ruchu i najpierw gracz X wybiera $k \in \{0, 1, 2\}$ i zastępuje wszystkie wystąpienia zmiennej x_i przez liczbę k, a następnie gracz Y wybiera $l \in \{0, 1, 2\}$ i zastępuje wszystkie wystąpienia zmiennej y_i przez liczbę l. Gracz X wygrywa, jeśli formuła bez zmiennych w jaką zmieni się ϕ po ostatnim ruchu gracza Y, jest prawdziwa.

Niech 3GRA będzie problemem rozstrzygnięcia, dla danej formuły ϕ, czy gracz X ma strategię wygrywającą w grze $G(\phi)$. Udowodnij, że QBF$\leqslant_p$3GRA.

Zadanie 209. What is the complexity of QBF restricted to instances with at most two occurrences of a general quantifier?

Zadanie 210. Jaka jest złożoność problemu rozstrzygnięcia, dla danych dwóch deterministycznych automatów skończonych M_1 i M_2, czy język $L_{M_1} \cap L_{M_2}$ jest niepusty? (wielkością zadania jest tu łączna liczba stanów tych automatów).

Zadanie 211. 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).

An instance of the Sightseeing Game (in next three problems) is a directed graph $G = \langle V, E\rangle$, a set $S \subseteq V$ of "our positions", a set $T \subseteq V$ "goal positions" and an "initial vertex" $v_0 \in V$.

For a given instance of the Sightseeing Game the game proceeds as follows. At the beginning we are in v_0. Then, in each subsequent move

- If we are currently in $w \in S$ then we can move to the w' of our choice, provided that $E(w, w')$, and that w' was not so far visited;

- If we are currently in $w \in V \setminus S$ then the Evil Guide can move us to a vertex w' of her choice, provided that $E(w, w')$, and that w' was not visited so far.

Game is over when a move cannot be made according to the above rules. If before that we visited all the vertices in T then we won. Otherwise we lost.

Bu GwZ (Gra w Zwiedzanie) we denote the problem of deciding, for a given instance of Sightseeing Game, whether we have a winning strategy. By GwZ_k we denote GwZ restricted to instances when $|S|, |T| \leqslant k$.

Zadanie 212. Show that GwZ is PSPACE-complete.

Zadanie 213. Show that there is no k (to fix attention you can imagine that k=7), such that GwZ_k is PSPACE-complete.

Remark: Assume that NP $\neq$ PSPACE.

Zadanie 214. Show that there is no k, such that GwZ_k is in PTIME.

Remark: Assume that NP $\neq$ PTIME.

Miasta każdego kraju planety Melmak tworzą, wraz z połączeniami drogowymi, graf nieskierowany. I w każdym z tych krajów panuje następujący obyczaj. Król wyznacza – jeśli to możliwe – każdemu miastu jeden z roboczych dni tygodnia (tzn od poniedziałku do piątku) jako dzień targowy, to znaczy ten, w którym w tym mieście odbywają się targi. Stara się jednak przestrzegać reguły, że *dwa miasta połączone bezpośrednią drogą nie mogą odbywać targów w ten sam dzień, ani nawet w dwa kolejne dni.* Problem rozstrzygnięcia, czy spełnienie powyższego jest możliwe, będziemy nazywać Problemem Króla na Melmak.

Zadanie 215. Udowodnij, że Problem Króla na Melmak jest NP-zupełny.

Zadanie 216. Niech L będzie pewną raz na zawsze ustaloną liczbą naturalną.

Kraj nazywamy *nadmorskim* jeśli jego miasta można ustawić w ciąg $a_1, a_2, \ldots a_n$, taki, że dla każdego $i \in \{1, 2, \ldots n\}$ liczba takich par połączonych bezpośrednią drogą miast $[a_j, a_k]$, że $j \leqslant i \leqslant k$, jest nie większa od L.

Skorzystaj z faktu, że dla każdego ustalonego DFA $\mathcal{A}$, problem czy dla danego słowa w automat $\mathcal{A}$ akceptuje słowo w rozwiązuje się w czasie liniowym, względem długości w, aby skonstruować algorytm szybko rozwiązujący Problem Króla na Melmak ograniczony do krajów nadmorskich.

Zadanie 217. W soboty i w niedziele, które są słusznie wolne od targów, 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 $\psi^{[**\ldots*]}$, 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 $\psi^{[dd\ldots d]}$, 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?