Rozwiązania
stu łatwych zadań z języków formalnych i złożoności obliczeniowej

przygotował Jakub „TheCat” Kowalski

jako kontrmateriał do kursu
prowadzonego w Instytucie Informatyki UWr
w latach 2008 – 2010$\setminus\{E, P\}$
przez Jerzego Marcinkowskiego

5 kwietnia 2010
wersja 36/55

Wersja okrojona, na potrzeby pierwszej części egzaminu w roku 2010.

Od autora

Info
Zbiór ten powstaje z przerwami od roku 2008, kiedy to miałem przyjemność robienia kursu języków formalnych i złożoności obliczeniowej w ii na UWr[1]. Jego celem jest zebranie wiedzy dotyczącej rozwiązywania problemów z jfizo przedstawionych na listach ćwiczeniowych i egzaminacyjnych, a tym samym pomoc studentom w przygotowaniach do ćwiczeń i egzaminu. Poza tym, tak naprawdę, to nie wiem czemu, ale mnie to strasznie rajcuje. Jest to też mój pierwszy samodzielny projekt takich rozmiarów.

Lojalnie ostrzegam, że spisywanie rozwiązań bez uprzedniej próby samodzielnego rozwiązania zadania nie jest dobrą taktyką, ponieważ egzamin weryfikuje zdolność do **samodzielnego** rozwiązywania problemów. Również na ćwiczeniach poprawne przedstawienie rozwiązania na tablicy wymaga jego zrozumienia, a nie jedynie przeczytania rozwiązania. Warto jednak zapoznać się z przedstawionymi rozwiązaniami, aby mieć szersze pojęcie o możliwych technikach rozwiązań, czy też po prostu wiedzieć, jakie nieporuszone na ćwiczeniach zagadnienia również należą do danego działu.

Zbiór ten powinien być traktowany jako swego rodzaju kompendium i wsparcie, a nie przepis na zaliczenie przedmiotu (którym definitywnie nie jest).

Aktualna wersja.
Aktualna wersja dokumentu zawiera spisane wszystkie zadania z ćwiczeń i egzaminów z lat: 2008, 2009, oraz zadania z ćwieczeń z roku 2010.

Oznaczenia zadań.
Każde zadanie posiada informację o tym przy jakich okazjach (od roku 2008) było wykorzystywane w ramach przedmiotu języki formalne i złożoność obliczeniowa. Informacja ta posiada następujący format: Rok (w formacie czterocyfrowym), kropka i numer zadania jeśli było to zadanie z listy lub `E` i numer zadania jeśli było to zadanie z egzaminu zasadniczego lub `P` i numer zadania jeśli było to zadanie z egzaminu poprawkowego.

Oznaczenia zadań są jednoznaczne wobec czego mogą służyć do szybkiego znalezienia konkretnego zadania w dokumencie.

Wewnętrzna numeracja zadań.
Stosowana przeze mnie numeracja zadań jest niezmienna, tzn. rozrost dokumentu powoduje jedynie dodanie nowych zadań do odpowiednich działów, nie modyfikując oznaczeń wcześniej umieszczonych zadań.

Autorzy rozwiązań.
Nie jestem autorem wszystkich zamieszczonych tutaj rozwiązań, często też przy rozwiązywaniu zadań kluczowe okazywały się pomysły na które wpadali inni. W wszystkich takich przypadkach (które miałem zanotowane, bądź byłem sobie w stanie o tym przypomnieć) przy rozwiązaniach tych zadań znajdują się odpowiednie adnotacje oddające autorowi lub pomysłodawcy należną mu cześć i chwałę.

Prawo do powielania.
Każdy może powielać i kopiować zamieszczone w tym zbiorku treści na swój prywatny użytek. Publiczne wykorzystanie wymaga podania źródła, autora zbiorku, oraz autora rozwiązania (jeśli dotyczy).

[1]Strona kursu: http://www.ii.uni.wroc.pl/~jma/kurs.phtml

Grafika.
Rysunki automatów wykonane zostały przy użyciu pakietu Graphviz i generowane przez `neato`.

Rozwój.
Obecnie rozwój zbioru zależy wyłącznie od moich chęci i czasu, a priorytetem jest udostępnianie rozwiązań przed kolejnymi częściami egzaminów.

Nie został jeszcze osiągnięty etap w którym spisałem wszystkie zadania które umiem rozwiązać – mam jeszcze stos notatek do posegregowania. Gdy to kiedyś nastąpi – zacznie się etap wymyślania.

Strona projektu
Oficjalna strona zbiorku, to `http://kot.rogacz.com/page/projects/jfizo`. Będzie się tam pojawiała aktualna wersja, oraz lista „most wanted”, zadań rozwiązania których szczególnie by mi się przydały.

Kontakt z autorem.
Wszelkie uwagi dotyczące zbiorku, znalezione błędy, oferty pracy i matrymonialne, a także własne rozwiązania proszę wysyłać na mój adres e–mail cutterthecat@gmail.com, w temacie dopisując `zbiór JFIZO`.

Jakub „TheCat” Kowalski
http://kot.rogacz.com

Spis treści

1 Deterministyczne Automaty Skończone

Zadanie 1.1. *2008.1; 2009.1; 2010.1*

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

Rozwiązanie.
Rozważmy język L nad dowolnym alfabetem Σ. Wszystkie litery należące do Σ i nie będące zerem możemy utożsamiać z jedynką. Przeprowadzimy dowód nie wprost. W tym celu załóżmy, że istnieje deterministyczny automat skończony $\mathcal{A}$ rozpoznający L i mający mniej niż 1024 stany.

Obserwacja 1. Ciągów o długości 10 nad alfabetem dwuliterowym jest 1024. W związku z tym (zgodnie z zasadą szufladkową Dirichleta) istnieją dwa różne ciągi u i v takie, że $|u| = |v| = 10$ dla których automat $\mathcal{A}$ zatrzyma się w tym samym stanie.

Załóżmy, że wyrazy u, v różnią się po raz pierwszy na k-tej pozycji. Rozpatrzmy więc słowa $u0^{k-1}$ i $v0^{k-1}$. Stan końcowy automatu $\mathcal{A}$ dla obu tych słów będzie tym samym stanem, zaś dziesiąty symbol od końca (k-ty symbol od początku) będzie różny w obu tych ciągach i tylko w jednym będzie wynosił 0. Z tego wynika, że dla jednego z tych wyrazów automat zwróci złą odpowiedź.

Uzyskaliśmy więc sprzeczność z założeniem, że $\mathcal{A}$ rozpoznaje język L. Oznacza to, że automat rozpoznający L musi więc mieć co najmniej 1024 stany.

Zadanie 1.2. *2008.2; 2009.2; 2010.2*

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

Rozwiązanie.
Załóżmy nie wprost, że L jest regularny. Spełnia wtedy lemat o pompowaniu dla języków regularnych. Weźmy więc n z tego lematu i słowo $w = a^n b^{2n}$, które spełnia założenia lematu ($|w| \geqslant n$ i $w \in L$ dla $n \in \mathbb{N}$).

Dokonajmy podziału słowa w, tzn. niech $w = xyz$ tak, że $|xy| \leqslant n$ i $y \neq \varepsilon$. Z postaci w wynika, że słowo $xy = a^{|xy|}$, zaś $y = a^{|y|}$. Jeśli napompujmy jednokrotnie y otrzymamy nowe słowo $w' = a^{|x|}a^{2|y|}a^{n-|xy|}b^{2n} = a^n a^{|y|} b^{2n}$. Założyliśmy, że słowo y jest niepuste, z czego wynika, że słowo w' nie należy do języka L.

Otrzymaliśmy sprzeczność z lematem o pompowaniu, język L nie jest więc regularny.

Zadanie 1.3. *2008.3; 2009.3; 2010.3*

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

Rozwiązanie.[2]
Załóżmy, że L jest regularny – spełnia wtedy lemat o pompowaniu. Weźmy więc n z tego lematu i słowo w będące reprezentacją binarną liczby pierwszej p takiej, że $p > 2^n$ (wiemy, że taka liczba istnieje dla dowolnego $n \in \mathbb{N}$).

Na mocy lematu istnieją więc takie x, y, z, że $w = xyz$, $y \neq \epsilon$ i $\forall k \in \mathbb{N}\ xy^kz$ jest reprezentacją pewnej liczby pierwszej. Niech b_x, b_y i b_z oznaczają wartości odpowiednio x, y i z traktowanych jako liczby binarne.

Weźmy $k = p$, wtedy $xy^pz = q$, gdzie q jest liczba pierwszą której wartość wynosi

$$b_x 2^{p|y|+|z|} + b_y 2^{|z|}(1 + 2^{|y|} + \ldots + 2^{(p-1)|y|}) + b_z.$$

Zgodnie z małym twierdzeniem Fermata mamy $p \mid 2^{p-1} - 1$ dla dowolnej liczby pierwszej p takiej, że $p > 2$. Równoważnie możemy tą własność zapisać jako $2^{p-1} \equiv 1 \mod p$, co po podniesieniu obu stron do potęgi $|y|$ daje nam $2^{(p-1)|y|} \equiv 1 \mod p$.

Tym samym mamy

$$2^{p|y|} = 2^{(p-1)|y|} 2^{|y|} \equiv 2^{|y|} \mod p.$$

Niech $s = 1 + 2^{|y|} + \ldots + 2^{(p-1)|y|}$, wtedy

$$(2^{|y|}-1)s = 2^{|y|}-1+2^{2|y|}-2^{|y|}+2^{3|y|}-2^{2|y|}+\ldots+2^{p|y|}-2^{(p-1)|y|} = 2^{p|y|}-1 \equiv 2^{|y|}-1 \mod p.$$

Z tego wynika, że $p \,|\, (2^{p-1})(s-1)$. Jako że $1 \leqslant |y| \leqslant n \Rightarrow 2 \leqslant 2^{|y|} \leqslant q2^n < p$, p nie może dzielić 2^{p-1}, a więc $s \equiv 1 \mod p$.

Mamy

$$q = b_x 2^{p|y|+|z|} + b_y 2^{|z|}(1 + 2^{|y|} + \ldots + 2^{(p-1)|y|}) + b_z,$$

czyli

$$q \equiv b_x 2^{|y|+|z|} + b_y 2^{|z|} + b_z \mod p.$$

Prawa strona jest równa numerycznej wartości p, co oznacza, że $q \equiv p \mod p$, a więc $p \,|\, q$. q nie może być liczbą pierwszą i tym samym $q \notin L$.

Otrzymaliśmy sprzeczność z lematem o pompowaniu, więc L nie jest językiem regularnym.

Zadanie 1.4.

2008.4; 2009.4; 2010.4

Niech L będzie dowolnym podzbiorem 0^*. Udowodnij, że L^* jest językiem regularnym.

Rozwiązanie.[3]
Rozpatrzmy dowolny język $L \subseteq 0^*$. Rozważmy trzy przypadki:

1. L jest pusty. Wtedy L^* jest regularny.
2. L jest niepusty i zawiera wyłącznie słowo puste. Wtedy L^* również składa się wyłącznie ze słowa pustego – jest więc regularny.

[2] Autor: Hopcroft, Ullman, *Wprowadzenie do teorii automatów, języków i obliczeń*, Wydawnictwo Naukowe PWN, Warszawa 2003
[3] Autor: nieznany

3. L jest niepusty i istnieje najkrótsze niepuste słowo v.

   Niech $n = |v|$ i niech $L_k = \{w : w \in L^* \ \wedge\ |w| \mod n = k\}$. Pokażemy, że dla każdego k język L_k jest regularny, co znaczy, że L^* jest regularny (jako suma skończonej ilości języków regularnych).

   Weźmy dowolne k. Rozważmy przypadki:

   (a) L_k jest pusty. Wtedy oczywiście L_k jest regularny

   (b) L_k nie jest pusty – wtedy istnieje najkrótsze słowo v', które należy do języka L_k.

   Do języka L_k należą wszystkie wyrazy długości $|v'| + rn$, dla $r \in \mathbb{N}$ (do każdego wyrazu możemy „dokleić” wyraz v).

   Co oznacza, że L_k jest regularny, ponieważ opisuje go wyrażenie $v'(0^n)^*$.

Zadanie 1.5. *2008.5; 2009.5; 2010.5*

Czy język $\{ww^R x : w, x \in \{0,1\}^* \wedge w, x \neq \varepsilon\}$ jest regularny?
Czy język $\{xwx : w, x \in \{0,1\}^* \wedge x \neq \varepsilon\}$ jest regularny?

Rozwiązanie.[4]
Lemat 1. Język $L_1 = \{ww^R x \mid w, x \in \{0,1\}^* \wedge w, x \neq \varepsilon\}$ nie jest językiem regularnym.

Aby pokazać, że język ten nie jest regularny posłużymy się twierdzeniem o indeksie, tzn. pokażemy, że pewien podzbiór tego języka posiada nieskończoną ilość klas abstrakcji względem relacji $\sim_{L_1}$. Weźmiemy słowa postaci $w_i = (01)^i$ i pokażemy, że dla każdych $i, j \in \mathbb{N},\ i \neq j$ zachodzi $w_i \nsim_{L_1} w_j$ z czego będzie wynikało, że każde takie słowo tworzy inną klasę abstrakcji relacji $\sim_{L_1}$, więc jest tych klas nieskończenie wiele.

Załóżmy, że dla pewnych i, j (bez straty ogólności możemy przyjąć, że $i < j$) zachodzi $w_i \sim_{L_1} w_j$. Pokażemy, że istnieje słowo $y \in \{0,1\}^*$ takie, że $w_i y \in L_1$ i $w_j y \notin L_1$.

Niech $y = w_i^R 0 = (10)^i 0$, wtedy $w_i y = (01)^i (10)^i 0 \in L_1$. Należy się zastanowić czy słowo $w_j y = (01)^j (10)^i 0$ może należeć do L_1, czyli czy istnieją takie niepuste w, x, że $u = ww^R x$. Słowo w^R musi zaczynać się na tą samą literę na którą kończy się w (innymi słowy są obok siebie dwie takie same litery) – zwróćmy uwagę, że taka sytuacja występuje w $w_j y$ tylko w jednym miejscu, po wyrazie w_j. Jednak słowo w_i jest krótsze od w_j nie jest więc możliwe aby zawierało odwrócone słowo w – sprzeczność.

Lemat 2. Język $L_2 = \{xwx \mid w, x \in \{0,1\}^* \wedge x \neq \varepsilon\}$ nie jest językiem regularnym.

Dowód ponownie będzie polegał na pokazaniu sprzeczności z twierdzeniem o indeksie. Niech $w_i = 10^i$ dla $i \in \mathbb{N}$. Pokażemy, że dla każdych $i, j \in \mathbb{N}$ takich, że $j < i$ zachodzi $w_i \nsim_{L_2} w_j$.

Weźmy $y = w_i$, wtedy $w_i y = 10^i 10^i \in L_2$, natomiast $u = w_j y = 10^j 10^i$. Gdyby $u = xwx$ dla pewnego $x \neq \varepsilon$, to w u wystąpiłaby dwa razy pierwsza litera słowa x. Ale w u jest tylko jedna pozycja, różna do pierwszej, na której występuje jedynka – w miejscu w którym zaczyna się słowo 10^i. Jednak nie może być to litera rozpoczynająca drugie wystąpienie słowa x ponieważ zgodnie z założeniem $|10^j| < |10^i|$.

[4] Autor: Rafał Nowak

Tym sposobem doprowadziliśmy do upragnionej sprzeczności, co kończy dowód.

Zadanie 1.6. *2008.6; 2009.9; 2010.9*

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

Zadanie 1.7. *2008.7; 2009.6; 2010.6*

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

Rozwiązanie.
Niech L będzie językiem opisanym w treści zadania, niech $\Sigma = \{a, b, c\}$ i $f : \Sigma^* \to \Sigma^*$ będzie funkcją przyporządkowującą dowolnemu słowu nad Σ^* jego najdłuższy sufiks w którym nie powtarza się żadna litera.

Obserwacja 1. Zbiór wartości funkcji f jest równy

$$S = \{\varepsilon, a, b, c, ab, ac, ba, bc, ca, cb, abc, acb, bac, bca, cab, cba\}.$$

Zbiór S zawiera wszystkie możliwe różnoliterowe prefiksy które mogą przyjąć słowa nad Σ^*.

Lemat 1. Dla każdych $w, v \in S$ $w \neq v \Rightarrow [w]_{\sim_L} \neq [v]_{\sim_L}$, gdzie $w \sim_L w'$ w.t.w. gdy $\forall u \in \Sigma^*.\ wu \in L \Leftrightarrow w'u \in L$.

Dowód lematu wymaga pokazania, dla każdej pary słów w, v z S, odpowiedniego kontrprzykładu, to znaczy takiego $u \in \Sigma^*$, że $wu \in L \wedge vu \notin L$.

Przykładowo dla $w = a$, $v = bc$, mamy $u = bc$. Wyraz $wu = abc$ należy do L, natomiast $vu = bcbc$ nie należy do L. Analogicznie pokazujemy pozostałe przypadki.

Wniosek 1. Z obserwacji 1 wynika, że liczba klas abstrakcji relacji $\sim_L$ jest nie większa niż moc zbioru S, natomiast z lematu 1 otrzymujemy, że nie może być mniejsza niż $|S|$. Tak więc z twierdzenia o indeksie otrzymujemy, że minimalna liczba stanów automatu rozpoznającego język L wyniosi $|S|$, czyli 16.

Zadanie 1.8. *2008.8; 2009.7; 2010.7*

Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający język tych słów nad alfabetem $\{a, b, c\}$, które mają przynajmniej 4 symbole i których ostatnie 4 symbole są jednakowe?

Rozwiązanie.
Niech L będzie językiem opisanym w treści zadania, niech $\Sigma = \{a, b, c\}$ i $f : \Sigma^* \to \Sigma^*$ będzie funkcją przyporządkowującą dowolnemu słowu nad Σ^* jego najdłuższy (ale o długości nie większej niż 4) sufiks zawierający tylko jedną literę alfabetu.

Obserwacja 1. Zbiór wartości funkcji f jest równy

$$S = \{\varepsilon, a, b, c, aa, bb, cc, aaa, bbb, ccc, aaaa, bbbb, cccc\}.$$

Zbiór S zawiera wszystkie możliwe jednakowoliterowe prefiksy, długości nie większej niż cztery, które mogą przyjąć słowa nad Σ^*.

Lemat 1. Dla każdych $w, v \in S$ $w \neq v \Rightarrow [w]_{\sim_L} \neq [v]_{\sim_L}$, gdzie $w \sim_L w'$ w.t.w. gdy $\forall u \in \Sigma^*.\ wu \in L \Leftrightarrow w'u \in L$.

Dowód lematu wymaga pokazania, dla każdej pary słów w, v z S, odpowiedniego kontrprzykładu, to znaczy takiego $u \in \Sigma^*$, że $wu \in L \wedge vu \notin L$.

Przykładowo dla $w = aa$, $v = bb$, mamy $u = aa$. Wyraz $wu = aaaa$ należy do L, natomiast $vu = bbaa$ nie należy do L. Analogicznie pokazujemy pozostale przypadki.

Wniosek 1. Z obserwacji 1 wynika, że liczba klas abstrakcji relacji $\sim_L$ jest nie większa niż moc zbioru S, natomiast z lematu 1 otrzymujemy, że nie może być mniejsza niż $|S|$. Tak więc z twierdzenia o indeksie otrzymujemy, że minimalna liczba stanów automatu rozpoznającego język L wyniosi $|S|$, czyli 13.

Zadanie 1.9. *2008.9*

Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający język tych słów nad alfabetem zerojedynkowym, które spośród czterech ostatnich symboli mają dokładnie dwie jedynki (słowo 010 ma jedną jedynkę wśród czterech ostatnich symboli)?

Zadanie 1.10. *2008.10; 2009.8; 2010.8*

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 ilość wystąpień wzorca $b_1 b_2$. Niech język L_n^2 składa się z tych słów nad A_n, które mają parzystą ilość wystąpień wzorca $b_2 b_3$ i tak dalej. Niech wreszcie język L_n^n składa się z tych słów nad A_n, które mają parzystą ilość 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 ilość stanów musi mieć deterministyczny automat skończony rozpoznający L_n?

Zadanie 1.11. *2008.11*

Przez *DFA ze stacją benzynową* będziemy w tym zadaniu rozumieć zwykły deterministyczny automat skończony wraz z jednym dodatkowym wyróżnionym stanem q_{BP} i liczbą naturalną

b. Powiemy, że DFA ze stacją benzynową M akceptuje słowo wejściowe w, jeśli M akceptuje w w zwykłym sensie, a spośród każdych b kolejnych stanów napotkanych w czasie wczytywania w przez M przynajmniej jeden jest stanem q_{BP}. Udowodnij, że nie każdy język regularny może być rozpoznany przez jakiś DFA ze stacją benzynową.

Rozwiązanie.[5]
Niech φ będzie wyrażeniem regularnym postaci 0^*1^*, oczywiście $L(\varphi)$ jest językiem regularnym. Załóżmy nie wprost, że istnieje automat deterministyczny ze stacją benzynową $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ który rozpoznaje ten język.

Obserwacja 1. Istnieje $\alpha \leqslant b$ (gdzie b jest stałą z definicji automatu ze stacją benzynową) takie, że $\hat{\delta}(q_0, 0^\alpha) = q_{BP}$. W przeciwnym wypadku automat nie akceptowałby słowa 0^b, które należy do języka.

Obserwacja 2. Istnieje $\beta \leqslant b$ takie, że $\hat{\delta}(q_0, 1^\beta) = q_{BP}$. W przeciwnym wypadku automat nie akceptowałby słowa 1^b, które należy do języka.

Obserwacja 3. Istnieje takie γ, że $\hat{\delta}(q_{BP}, 0^\gamma) \in F$. W przeciwnym wypadku automat nie akceptowałby słowa $0^{\alpha+\gamma}$, które także należy do języka.

Rozpatrzmy słowo $w = 1^\beta 0^\gamma$, słowo to oczywiście nie należy do języka $L(\varphi)$, ale z wcześniej poczynionych obserwacji wynika, że jest akceptowane przez automat $\mathcal{A}$.

Doszliśmy więc do sprzeczności – nie istnieje DFA ze stacją benzynową akceptujący język regularny $L(0^*1^*)$.

Zadanie 1.12. *2008.P2*

Cykliczny deterministyczny automat skończony (CDFA, nazwa na użytek tej listy zadań) jest bardzo podobny do zwykłego DFA, ma skończony zbiór stanów, stan początkowy, stany akceptujące i funkcję przejścia. Jedyna różnica z DFA polega na sposobie traktowania słowa wejściowego: na końcu słowa, które otrzymuje do wczytania CDFA, dokleja się symbol końca słowa @, a następnie skleja się ten koniec z początkiem słowa, tworząc w ten sposób pętlę po której automat może chodzić bez końca (każde kolejne „wczytanie” słowa wejściowego CDFA zaczyna od stanu, w którym zakończył swoje poprzednie przejście, dzięki temu może „pamiętać” część informacji jaką pozyskał w poprzednich przejściach). CDFA akceptuje słowo wejściowe, jeśli znajdzie się kiedyś nad symbolem @ i będzie wtedy w stanie akceptującym. W przeciwnym razie odrzuca słowo wejściowe (zwróć uwagę, że równoważnie moglibyśmy powiedzieć, że automat odrzuca słowo wejściowe, kiedy po raz drugi znajdzie się nad @ w tym samym stanie nieakceptującym).

Pokaż, że jeśli dla języka L istnieje rozpoznający go $CDFA$, to język ten jest regularny.

[5] Autor: Konrad Drukała

Zadanie 1.13. *2008.P2*

Pokaż, że dla każdego $n \in \mathbb{N}$ istnieją język L_n i rozpoznający go *cykliczny deterministyczny automat skończony* C_n o mniej niż $3n$ stanach takie, że najmniejszy DFA rozpoznający L_n ma więcej niż 2^n stanów.

Zadanie 1.14. *2009.E2*

Niech $p \geqslant 5$ będzie liczba 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 $\{1, 2, \ldots, \frac{p-1}{2}\}$, przy czym liczby czytamy „od prawej”, czyli od najmniej znaczącego bitu (to znaczy pierwszy znak słowa jest ostatnią cyfrą liczby).

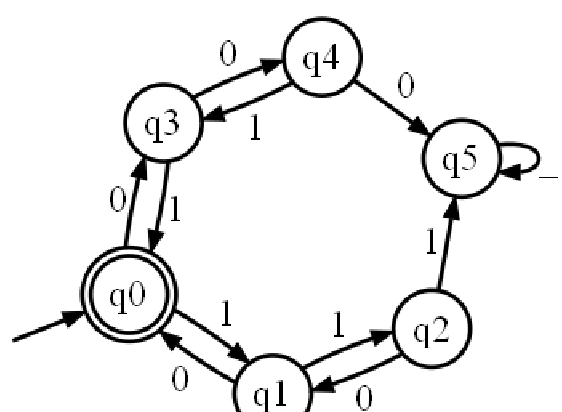
Czy istnieje deterministyczny automat skończony o mniej niż $2p$ stanach rozpoznający język L_p?

2 Wyrażenia regularne

Zadanie 2.1. *2008.12; 2009.10; 2010.10*

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

Rozwiązanie.
Niech automat $\mathcal{A}$ będzie następujący:



Obserwacja 1. $\mathcal{A}$ rozpoznaje język L opisany w treści zadania.

Niech $F = \{q_0\}$ będzie zbiorem stanów akceptujących automatu $\mathcal{A}$. Niech $reg(q_a \xrightarrow{\setminus X} q_z)$ oznacza wyrażenie regularne odpowiadające językowi wszystkich wyrazów powstałych w $\mathcal{A}$ podczas przechodzenia ze stanu q_a do q_z z pominięciem stanów należących do X.

Na podstawie automatu, korzystając z algorytmu elminacji stanów, konstruujemy wyrażenie regularne rozpoznające L:

$$\begin{aligned} reg(q_1 \xrightarrow{\setminus\{q_0\}} q_1) &= w_1 = (10)^* \\ reg(q_3 \xrightarrow{\setminus\{q_0\}} q_3) &= w_2 = (01)^* \\ reg(q_0 \longrightarrow F) = w_3 &= (1w_1 0 + 0w_2 1)^* \end{aligned}$$

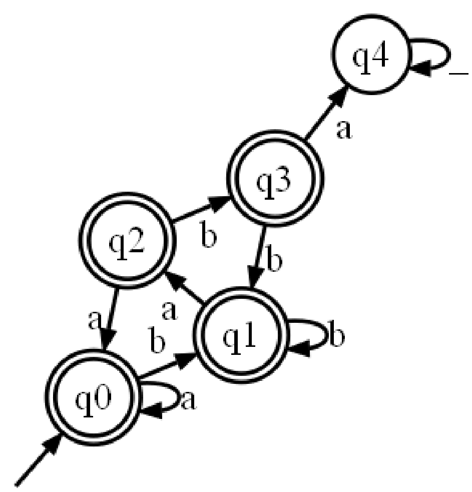
Obserwacja 2. Otrzymane wyrażenie regularne w_3 definiuje język L.

Zadanie 2.2.

2008.13; 2009.11; 2010.11

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

Rozwiązanie.
Niech automat $\mathcal{A}$ będzie następujący:



Obserwacja 1. $\mathcal{A}$ rozpoznaje język L opisany w treści zadania.

Niech $F = \{q_0, q_1, q_2, q_3\}$ będzie zbiorem stanów akceptujących automatu $\mathcal{A}$. Na podstawie automatu, korzystając z algorytmu elminacji stanów, konstruujemy wyrażenie regularne rozpoznające L:

$$\begin{aligned} reg(q_1 \xrightarrow{\setminus\{q_0\}} q_1) &= w_1 = (b + abb)^* \\ reg(q_1 \xrightarrow{\setminus\{q_0\}} F \setminus \{q_0\}) &= w_2 = w_1(\varepsilon + a + ab) \\ reg(q_0 \longrightarrow q_0) &= w_3 = (a + bw_1 aa)^* \\ reg(q_0 \longrightarrow F) &= w_4 = w_3(\varepsilon + bw_2) \end{aligned}$$

Obserwacja 2. Otrzymane wyrażenie regularne w_4 definiuje język L.

Zadanie 2.3. *2008.14; 2009.12; 2010.12*

Dodanie do definicji wyrażeń regularnych pozwolenia na użycie symbolu $\cap$, oznaczającego przekrój języków nie umożliwia reprezentowania nowych zbiorów, wyrażenia stają się jednak krótsze.
Udowodnij, że użycie $\cap$ może wykładniczo skrócić wyrażenie.
Wskazówka: rozważyć język składający się z jednego słowa $(\ldots((a_0a_1)^2a_2)^2\ldots)^2$

Rozwiązanie.
Rozpatrzmy języki nad alfabetem $\Sigma = \{a_0, a_1, \ldots, a_n\}$ takie, że $L_k = L(w_k)$, tworzone za pomocą następujących reguł:

$$\begin{aligned} w_1 &= (a_0a_1)^2 \\ w_2 &= (w_1a_2)^2 = ((a_0a_1)^2a_2)^2 \\ w_n &= (w_{n-1}a_n)^2 = ((\ldots((a_0a_1)^2a_2)^2\ldots)^2a_n)^2 \end{aligned}$$

Obserwacja 1. Wyrażenie opisujące słowo w_n ma długość $\Theta(2^{n+1})$, gdzie przez długość wyrażenia regularnego przyjmujemy ilość występujących w nim symboli (w tym wypadku, po rozwinięciu skrótu notacyjnego ^{2}, jedynymi symbolami występującymi w wyrażeniu będą litery należące do Σ).

Skrócenie wyrażenia poprzez zastosowanie operacji $+$ i * nie jest możliwe, ponieważ sprawiłyby one, że z wyrażenia dałoby się wyprowadzić więcej niż jedno słowo, a więc nie definiowałoby już języka L_n, który zawiera wyłącznie jeden wyraz.

Lemat 1. Istnieje wyrażenie regularne wykorzystujące operację przecięcia, definiujące język L_n i posiadające wielomianową długość.

Niech φ_k będzie wyrażeniem postaci $a_0+\ldots+a_{k-1}+a_{k+1}+\ldots+a_n$, a więc zbiorem wszystkich liter alfabetu poza a_k. Z tego wynika, że φ_k^* generuje wszystkie słowa nad tym alfabetem nie zawierające litery a_k.

Dla $k > 1$ (przyjmujemy, że $w'_1 = w_1$) możemy więc zapisać wyrażenie

$$w'_k = (w'_{k-1}a_k)^* \cap (\varphi_k^*a_k\varphi_k^*a_k)$$

którego długość jest wielomianowa.

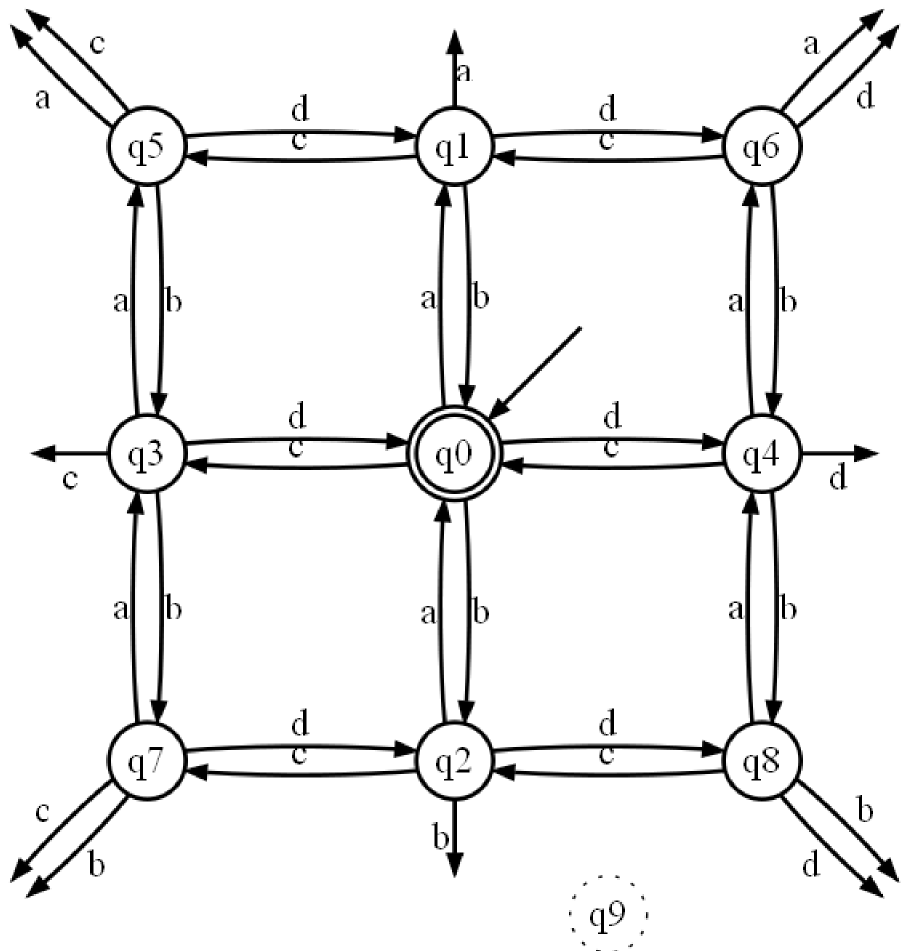
Obserwacja 2. Wyrażenie w' definiuje zadany język, tzn $L_n = L(w'_n)$.

Zauważmy, że lewa strona przekroju generuje wszystkie potęgi słowa $s = w'_{k-1}a_k$, więc wyrażenie po prawej stronie musi być tak dobrane aby poprawne było tylko słowo s^2. Wiemy, że słowo w'_{k-1} nie zawiera litery a_k, więc przekrój z wyrażeniem postaci $\varphi_k^*a_k\varphi_k^*a_k$ wybierze takie potęgi i słowa s, że litera a_k występuje w s^i dokładnie dwa razy. Tak jest tylko dla $i = 2$, co kończy dowód.

Zadanie 2.4. *2008.15; 2009.13; 2010.13*

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

Rozwiązanie.
Automat $\mathcal{A}$, rozpoznający zadany język L ma następującą postać:



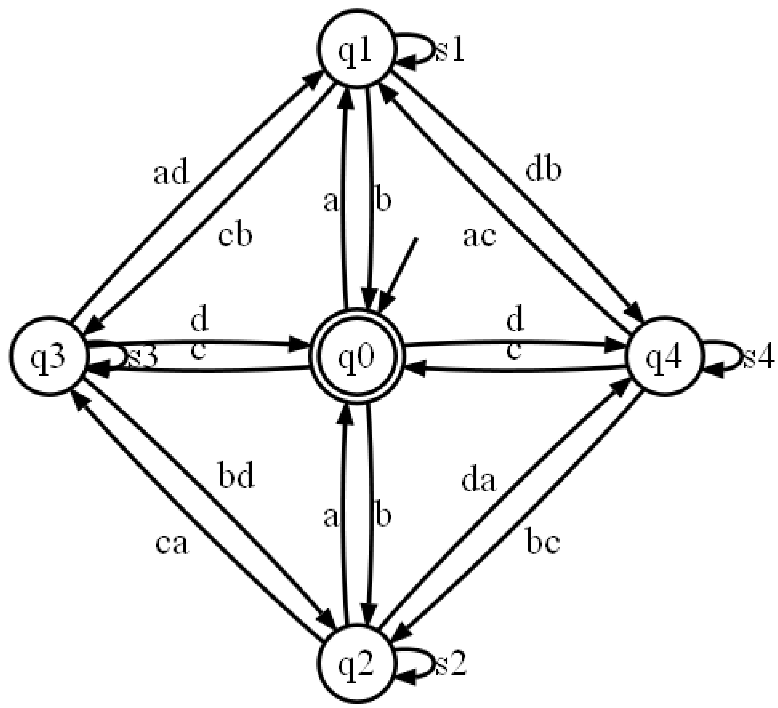
Niech $reg(q_a \xrightarrow{q_v} q_z)$ oznacza wyrażenie regularne odpowiadające językowi wszystkich wyrazów powstałych w automacie podczas przechodzenia ze stanu q_a do q_z przez stan q_v.

Przekształcimy powyższy automat na odpowiadające mu wyrażenie regularne metodą eliminacji stanów, tzn. krawędzie nowo tworzonych równoważnych automatów będziemy etykietować odpowiednimi wyrażeniami regularnymi.

Najpierw wyeliminujemy stany q_5, q_6, q_7, q_8 oraz „otchłań” q_9, zastępując je odpowiednimi wyrażeniami.

$$
\begin{aligned}
reg(q_1 \xrightarrow{q_5,q_6} q_1) &= s_1 = cd + dc \\
reg(q_2 \xrightarrow{q_7,q_8} q_2) &= s_2 = dc + cd \\
reg(q_3 \xrightarrow{q_5,q_7} q_3) &= s_3 = ab + ba \\
reg(q_4 \xrightarrow{q_8,q_6} q_4) &= s_4 = ab + ba
\end{aligned}
$$

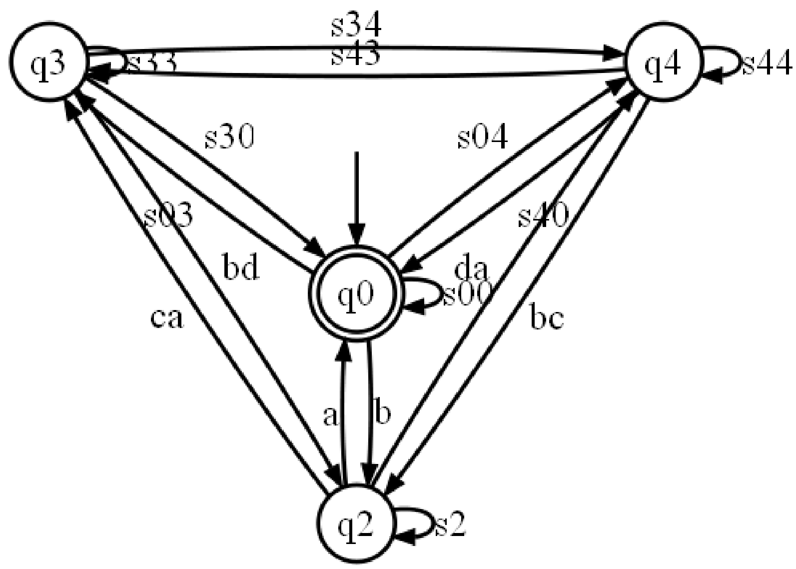
W ten sposób otrzymamy następujący automat $\mathcal{A}'$:



Następnie eliminujemy stan q_1:

$$
\begin{aligned}
reg(q_4 \xrightarrow{q_1} q_3) &= s_{43} = ac(s_1 + ba)^* cb \\
reg(q_3 \xrightarrow{q_1} q_4) &= s_{43} = ad(s_1 + ba)^* db \\
reg(q_0 \xrightarrow{q_1} q_4) &= s_{04} = d + a(ba + s1)^* db \\
reg(q_0 \xrightarrow{q_1} q_3) &= s_{03} = c + a(ba + s1)^* cb \\
reg(q_4 \xrightarrow{q_1} q_0) &= s_{40} = c + ac(ba + s1)^* b \\
reg(q_3 \xrightarrow{q_1} q_0) &= s_{30} = d + ad(ba + s1)^* b \\
reg(q_4 \xrightarrow{q_1} q_4) &= s_{44} = (s_4 + acs_1^* db) \\
reg(q_3 \xrightarrow{q_1} q_3) &= s_{33} = (s3 + ads_1^* cb) \\
reg(q_0 \xrightarrow{q_1} q_0) &= s_{00} = as_1^* b
\end{aligned}
$$

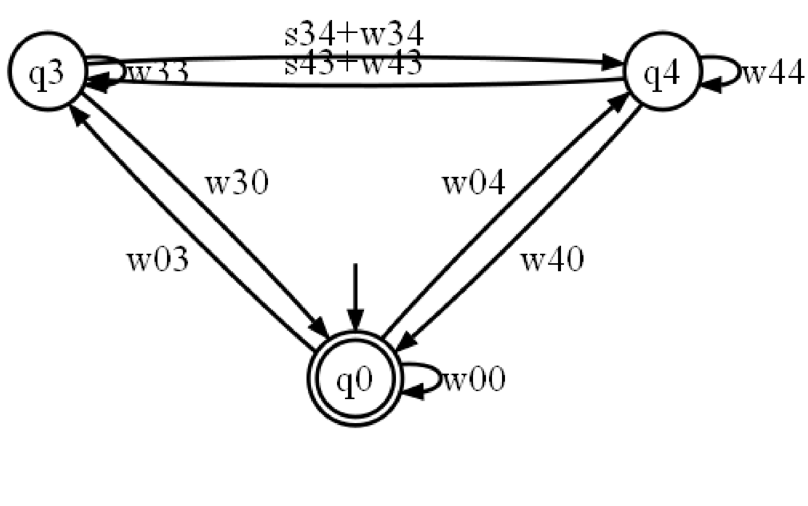
otrzymując automat $\mathcal{A}''$



Eliminujemy stan q_2:

$$
\begin{aligned}
reg(q_4 \xrightarrow{q_2} q_3) &= w_{43} = bc(s_2 + ab)^* ca \\
reg(q_3 \xrightarrow{q_2} q_4) &= w_{34} = bd(s_2 + ab)^* da \\
reg(q_0 \xrightarrow{q_2} q_4) &= w_{04} = s_{04} + b(ab + s_2)^* da \\
reg(q_0 \xrightarrow{q_2} q_3) &= w_{03} = s_{03} + b(ab + s_2)^* ca \\
reg(q_4 \xrightarrow{q_2} q_0) &= w_{40} = s_{40} + bc(ab + s2)^* a \\
reg(q_3 \xrightarrow{q_2} q_0) &= w_{30} = s_{30} + bd(ab + s2)^* a \\
reg(q_4 \xrightarrow{q_2} q_4) &= w_{44} = s_{44} + bcs_2^* da \\
reg(q_3 \xrightarrow{q_2} q_3) &= w_{33} = s_{33} + bds_2^* ca \\
reg(q_0 \xrightarrow{q_2} q_0) &= w_{00} = s_{00} + bs_2^* a
\end{aligned}
$$

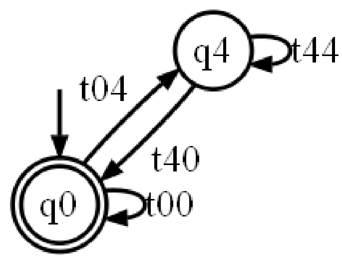
Powstały w ten sposób graf wygląda następująco:



Ostatecznie eliminujemy również stan q_3:

$$\begin{aligned}
reg(q_0 \xrightarrow{q_3} q_4) &= t_{04} = w_{04} + w_{03}w_{33}^*(s_{34} + w_{34}) \\
reg(q_4 \xrightarrow{q_3} q_0) &= t_{40} = w_{40} + (s_{43} + w_{43})w_{33}^*w_{30} \\
reg(q_4 \xrightarrow{q_3} q_4) &= t_{44} = w_{44} + (s_{43} + w_{43})w_{33}^*(s_{34} + w_{34}) \\
reg(q_0 \xrightarrow{q_3} q_0) &= t_{00} = w_{00} + w_{03}w_{33}^*w_{30}
\end{aligned}$$

Co daje automat



Na podstawie którego konstruujemy ostateczne wyrażenie generujące język L którym jest

$$w = (t_{00}^*(t_{04}t_{44}^*t_{40})^*)^*$$

Zadanie 2.5. *2008.16; 2009.15; 2010.15*

A. Które z poniższych wyrażeń są deterministyczne, a które są deterministyczne on-line?
i. $0^*10^* + 0^*$ **ii.** $(0+1)^*1(0+1)$ **iii.** $(0+1)(0+2)^* + (1+2)(0+1)^* + (0+2)(1+2)^*$
B. Znajdź deterministyczne wyrażenie regularne oznaczające język tych wszystkich słów nad alfabetem zerojedynkowym, które zawierają wzorzec 101.

Rozwiązanie.
Lemat 1. Wyrażenie $0^*10^* + 0^*$ jest wyrażeniem deterministycznym, ale nie jest wyrażeniem deterministycznym on-line.

W zależności od tego, czy w rozpatrywanym słowie znajduje się litera 1 czy nie, możemy wyznaczyć mapowanie dla tego słowa na dokładnie jeden sposób, wyrażenie jest więc deterministyczne.

Wyrażenie nie jest deterministyczne on-line ponieważ istnieją dwa słowa 00 i 01, dla których mapowania różnią się na prefiksie 0.

Lemat 2. Wyrażenie $(0+1)^*1(0+1)$ jest wyrażeniem deterministycznym, ale nie jest wyrażeniem deterministycznym on-line.

Wyrażenie jest deterministyczne, ponieważ ostatnie dwie litery słowa (które możemy jednoznacznie zmapować na symbole z wyrażenia) determinują mapowanie pozostałej jego części również w sposób jednoznaczny.

Nie jest to wyrażenie deterministyczne on-line, ponieważ poprawne mapowania słów 011 i

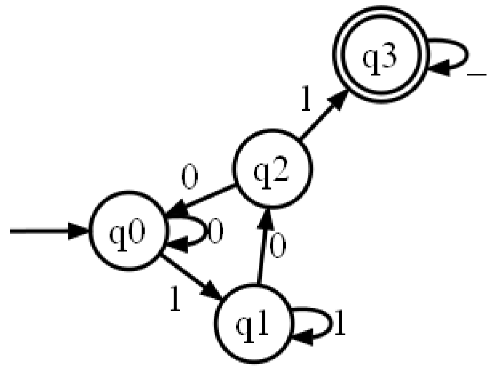
0111 różnią się na prefiksie 01.

Lemat 3. Wyrażenie $(0+1)(0+2)^* + (1+2)(0+1)^* + (0+2)(1+2)^*$ nie jest wyrażeniem deterministycznym.

Istnieje słowo $w = 0$, które posiada więcej niż jedno poprawne mapowanie. Słowo to można wyprowadzić z wyrażenia na dwa sposoby –jako pierwszy lub trzeci element sumy.

Lemat 4. Wyrażenie regularne $w_4 = (0 + 11^*00)^*11^*01(0+1)^*$ jest deterministycznym wyrażeniem definiującym język L tych wszystkich słów nad alfabetem $\{0,1\}$, które zawierają wzorzec 101.

Niech automat $\mathcal{A}$ będzie następujący:



Obserwacja 1. $\mathcal{A}$ rozpoznaje język L.

Niech $F = \{q_3\}$ będzie zbiorem stanów akceptujących automatu $\mathcal{A}$. Na podstawie automatu, korzystając z algorytmu elminacji stanów, konstruujemy wyrażenie regularne rozpoznające L:

$reg(q_1 \xrightarrow{\setminus\{q_0\}} q_1) = w_1 = 1^*$
$reg(q_1 \xrightarrow{\setminus\{q_0\}} F) = w_2 = w_1 01(0+1)^*$
$reg(q_0 \longrightarrow q_0) = w_3 = (0 + 1w_1 00)^*$
$reg(q_0 \longrightarrow F) = w_4 = w_3 1 w_2$

Obserwacja 2. Wyrażenie w_4 jest deterministyczne.

Potrafimy przypisać początkowi dowolnego słowa odpowiednie wyrażenia z nawiasu – jakiekolwiek zero nie należące do ciągu 11^*00 powstało poprzez zastosowanie alternatywnej reguły 0. Następnie znajdujemy (po ewentualnym ciągu jedynek) pierwsze wystąpienie klucza 101. Następnie każdą literę mapujemy jednoznacznie, na odpowiadający jej element w wyrażeniu $(0+1)^*$.

Zadanie 2.6.

2008.17; 2009.16; 2010.16

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

Zadanie 2.7. ***2008.18; 2009.17; 2010.17***

Znajdź deterministyczne on-line wyrażenie regularne oznaczające język tych wszystkich słów nad alfabetem zerojedynkowym, które zawierają jedną lub dwie jedynki.

Rozwiązanie.
Niech $\varphi = 0^*10^*(\varepsilon + 10^*)$.

Obserwacja 1. $L(\varphi)$ jest językiem wszystkich słów nad alfabetem $\{0, 1\}$ które mają jedną lub dwie jedynki.

Obserwacja 2. Wyrażenie φ jest deterministyczne on-line.

Każde zero występujące na początku wyrazu jest jednoznacznie zmapowane. Następująca po tym (być może pustym) ciągu zer jedynka również. Kolejny (być może pusty) ciąg zer także ma jednoznaczne odwzorowanie na odpowiedni fragment wyrażenia.

W pewnym momencie ten ciągu zer się kończy. Następuje wtedy koniec wyrazu, który zostaje zmapowany na ε i wyraz jest akceptowany, lub druga jedynka. W przypadku jedynki, następuje po niej kolejny (być może pusty) ciąg zer i ta końcówka słowa musi zostać jednoznacznie zmapowana na podwyrażenie 10^*.

Dowolne funkcje mapujące zgadzają się na identycznych prefiksach słów, φ jest więc wyrażeniem deterministyczm on-line.

3 Niedeterministyczne Automaty Skończone

Zadanie 3.1. ***2008.19; 2009.18; 2010.18***

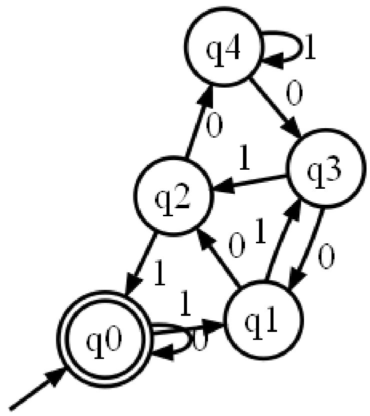
Skonstruuj niedeterministyczny automat skończony rozpoznający język tych słów nad $\{0, 1\}^*$ które jako liczba w systemie dwójkowym dzielą się przez 5, przy czym liczba jest wczytywana

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

Rozwiązanie.[6]
Lemat 1. Istnieje automat rozpoznający język L_a liczb podzielnych przez 5 wczytywanych w systemie dwójkowym od najbardziej znaczącego bitu.

Przy konstrukcji automatu skorzystamy z faktu, że $n_{w0} = 2n_w$ oraz $n_{w1} = 2n_w + 1$, gdzie n_w oznacza wartość słowa w interpretowanego jako liczba binarna. Stany Q_i dla $i = 0, \ldots, 4$ będą oznaczały, że wczytane słowo przystaje do i modulo 5.

[6] Autor: Wojtek „stdin” Iżykowski



Obserwacja 1. Powyższy DFA $\mathcal{A}$ rozpoznaje język L_a.

Lemat 2. Istnieje automat rozpoznający język L_b liczb podzielnych przez 5 wczytywanych w systemie dwójkowym od najmniej znaczącego bitu.

Zauważmy, że $2^0 \equiv 1 \mod 5$, $2^1 \equiv 2 \mod 5$, $2^2 \equiv 4 \mod 5$, $2^3 \equiv 3 \mod 5$, $2^4 \equiv 1 \mod 5$, itd. Dodanie cyfry 0 nie zmieni nam reszty z dzielenia wczytywanej liczby przez 5, natomiast wczytanie liczby 1 zmieni nam wczytywaną liczbę o 1, 2, 4 lub 3 w zależności od reszty z dzielenia pozycji wczytywanej jedynki przez 4.

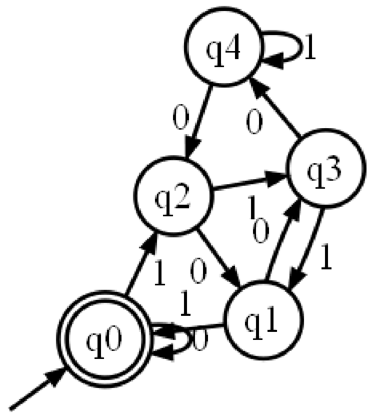
Na podstawie tych obserwacji skonstruujemy NDFA $M = \langle \Sigma, Q, q_0, F, \delta \rangle$, gdzie:

$$
\begin{aligned}
\Sigma &= \{0, 1\} \\
Q &= \langle \mathbb{Z}_5, \mathbb{Z}_4 \rangle \\
q_0 &= \langle 0, 0 \rangle \\
F &= \{\langle 0, i \rangle : i \in \mathbb{Z}_4\} \\
\delta &= \{(\langle i, j \rangle, 1, \langle (i + val(j)) \mod 5, (j+1) \mod 4 \rangle) : i \in \mathbb{Z}_5, j \in \mathbb{Z}_4\} \\
&\quad \cup \{(\langle i, j \rangle, 0, \langle i, (j+1) \mod 4 \rangle) : i \in \mathbb{Z}_5, j \in \mathbb{Z}_4\} \\
&\quad \text{gdzie } val = \{\langle 0, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 4 \rangle, \langle 3, 3 \rangle\}
\end{aligned}
$$

W stanie $q = \langle i, j \rangle$ pamiętamy resztę i z dzielenia wczytanej liczby modulo 5 oraz resztę j z dzielenia przez 4 pozycji wczytywanej liczby.

*Lemat 3 (alternatywne rozwiązanie podpunktu **b**).* Istnieje automat rozpoznający język L_b wykorzystujący konstrukcję przedstawioną w lemacie 1.

Automat taki można stworzyć (powołując się na następne zadanie) zamieniając stany początkowe z końcowymi (automat $\mathcal{A}$ po tej operacji pozostaje bez zmian), oraz odwracając funkcję przejścia. Tak skonstruowany automat $\mathcal{B}$ wygląda następująco:



Obserwacja 1. Automat $\mathcal{B}$ rozpoznaje język L_b.

Zadanie 3.2.

2008.20; 2009.19; 2010.19

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

Rozwiązanie.
Mamy dany NDFA $M = \langle \Sigma, Q, q_0, F, \delta \rangle$ rozpoznający język L. Konstruujemy NDFA $M' = \langle \Sigma, Q', p_0, F', \delta' \rangle$ w następujący sposób:

1. Odwracamy wszystkie łuki w diagramie przejść M.
2. Czynimy stan początkowy M jedynym stanem akceptującym nowego automatu.
3. Tworzymy nowy stan p_0 z ε-przejściami do wszystkich stanów z F.

Mamy więc $Q' = Q \cup \{p_0\}$, $F' = \{q_0\}$ i $\delta' = \{\langle q_i, a, q_j \rangle : \delta(q_j, a, q_i)\} \cup \{\langle p_0, \varepsilon, q \rangle : q \in F\}$.

Lemat 1. $\hat{\delta}(q_i, w, q_j) \Leftrightarrow \hat{\delta}'(q_j, w^R, q_i)$.

Dowód (indukcyjny po długości słowa):

1. Niech $|w| \leqslant 1$. Wtedy $w = a$ dla pewnego $a \in \Sigma$.

   Mamy więc $\hat{\delta}(q_i, w, q_j) \Leftrightarrow \delta(q_i, a, q_j) \Leftrightarrow \delta'(q_j, a^R, q_i) \Leftrightarrow \hat{\delta}'(q_j, w^R, q_i)$.

2. Załóżmy, że dla wszystkich słów w' krótszych niż w twierdzenie jest prawdziwe. Pokażemy, że dla w również jest ono prawdziwe. Niech $w = aw'$, gdzie $a \in \Sigma$. Wtedy:

   $\hat{\delta}(q_i, w, q_j) \Leftrightarrow \hat{\delta}(q_i, aw', q_j) \Leftrightarrow \exists q_k.\ \delta(q_i, a, q_k) \wedge \hat{\delta}(q_k, w', q_j) \overset{\text{zał.ind.}}{\Leftrightarrow} \exists q_k.\ \hat{\delta}'(q_j, {w'}^R, q_k) \wedge \delta'(q_k, a, q_i) \Leftrightarrow \hat{\delta}'(q_j, {w'}^R a, q_i) \Leftrightarrow \hat{\delta}(q_j, w^R, q_i)$.

Lemat 2. Automat M' rozpoznaje język L^R.

Dowód równoważności $w \in L(M) \Leftrightarrow w^R \in L(M')$:

(⇒) Weźmy dowolne słowo $w \in L(M)$. Wtedy spełniona jest relacja $\hat{\delta}(q_0, w, q_k)$, gdzie $q_k \in F$. Zatem, na mocy lematu 1, $\hat{\delta}'(q_k, w^R, q_0)$ i $\delta'(p_0, \varepsilon, q_k)$, stąd $\hat{\delta}'(p_0, w, q_0)$. $q_0 \in F'$, więc słowo w^R jest akceptowalne przez M', czyli $w^R \in L(M')$.

(⇐) Weźmy dowolne słowo $w \in L(M')$. Wtedy spełniona jest relacja $\hat{\delta}'(p_0, w, q_0)$ (ponieważ mamy tylko jeden stan akceptujący q_0). Z p_0 mamy jedynie ε-przejścia do stanów z F, zatem $\delta'(p_0, \epsilon, q_k)$ oraz $\hat{\delta}'(q_k, w, q_0)$, dla pewnego $q_k \in F$. Wtedy, na mocy lematu 1, $\hat{\delta}(q_0, w^R, q_k)$. $q_k \in F$, więc słowo w^R jest akceptowalne przez M, czyli $w^R \in L(M)$

Zadanie 3.3. *2008.21; 2009.20; 2010.20*

Wiadomo, że język L jest językiem regularnym. Pokaż, że język
$\{w : \exists n \in \mathbb{N}.\ \exists v \in L.\ w^n = v\}$
jest językiem regularnym. Przez w^n rozumiemy słowo w skonkatenowane ze sobą n razy.

Rozwiązanie.
Niech $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ będzie automatem rozpoznającym język L, L' będzie językiem z treści zadania tzn. $L' = \{w : \exists n \in \mathbb{N}.\ \exists v \in L.\ w^n = v\}$ i niech $n = |Q|$.

Konstruujemy automat $\overline{\mathcal{A}} = \langle \Sigma, \overline{Q}, \overline{q}_0, \overline{F}, \overline{\delta} \rangle$ taki, że:

$$\begin{aligned} \overline{Q} &= Q^n \\ \overline{q}_0 &= \langle q_0, q_1, \ldots, q_{n-1} \rangle \\ \overline{F} &= \{\langle r_0, \ldots, r_{n-1} \rangle : \\ &\qquad \exists k_0, \ldots, k_i < n, i \in \mathbb{N}, i \leqslant n.\ k_0 = 0 \wedge \forall j < i.\ r_{k_j} = q_{k_{j+1}} \wedge q_{k_i} \in F\} \\ \overline{\delta}(\langle r_0, \ldots, r_{n-1} \rangle, a) &= \langle \delta(r_0, a), \ldots, \delta(r_{n-1}, a) \rangle. \end{aligned}$$

Obserwacja 1. Automat ten skonstruowany jest w taki sposób, że jego stany to krotka wszystkich stanów automatu $\mathcal{A}$. Po wczytaniu pewnego słowa w, automat $\overline{\mathcal{A}}$ zawiera informację o stanach w których znalazłby się $\mathcal{A}$ po wczytaniu tego słowa rozpoczynając od dowolnego stanu z Q. Stan $\overline{\mathcal{A}}$ jest więc stanem akceptującym, gdy istnieje ciąg i stanów pośrednich takich, że $\mathcal{A}$ po wczytaniu w ze stanu q_0 znajduje się w stanie q_j, po wczytaniu w ze stanu q_j znajduje się w stanie q_k i tak, aż do i-tego stanu w ciągu, który jest stanem akceptującym w $\mathcal{A}$. Ponieważ w takim przypadku $w^i \in L$, więc w należy do języka L'.

Dowód równoważności $w \in L' \Leftrightarrow w \in L(\overline{\mathcal{A}})$.

(⇒) Załóżmy, że mamy słowo w które należy do L'. Skoro tak, to $\exists i \in \mathbb{N}.\ w^i \in L$. Muszą więc w automacie $\mathcal{A}$ istnieć stany $q_1, \ldots, q_i$ takie, że $\hat{\delta}(q_0, w) = q_1, \ldots, \hat{\delta}(q_{i-1}, w) = q_i$ i q_i jest stanem akceptującym. Ale w takim razie automat $\overline{\mathcal{A}}$ przyjmie po wczytaniu słowa w stan $\langle q_1, \ldots, q_i, \ldots \rangle$ w której to krotce będą się znajdować wszystkie ze stanów $q_1, \ldots, q_i$, a więc zgodnie z definicją zbioru $\overline{F}$ stan ten będzie stanem akceptującym i w zostanie przez automat $\overline{\mathcal{A}}$ zaakceptowane.

(⇐) Załóżmy, że automat $\overline{\mathcal{A}}$ zaakceptuje słowo w. Oznacza to, że istnieje odpowiedni ciąg $k_0, \ldots, k_i$ dla pewnego $i \in \mathbb{N}$, a więc słowo w^i należy do języka L, co oznacza że słowo w należy do języka L'.

Zadanie 3.4. *2008.22; 2009.21; 2010.21*

Udowodnij, że jeśli L_1 i L_2 są językami regularnymi nad pewnym alfabetem $\mathcal{A}$ to również języki $L_1 \cup L_2$, $L_1 \cap L_2$ i $\mathcal{A}^* - L_1$ są językami regularnymi.

Rozwiązanie.
Niech L_1, L_2 będą językami regularnymi nad alfabetem Σ.

Lemat 1. Język $\overline{L_1} = \Sigma^* \setminus L_1$ jest językiem regularnym

Skoro L_1 jest regularny, to istnieje DFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ taki, że $L_1 = L(\mathcal{A})$. Wtedy możemy stworzyć automat $\overline{\mathcal{A}} = \langle \Sigma, Q, q_0, Q \setminus F, \delta \rangle$ który akceptuje język $\overline{L_1}$.

Zauważmy, że jeśli pewne słowo należało do języka L_1 to nie zostanie zaakceptowane przez $\overline{\mathcal{A}}$, zaś jeśli nie należało do L_1 to na pewno należy do $\overline{L_1}$ (zostanie zaakceptowane przez $\overline{\mathcal{A}}$).

Lemat 2. Język $L_1 \cup L_2$ jest językiem regularnym

Skoro L_1 i L_2 są regularne, to istnieją wyrażenia regularne φ_1 i φ_2 takie, że $L_1 = L(\varphi_1)$ oraz $L_2 = L(\varphi_2)$.

Wtedy $\psi = \varphi_1 + \varphi_2$ jest wyrażeniem regularnym takim, że $L_1 \cup L_2 = L(\varphi_1) \cup L(\varphi_2) = L(\psi)$, a więc $L_1 \cup L_2$ jest językiem regularnym.

Lemat 3. Język $L_1 \cap L_2$ jest językiem regularnym

Skoro L_1 i L_2 są regularne, to istnieją DFA $A_1 = \langle \Sigma, Q_1, q_{10}, F_1, \delta_1 \rangle$ i $A_2 = \langle \Sigma, Q_2, q_{20}, F_2, \delta_2 \rangle$ takie, że $L_1 = L(A_1)$ oraz $L_2 = L(A_2)$.

Wtedy automat $A = \langle \Sigma, Q_1 \times Q_2, \langle q_{10}, q_{20} \rangle, F_1 \times F_2, \delta \rangle$, gdzie

$$\delta(\langle q_1, q_2 \rangle, a) = \langle \delta_1(q_1, a), \delta_2(q_2, a) \rangle$$

akceptuje język $L_1 \cap L_2$.

Alternatywnie można skorzystać z lematów 1 i 2 oraz praw DeMorgana, a więc wykorzystać własność

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

.

Zadanie 3.5. *2008.23; 2009.22; 2010.22*

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

Rozwiązanie.[7]
Niech $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ będzie automatem rozpoznającym język L i niech $n = |Q|$.

[7] Pomysł: Kuba „q” Michaliszyn

Wtedy konstruujemy automat $\overline{\mathcal{A}} = \langle \Sigma, \overline{Q}, \overline{q}_0, \overline{F}, \overline{\delta} \rangle$ taki, że:

$$\begin{aligned}
\overline{Q} &= Q^n \times 2^Q \\
\overline{q}_0 &= \langle q_0, q_1, \ldots, q_{n-1}, \{q_0\} \rangle \\
\overline{F} &= \{ \langle r_0, \ldots, r_{n-1}, P \rangle | \exists k < n q_k \in P \wedge r_k \in F \} \\
\overline{\delta}(\langle r_0, \ldots, r_{n-1}, P \rangle, a) &= \langle \delta(r_0, a), \ldots, \delta(r_{n-1}, a), \{ q \in Q | \exists p \in P.\ \exists b \in \Sigma.\ \delta(p, b) = q \} \rangle
\end{aligned}$$

Automat ten skonstruowany jest w taki sposób, że jego stany to krotka: zbiór stanów Q w których mógłby się znaleźć automat rozpoznający przynależność słowa v do języka L po $k \in \mathbb{N}$ krokach, wszystkie możliwe stany w których mógłby się znaleźć $\mathcal{A}$ po przeczytaniu słowa w zaczynając od dowolnego stanu. Stan jest akceptujący gdy istnieje taki stan q_k że $\mathcal{A}$ po przeczytaniu v znalazłby się w tym stanie oraz gdyby stanem początkowym w $\mathcal{A}$ dla słowa w był stan q_k to vw należałby do L.

Lemat 1. Automat $\overline{\mathcal{A}}$ rozpoznaje język $L/2$, tzn zachodzi równoważność: w należy do języka $L/2$ wtw. $\overline{\mathcal{A}}$ akceptuje słowo w.

($\Rightarrow$) Załóżmy, że mamy słowo w które należy do języka $L/2$. Skoro $w \in L/2$ to $vw \in L$ i $|v| = |w| = k$. Wiemy więc, że istnieje stan $q_i \in Q$ w automacie $\mathcal{A}$, że $\delta(q_0, v) \to q_i$ i $\delta(q_i, w) \to q_f$ gdzie $q_f \in F$ ponieważ słowo vw zostało zaakceptowane.

Ale jeśli taki stan by istniał to w i-tym kroku automatu $\overline{\mathcal{A}}$ stan q_i należałby do zbioru P ponieważ z założenia istnieje ciąg liter z Σ którymi da się dojść z q_0 do q_i - ten ciąg to v. Istniałby także stan $r_i = q_f \in F$ do którego można dojść słowem w rozpoczynając od stanu q_i. Z tego wynika, że stan przyjęty przez automat $\overline{\mathcal{A}}$ byłby dla słowa w akceptujący.

($\Leftarrow$) Załóżmy, że automat $\overline{\mathcal{A}}$ zaakceptuje słowo w długości k. Jeśli $\overline{\mathcal{A}}$ zaakceptował w to znaczy, że dla pewnego stanu q_i należącego do automatu $\mathcal{A}$ istnieje słowo v długości k (jego utworzenie wymagało tylu znaków co przeczytanie słowa w) takie, że po jego wczytaniu $\mathcal{A}$ znajdzie się w tym stanie. Wiemy także, że słowo w zaaplikowane do automatu $\mathcal{A}$ poczynając od stanu q_i zostanie zaakceptowane. A więc słowo vw należy do L i tym samym słowo w należy do $L/2$.

Wniosek 1. Język $L/2$ jest językiem regularnym ponieważ istnieje DFA który go rozpoznaje.

Zadanie 3.6.

2008.24; 2009.23; 2010.23

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

Rozwiązanie.
Niech $\mathcal{A}_1$ i $\mathcal{A}_2$ będą dowolnymi automatami deterministycznymi i $L_1 = L(\mathcal{A}_1)$ oraz $L_2 = L(\mathcal{A}_2)$.

Fakt 1. Z udowodnionych w poprzednich zadaniach własności wiemy, że język $L' = (L_1 \cup L_2) \cap (\overline{L_1} \cup \overline{L_2})$ jest regularny i mając automaty $\mathcal{A}_1$ i $\mathcal{A}_2$ jesteśmy w stanie skonstruować DFA $\mathcal{A}'$ rozpoznający język L'.

Obserwacja 1. Zauważmy, że $L_1 = L_2 \Leftrightarrow L' = \emptyset$.

Korzystając z przedstawionych własności możemy skonstruować algorytm rozstrzygający równość języków L_1 i L_2. Traktujemy automat $\mathcal{A}'$ jako graf o wyróżnionym wierzchołku początkowym i przeszukujemy go (na przykład wszerz) spamiętując odwiedzone wierzchołki.

Jeśli algorytm odnalzał ścieżkę ze stanu startowego do stanu akceptujacego, znaczy to, że $L' \neq \emptyset$ i tym samym $L_1 \neq L_2$. Jeśli natomiast algorytm odwiedzi wszystkie wierzchołki osiągalne ze stanu startowego i nie znajdzie stanu akceptującego, to języki L_1 i L_2 są równe.

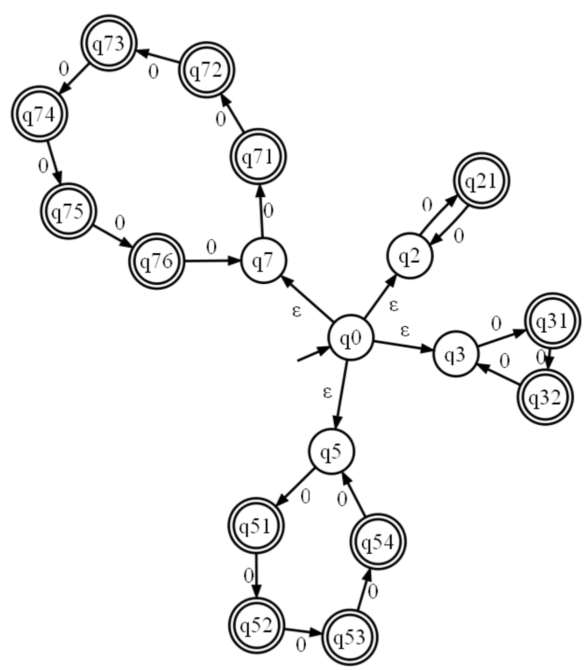
Zadanie 3.7. *2008.25; 2009.24; 2010.24*

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

Rozwiązanie.
Niech $L \subseteq 0^*$ taki, że $L = \{0^n : 210 \not| \, n\}$.

Obserwacja 1. Korzystając z tego, że $210 = 2 \cdot 3 \cdot 5 \cdot 7$, możemy stworzyć NDFA rozpoznający jezyk L i posiadający 18 stanów.

Automat ten ma następującą postać:



Lemat 1. Nie istnieje $NDFA$ o mniej niż 200 stanach, akceptujący język $\overline{L} = \{0^n : 210 \mid \, n\}$.

Załóżmy nie wprost, że taki NDFA $\mathcal{A}$ istnieje. W takim razie akceptuje słowo $w = 0^{210}$ jako należące do języka. Ponieważ długość tego słowa jest większa niż liczba stanów $\mathcal{A}$, z zasady szufladkowej Dirichleta wynika, że istnieje stan q_k który występuje w ścieżce akceptującej w dwukrotnie. Ścieżka ta jest więc postaci $q_0 \to \dots q_p \to q_k \to \dots \to q_k \to q_n \dots \to q_f$.

To oznacza, że pewne słowo v krótsze od w, o ścieżce $q_0 \to \dots q_p \to q_k \to q_n \dots \to q_f$ zostanie zaakceptowane przez automat $\mathcal{A}$ pomimo, iż nie należy do języka $\overline{L}$.

Dowiedliśmy więc, że nie istnieje automat niedeterministyczny o mniej niż 200 stanach rozpoznający dopełnienie języka L.

Zadanie 3.8.

2008.26; 2009.14; 2010.14

Czy istnieje wyrażenie regularne ϕ, takie, że $L_{a\phi} = L_{\phi b}$? Czy istnieje wyrażenie regularne ϕ, takie, że $L_{a^*\phi} = L_{\phi b^*}$?

Rozwiązanie.
Lemat 1. Nie istnieje wyrażenie regularne ϕ takie, że $L_{a\phi} = L_{\phi b}$.

Weźmy słowo $w \in L_\phi$ takie, że na początku w znajduje się najmniejsza ilość liter a (oznaczmy ich liczbę przez k).

Wtedy słowa wb należącego do języka $L_{\phi b}$ nie da się wygenerować za pomocą wyrażenia $a\phi$, ponieważ będzie miało k liter a z przodu, zaś każde słowo w $a\phi$ będzie miało na początku co najmniej $k+1$ liter a. Z tego wynika, że takie wyrażenie regularne ϕ nie istnieje.

Lemat 2. Istnieje wyrażenie regularne ϕ takie, że $L_{a^*\phi} = L_{\phi b^*}$.

Dla $\phi = a^*b^*$ mamy następujący ciąg równoważności:

$$a^*\phi \Leftrightarrow a^*a^*b^* \Leftrightarrow a^*b^* \Leftrightarrow a^*b^*b^* \Leftrightarrow \phi b^*.$$

Skoro powyższe wyrażenia regularne są równoważne, to języki $L_{a^*\phi}$ i $L_{\phi b^*}$ generowane przez te wyrażenia są równe.

Zadanie 3.9.

2008.27

Przez *NDFA ze stacją benzynową* będziemy w tym zadaniu rozumieć zwykły niedeterministyczny automat skończony wraz z jednym dodatkowym wyróżnionym stanem q_{BP} i liczbą naturalną b. Powiemy, że NDFA ze stacją benzynową M akceptuje słowo wejściowe w, jeśli istnieje taka ścieżka akceptująca słowo w w M w zwykłym sensie, że spośród każdych b kolejnych stanów na tej ścieżce, przynajmniej jeden jest stanem q_{BP}. Udowodnij, że nie każdy język regularny może być rozpoznany przez jakiś NDFA ze stacją benzynową.

Rozwiązanie.[8]
Niech φ będzie wyrażeniem regularnym postaci 0^*1^*, oczywiście $L(\varphi)$ jest językiem regularnym. Załóżmy nie wprost, że istnieje automat niedeterministyczny ze stacją benzynową

[8] Autor: Konrad Drukała

$\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ który rozpoznaje ten język.

Obserwacja 1. Istnieje $\alpha \leqslant b$ (gdzie b jest stałą z definicji automatu ze stacją benzynową) takie, że $\hat{\delta}(q_0, 0^\alpha) = q_{BP}$. W przeciwnym wypadku automat nie miałby możliwości zaakceptowania słowa 0^b, które należy do języka.

Obserwacja 2. Istnieje $\beta \leqslant b$ takie, że $\hat{\delta}(q_0, 1^\beta) = q_{BP}$. W przeciwnym wypadku automat nie miałby możliwości zaakceptowania słowa 1^b, które należy do języka.

Obserwacja 3. Istnieje takie γ, że $\hat{\delta}(q_{BP}, 0^\gamma) \in F$. W przeciwnym wypadku automat nie miałby możliwości zaakceptowania słowa $0^{\alpha+\gamma}$, które także należy do języka.

Rozpatrzmy słowo $w = 1^\beta 0^\gamma$, słowo to oczywiście nie należy do języka $L(\varphi)$, ale z wcześniej poczynionych obserwacji wynika, że w automacie $\mathcal{A}$ istnieje scieżka akceptująca to słowo.

Doszliśmy więc do sprzeczności – nie istnieje DFA ze stacją benzynową akceptujący język regularny $L(0^*1^*)$.

Zadanie 3.10. ***2008.E1; 2009.25; 2010.25***

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 niedeterministycznycgo 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.

Rozwiązanie.[9]
Niech $P = \{p : p \text{ jest liczbą pierwszą}\}$ i $R = \{2 \cdot p : p \text{ jest liczbą pierwszą} \wedge p > 3\}$.

Fakt 1. Zbiory P i R są nieskończone.

Lemat 1. Dla każdego k, $d(L_k) = k$.

Konstrukcja automatu $\mathcal{A} = \langle \{0\}, Q, q_0, F, \delta \rangle$ o k stanach rozpoznającego L_k jest prosta i wygląda następująco:

$$\begin{aligned} Q &= \{q_0, \ldots, q_{k-1}\} \\ F &= \{q_1, \ldots, q_{k-1}\} \\ \delta(\langle q_i, 0 \rangle) &= \begin{cases} q_{i+1} & \text{dla } i < k-1 \\ q_0 & \text{dla } i = k-1 \end{cases} \end{aligned}$$

Załóżmy nie wprost, że istnieje DFA $\mathcal{B}$ o mniej niż k stanach, rozpoznający język L_k. Wtedy z zasady szufladkowej Dirichleta, ścieżka odrzucająca 0^k w $\mathcal{B}$ ma postać $q_0 \to \ldots \to q_{i-1} \to q_i \to \ldots \to q_i \to q_{i+1} \to \ldots \to q_n$, gdzie q_0 jest stanem początkowym, a $q_n \notin F$, a więc powien stan q_i występuje na tej ścieżce więcej niż raz.

Istnieje więc słowo w^l, $0 < l < k$, takie że w automacie $\mathcal{B}$ istnieje ścieżka $q_0 \to \ldots \to q_{i-1} \to q_i \to q_{i+1} \to \ldots \to q_n$ odrzucająca słowo 0^l. Ale 0^k jest z definicji języka najkrótszym niepustym słowem nad 0^* które do niego nie należy.

[9] Autor: Aleksander „alistra” Balicki

Nie istnieje więc automat deterministyczny rozpoznający L_k o mniej niż k stanach.

Lemat 2. Jeśli $k \in P$, to $n(L_k) = k$.

Z lematu 1 wynika, że potrafimy skonstruować DFA (a tym samym NDFA) o k stanach rozpoznający L_k.

Załóżmy nie wprost, że potrafimy skonstruować NDFA $\mathcal{C}$ rozpoznający L_k o mniej niż k stanach. Wtedy, z zasady szufladkowej Dirichleta wynika, że ścieżka akceptująca dla słowa $w = 0^{k+1} \in L_k$ jest postaci $q_0 \to \ldots \to \overbrace{q_i \to \ldots}^{c} \to q_i \to \ldots \to q_f$, a więc zawiera jakiś cykl długości c.

Pokażemy, że powtarzając ten cykl r razy zaakceptujemy słowo o długości będącej wielokrotnością k. Tzn, że prawdziwa jest formuła

$$\exists n \in \mathbb{N}, r \in \mathbb{N}.\ (k + 1 - c) + r \cdot c = n \cdot k.$$

Którą równoważnie możemy przedstawić jako

$$\exists u \in \mathbb{N}.\ 1 + u \cdot c \equiv_k = 0 \iff \exists u \in \mathbb{N}.\ u \cdot c \equiv_k k - 1.$$

W grupie addytywnej $\mathbb{Z}_k$ dla k będących liczbami pierwszymi, każdy element jest generatorem. W szczególności oznacza to, że istnieje takie u które pomnożone przez c da element przystający do $k - 1$.

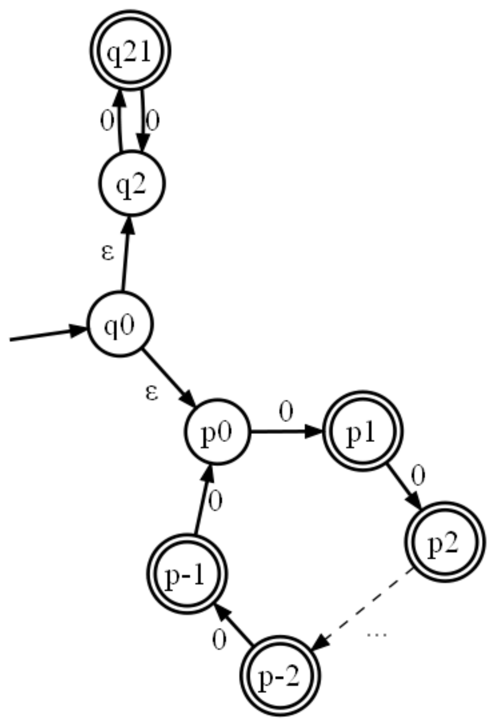
Oznacza to, że w automacie $\mathcal{C}$ istnieje ścieżka akceptująca pewne słowo postaci $0^{n \cdot k}$, a więc takie, które nie należy do języka L_k. Nie istnieje więc automat niedeterministyczny rozpoznający L_k dla $k \in P$ o mniej niż k stanach.

Wniosek 1. Z lematów 1 i 2 wynika, że dla każdego $k \in P$, zachodzi $d(L_k) = n(L_k)$

Lemat 3. Jeśli $k \in R$, to $n(L_k) \leqslant 1 + 2 + \frac{k}{2}$.

Niech $k \in R$. Wtedy istnieje takie $p \in P$, że $k = 2 \cdot p$. Oznacza to, że dla dowolnego n naturalnego mamy $k \mid n \Leftrightarrow 2 \mid n \wedge p \mid n$.

Konstrukcja NDFA o $1+2+\frac{k}{2}$ stanach, rozpoznającego język L_k i korzystającego z powyższego spostrzeżenia wygląda następująco:



Wniosek 2. Z lematów 1 i 3 wynika, że dla każdego $k \in R$, zachodzi $d(L_k) \neq n(L_k)$

Zadanie 3.11. ***2009.E1***

Niech $p \geqslant 5$ będzie liczba 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 $\{1, 2, \ldots, \frac{p-1}{2}\}$, przy czym liczby czytamy „od prawej”, czyli od najmniej znaczącego bitu (to znaczy pierwszy znak słowa jest ostatnią cyfrą liczby).

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

Zadanie 3.12. ***2009.P2***

Niech M_n będzie językiem tych słów nad alfabetem $\{1, 2, \ldots, n\}$ (gdzie n jest pewną liczbą parzystą) w których występują wszystkie litery alfabetu oprócz być może jednej. Przez $\bar{M}_n$ rozumiemy dopełnienie języka M_n do zbioru $\{1, 2, \ldots, n\}^*$.

a. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający $\bar{M}_n$?

b. Jaką minimalną liczbę stanów musi mieć niedeterministyczny automat skończony rozpoznający $\bar{M}_n$?

Zadanie 3.13. *2009.P3*

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 $1 \leqslant k \leqslant \frac{n}{2}$ nazwijmy parę liczb $\{2k-1, 2k\}$ rodziną. Powiemy, że słowo $x \in \{1, 2, \ldots, n\}^*$ 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 należących do M_n, które są rosnące i nie rozdzielają rodzin?

4 Gramatyki bezkontekstowe i automaty ze stosem

Zadanie 4.1. *2008.28; 2009.26; 2010.26*

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

Rozwiązanie.
Niech G będzie gramatyką podaną w treści zadania i niech $\mathcal{A} = \langle \Sigma, \{[,], (,), \perp\}, q, q, \delta\rangle$ będzie automatem o następującej relacji przejścia:
$\delta = \{\langle q, \epsilon, \perp q, \epsilon\rangle$
$\langle q, [, Z, q, [Z\rangle$
$\langle (q, (, Z, q, (Z\rangle$
$\langle q,], [, q, \epsilon\rangle$
$\langle (q,), (, q, \epsilon\rangle\}$
Automat ten akceptuje słowa po wczytaniu których stos jest pusty.

Lemat 1. $L(\mathcal{A}) = L(G)$

(⊆) Dowód indukcyjny po długości wyprowadzenia.

Jeśli przed wczytaniem słowa ε na stosie automatu znajdował się symbol $\perp$ to po wczytaniu tego słowa stan stosu się nie zmieni.

Załóżmy teraz, że dla każdego n jeśli słowo w zostało wyprowadzone z S w nie więcej niż n krokach to stan stosu po wczytaniu w jest taki sam jak przed jego wczytaniem.

Rozpatrzmy wyprowadzenia z S o długości $n+1$:

$S \to SS$ Jeśli jeden z tych nieterminali został przekształcony w ε to nie bierzemy go pod uwagę i rozpatrujemy drugi. W przeciwnym wypadku wyprowadzenia z obu tych nieterminali mają długość mniejszą niż $n+1$ i może zostać do nich zastosowane założenie indukcyjne. Zgodnie z nim, jeśli przed zastosowaniem tej produkcji na stosie mieliśmy słowo r, to po zaaplikowaniu słowa wyprowadzonego z pierwszego S dalej mamy r, więc po zaaplikowaniu słowa wyprowadzonego z drugiego S również mamy r.

$S \to [S]$ Załóżmy, że przed zastosowaniem tej produkcji na stosie automatu znajdowało się słowo r. Po wczytaniu symbolu [na stosie będzie się znajdowało słowo postaci $[r$. Jako, że długość wyprowadzania słowa z nieterminalu S będzie nie większa niż n, na mocy założenia indukcyjnego wiemy, że po zaaplikowaniu do automatu słowa wyprodukowanego z tego nieterminala stan stosu nie ulegnie zmianie i dalej będzie się na nim znajdować słowo $[r$. Po wycztaniu symbolu terminalnego] zawartość stosu ulegnie zmianie i będzie się na nim znajdowało słowo r.

$S \to (S)$ Rozmumowanie przebiega analogicznie jak w poprzednim podpunkcie.

Udowodniliśmy, że dowolnie długie wyprowadzenie słowa z S nie zmienia zawartości stosu automatu. Z tego, że S jest stanem początkowym gramatyki i że na początku stos automatu jest pusty wnioskujemy, że każde słowo wyprowadzalne z gramatyki $\mathcal{G}$ jest akceptowane przez automat $\mathcal{A}$.

($\supseteq$) Niech w będzie najkrótszym wyrazem akceptowanym przez automat $\mathcal{A}$ którego nie da się wyprowadzić z gramatyki. Wiemy, że aby automat zaakceptował jakieś słowo jego pierwszym znakiem musi być nawias otwierający, zaś ostatnim zamykający. Mamy dwie możliwości:

- W trakcie wczytywania słowa w automat ani razu nie miał pustego stosu. Wtedy (wynika to z konstrukcji automatu) słowo w jest postaci (v) lub $[v]$. Jako, że v jest krótsze od w oznacza to, że jest wyprowadzalne z S. Ale z tego wynika, że w należy do $L(\mathcal{G})$ ponieważ można je wyprowadzić za pomocą odpowiedniej produkcji $S \to (S) \mid [S]$. Otrzymaliśmy sprzeczność z założeniem.
- Istnieje taki podział w na niepuste podsłowa, że $w = w_1 w_2$ i po wczytaniu słowa w_1 automat miał pusty stos. Skoro słowo w jest najkrótszym słowem akceptowanym przez automat które nie należy do $L(\mathcal{G})$ to znaczy, że słowa w_1, w_2 są wyprowadzalne z gramatyki. Ale to znaczy, że słowo w też jest wyprowadzalne z gramatyki za pomocą produkcji $S \to SS$. Uzyskaliśmy w ten sposób sprzeczność z założeniem.

Pokazaliśmy więc, że takie słowo w nie istnieje, co kończy dowód zawierania, a tym samym całego lematu.

Zadanie 4.2. *2008.29; 2009.27; 2010.27*

Zbuduj gramatykę bezkontekstową generującą język: $\{w \in \{0,1\}^* : |w|_0 = 2|w|_1 \wedge 2 \mid |w|_1\}$.

Rozwiązanie.
Niech $L = \{w \in \{0,1\}^* : |w|_0 = 2|w|_1 \wedge 2||w|_1\}$ i $L' = \{w \in \{0,1\}^* : |w|_0 = 2|w|_1 \wedge 2 \not| |w|_1\}$.
Zdefiniujmy gramatykę $G = \langle \Sigma, N, S, \Pi \rangle$ i $G' = \langle \Sigma, T, S, \Pi \rangle$, gdzie:

$$\begin{aligned}
\Sigma &= \{0,1\},\\
N &= \{S,T\},\\
\Pi &= S \to 0S0S1T \mid 0S0T1S \mid 0T0S1S \mid 0S1S0T \mid 0S1T0S \mid 0T1S0S \mid 1S0S0T \mid 1S0T0S \mid 1T0S0S \mid \epsilon,\\
&\ T \to 0S0S1S|0S1S0S|1S0S0S\}.
\end{aligned}$$

Udowodnimy, że gramatyka G generuje język L, a gramatyka G' generuje język L'.

Lemat 1. $L(G) \subseteq L$ oraz $L(G') \subseteq L'$.

Dowód będzie indukcyjny po wysokości h drzewa wyprowadzenia z gramatyki.

– $h = 1$. Wtedy z G możemy wyprowadzić tylko ϵ, a $\epsilon \in L$.

  $h = 2$. Wtedy z G' możemy wyprowadzić jedynie 001 lub 010 lub 100, które to słowa należą do G'.

– Zakładamy, że przy wyprowadzeniu wysokości $h \leqslant k$ z G wyprowadzamy jedynie słowa z L, a z G' jedynie słowa z L'.

– Pokażemy, że dla $h = k+1$ słowa wyprowadzone z G należą do L, a słowa wyprowadzone z G' należą do L'.

  1. $L(G) \subseteq L$

     W korzeniu drzewa wyprowadzenia z G znajduje się jedna spośród produkcji $S \rightarrow 0S0S1T$, $S \rightarrow 0S1T0S$, ... stąd mamy 9 przypadków do rozważenia.

     Bez straty ogólności ze względu na symetryczność przypadków, możemy założyć, że w korzeniu mamy produkcję $S \rightarrow 0S0S1T$. Wtedy, na mocy założenia indukcyjnego, z poddrzew wyprowadzamy słowa $w_1, w_2 \in L, w_3 \in L'$. Mamy $w = 0w_10w_21w_3$, a stąd $|w|_0 = 2 + |w_1|_0 + |w_2|_0 + |w_3|_0$ oraz $|w|_1 = 1 + |w_1|_1 + |w_2|_1 + |w_3|_1$.

     Na mocy założenia indukcyjnego $|w_1|_0 = 2|w_1|_1$, $|w_2|_0 = 2|w_2|_1$, $|w_3|_0 = 2|w_3|_1$, skąd $|w|_0 = 2|w|_1$.

     Na mocy założenia indukcyjnego mamy $2 \mid |w_1|_1$, $2 \mid |w_2|_1$, $2 \nmid |w_3|_1$, a stąd $2 \mid |w|_1$.

  2. $L(G') \subseteq L'$

     W korzeniu drzewa wyprowadzenia z G' znajduje się jedna spośród produkcji $T \rightarrow 0S0S1S, T \rightarrow 0S1S0S, T \rightarrow 1S0S0S$, stąd mamy do rozważenia trzy przypadki.

     Bez straty ogólności ze względu na symetryczność przypadków, możemy założyć, że w korzeniu mamy produkcję $T \rightarrow 0S0S1S$. Wtedy, na mocy założenia indukcyjnego, z poddrzew wyprowadzamy słowa $w_1, w_2, w_3 \in L$. Mamy $w = 0w_10w_21w_3$, a stąd $|w|_0 = 2 + |w_1|_0 + |w_2|_0 + |w_3|_0$ oraz $|w|_1 = 1 + |w_1|_1 + |w_2|_1 + |w_3|_1$.

     Na mocy założenia indukcyjnego $|w_1|_0 = 2|w_1|_1$, $|w_2|_0 = 2|w_2|_1$, $|w_3|_0 = 2|w_3|_1$, skąd $|w|_0 = 2|w|_1$.

     Na mocy założenia indukcyjnego mamy $2 \mid |w_1|_1$, $2 \mid |w_2|_1$, $2 \mid |w_3|_1$, a stąd $2 \nmid |w|_1$.

Lemat 2. $L \subseteq L(G)$ oraz $L' \subseteq L(G')$.

Zdefiniujmy funkcję b:

$$\begin{aligned} b(\varepsilon) &= 0 \\ b(w0) &= b(w) - 1 \\ b(w1) &= b(w) + 2 \end{aligned}$$

Dowód będzie indukcyjny po długości d słowa.

– Dla $d = 0$ mamy słowo $\epsilon \in L(G)$.

  Dla $d = 3$ mamy słowa $001, 010, 100 \in L(G')$.

– Zakładamy, że każde słowo długości $d \leqslant k$ z L jest wyprowadzalne z G, a z L' jest wyprowadzalne z G'.

– Pokażemy, że tak jest dla wszystkich słów długości $k + 1$.

  1. Dla $w \in L$ rozważmy wszystkie możliwe przypadki
     (a) Jeśli dla każdego właściwego prefiksu v słowa w zachodzi $b(v) > 0$, to oznacza, że $w = 1u00$ dla $u \in L'$. Korzystając z założenia indukcyjnego możemy więc wyprowadzić w stosując produkcję $S \to 1T0S0S$.
     (b) Jeśli dla każdego właściwego prefiksu v słowa w zachodzi $b(v) < 0$, to oznacza, że $w = 00u1$ dla $u \in L'$. Korzystając z założenia indukcyjnego możemy więc wyprowadzić w stosując produkcję $S \to 0S0T1S$.
     (c) Jeśli $w = uv$ dla pewnych $u, v \neq \epsilon$ takich, że $b(u) = 0$, to w zależności od tego czy liczba jedynek w u jest parzysta czy nie należy wyprowadzić w z produkcji typu $S \to xTySzS$ lub $S \to xSySzT$, gdzie $x, y, z \in \Sigma$.
     (d) Pozostałe przypadki obejmują takie słowa w, że funkcja b dla pewnego prefiksu w zmienia znak, ale nie osiąga zera, co musi być spowodowane jedynką w środku słowa – w jest więc postaci $0u1v0$. Funkcja b może zmieniać znak tylko z ujemnego na dodatni więc na początku i na końcu słowa jest 0, a w u i v nie ma już zmian znaku. Następnie w zależności od tego czy liczba jedynek w u jest parzysta czy nie, wyprowadzenie rozpoczynamy od $S \to 0S1T0S$ lub $S \to 0T1S0S$.
  2. Dla $w \in L'$ rozważane przypadki są bardzo podobne do tych z poprzedniego podpunktu.

Wniosek 1. G jest gramatyką bezkontekstową generująca język L.

Zadanie 4.3. *2008.30; 2009.29; 2010.29*

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

Rozwiązanie.[10]

Załóżmy, że dany język (nazwijmy go L) jest bezkontekstowy – spełnia wtedy lemat o pompowaniu. Niech m będzie stałą z lematu o pompowaniu.

Weźmy słowo $w = (1^{2m+1}0)^{2m+1}$, $|w|_1 = (2m+1)|w|_0$ więc słowo to należy do języka L. Tworzymy podział w na $ABCDE$. Z postaci słowa w i tego, że $|BCD| \leqslant m$ wynika fakt, iż słowa B, D łącznie posiadają co najwyżej jedno zero i nie więcej niż m jedynek.

[10] Autor: Paweł „uzi" Laskoś–Grabowski

Niech w' oznacza słowo w po procesie pompowania. Rozpatrzymy wszystkie możliwe przypadki (zgodnie z lematem o pompowaniu słowo BD nie może być puste):

– Jeśli BD zawiera b jedynek i żadnego zera, to bierzemy potęgę k równą 2.

  Mamy wtedy $|w'|_1 = (2m+1)^2 + b \in ((2m+1)^2, (2m+1)(2m+2))$ i $|w'|_0 = 2m+1$. Słowo w' nie należy więc do L.

– Jeśli BD zawiera $1 < b < m$ jedynek i jedno zero, to bierzemy potęgę k równą 0.

  Mamy wtedy $|w'|_1 = (2m+1)^2 - b \in (2m(2m+1), 2m(2m+2))$ i $|w'|_0 = 2m$. Co oznacza, że słowo w' nie należy do L.

– Jeśli BD zawiera jedno zero lub zero i jedną jedynkę, to bierzemy potęgę k równą 3.

  Mamy wtedy $|w'|_1 = (2m+1)^2[+2$[11]$] \in ((2m+3)(2m-1), (2m+3)2m)$ (mamy $\forall c \in \mathbb{N}.\ c > 2 \Rightarrow 2c > 3$) i $|w'|_0 = 2m+3$. Słowo w' nie należy do L.

Otrzymaliśmy sprzeczność z lematem o pompowaniu – język L nie jest więc bezkontekstowy.

Zadanie 4.4. *2008.31; 2009.30; 2010.30*

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

Rozwiązanie.
Rozważmy następujący algorytm $\mathbb{A}$:

1. Przekształć podaną gramatykę do postaci Chomsky'ego
2. Jeżeli istnieje produkcja $S \to t$ gdzie $t \in \Sigma$, to język jest niepusty
3. Jeżeli istnieje nieterminal X taki, że $X \xrightarrow{*} t$, gdzie $t \in \Sigma$, to zamień X w każdej produkcji zawierającej X po prawej stronie na t, oraz usuń wszystkie produkcje zawierające X po lewej stronie.
4. Jeśli w kroku 3 nie istniał taki nieterminal, to język jest pusty. W przeciwnym wypadku wróć do kroku 2.

Obserwacja 1. Algorytm $\mathbb{A}$ zatrzyma się, ponieważ zbiór nieterminali jest skończony, a przy każdym wykonaniu kroku 3 usuwany jest jeden z nieterminali.

Obserwacja 2. $L(G) \neq \emptyset$ wtedy i tylko wtedy gdy $L(G') \neq \emptyset$, gdzie G' jest gramatyką po modyfikacjach z punktu 3 algorytmu.

Zadanie 4.5. *2008.32; 2009.31; 2010.31*

Niech G będzie gramatyką generującą poprawnie zbudowane formuły rachunku zdań ze zmiennymi p i q. Symbolami nieterminalnymi G są p, q, (,), $\neg$ i $\Rightarrow$ a produkcjami

[11] jeśli słowo BD zawiera jedynkę

$S \to \neg S \mid (S \Rightarrow S) \mid p \mid q$
Znajdź gramatykę w postaci normalnej Chomsky'ego generującą tem sam język.

Rozwiązanie.
Dla podanej gramatyki G równoważna gramatyka $\overline{G}$ w postaci normalnej Chomsky'ego ma postać:

$$
\begin{array}{rcl}
S & \to & XS \mid ZR \mid p \mid q \\
X & \to & \neg \\
Z & \to & MI \\
M & \to & LS \\
L & \to & (\\
I & \to & \Rightarrow \\
R & \to & SP \\
P & \to &)
\end{array}
$$

Dowód równoważności $w \in L(G) \Leftrightarrow w \in L(\overline{G})$:

($\Rightarrow$) Indukcja po długości wyprowadzenia.

Założmy, że mamy słowo w które należy do języka $L(G)$. Jeśli jest ono wyprowadzalne z symbolu startowego w jednym kroku to jest to p lub q, które to słowa są również wyprowadzalne z $\overline{G}$.

Załóżmy, że dla każdego $n \in \mathbb{N}$ słowo o wyprowadzeniu długości n należy również do $L(\overline{G})$ i rozpatrzmy słowa o długości wyprowadzenia $n+1$, a więc te produkcje w których jedyny nieterminal S występuje również po prawej stronie. Mamy wtedy:
$S \to \neg S \equiv S \to XS \to \neg S$
$S \to (S \Rightarrow S) \equiv$
$S \to ZR \to MIR \to LSIR \to (SIR \to (S \Rightarrow R \to (S \Rightarrow SP \to (S \Rightarrow S)$

($\Leftarrow$) Indukcja po długości wyprowadzenia.

Założmy, że mamy słowo w które należy do języka $L(\overline{G})$. Jeśli jest ono wyprowadzalne z symbolu startowego w jednym kroku to jest też wyprowadzalne z G.

Załóżmy, że dla każdego $n \in \mathbb{N}$ słowo o wyprowadzeniu długości n należy również do $L(G)$ i rozpatrzmy słowa o większej długości, które można wyprowadzić ze słowa startowego i których jedynym możliwym nieterminalem jest S.
$S \to XS \to \neg S \equiv S \to \neg S$
$S \to ZR \to MIR \to LSIR \to (SIR \to (S \Rightarrow R \to (S \Rightarrow SP \to (S \Rightarrow S) \equiv$
$S \to (S \Rightarrow S)$

Zadanie 4.6.

2008.33; 2009.35; 2010.35

Czy istnieje gramatyka bezkontekstowa generująca zbiór tych słów nad alfabetem zerojedynkowym, które nie są postaci vww dla żadnych słów w, v takich, że $|v| = |w|$?

Zadanie 4.7. *2008.34; 2009.36; 2010.36*

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?

Rozwiązanie.
Rozważmy język $L' = L \cap 0^*1^*0^*$. Zakładamy nie wprost, że język L jest bezkontekstowy, z czego wynika, że język L' jest również bezkontekstowy. Niech $w = 0^c1^c0^c = w_1vw_2$ gdzie c stała z lematu o pompowaniu i $w_1 = w_2$. Oczywiście $w \in L'$.

Rozpatrzmy więc dowolny podział $w = ABCDE$ taki, że $BD \neq \varepsilon$ i $BCD \leqslant c$. Zauważmy, że jeśli jedno ze słów B lub D miałoby wspólne litery równocześnie z w_1 i v lub v i w_2, to po jednokrotnym dopompowaniu słowo nie byłoby już postaci $0^*1^*0^*$ więc nie należałoby do L'.

Niech $w' = w_1'v'w_2'$ będzie słowem powstałym poprzez jednokrotne napompowanie słowa w. Rozpatrzmy wszystkie możliwe przypadki, ze względu na położenie słów B i D:

– Oba te słowa leżą w w_1. Wtedy dopompowanie słowa w zwiększy ilość zer o pewną liczbę n z przedziału $\{1, \ldots, c\}$. Jeśli $3 \not| \; n$ to słowo w' nie da się podzielić na trzy podsłowa o jednakowej długości więc $w' \notin L$. W przeciwnym wypadku długość każdego z podsłów się zwiększy i podział będzie musiał wypadać tak, że słowo v' będzie zaczynać się $\frac{2n}{3}$ zerami. Z tego wynika, że w_2' musi się zaczynać ciągiem jedynek długości $\frac{n}{3}$. Ale to oznacza, że w_1' zaczyna się zerem, zaś w_2' jedynką więc $w_1' \neq w_2'$ i $w' \notin L'$.

– Oba te słowa leżą w w_2. Wtedy rozumowanie jest analogiczne jak w powyższym przypadku z tym, że teraz w_2' kończy się zerem, zaś w_1' kończy się jedynką.

– Oba te słowa leżą w v. Wtedy dopompowanie słowa zwiększy ilość jedynek o n i nowe podsłowa (o ile da się je utworzyć tak aby miały równą długość) będą miały następującą postać: $w_1' = 0^c1^{\frac{n}{3}}$, $v' = 1^c1^{\frac{n}{3}}$, $w_2' = 1^{\frac{n}{3}}0^c$. Ponownie doszliśmy do sprzeczności postaci $w_1' \neq w_2'$.

– $B \in w_1$ i $D \in v$. Wtedy, jeśli $B \neq \varepsilon$ to zwiększy się liczba zer w w_1 i na mocy rozumowania z pierwszego podpunktu w_2' będzie się zaczynało od jedynki, co w znany sposób prowadzi do sprzeczności. W przeciwnym przypadku B nie ma znaczenia i zachowujemy się tak jakby oba podsłowa należały do v (podpunkt trzeci).

– $B \in v$ i $D \in w_2$. Wtedy, jeśli $D \neq \varepsilon$ to zwiększy się liczba zer w w_2 i na mocy rozumowania z drugiego podpunktu w_1' będzie się kończyło jedynką, z czego wynika, że $w_1' \neq w_2'$.W przeciwnym przypadku D nie ma znaczenia i zachowujemy się tak jakby oba podsłowa należały do v (podpunkt trzeci).

Uzyskaliśmy, dla języka L', sprzeczność z lematem o pompowaniu dla języków bezkontekstowych. Z czego wynika, że L również nie jest bezkontekstowy.

Zadanie 4.8. *2008.35; 2009.37; 2010.37*

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

Rozwiązanie.
Pokażemy, że język $\overline{L}$ jest bezkontekstowy przedstawiając gramatykę $\mathcal{G} = \langle\{0,1\}, \{S, X, K, W, L_1, L_2, R_1, R_2\}, S, \Pi\rangle$ która go generuje.

$$\Pi = \{ \quad \begin{aligned} S &\to X \mid XX \mid K \\ X &\to WWWX \mid W \\ W &\to 0 \mid 1 \\ K &\to L_1WR_1 \mid L_2WR_2 \\ L_1 &\to WL_1WW \mid 0 \\ L_2 &\to WL_2WW \mid 0 \\ R_1 &\to WWR_1W \mid 1 \\ R_2 &\to WWR_2W \mid 1 \end{aligned} \quad \}$$

Dowód równości $\overline{L} = L(\mathcal{G})$.

($\subseteq$) Niech $w \in \overline{L}$ pokażemy, że w jest wyprowadzalne z G. Niech n będzie długością w, mamy wtedy trzy przypadki:

- $n \mod 3 = 1$. Wtedy z symbolu startowego wyprowadzamy X, z którego da się wyprowadzić każde słowo nad $\{0,1\}^*$ o długości modulo 3 równej 1.
- $n \mod 3 = 2$. Korzystamy z produkcji $S \to XX$ i obserwacji uczynionej w poprzednim podpunkcie.
- Gdy długość słowa jest podzielna przez trzy i słowo to należy do $\overline{L}$ oznacza to, że gdy dokonamy podziału $w = w_1vw_2$ takiego, że $|w_1| = |w_2| = |v|$ to $w_1 \neq w_2$. Czyli istnieje takie $i \leqslant |w_1|$, że $w_1[i] \neq w_2[i]$, gdzie $w[i]$ oznacza i-tą literę słowa w. Weźmy najmniejsze takie i i załóżmy, że $w_1[i] = 0$ (drugi przypadek dowodzi się analogicznie).

  Korzystając z odpowiedniego ciągu produkcji mamy

  $$S \xrightarrow{*} L_1WR_1 \to W^{i-1}0(WW)^{i-1}WR_1.$$

  Weźmy teraz j takie, że $3(i-1)+3j+3 = n$ i kontynuujmy wyprowadzanie aż do uzyskania słowa $W^{i-1}0(WW)^{i-1}W(WW)^j1W^j$. Jedynka po lewej stronie znajduje się na $3i+2j+3$ pozycji. Wyprowadzając terminale z pozostałych produkcji W możemy utworzyć dowolne słowo w' długości n takie, że $w'[i] = 0$ i $w'[3i+2j+3] = 0$ w szczególności możemy wyprowadzić w.

($\supseteq$) Dowód niewprost.
Załóżmy, że istnieje takie $w \in L$, że w jest wyprowadzalne z $\mathcal{G}$. Z prostej obserwacji wynika, że słowo to nie mogło zostać wyprowadzone z produkcji $S \to X \mid XX$. Z analizy gramatyki wynika, że z nieterminalu K z którego musiało w takim razie zostać wyprowadzone w, można wyprowadzić jedynie słowa postaci $W^iaW^iW^iWW^jW^jbW^j$ dla dowolnych $i, j \in \mathbb{N}$ i $a, b \in \Sigma$, $a \neq b$.

Wyprowadzane słowa mają długość $3i + 3j + 3$ z czego wynika, że przy podziale na trzy podsłowa równej długości i każde z nich będzie miało długość $i + j + 1$. Wiemy, że pierwsze z tych podsłów na i-tym miejscu ma literę a. Zauważmy, że i-ta pozycja w ostatnim słowie odpowiada $3i + 2j + 3$ pozycji w całym słowie. Z postaci słowa wyprowadzonego z K wynika, że na tej pozycji znajduje się litera b. Wiemy jednak, że $b \neq a$ co oznacza, że pierwsze i trzecie podsłowo różnią się na i-tej pozycji. Więc żadne słowo wyprowadzone z $\mathcal{G}$ nie należy do języka L, co kończy dowód.

Zadanie 4.9. *2008.36; 2009.32; 2010.32*

Czy język

$$L_3 = \{w \in \{0,1,2\}^* : \neg \exists x \in \{0,1,2\}^*.\ w = xx\}$$

jest bezkontekstowy?

Rozwiązanie.[12]
Język ten jest bezkontekstowy i generowany przez gramatykę $\mathcal{G} = \langle \Sigma, N, S, \Pi \rangle$, gdzie:
$\Sigma = \{0, 1, 2\}$
$N = \{S, X, A, B, C\}$
$\Pi = \{$

$$\begin{array}{rcl} S & \to & A \mid B \mid C \mid AB \mid AC \mid BA \mid BC \mid CA \mid CB \\ X & \to & 0 \mid 1 \mid 2 \\ A & \to & XAX \mid 0 \\ B & \to & XBX \mid 1 \\ C & \to & XCX \mid 2 \end{array}$$

$\}$

Dowód równości $L(\mathcal{G}) = L_3$

($\subseteq$) Niech $w \in L(\mathcal{G})$. Jeśli wyprowadzenie w zaczyna się od jednej z produkcji $S \to A \mid B \mid C$ to możemy zauważyć, że $|w|$ będzie nieparzyste więc nie istnieje taki wyraz u nad danym alfabetem, żeby $w = uu$.

W przeciwnym wypadku (bez straty ogólności z powodu symetryczności przypadków) możemy założyć, że w zostało wyprowadzone z produkcji $S \to AC$. Słowo w ma więc postać $x_1 0 x_2 z_1 2 z_2$, gdzie $|x_1| = |x_2| = i$ oraz $|z_1| = |z_2| = j$, $x_1, x_2, z_1, z_2 \in \Sigma^*$. $|w| = 2i + 2j + 2$ więc długość ewentualnego u takiego, że $w = u_1 u_2$ i $|u_1| = |u_2|$ wynosi $i + j + 1$. Ale nie da się dokonać takiego podziału ponieważ na $i + 1$ pozycji w u_1 znajduje się litera 0, zaś w u_2 2 (jest to $2i + j + 2$ pozycja w całym słowie).

($\supseteq$) Weźmy najkrótsze $w \in L_3$ takie, że w jest niewyprowadzalne z $\mathcal{G}$. Mamy dwa przypadki.

– Jeśli $|w|$ jest nieparzyste, to patrzymy na literę na środkowej pozycji w w i wyprowadzamy słowo z odpowiedniego pojedynczego nieterminalu.

[12] Autor: Konrad Drukała

– Jeżeli $|w|$ jest parzyste, to można je przedstawić w postaci $x_1ax_2z_1bz_2$, gdzie $|x_1| = |x_2| = i$ oraz $|z_1| = |z_2| = j$, $x_1, x_2, z_1, z_2 \in \Sigma^*$, $a, b \in \Sigma$, $a \neq b$. Wtedy na $i + j + 1$ pozycji w znajduje się litera a, zaś na $2i + j + 2$ pozycji litera b (przy równym podziale słowa w litery te znajdują się na tej samej pozycji, więc jeśli $w \in L_3$ to rozważane przedstawienie musi istnieć dla pewnych i, j).

  W zależności od konkretnych wartości a i b do rozpatrzenia są symetryczne przypadki. Rozważmy jeden: jeśli $a = 0$, $b = 2$ wtedy $S \to XZ \xrightarrow{*} x_1Xx_2z_1Zz_2 \xrightarrow{2} x_10x_2z_12z_2 \xrightarrow{*} w$.

We wszystkich przypadkach otrzymaliśmy sprzeczność z założeniem – słowo w daje się więc wyprowadzić z gramatyki $\mathcal{G}$.

Co kończy dowód równości języków.

Zadanie 4.10. *2008.37; 2009.33; 2010.33*

Czy dopełnienie języka L_3, język

$$L_4 = \{w \in \{0, 1, 2\}^* : \exists x \in \{0, 1, 2\}^*.\ w = xx\}$$

jest bezkontekstowy?
Wskazówka: (1) Rozważ język $L_4 \cap L$ gdzie $L = L_{0^*10^*10^*10^*1}$. (2) Skorzystaj z lematu o pompowaniu, pamiętaj, że podział $w = xyztr$ którego istnienie postuluje lemat jest taki, że $|yzt| \leqslant c$, gdzie c jest stałą z lematu.

Rozwiązanie.
Rozważmy język L' stanowiący przecięcie języka L_4 z językiem regularnym opisywanym przez wyrażenie $0^*10^*10^*10^*1$. Wiemy, że L' jest językiem bezkontekstowym. Weźmy $w = x_1x_2 = 0^c10^c10^c10^c1$ gdzie c jest stałą z lematu o pompowaniu i $x_1 = x_2$. Oczywiście w należy do L'.

Rozważmy dowolny podział $w = ABCDE$ taki, że $|BCD| \leqslant c$ i $BD \neq \varepsilon$ i zwróćmy uwagę na fakt, że jedyną literą której ilość możemy zwiększyć pompując słowo jest 0 (zwiększenie ilości jedynek sprawi, że wyraz przestanie należeć do języka danego wyrażeniem regularnym).

Mamy do rozpatrzenia następujące przypadki (załóżmy, że za każdym razem pompujemy słowo raz, zwiększając liczbę zer o $n \in \{1, \ldots, c\}$):

– Jeśli BCD znajduje się w całości w jednym z słów x_i, to napompowanie słowa w spowoduje jeden z dwóch przypadków:
  – napompowany zostanie ciąg zer pomiędzy jedynkami, wtedy albo słowa różnią się ilością jedynek, albo odległość między dwiema jedynkami jest w obu słowach różna.
  – napompowany zostanie ciąg zer, znajdujący się na początku lub na końcu jednego z wyrazów. Rozpatrzmy jeden z tych przypadków (drugi jest symetryczny). Jeśli ilość zer na początku wyrazu x_1 zwiększy się o n, to długość tego wyrazu zwiększy się o $\frac{n}{2}$, zaś końcowe $\frac{n}{2}$ zer dawnego wyrazu x_1 (co najmniej 1) stanie się prefiksem słowa x_2. Jako, że oba słowa różnią się na pierwszej pozycji, mamy, że $x_1 \neq x_2$ z czego wynika, że napompowane słowo w nie należy do L'.

– Sytuacja gdy $B \in x_1$ i $D \in x_2$ Zwróćmy uwagę, że jeśli B lub D miałoby części wspólne z oboma słowami x oznaczałoby to, że zawiera literę 1. Jeśli $B = \varepsilon$ lub $D = \varepsilon$ to aktualny przypadek degeneruje się do tego rozważanego w poprzednim podpunkcie.

  Jeśli zaś B jest niepustym ciągiem zer, to napompowanie zmienia ilość zer pomiędzy dwiema pierwszymi jedynkami, która od tej pory staje się różna od ilości zer między trzecią a czwartą jedynką, bez względu na zawartość słowa D (którego napompowanie może zmodyfikować liczbę zer jedynie między drugą a trzecią jedynką w słowie). Ponownie więc albo słowa różnią się ilością jedynek, albo odległość między dwiema jedynkami jest w obu słowach różna.

W ten sposób udowodniliśmy, dochodząc do sprzeczności z założenia o bezkontekstowości L', że język L_4 nie jest bezkontekstowy.

Zadanie 4.11. *2008.38; 2009.34; 2010.34*

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

Zadanie 4.12. *2008.39; 2009.38; 2010.38*

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

Rozwiązanie.[13]
Lemat 1. $L = \{vww : v, w \in \{a,b,c\}^* \wedge |w| > 0\}$ nie jest językiem bezkontekstowym.

Załóżmy niewprost, że L jest językiem bezkontekstowym. Niech $L' = L \cap L_{ca^*b^*cca^*b^*c}$. L' jako przecięcie bezkontekstowego z regularnym jest językiem bezkontekstowym. Niech N będzie stałą z lematu o pompowaniu. Niech $w = ca^N b^N c$ i niech $u = ww = ca^N b^N cca^N b^N c$. Na mocy lematu o pompowaniu istnieje rozbicie u na $ABCDE$ takie, że $|BCD| \leqslant N$ i $|BD| \geqslant 1$ oraz $AB^iCD^iE \in L'$ dla każdego $i \in \mathbb{N}$.

Zauważmy, że w B i D nie występuje litera c (gdyby występowała możnaby ją powielić więcej niż 4 razy i słowo przestałoby należeć do języka). Rozważmy pozostałe przypadki:

– pierwsze słowo w jest prefiksem A – oznacza to, że pompujemy pomiędzy trzecią a czwartą literą c. Jeżeli teraz zwiększymy ilość liter pomiędzy nimi, to słowo przestanie należeć do języka (odległość między literami c będzie różna w obu podsłowach u, podsłowa te przestaną więc być sobie równe);

– drugie słowo w jest sufiksem E (pompujemy pomiędzy pierwszą a drugą literą c) – sytuacja analogiczna do tej rozpatrzonej w poprzednim podpunkcie;

– C zawiera dwie litery c (czyli A i E po jednej) czyli pompujemy pomiędzy pierwszą i drugą oraz trzecią i czwartą literą c. Zwiększamy z jednej strony ilość liter b, a z drugiej

[13] Autor: Mateusz Kocielski

a w konsekwencji nie będzie dopasowania między pierwszym i trzecim oraz drugim i czwartym c (pomiędzy nimi muszą być te same litery w takiej samej ilości i kolejności).

Otrzymaliśmy sprzeczność z założeniem o bezkontekstowości języka L', z czego wynika, że L również nie jest językiem bezkontekstowym.

Zadanie 4.13.

2008.40; 2009.28; 2010.28

Czy język: $\{w \in \{0,1\}^* : |w|_0 \leqslant |w|_1 \leqslant 2|w|_0\}$ jest bezkontekstowy?

Zadanie 4.14.

2008.41; 2009.39; 2010.39

Niech $L_=$ będzie językiem tych słów nad alfabetem zerojedynkowym, 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 4.15.

2008.42; 2009.40; 2010.40

Niech $L_=$ będzie językiem tych słów nad alfabetem zerojedynkowym, które mają tyle samo zer co jedynek, a L_R niech będzie językiem wszystkich palindromów. Czy przekrój $L_=$ z L_R jest językiem bezkontekstowym?

Rozwiązanie.
Niech $L = L_= \cap L_R$ będzie językiem opisanym w treści zadania. Korzystając z lematu o pompowaniu dla języków bezkontekstowych pokażemy, że L nie jest językiem bezkontekstowym. Niech N będzie stałą z lematu o pompowaniu. Wtedy $w = 0^N 1^{2N} 0^N$ należy do języka L.

Rozważmy dowolny podział $w = ABCDE$ taki, że $|BCD| \leqslant N$ i $BD \neq \varepsilon$. Pokażemy, że jednokrotne napompowanie słowa w sprawi, że przestanie ono należeć do języka. Rozpatrzmy wszystkie możliwe przypadki:

- Jeśli napompowaliśmy jedynie ciągi zer, to słowo przestaje mieć tyle samo zer co jedynek, nie należy więc do $L_=$.
- Jeśli napompowaliśmy jedynie ciągi jedynek, to również zaburzony zostaje warunek narzucony przez $L_=$.
- Jeśli napompowaliśmy zarówno zera jak i jedynki, to z założonej nierówności $|BCD| \leqslant N$ wiemy, że dopompowaniu mógł ulec tylko jeden z dwóch ciągów zer w słowie w. Po dopompowaniu, początkowy i końcowy ciąg zer nie są więc sobie równe, i tym samym słowo nie należy do języka L_R.

Co kończy dowód bezkontekstowości przecięcia języków $L_=$ i L_R.

Zadanie 4.16. *2008.E2; 2009.41; 2010.41*

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

Zadanie 4.17. *2008.E3; 2009.42; 2010.42*

Czy język $\{0^n 1^{n^2} : n \in \mathbb{N}\}$ jest bezkontekstowy?

Rozwiązanie.
Załóżmy nie wprost, że język $L = \{0^n 1^{n^2} : n \in \mathbb{N}\}$ jest bezkontekstowy i niech $w = 0^c 1^{c^2} \in L$, gdzie c jest stałą z lematu o pompowaniu dla języków bezkontekstowych.

Rozpatrzmy dowolny podział $w = ABCDE$ taki, że $BD \neq \varepsilon$ i $BCD \leqslant c$.

BD niech będzie takie, że zawiara n zer i m jedynek, z założeń lematu wynika, że $0 < n + m \leqslant c$. Rozważmy słowo w' powstałe przez jednokrotne napompowanie słowa w. Mamy $w' = 0^{c+n} 1^{c^2+m}$.

Wyliczmy jakie powinny być n i m, żeby to słowo należało do języka, tzn. istniało takie k naturalne, że $w' = 0^k 1^{k^2}$. Skoro $k = c+n$, to oznacza, że $k^2 = (c+n)^2 = c^2 + 2cn + n^2 = c^2 + m$, a więc $m = 2cn + n^2$. Z tego wynika, że aby w' należało do L to musi zachodzić $n = m = 0$ lub $n + m > c$. Oba te przypadki są jednak sprzeczne z założeniami o podziale słowa w.

Uzyskaliśmy sprzeczność z lematem o pompowaniu, co implikuje, że język L nie jest językiem bezkontekstowym.

Zadanie 4.18. *2008.P1*

Czy język składający się z tych słów w nad alfabetem $\{a, b, c\}$, które zaczynają się od palindromu parzystej długości i dłuższego niż połowa długości w, jest bezkontekstowy?

Zadanie 4.19. *2009.43; 2010.43*

Splecenie definiujemy w tym zadaniu jako najmniejszą relację ternarną na słowach nad pewnym ustalonym alfabetem $\mathcal{A}$ spełniającą następujące warunki:

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

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

Czy splecenie dwóch języków regularnych zawsze jest językiem regularnym?

Czy splecenie dwóch języków bezkontekstowych zawsze jest językiem bezkontekstowym?

Zadanie 4.20. *2009.E3*

Rozważmy język $L = \{w \in \{0,1\}^* : |w|_0 = |w|_1\}$.
a) Podaj gramatykę bezkontekstową G w postaci normalnej Greibach, taką, że $L = L_G$.
b) Podaj gramatykę bezkontekstową G w postaci normalnej Chomsky'ego, taką, że $L = L_G$.
c) Wyjaśnij w jednym zdaniu, po co potrzebujemy twierdzenia o postaci normalnej Greibach.
d) Wyjaśnij w jednym zdaniu, po co potrzebujemy twierdzenia o postaci normalnej Chomsky'ego.

Zadanie 4.21. *2009.P1*

Nich $L(L_1, L_2, L_3) = \{xyz \in \{0,1\}^* : x \in L_1 \wedge y \in L_2 \wedge z \in L_3 \wedge |x| = |y| = |z|\}$. Czy $L(L_1, L_2, L_3)$ jest bezkontekstowy jeśli:
a. $L_1 = L_3 = L_{0^*}$, zaś $L_2 = L_{0^*10^*10^*}$?
b. $L_1 = L_2 = L_{0^*}$, zaś $L_3 = L_{0^*10^*10^*}$?