

Zadanie 1 Pokaż, że problem spełnialności układów równań słów jest NP-trudny także w przypadku, gdy $\Sigma = \{a\}$.

Wskazówka: Redukcja z innego typu znanych równań.

Zadanie 2 Pokaż, że problem spełnialności równań słów jest NP-trudny również wtedy, gdy rozważamy systemy w których wszystkie prawe strony nie zawierają zmiennych.

(Uwaga: w tym zadaniu może być prościej, jeśli dopuszczimy podstawianie ϵ za zmienną).

Zadanie 3 Pokaż, że problem spełnialności układu równań słów można sprowadzić do problemu spełnialności jednego równania słów, jeśli możemy dodawać litery do alfabetu. Pokaż też, że dodawanie liter nie jest potrzebne, jeśli wejściowy alfabet zawierał przynajmniej dwie różne litery.

Zadanie 4 Zredukuj spełnialność równań w słowach do spełnialności kubicznych równań w słowach.

Zadanie 5 Pokaż, że spełnialność równania w słowach z więzami bezkontekstowymi jest nierozstrzygalna. (Tzn. pozwalamy na więzy przy użyciu języków bezkontekstowych).

Zadanie 6 Pokaż, że problem niepustości przecięcia języków deterministycznych automatów skończonych:

Problem: Niepustość przecięcia języków deterministycznych automatów skończonych

Wejście: deterministyczne automaty skończone $D_1, D_2, \dots, D_m$

Cel: Rozstrzygnij, czy istnieje słowo akceptowane przez wszystkie te automaty.

jest PSPACE-trudny.

Pokaż, że ten problem jest w PSPACE nawet jeśli wejściowe automaty są niedeterministyczne.

Wywnioskuj z tego, że problem równań słów z więzami regularnymi jest PSPACE-trudny.

Zadanie 7 (Długie i trochę żmudne, choć nietrudne: dwa punkty) Niech $\langle \mathbb{M}, \cdot \rangle$ będzie zbiorem macierzy kwadratowych o wyznaczniku 1 oraz wyrazach naturalnych z operacją mnożenia.

Rozważmy funkcję z $\Sigma = \{a, b\}$ w $\mathbb{M}$, zadaną przez

$$\varphi(a) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } \varphi(b) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Rozszerzamy ją do funkcji z $\langle \Sigma^*, \cdot \rangle$ w $\langle \mathbb{M}, \cdot \rangle$ w naturalny sposób, tj. jako homomorfizm: konkatenacja przechodzi na mnożenie.

Pokaż, że dla każdego $w \in \Sigma^*$ jego obraz ma wyznacznik 1.

Pokaż, że ta funkcja jest różnowartościowa; w tym celu rozważ, jak wyglądają wiersze macierzy $\varphi(a) \begin{bmatrix} n & n' \\ m & m' \end{bmatrix}$, a jak wiersze macierzy $\varphi(b) \begin{bmatrix} n & n' \\ m & m' \end{bmatrix}$. Wywnioskuj z tego, że jeśli $M_w = \varphi(w)$ to z postaci M można wywnioskować, jaka jest pierwsza litera w .

Pokaż, że daną macierz kwadratową $M \in \mathbb{M}$ można przedstawić albo w postaci $\varphi(a)M'$ albo $\varphi(b)M'$, gdzie $M' \in \mathbb{M}$ albo $M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ (znów: popatrz na wiersze, rozważ wyznacznik).

Wywnioskuj z tego, że φ jest izomorfizmem między $\langle \Sigma^*, \cdot \rangle$ a $\langle \mathbb{M}, \cdot \rangle$.

Wywnioskuj z tego, że spełnialność równań w słowach nad $\Sigma = \{a, b\}$ redukuje się do spełnialności równań Diofantycznych (tj. nad liczbami naturalnymi); w tym celu wyraż $\varphi(X)$ jako macierz składającą się z 4 zmiennych naturalnych, zapisz warunek na wyznacznik równy 1.

Zadanie 8 Podaj algorytm przekształcający composition system rozmiaru n na SLP wielomianowego rozmiaru. Jak mały może być stopień tego wielomianu? Jak szybki jest Twój algorytm?

Zadanie 9 Podaj algorytm PSPACE dla poniższego problemu równoważności dwóch composition systems.

Problem: Równoważność Compositions Systems

Wejście: Dwa composition systems $\mathcal{C}$ i $\mathcal{C}'$.

Cel: Rozstrzygnij, czy $\text{val}(\mathcal{C}) = \text{val}(\mathcal{C}')$.

Podaj również algorytm w klasie **coNP** dla tego problemu.

Zadanie 10 Udowodnij, że równanie słów z n wystąpieniami zmiennych i ustalonym rozwiązaniem s ma najwyżej $2n$ różnych par krzyżujących i najwyżej $2n$ różnych liter krzyżujących.

Zadanie 11 Niech s będzie najkrótszym rozwiązaniem równania słów $u = v$. Pokaż że:

- Niech para ab dla $a \neq b$ występuje w $s(u)$. Udowodnij, że ab ma wystąpienie jawne lub krzyżujące w $s(u)$ lub $s(v)$ (dla s).
- Jeśli równanie zawiera stałe, to istnieje rozwiązanie najkrótsze, takie że dla a występującego w $s(u)$ a występuje również w u lub v , tj. ma wystąpienie jawne.
- Niech a^ℓ będzie maksymalnym blokiem w $s(u)$. Pokaż, że a^ℓ ma wystąpienie jawne, krzyżujące lub jest prefiksem/sufiksem $s(X)$ dla jakiejś zmiennej X . (Czyli innymi słowy dotyka cięcia). Może być pomocne patrzenie na $ba^\ell c$ zamiast na samo a^ℓ .

Zadanie 12 Pokaż, że możemy odkrzyżować i skompresować wszystkie bloki liter jednocześnie, tj. w ramach jednej procedury, która wypycha najwyżej jeden prefiks i sufix z jednego wystąpienia zmiennej.

Zadanie 13 Rozważmy słowo $w \in \Sigma^*$ takie, że żadne jego dwie kolejne litery nie są takie same. Wystąpienie pary ab w w jest *pokryte* przez partycję (Σ_1, Σ_2) jeśli $a \in \Sigma_1$ i $b \in \Sigma_2$. Pokaż, że istnieje partycja Σ pokrywająca przynajmniej $\frac{|w|-1}{2}$ liter w w . Podaj liniowy algorytm obliczający taką partycję. (Możesz założyć, że litery w w można sortować w liniowym czasie przy użyciu RadixSort).

Uogólnij tę obserwację na równanie słów z rozwiązaniem s (i nie więcej niż n wystąpieniami zmiennych).

Wskazówka: Sprowadź problem do problemu konstrukcji maksymalnego (ważonego) cięcia w grafie. Istnieje dla niego prosty algorytm zrandomizowany, można go łatwo zderandomizować używając metody warunkowych wartości oczekiwanych. Opis można znaleźć w: Michael Mitzenmacher, Eli Upfal *Probability and computing* jak również w Vijay Vazirani *Approximation algorithms*.

Zadanie 14 Partycję alfabetu Σ jest para alfabetów (Σ_1, Σ_2) t. że $\Sigma_1 \cup \Sigma_2 = \Sigma$ oraz $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Pokaż, że możemy odkrzyżować i skompresować zbiór par $\{a_i b_i\}_{i \in I}$ równoległe, o ile $a_i \in \Sigma_1$ oraz $b_i \in \Sigma_2$ dla każdego $i \in I$.

Zadanie 15 Używając Zadań 12–13 podaj algorytm dla równań słów który przechowuje równanie liniowej (względem wejściowego równania) długości; algorytm może używać dodatkowej pamięci w czasie pracy, ponadto traktujemy bloki a^{cn} tak jakby miały długość 1. (To drugie to pewne oszustwo, ale da się je obejść, dokładne metody poznamy później.)

Zadanie 16 To zadanie jest trochę niedookreślone, ale powinno dać się je zrobić ze wskazówkami w treści.

Pokaż, że dla odpowiednio dużej stałej c , jeśli równanie słów ma długość najwyżej cn^2 oraz najwyżej n wystąpień zmiennych to dla najkrótszego rozwiązania s zachodzi jeden z poniższych warunków:

- istnieje para ab w $s(u)$ która jest niekrzyżująca;
- istnieje a niekrzyżujące, przy czym $s(u)$ zawiera maksymalny blok a^ℓ dla pewnego $\ell > 1$;
- istnieje krzyżujące ab takie że odkrzyżowanie go i skompresowanie pary ab równanie wciąż ma długość nie większą niż cn^2 ;
- istnieje krzyżujące a takie że po odkrzyżowaniu i skompresowaniu jego bloków równanie wciąż ma długość nie większą niż cn^2 .

Wykorzystaj tę obserwację do konstrukcji algorytmu dla równań słów działającego w niedeterministycznej pamięci wielomianowej.

Zadanie 17 (2 punkty) Celem tego zadania jest podanie algorytmu, który wykonuje jedynie kompresję par, w tym dla par aa .

Powodem, dla którego nie możemy użyć kompresji par aa jest nakładanie się par aa , które powoduje niejednoznaczność rozkładu na takie pary; np. w równaniu $aX = Xa$ wszystkie rozwiązania są postaci $s(X) = a^k$ i dla żadnego rozwiązania nie możemy sparować liter w X oraz aX tak samo.

Ten problem można jednak obejść: zauważmy, że a oraz X są przemienne, gdyż oba reprezentują bloki a . W związku z tym możemy zamienić aX w Xa po lewej stronie równania, co nie zmienia zbioru rozwiązań.

Pokaż, że jeśli dla ustalonej litery a dla każdej zmiennej X albo:

1. $s(X)$ nie ma a -prefiksu i a -sufiksu *albo*
2. $s(X)$ jest blokiem a

to wtedy możemy poprzestawiać w równaniu zmienne i litery tak że dostaniemy równanie dla którego s wciąż jest rozwiązaniem a kompresja par aa jest jednoznaczna. W szczególności taka kompresja powinna wypychać po stałej liczbie liter z każdej strony zmiennej.

Pokaż, że potem 1–2 jest spełnione dla a' , które jest literą zastępującą aa .

Aby uzyskać równanie spełniające 1–2 wypychamy a -prefiksy i a -sufiksy ze zmiennych, ale oznaczamy je jako nowe zmienne.

Pojawia się dodatkowy problem: w równaniu akumulują się litery powstałe wskutek wypychania liter z tych zmiennych. Aby się z tym uporać *typujemy* litery które reprezentują skompresowane bloki a : początkowo typujemy tylko litery a i zmienne spełniające 2; potem dodatkowo dokonujemy kompresji par w których obie litery są typowane, uzyskane litery też są typowane. Pokaż, że 1 można uogólnić: zmienna nie ma prefiksu ani sufiksu typowanej litery.

Zadanie 18 Załóżmy, że rozwiązaliśmy Zadanie 17. Korzystając z tego algorytmu pokaż, że najkrótsze rozwiązanie równania w słowach ma długość najwyżej podwójnie wykładniczą.

Na czym polega problem z uogólnieniem tego argumentu również na oryginalny algorytm?

Dotąd zrobiliśmy wszystkie zadania

Zadanie 19 Mówimy, że rozwiązanie s równania słów $u = v$ jest dobre, jeśli:

- dla każdej pary ab z $s(u)$ para ta ma w $s(u)$ lub $s(v)$ wystąpienie krzyżujące lub explicite;
- dla każdego maksymalnego bloku a^ℓ , gdzie $\ell \geq 2$, z $s(u)$ blok ten ma w $s(u)$ lub $s(v)$ wystąpienie dotknięte (tj. krzyżujące lub explicite lub jest prefiksem lub sufiksem jakiejś zmiennej);
- każda litera z $s(u)$ występuje w u lub v .

Pokaż, że jeśli algorytm z drugiego wykładu rozpocznie od najkrótszego rozwiązania to odpowiadające rozwiązania uzyskane w każdym kolejnym kroku są dobre.

Zadanie 20 Pokaż że któryś z rozważanych wariantów algorytmu dla równań w słowach (wybierz który chcesz) może być użyty do wygenerowania SLP rozmiaru $\text{poly}(n, \log N)$ dla najkrótszego rozwiązania, gdzie N jest długością tego rozwiązania. Jak mały może być wykładnik dla $\log N$?

Wskażówka: Zadanie 19 oraz 19 powinno być pomocne.

Zadanie 21 (zrobione) Udowodnij nierówność Hadamarda: dla macierzy kwadratowej $N = (n_{i,j})_{i,j=1}^k$ zachodzi

$$|\det(N)| \leq \prod_{j=1}^k \sqrt{\sum_{i=1}^k n_{i,j}^2}.$$

Zadanie 22 (zrobione) Pokaż, że wykładnicze ograniczenie na wykładnik okresowości jest ścisłe, z dokładnością do stałej w wykładniku.

Zadanie 23 Udowodnij, że możemy użyć parametryzowania długości maksymalnych bloków i zapisywania równości między nimi również w algorytmie rozwiązującym równania w słowach: zamiast zgadywać długości maksymalnych bloków w czasie kompresji, określ je jako ℓ_X i r_X będące zmiennymi naturalnymi i zapisz równość bloków jako układ liniowych równań Diofantycznych.

Pokaż, że jeśli równanie słów jest zakodowane przy użyciu m bitów (w naturalnym kodowaniu) to stworzony układ równań liniowych ma $\mathcal{O}(m)$ bitów.

choć jest mniej wydajne.

Wskażówka: Unarne kodowanie stałych, tj. p kodowane jako p bitów, może być prostsze w dowodzie,

Zadanie 24 (zrobione) Udowodnij, że możemy sprawdzić spełnialność układu liniowych równań Diofantycznych w którym wszystkie stałe są zakodowane unarnie w niedeterministycznej pamięci liniowej. Zarówno pamięć wejścia jak i pamięć algorytmu liczymy w bitach.

Wskażówka: Czy widzisz związek z kompresją bloków?

Wskażówka: Zgaduj parzystość obu stron równania i dziel przez 2, odpowiednio przekształcając

Zadanie 25 Używając ograniczenia na minimalne rozwiązanie układu liniowych równań Diofantycznych udowodnij, że podwójnie wykładnicze ograniczenie na wielkość minimalnego rozwiązania wynika już z pierwszej wersji algorytmu dla równań w słowach, prezentowanego na wykładzie.

Wskażówka: Być może Zadanie 19 jest pomocne.

Zadanie 26 Pokaż, że wykładnicze oszacowanie na wykładnik periodyczności (postaci $2^{\text{poly}(n)}$), lecz nie postaci $2^{\mathcal{O}(n)}$ może zostać wyprowadzone z ograniczenia na długości maksymalnych bloków dla najkrótszych rozwiązań oraz algorytmu dla sprawdzania spełnialności równań w słowach.

Wskażówka: Jak zmienia się wykładnik periodyczności po jednym kroku kompresji? Jaka jest różnica między kompresją par i kompresją bloków? Być może Zadanie 19 jest pomocne.

Zadanie 27 (zrobione) Przypomnij sobie alternatywne sformułowania Lematu o okresowości:

- Jeśli słowo w ma okresy p, q takie że $|p| + |q| \leq |w|$ to w ma też okres długości $\gcd(|p|, |q|)$.

- Dla dwóch słów u, v jeśli $uv = vu$ to istnieje słowo w oraz liczby n, m takie że $u = w^n$, $v = w^m$, tzn. są one potęgami tego samego słowa.

Udowodnij równoważność tych sformułowań.

Przypomnij silne sformułowanie Lematu o okresowości w dwóch wariantach (jak powyżej), pokaż ich równoważność.

Zadanie 28 (zrobione) Pokaż, że silny Lemat o okresowości (w którymś ze sformułowań) wynika ze swojej wersji w której $\gcd$ wynosi 1.

Zadanie 29 (zrobione) Udowodnij silną wersję Lematu o okresowości w wersji z $uv = vu$.

Zadanie 30 (Inny dowód Lematu o okresowości), (zrobione) Dla danego słowa $w[1..p+q]$ o okresach p, q takich że $\gcd(p, q) = 1$ zdefiniujmy graf na pozycjach tego słowa poprzez: $\{i, j\}$ jest krawędzią wtw. $|i - j| \in \{p, q\}$. Pokaż, że ten graf jest cyklem. Wywnioskuj z tego, że $w \in a^*$ dla odpowiedniego a .

Wzmocnij dowód tak, aby działał też dla słowa $w = w[1..p+q-1]$

Wskazówka: Ten graf nie jest cyklem. Co się stanie, gdy usuniemy z poprzedniego grafu jeden wierzchołek?

Zadanie 31 (zrobione) Udowodnij, że u jest słowem pierwotnym wtw. wszystkie jego sprzężenia są parami różne.

Niech u, u' będą sprzężone. Pokaż, że u jest pierwotne wtw. u' jest pierwotne.

Zadanie 32 (zrobione) Niech P będzie słowem pierwotnym, W, W' dwoma słowami o P -prezentacjach (W) i (W') . Pokaż, że P -prezentacja WW' jest jednej z poniższych postaci:

- (WW') ;
- (U, V) , gdzie $WW' = UV$;
- (U, V) , gdzie $WW' = UPV$;
- (U, Z, V) , gdzie $WW' = UZV$.

Zadanie 33 (zrobione) Pokaż, że dla P — słowa pierwotnego i W — słowa, P -prezentacja istnieje, jest jedyna i może być obliczona zachłannie.

Wskazówka: Użyj Zadania 32

Zadanie 34 (zrobione) Niech P będzie słowem pierwotnym, W, W' dwoma słowami o P -prezentacjach $U_0, \dots, U_u$ i $V_0, \dots, V_v$ (dla ciągów $k_1, \dots, k_u$ i $\ell_1, \dots, \ell_v$). Pokaż, że P -prezentacja słowa WW' jest jednej z poniższych postaci i że ta postać zależy tylko od U_u and V_0 .

- $WW' = [U_0, \dots, U_{u-1}, V_1, \dots, V_v](k_1, \dots, k_{u-1}, k_u + \ell_1 + c, \ell_2, \dots, \ell_v)$ dla pewnego $c \leq 3$.
- $WW' = [U_0, \dots, U_{u-1}, U', V_1, \dots, V_v](k_1, \dots, k_{u-1}, k_u + c, \ell_1 + c', \ell_2, \dots, \ell_v)$ dla pewnych $c + c' \leq 2$.
- $WW' = [U_0, \dots, U_{u-1}, U', V', V_1, \dots, V_v](k_1, \dots, k_{u-1}, k_u + c, c', \ell_1 + c'', \ell_2, \dots, \ell_v)$ dla pewnych $c + c' + c'' \leq 2$.
- $WW' = [U_0, \dots, U_{u-1}, U', U'', V', V_1, \dots, V_v](k_1, \dots, k_{u-1}, k_u, 0, 0, \ell_1, \ell_2, \dots, \ell_v)$.

Wskazówka: Użyj Zadania 32 i Zadania 33

Zadanie 35 (zrobione) Udowodnij, że algorytm dla kwadratowych równań słów można ulepszyć tak, aby zwracał skończoną reprezentację wszystkich rozwiązań.

Zadanie 36 (zrobione) Rozważmy następującą ograniczoną klasę równań słów:

regularne każda zmienna występuje najwyżej raz w każdej stronie równania;

zorientowane jeśli dwie zmienne X, Y występują po obu stronach równania to występują w tej samej kolejności, tj. X jest przed Y po lewej stronie wtedy i tylko wtedy, gdy X występuje przed Y po prawej stronie.

Pokaż, że rozstrzygalność takich równań jest w NP.

Wskazówka: Użyj algorytmu dla równań kwadratowych.

Zadanie 37 (zrobione) Dla równania słów

$$A_0 X A_1 \dots A_{n_A-1} X A_{n_A} = X B_1 \dots B_{n_B-1} X B_{n_B}, \quad (1)$$

pokaż, że jeśli liczby wystąpień zmiennej po obu stronach równania są różne, innymi słowy: $n_A \neq n_B$ to równanie ma najwyżej jedno rozwiązanie i można je podać i zweryfikować w liniowym czasie.

Zadanie 38 (zrobione) Pokaż, że jeśli mamy dany układ równań słów z jedną zmienną to możemy w liniowym czasie podać i zweryfikować jedyne rozwiązanie lub podać jedno, równoważne równanie, zmieniając sumaryczną długość jedynie o stałą.

Zadanie 39 (zrobione) Pokaż, że jeśli w równaniu $\mathcal{A} = \mathcal{B}$ w postaci (1) obie strony równania mają tyle samo wystąpień zmiennej oraz ostatnie słowo po prawej stronie jest niepuste a ostatnie po lewej puste, innymi słowy $A_{n_A} = \epsilon \neq B_{n_B}$ to to równanie można podzielić na dwa równoważne wejściowe, tj. jest równoważne pewnym dwóm równaniom postaci $\mathcal{A}' = \mathcal{B}'$, $\mathcal{A}'' = \mathcal{B}''$, gdzie $\mathcal{A} = \mathcal{A}' \mathcal{A}''$, $\mathcal{B} = \mathcal{B}' \mathcal{B}''$.

Wskazówka: Zdefiniuj bilans dowolnego prefiksu $\mathcal{A}$ i analogicznie $\mathcal{B}$ jako parę (liczba wystąpień zmiennej, liczba wystąpień liter). Pokaż, że istnieją prefiksy obu stron, które mają ten sam bilans.

Zadanie 40 (zrobione) Załóżmy, że równanie słów $Xp = qX$ jest spełnialne, gdzie X jest zmienną a p, q — słowami. Pokaż, że zachodzą wtedy następujące warunki:

- p, q są sprzężone; w szczególności również ich słowa pierwotne są sprzężone.
- Istnieje dokładnie jedna para słów (u, v) i liczba naturalna $k \geq 1$, takie że uv, vu są pierwotne i $p = (uv)^k$, $q = (vu)^k$;
- zbiór rozwiązań tego równania jest postaci $(vu)^*v$; tj. każde podstawienie $s(X) \in (vu)^*v$ jest rozwiązaniem oraz każde rozwiązanie podstawia pod X słowo z $(vu)^*v$.

Zadanie 41 (zrobione) Podaj liniowy algorytm konstruujący tablicę MP π_w dla danego słowa w . Dla przypomnienia: dla słowa $w[1..n]$ definiujemy

$$\pi[i] = \max\{j < i : w[1..j] \text{ jest prefikso-sufiksem } w[1..i]\}.$$

Zadanie 42 (2 punkty), (w trakcie robienia) Niech u, v, w będą słowami pierwotnymi, takimi że u^2 jest prefiksem v^2 oraz v^2 jest prefiksem w^2 . Udowodnij, że $|u| + |v| \leq |w|$.

Wskazówka: To nie jest bardzo trudne, ale wymaga rozważenia paru przypadków. W ostatczkości w miarę krótki dowód jest w M. Lothaire: "Algebraic Combinatorics on Words", Tw. 8.1.14, link

Zadanie 43 (zrobione) Załóżmy, że dla danego słowa w w czasie $\mathcal{O}(n)$ jesteśmy w stanie skonstruować strukturę, która dla dwóch danych indeksów i, j w czasie $\mathcal{O}(1)$ zwróci długość najdłuższego wspólnego prefiksu słów $w[i..n]$ oraz $w[j..n]$. Pokaż, że w takim razie jesteśmy w stanie w czasie $\mathcal{O}(n + n_X \log n)$ zweryfikować $\mathcal{O}(\log n)$ kandydatów na rozwiązanie równania w słowach z jedną zmienną, z których każdy jest prefiksem A_0 , zdefiniowanego jak w (1); dla przypomnienia n_X jest ilością wystąpień zmiennej w równaniu.

Zadanie 44 (zrobione) Dla danych słów pierwotnych $P_1, \dots, P_k$ takich że dla każdego i zachodzi $P_i^2 \subseteq P_{i+1}^2$ oraz słowa w podaj algorytm o czasie działania $\mathcal{O}(|P_k| + |w|)$ który zwróci dla wszystkich P_i z $P_1, \dots, P_k$ długość P_i -prefiksu słowa w .

Zadanie 45 (zrobione) Udowodnij, że:

- dla dowolnego niepustego, czyli $s(X) \neq \epsilon$, rozwiązania równania (1) pierwsza litera $s(X)$ to pierwsza litera A_0 a ostatnia litera to ostatnia litera B_{n_B} .
- jeśli $A_0 \in a^+$ to $s(X) \in a^+$ dla każdego niepustego rozwiązania (1).
- jeśli pierwszą literą A_0 jest a oraz $A_0 \notin a^+$ to istnieje najwyżej jedno rozwiązanie $s(X) \in a^+$, istnienie tego rozwiązania można sprawdzić w czasie liniowym: $\mathcal{O}(|\mathcal{A}| + |\mathcal{B}|)$. Co więcej, dla $s(X) \notin a^+$ długości a -prefiksów $s(X)$ oraz A_0 są tej samej długości.

Zadanie 46 (zrobione) Niech $A_0 \in a^+$. Pokaż, jak policzyć wszystkie rozwiązania równania w czasie liniowym.

Wskazówka: Popatrz na pierwszą literę a po obu stronach równania. Rekurencyjnie dla reszty.

Zadanie 47 (zrobione) Na podstawie drugiego algorytmu dla równań słów z jedną zmienną (tj. kompresującego i sprawdzającego rozwiązania w czasie swej pracy) podaj alternatywny dowód faktu, że równanie słów z jedną zmienną ma najwyżej $\mathcal{O}(\log n)$ rozwiązań oraz jedną rodzinę rozwiązań postaci $\{w^k w' : k \geq 1\}$, gdzie w' jest prefiksem w .

Wskazówka: Pokaż najpierw, że w kolejnych iteracjach słowo A_0 uzyskujemy przez kompresję przesunięcia cyklicznego oryginalnego A_0 .

Zadanie 48 (zrobione) Rozważmy równanie słów $u = v$ z wieloma zmiennymi, takie że:

- każda użyta zmienna występuje dokładnie raz w u i dokładnie raz w v ;
- jeśli X, Y występują po obu stronach równania, to występują w nich w tej samej kolejności
- jest po jednym pozytywnym więzie regularnym na zmienną, tj. dla każdej zmiennej X mamy NFA N_X (N_X jest częścią wejścia) i wymagamy, aby w rozwiązaniu $s(X) \in L(N_X)$.

Pokaż, że spełnialność takich równań jest w NP.

Zadanie 49 (2 punkty), (zrobione) Rozważmy system $\mathcal{S}_2$ równań postaci:

$$\begin{cases} YAX &= XBY \\ YCX &= XDY \end{cases}, \quad (\mathcal{S}_2)$$

gdzie dodatkowo $|A| = |B| \leq |C| = |D|$ oraz zakładamy, że $|s(Y)| < |s(X)|$. Oznaczmy też $n = |ABCD|$.

Pokaż, że zbiór podstawień pod X w rozwiązaniach tego układu zawiera się w sumie następujących języków:

- $\{(p^{j+a}q)^k p^j p' : j, k \geq 0\}$, gdzie p, q : słowa, $\neg(p \sqsubseteq q)$, $p' \sqsubseteq p$, $|p|, |q| \in \mathcal{O}(n)$ oraz $0 \leq a \leq n$.
- $\mathcal{O}(n)$ języków postaci $\{p^j q : j \geq 0\}$, gdzie p, q : słowa, $\neg(p \sqsubseteq q)$, $|p|, |q| \in \mathcal{O}(n)$.
- $\mathcal{O}(n \log n)$ języków postaci $\{p\}$, gdzie p : słowo, $|p| \in \mathcal{O}(n)$.

Wskazówka: Postępuj podobnie jak w przypadku kwadratowych: rozpatrz osobno przypadki $|s(X)| \leq |s(Y)|$ oraz $|s(X)| < |s(Y)|$. Pierwszy jest prosty, drugi sprowadź to do systemu $\mathcal{S}_1$ z wykładu. Robi się to przez podstawienia, są lekko trikowe.

Zadanie 50 (zrobione) Pokaż jak znaleźć wszystkie rozwiązania równania $\mathcal{S}_3$:

$$YAX = XBY, \quad (\mathcal{S}_3)$$

gdzie $A \neq B$.

Wskazówka: To jest równanie kwadratowe.

Zadanie 51 Pokaż że dla równania $\mathcal{S}_4$:

$$YAX = XAY \quad (\mathcal{S}_4)$$

każde rozwiązanie jest albo postaci

$$s(X) = (Zp)^j Z \quad s(Y) = (pZ)^k p$$

gdzie p jest słowem, $|p| \in \mathcal{O}(|A|)$ zaś Z jest nową zmienną zaś j, k : liczbami naturalnymi, lub istnieją słowa p, q , $|pq| \in \mathcal{O}(n)$ takie że podstawienie pod X (lub pod Y) w rozwiązaniu jest z języka p^*q .

możliwe długości słowa, którego potęgę uzyskasz.

Wskazówka: Dopisz A do obu stron po prawej stronie. Czy widzisz komutację? Rozpatrz osobno

Zadanie 52 (zrobione) Rozważmy równanie słów z jedną zmienną, n oznacza jego długość, n_X : liczbę wystąpień X po jednej stronie równania. Równanie to dodatkowo zawiera n_Y wystąpień znaku Y , który oznacza dane na wejściu słowo p , gdzie $|p| \in \mathcal{O}(n)$.

Podaj algorytm, który podaje wszystkie rozwiązania tego równania w czasie $\mathcal{O}(n(n_X + n_Y))$.

słów.

To nie jest trudne, ale trzeba wyłumaczyć usprawnienia kroków drugiego algorytmu dla równań

prawy przy weryfikacji chcemy poświęcić stały czas na jej przejście. krzyżujące, należy na początku każdej fazy wypchnąć z X do równania jedną literę na lewo i jedną na

Wskazówka: Myśl o wystąpieniach X jak o zmiennej, dla której znasz podstawienie: też definiuje parę

Zadanie 53 (2 punkty) Jak w poprzednim zadaniu, rozważmy równanie słów z jedną zmienną, n oznacza jego długość, n_X : liczbę wystąpień X po jednej stronie równania. Równanie to dodatkowo zawiera n_Y wystąpień znaku Y , który oznacza dane na wejściu słowo parametryczne $p^j d$, gdzie $|d|, |p| \in \mathcal{O}(n)$.

Podaj algorytm, który podaje wszystkie rozwiązania tego równania. Przez rozwiązanie rozumiemy takie podstawienie s , że dla pewnej wartości $j_0 \in \mathbb{N}$ podstawienie $s(X)$ jest rozwiązaniem dla równania w którym wszystkie wystąpienia znaku Y zastąpimy słowem $p^{j_0} d$.

Jak dobrze potrafisz ograniczyć czas działania tego algorytmu?

Jeśli to pomoże, to możesz założyć, że P jest słowem pierwotnym.

skonstruować.

Przy testowaniu rozwiązań możesz wykorzystywać strukturę z Zadania 43, nawet jeśli nie potrafisz jej kompresji tego p^j .

W pewnym momencie p zostanie zredukowane do litery i całe p^j zastąpione nową literą a_j . Zastanów się, co się dzieje, jeśli są lub nie ma wystąpienia tej litery w tym równaniu inne niż te uzyskane z poszczególnymi p w czasie kompresji?

Wskazówka: Należy wcześniej rozwiązać Zadanie 52. Małe wartości j_0 należy rozpatrzeć oddzielnie i rozwiązać wg. Zadania 52. Dla większych wartości zastanów się, jak wygląda interakcja pomiędzy

Zadanie 54 Niech jak w dowodzie dla systemu $\mathcal{S}_1$, CD będzie słowem pierwotnym, zdefiniujmy $v = (CD)^i CA$ oraz $w = B(CD)^i C$, przy czym $i > 1 + 3[|CA|/|CD|]$. Załóżmy, że v, w są pierwotne (w szczególności CD nie jest pierwiastkiem pierwotnym v) oraz $v = gh$ oraz $w = hg$. Pokaż, że prefiks v o okresie $|CD|$ i sufix (też v) o okresie $|DC|$ nie mogą na siebie zająć o $|CD|$ (lub więcej) liter. Wynioskuj z tego, że $|g| \geq |CD|$. Podobnie pokaż, że CD jest okresem g .

Zadanie 55 (zrobione) Skonstruuj nieskończony układ równań słów (z nieskończeniem wieloma zmiennymi), taki że każdy jego skończony podzbiór jest spełnialny, natomiast cały układ spełnialny nie jest.

Zadanie 56 (zrobione) Niech dany będzie układ równań słów postaci

$$X_0^k = X_1^k X_2^k \cdots X_n^k$$

dla $k = k_1, k_2, k_3$, gdzie $0 < k_1 < k_2 < k_3$ są liczbami naturalnymi.

Pokaż, że jeśli równanie to ma nietrywialne rozwiązanie s nad dowolnym alfabetem (tj. takie, że nie istnieje w takie że $s(X_0), s(X_1), \dots, s(X_n) \in w^*$) to ma też nietrywialne rozwiązanie nad alfabetem dwuliterowym.

Zadanie 57 (zrobione na wykładzie) Pokaż, że dla zadanego słowa $w \in \Gamma^*$, gdzie identyfikujemy $\Gamma \subseteq \mathbb{R}$ z podzbiorem liczb rzeczywistych możemy zmienić alfabet (formalnie: możemy podać bijekcję między Γ a Γ') tak aby $\sum(w) = 0$.

Zadanie 58 (zrobione) Mówimy, że słowo w ma prefikso-sufiks, jeśli istnieje $v \neq \epsilon$ takie że $w = vw' = vw''$ dla pewnych w', w'' . Pokaż, że jeśli w ma prefikso-sufiks, to można je przedstawić jako $w = vuv$ dla pewnego $v \neq \epsilon$ oraz pewnego u .

Zadanie 59 (zrobione) Niech w, w' będą niepustymi słowami sprzężonymi oraz $w = v^i$ dla pewnego naturalnego $i \geq 1$. Pokaż, że istnieją p, q , takie że $v = pq$ oraz $w' = (qp)^i$.

Zadanie 60 (zrobione) Pokaż, że słowo Lyndona nie ma prefikso-sufiksu.

Zadanie 61 (zrobione) Podaj elementarny algorytm, który dla danego słowa pierwotnego wyznacza jego słowo Lyndona w liniowym czasie.

Zadanie 62 (zrobione) Pokaż, że słowo w ma prefikso-sufiks wtedy i tylko wtedy, gdy jest podsłowem słowa u^k dla pewnego $k \geq 1$ oraz u krótszego niż w , tj. $|u| < |w|$.

Zadanie 63 (zrobione) Rozważmy zbiór formuł postaci

$$\exists_{x_1, x_2, \dots, x_k} \varphi(x_1, x_2, \dots, x_k)$$

gdzie φ jest bez kwantyfikatorów, używa spójników logicznych ze zbioru $\{\wedge, \vee, \neg\}$ a atomy to równania używające skwantyfikowanych zmiennych $x_1, x_2, \dots, x_k$ i stałych z Σ^* .

Pokaż, że możemy rozstrzygać w PSPACE prawdziwość takich formuł.

Wskazówka: Algorytm zapewne będzie używał niedeterminizmu do sprowadzenia tych formuł do postaci, w której nie ma $\wedge$, czyli do układu równań i nierówności. Pokaż, jak sprowadzić nierówność do równości: wystarczy zlokalizować pierwszą różnicę — gdzie będzie niedeterminizm.

Zadanie 64 (zrobione) Pokaż, że nie problem sprawdzenia prawdziwości formuły postaci

$$\exists x_1, x_2, \dots, x_k \forall x_{k+1}, \dots, x_{k+\ell} \varphi(x_1, x_2, \dots, x_k)$$

gdzie φ jest bez kwantyfikatorów, używa spójników logicznych ze zbioru $\{\wedge, \vee\}$ a atomy to równania używające skwantyfikowanych zmiennych $x_1, x_2, \dots, x_{k+\ell}$ i stałych z Σ^* , jest nierozstrzygalny.

Jeśli nikomu nie uda się wersji z jedną alternacją kwantyfikatorów, to możemy zrobić z dwoma, tj. formuła jest postaci

$$\exists x_1, x_2, \dots, x_k \forall x_{k+1}, \dots, x_{k+\ell} \exists x_{k+\ell+1}, \dots, x_{k+\ell+r} \varphi(x_1, x_2, \dots, x_k)$$

lub

$$\forall x_1, x_2, \dots, x_k \exists x_{k+1}, \dots, x_{k+\ell} \forall x_{k+\ell+1}, \dots, x_{k+\ell+r} \varphi(x_1, x_2, \dots, x_k)$$

Wskazówka: Użyj najpierw równań — a potem wyeliminuj jak w Zadaniu 63.

Zadanie 65 (Lemat Newmana), (zrobione) System przepisywania termów $\mathcal{S} = \{(\ell_i, r_i)\}_{i \in I}$ jest *skracający*, jeśli $(\ell, r) \in \mathcal{S}$ implikuje $|\ell| > |r|$. $\mathcal{S}$ jest *konfluentny*, jeśli dla wszystkich s, t, u

spełniających $s \rightarrow_S^* t$ i $s \rightarrow_S^* u$ istnieje v , takie że $t, u \rightarrow_S^* v$; jest lokalnie konfluentny, jeśli dla wszystkich s, t, u spełniających $s \rightarrow_S t$ i $s \rightarrow_S u$ istnieje v takie że $t, u \rightarrow_S^* v$.

Pokaż, że jeśli $\mathcal{S}$ jest skracający to $\mathcal{S}$ jest konfluentny wtedy i tylko wtedy gdy jest lokalnie konfluentny.

Wynioskuj z tego, że grupa wolna jest dobrze zdefiniowana, w szczególności, $\text{IRR}(w)$ jest jednoznacznie wyznaczone.

Zadanie 66 (zrobione) Pokaż, że jeśli w grupie wolnej zachodzi $ab = c$, to istnieje r takie że

$$a = a'r \quad b = r^{-1}a' \quad a'b' = c$$

oraz $a'b'$ jest nieskracalne, tj. $\text{IRR}(a'b') = a'b'$.

Zadanie 67 (zrobione) Pokaż, że dla ustalonych λ, ℓ dla dostatecznie dużego p istnieje zbiór $R = \{r_0, \dots, r_\lambda\} \subseteq \{a, b, \bar{a}, \bar{b}\}^m$ który jest dostatecznie losowy, tj. dla każdego słowa $w \in \{a, b, \bar{a}, \bar{b}\}^*$ długości przynajmniej $\frac{m-\ell}{2}$ słowo w ma najwyżej jedno wystąpienie w $R \cup \bar{R}$.

dowodów.

Mozliwa jest też konstrukcja wprost, używająca np. słów de Bruïna. Postaraj się policzyć obie wersje.

Wskazówka: Najprostszy dowód używa argumentu prawdopodobieństwa/złożoności Kolmogorowa.

Zadanie 68 (zrobione) Pokaż, że podane na wykładzie systemy przepisowywania termów:

$$P_i = \{r_i \ell r_{i+1} \rightarrow r_i k r_{i+1}, \bar{r}_{i+1} \bar{\ell} \bar{r}_i \rightarrow \bar{r}_{i+1} \bar{k} \bar{r}_i\}$$

gdzie r_i, r_{i+1} są częścią zbioru dostatecznie losowego, zaś $|\ell| \geq 2$, mają postać normalną.

Wskazówka: Są skracające.

Zadanie 69 (zrobione) Pokaż, że problem równań słów w półgrupie wolnej z involucją (czyli $\Sigma \cup \bar{\Sigma}^*$) w której involucja posiada punkty stałe (tj. istnieje a takie że $a = \bar{a}$) można sprowadzić do problemu równań słów w (większej) półgrupie wolnej z involucją, która nie ma punktów stałych. Uogólnij swoje rozwiązanie tak, aby działało też w przypadku równań z więzami regularnymi.

Wskazówka: Dla litery a takiej że $\bar{a} = a$ zamień a na $\bar{a}$.

Zadanie 70 (zrobione) Pokaż, że jeśli dany jest homomorfizm z półgrupy wolnej z involucją $(\Sigma \cup \bar{\Sigma})^*$ w macierz Boolowskie (z operacjami logicznymi) $\rho : (\Sigma \cup \bar{\Sigma})^* \rightarrow \prod_i \mathbb{B}_{m_i \times m_i}$ to możemy zdefiniować na macierzach Boolowskich involucję i zadać homomorfizm $\rho' : (\Sigma \cup \bar{\Sigma})^* \rightarrow \prod_i \mathbb{B}_{m'_i \times m'_i}$ (czyli w większe macierze), który zachowuje involucję. Od involucji na macierzach wymagamy jedynie, by $\bar{\bar{M}} = M$, na wektorze macierzy powinna ona działać po współrzędnych.

Wskazówka: Wystarczy wziąć $m'_i = 2m_i$ i rozważać macierze z dwoma kłatkami $m_i \times m_i$. Potencjalna involucja to transpozycja, ale to może nie wystarczyć; zamienianie kłatek między sobą pomoże.

Zadanie 71 (2 punkty) (zrobione) Pokaż, że dla danego na wejściu równania słów z więzami regularnymi $\rho : \Sigma \rightarrow \prod_i \mathbb{B}_{m_i \times m_i}$ (dla uproszczenia: bez involucji, ale powinno działać też z involucją) możemy rozszerzyć alfabet wejściowy Σ o litery

$$\{a_\tau : \tau \in \prod_i \mathbb{B}_{m_i \times m_i} \text{ istnieje słowo } w \in \Sigma^* \text{ takie że } \rho(w) = \tau\}.$$

Pokaż, że ustalone równanie jest spełnialne nad Σ wtedy i tylko wtedy, gdy jest spełnialne nad tym rozszerzonym alfabetem. Zmodyfikuj algorytm sprawdzający spełnialność równań słów tak, że potrafi obsłużyć więzy regularne i ten alfabet. Czy działa on w PSPACE?

Zadanie 72 (zrobione) Niech $\mathbb{G}_1, \dots, \mathbb{G}_m \leq \mathbb{G}$ będą podgrupami $\mathbb{G}$, $\vec{z} \in \mathbb{G}$ ustalonymi elementami z $\mathbb{G}$, zaś $i : \mathbb{G} \rightarrow \mathbb{G}$ będzie automorfizmem $\mathbb{G}$ takimi że $i(\mathbb{G}_j) = \mathbb{G}_j$. Pokaż, że

$$\exists Y_1 \in \mathbb{G}_1 \exists Y_2 \in \mathbb{G}_2 \dots \exists Y_m \in \mathbb{G}_m \varphi(Y_1, \dots, Y_m, \vec{z})$$

zachodzi wtedy i tylko wtedy, gdy zachodzi

$$\exists Y_1 \in \mathbb{G}_1 \exists Y_2 \in \mathbb{G}_2 \dots \exists Y_m \in \mathbb{G}_m \varphi(Y_1, \dots, Y_m, i(\vec{z})) .$$

Zadanie 73 (zrobione) Niech $\mathbb{G} = \langle c_1, \dots, c_m \rangle$ będzie grupą wolną. Rozważmy $h : \mathbb{G} \rightarrow \mathbb{G}$ zdefiniowany jako $h(c_1) = gc_1g'$, gdzie $g, g' \in \langle c_2, \dots, c_m \rangle$, oraz h jest identycznością na pozostałych generatorach. Pokaż, że h jest automorfizmem $\mathbb{G}$ (czyli izomorfizmem przekształcającym $\mathbb{G}$ w $\mathbb{G}$).

Zadanie 74 (zrobione) Niech G będzie grupą wolną której elementami są nieskracalne słowa nad $\Sigma \cup \Sigma^{-1}$. Pokaż, że każda funkcja $f : \Sigma \rightarrow H$, gdzie H jest grupą, przedłuża się do homomorfizmu (tzn. istnieje homomorfizm $h : G \rightarrow H$ taki że $f = h \upharpoonright_\Sigma$).

Zadanie 75 Rozważmy równania słów nad $(\Sigma \cup \bar{\Sigma})^*$ z involucją bez punktów stałych. Dla litery a jej a -quasioblok to dowolne słowo postaci $(a\bar{a})^\ell$ dla $\ell > 1$. Wystąpienie quasibloku jest maksymalne (w słowie w), jeśli nie ma dłuższego wystąpienia quasibloku a zawierającego dane wystąpienie. Dla równania $u = v$ oraz podstawienia s maksymalne wystąpienie quasibloku jest jawne, jeśli pochodzi z równania, niejawne jeśli pochodzi z podstawienia za jedno wystąpienie zmiennej i krzyżujące w pozostałych przypadkach. Litera a ma krzyżujące quasibloki (dla podstawienia s), jeśli choć jedno wystąpienie quasibloku jest krzyżujące.

Pokaż, że jeśli dla rozwiązania s a nie ma krzyżujących quasibloków, to dla poniższego algorytmu

Algorithm 1 QuasiBlockCompNCr(a)

- 1: **for** każdego quasibloku a występującego w u lub v **do**
 - 2: **for** każdego ℓ **do**
 - 3: niech $c_\ell, \bar{c}_\ell \in \Sigma \cup \bar{\Sigma}$ będą nieużywanymi literami
 - 4: zastąp każde maksymalne wystąpienie $(a\bar{a})^\ell$ w u lub v przez $c_\ell \bar{c}_\ell$
-

zachodzi: równanie $U' = V'$ otrzymane po zastosowaniu algorytmu QuasiBlockCompNCr(a) ma rozwiązanie s' takie że $s'(U')$ jest otrzymane przez kompresję quasibloków a w $s(U)$. Co więcej, jeśli równanie $U' = V'$ ma rozwiązanie, to rozwiązanie ma również równanie $U = V$.

Zadanie 76 Rozważmy równania słów nad $(\Sigma \cup \bar{\Sigma})^*$ z involucją bez punktów stałych. Dla litery a oraz słowa w definiujemy $a\bar{a}$ prefiks w jako najdłuższy prefiks w z języka $(\bar{a} + \epsilon)(a\bar{a})^*$, analogicznie definiujemy $a\bar{a}$ -sufiks (dla języka $(a\bar{a})^*(\bar{a} + \epsilon)$).

Pokaż, że dla następującego algorytmu:

Algorithm 2 QuasiPop(a)

- 1: **for** $X \in \mathcal{X}$ **do**
 - 2: niech p, s będą $a\bar{a}$ -prefiksem i sufiksem $s(X)$ ▷ Zgadnij
 - 3: zamień każde X w u i v przez pXs
 - 4: **if** $s(X) = \epsilon$ **then** ▷ Zgadnij
 - 5: usuń X z u i v
-

jeśli $U = V$ miało rozwiązanie s to dla odpowiednich wyborów niedeterministycznych otrzymane równanie $U' = V'$ ma rozwiązanie s' takie, że a nie ma krzyżujących quasibloków oraz $s(U) = s'(U')$. Z drugiej strony, jeśli $U' = V'$ ma rozwiązanie, to ma je również $U = V$.

Zadanie 77 Poprawnie zbudowany term to ukorzenione, uporządkowane, etykietowane (elementami z Σ) drzewo, w którym wierzchołek o etykietce a ma $\text{ar}(a)$ dzieci, gdzie $\text{ar} : \Sigma \rightarrow \mathbb{N}$.

W problemie unifikacji (termów) rozważamy skończone układy równań postaci $\{u_i = v_i\}_{i \in I}$ gdzie u_i, v_i są termami nad sygnaturą $\Sigma \cup \mathcal{X}$, gdzie $\Sigma \cap \mathcal{X} = \emptyset$, a funkcja $\text{ar}(X) = 0$ dla każdej zmiennej $X \in \mathcal{X}$. Podstawienie s przypisuje każdej zmiennej poprawny term nad Σ (z funkcja arności $\text{ar} \upharpoonright_\Sigma$). Podstawienie jest *rozwiązaniem*, jeśli po zamianie każdego wystąpienia zmiennej X przez term $s(X)$ równanie $u = v$ jest spełnione.

Pokaż, że problem unifikacji termów można rozwiązać w czasie liniowym (być może będziesz potrzebować kilku rozsądnych założeń o modelu obliczeń — maszyna RAM).

Pokaż też, że jeśli Σ zawiera choć jeden symbol o arności większej niż 1 i choć jeden symbol o arności 0, to problem unifikacji układów równań można sprowadzić do problemu unifikacji jednego równania.

Zadanie 78 (1 punkt za pierwszy podpunkt, 1 za pozostałe) Rozważmy układy równań słów, w których dodatkowo pozwalamy na więzy postaci $|X|_a = |Y|_b$ dla dowolnych zmiennych X, Y oraz dowolnych liter $a, b \in \Sigma$, który oznacza, że od rozwiązania s wymagamy, aby liczba liter a w $s(X)$ była równa liczbie liter b w $s(Y)$. Pokaż, że:

- Te równania są nierozstrzygalne.
- Pozostają nierozstrzygalne, gdy więzy mogą być tylko postaci $|X|_a = |X|_b$, tj. porównujemy liczbę wystąpień liter w zmiennej.
- Pozostają nierozstrzygalne, gdy więzy mogą być tylko postaci $|X|_a = |Y|_a$, tj. porównujemy liczbę wystąpień ustalonej litery w zmiennych.

Wskazówka: Zredukuj 10. problem Hilberta.

Definicja 1. Rozważmy równanie w słowach $u = v$ (bez więzów czy inwolucji), niech Σ będzie alfabetem równania. Niech s będzie rozwiązaniem, zaś $\Sigma \cup \Sigma'$ alfabetem tego rozwiązania (tj. Σ' to litery spoza równania, które są użyte przez rozwiązanie). Jeśli $\Sigma' \neq \emptyset$ to s nazwiemy *unifikatorem z wolnymi literami* Σ' . Rozwiązanie s' jest *instancją* unifikatora s , jeśli dla każdej zmiennej X mamy $s'(X) = \varphi(s(X))$ gdzie $\varphi : (\Sigma \cup \Sigma') \rightarrow (\Sigma \cup \Sigma')^+$ przy czym φ nie jest permutacją na $\Sigma \cup \Sigma'$ oraz jest stałe na Σ (innymi słowy s' powstaje przez nietrywialne zastąpienie wszystkich liter z Σ' słowami z $\Sigma \cup \Sigma'$). Rozwiązanie jest *minimalne*, jeśli nie jest unifikatorem ani instancją unifikatora. Jest *minimalnym unifikatorem* jeśli jest unifikatorem i nie jest instancją innego unifikatora.

Zadanie 79 Rozważmy równanie $aX = Xa$. Pokaż, że wszystkie jego rozwiązania są minimalne.

Zadanie 80 Pokaż, że jeśli s jest unifikatorem to każda jego instancja jest rozwiązaniem.

Zadanie 81 Niech s będzie minimalnym unifikatorem równania $u = v$. Pokaż, że dla każdej wolnej litery a istnieje zmienna X taka, że a jest pierwszą lub ostatnią literą $s(X)$.

Wskazówka: Popatrz na jedną z sąsiadnych liter.

Zadanie 82 Używając poprzedniego zadania pokaż, że jeśli s jest minimalnym unifikatorem równania $u = v$ to dla równania $u' = v'$ uzyskanego z $u = v$ przez wypchnięcie z każdej zmiennej pierwszej i ostatniej litery (zgodnie z s) odpowiadające rozwiązanie s' (tzn. tego, dla którego $s'(u') = s(u)$) jest minimalne.

Zadanie 83 Niech s będzie minimalnym rozwiązaniem równania $u = v$ zaś s' będzie odpowiadającym rozwiązaniem równania $u' = v'$ otrzymanym po kompresji lub wypchnięciu liter ze zmiennej. Pokaż, że s' też jest minimalne.

Zadanie 84 Niech s będzie minimalnym rozwiązaniem równania $u = v$. Pokaż że :

- Jeśli ab jest pod słowem $s(u)$, gdzie $a \neq b$, to ab ma wystąpienie jawne lub krzyżujące w $s(u)$ lub $s(v)$.
- Analogicznie, jeśli a^k jest maksymalnym powtórzeniem w $s(u)$ to a występuje w $u = v$ i istnieje wystąpienie maksymalnego powtórzenia a^k w $s(u)$ lub $s(v)$, które jest jawne lub krzyżujące lub jest prefiksem lub sufixem jakiegś zmiennej.

Zadanie 85 (2 punkty) Pokaż alternatywny dowód, że wariant liniowej monadycznej unifikacji drugiego rzędu rozważany na wykładzie jest w NP; możesz bez dowodu skorzystać z faktu, że jeśli mamy SLP $\mathcal{A}$ to SLP dla jego infiksu od i -tej do j -tej pozycji można policzyć w czasie wielomianowym od rozmiaru SLP $\mathcal{A}$.

- Chcemy pokazać, że istnieje rozwiązanie, które jest wykładnicze; algorytm to dowodzący może mieć czas wykładniczy. Wtedy możemy użyć algorytmu dla równań słów (działa w czasie zależnym od $\log N$).
- Pokaż, że jeśli w grafie zależności jest cykl o konkatenacji etykiet w to za jedną ze zmiennych można podstawić $w^k w'$, gdzie w' jest prefiksem w .
- Podstaw to słowo (trzymane jako SLP).
- Iteruj.

Sam algorytm może być wielomianowy, jeśli przy wypychaniu liter ze zmiennej (gdy graf zależności ma krawędzie, ale nie cykle) wypychamy nie pojedyncze litery, ale dłuższe fragmenty, takie że po wypchnięciu pojawia się nowa krawędź albo zostanie usunięta zmienna.

Zadanie 86 Udowodnij, że wariant liniowej monadycznej unifikacji drugiego rzędu rozważany na wykładzie jest NP-trudny.

Zadanie 87 Rozszerz wariant liniowej monadycznej unifikacji drugiego rzędu rozważany na wykładzie w którym równania nie kończą się $\perp$, tj. dopuszczamy równania postaci $u = v$ dla $u, v \in (\Sigma \cup \mathcal{X})^*$. Za zmienne wciąż możemy podstawiać to, co poprzednio. Pokaż, że ten problem jest w NP.

Zadanie 88 Rozszerz wariant liniowej monadycznej unifikacji drugiego rzędu z Zadania 87: zmienna może użyć swego argumentu w środku, ale najwyżej raz. Formalnie, dopuszczamy też podstawienia λ -termów postaci

$$\lambda x. w x w' \quad \text{oraz} \quad \lambda x. w x w' \perp$$

gdzie w, w' są słowami symboli o arności 1. Np. podstawieniem za X może być $s(X) = aa \bullet bb$ i wtedy $s(Xcc) = aaccbb$. Pokaż, że ten problem jest w NP.

Zadanie 89 Rozszerz wariant liniowej monadycznej unifikacji drugiego rzędu rozważany na wykładzie: zmienna może mieć wiele argumentów, ale może użyć najwyżej jednego i najwyżej raz. Formalnie, podstawienia są λ -termami postaci

$$\lambda x_1 \lambda x_2 \cdots \lambda x_k. w x_i \quad \text{oraz} \quad \lambda x_1 \lambda x_2 \cdots \lambda x_k. w \perp$$

Wtedy każda zmienna ma określoną arność i ilość argumentów musi być prawidłowa, każde strona równania ma być poprawnie zbudowanym termem. Np. równanie (zmienna X ma arność 3)

$$X(aa\perp)(a\perp)(\perp) = aa\perp$$

ma rozwiązanie:

$$s_0(X) = \lambda x_1 \lambda x_2 \lambda x_3. aa\perp$$

$$s_1(X) = \lambda x_1 \lambda x_2 \lambda x_3. x_1$$

$$s_2(X) = \lambda x_1 \lambda x_2 \lambda x_3. ax_2$$

$$s_3(X) = \lambda x_1 \lambda x_2 \lambda x_3. aax_3$$

Pokaż, że ten problem jest w NP.

Zadanie 90 Niech s będzie najkrótszym rozwiązaniem wariantu liniowej monadycznej unifikacji drugiego rzędu rozważanego na wykładzie. Niech w^k będzie podciągami $s(X)$. Pokaż, że $k \leq 2^{cn}$ dla pewnej stałej c , gdzie n jest sumą długości wszystkich równań w instancji.

Zadanie 91 Pokaż, że systemy $\{u_i = v_i\}_{i \in I} \cup \{au = av\}$ oraz $\{u_i = v_i\}_{i \in I} \cup \{u = v\}$ są równoważne, jako instancje wariantu liniowej monadycznej unifikacji drugiego rzędu rozważanego na wykładzie.

Pokaż, że w ogólności nie jest to prawda dla systemów $\{u_i = v_i\}_{i \in I} \cup \{Xu = Xv\}$ oraz $\{u_i = v_i\}_{i \in I} \cup \{u = v\}$, gdzie X jest zmienną.

Zadanie 92 Niech w grafie zależności istnieje krawędź $X \xrightarrow{w} Y$. Pokaż, że dla każdego rozwiązania s tego systemu zachodzi:

- $s(X) = w'$ gdzie w' jest prefiksem w lub
- $s(X)$ ma prefiks w (ten przypadek obejmuje też $s(X) = w\perp$).

W szczególności, jeśli $X \xrightarrow{w} Y$ oraz $X \xrightarrow{w'} Y'$ to wtedy zachodzi przynajmniej jeden z warunków:

- w jest prefiksem w' lub
- w' jest prefiksem w lub
- $s(X)$ jest prefiksem w' oraz w .

Zadanie 93 Niech $\mathcal{A}$ będzie SLP a w słowem nad alfabetem Σ definiowanym przez $\mathcal{A}$. Załóżmy, że w w nie ma dwóch kolejnych identycznych liter. Podaj algorytm, który wyznaczy partycję Σ na Σ_0 i Σ_1 , takie że spośród $|w| - 1$ par $w[i, i + 1]$ dla $i = 1, \dots, |w| - 1$ przynajmniej $\frac{|w|-1}{4}$ z nich należy do zbioru $\Sigma_0\Sigma_1$.

Wskazówka: Model: maszyna RAM, jedno słowo maszynowe obsługujące liczby wielkości $|w|$.

Zadanie 94 Udowodnij, że dla danego słowa t oraz wzorca p oraz pozycji i pozycje wszystkich wystąpień p w t które zawierają pozycję i lub $i + 1$ stanowią jeden ciąg arytmetyczny.

Zadanie 95 Podaj procedurę, która dla dwóch danych ciągów arytmetycznych reprezentowanych jako trójki: (pierwszy element, wzrost, ostatni element) zwróci ciąg arytmetyczny ich elementów wspólnych.