Rozwiązania części z około dwustu łatwych zadań z języków formalnych i złożoności obliczeniowej i być może jednego chyba trudnego

Kamil Matuszewski

20 lutego 2017 — 23 marca 2017

Zadania, które zrobiliśmy wraz z Bartoszem Bednarczykiem na ćwiczenia z języków formalnych i złożoności obliczeniowej. Zadania były rozwiązywane **przed** ćwiczeniami, nie biorę żadnej odpowiedzialności za ich poprawność. Zadania które były zadane w naszej edycji oznaczone będą gwiazdką. Zadania warto zrobić samodzielnie a potem ewentualnie skonfrontować swoje rozwiązania z naszymi. Zadania w większości opisują idee, nie przedstawiają formalnego dowodu. Zapraszam również na github wiedzmac, gdzie będą pojawiały się zadania spisane przez Bartosza Bednarczyka.

1 Deterministyczne Automaty Skończone

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

Rozwiązanie:
Zadanie rozwiążemy nie wprost.

Dowód. Załóżmy nie wprost, że istnieje deterministyczny automat skończony $\mathcal{A}$ o mniej niż 1024 stanach rozpoznający nasz język L. Wiemy, że liczba dziesięcioliterowych słów nad alfabetem $\{0, 1\}$ to $2^{10} = 1024$. Wykorzystując zasadę szufladkową dochodzimy do wniosku, że są przynajmniej dwa różne słowa, które skończą w tym samym stanie. Niech te słowa nazywają się w i v. Skoro te słowa się różnią, to musi istnieć najmniejszy indeks k taki, że $w[k] \neq v[k]$. Skoro słowa są w tym samym stanie q, to po dopisaniu takiej samej liczby jedynek do obu słów oba będą w jakimś stanie q' (być może $q' = q$). Rozpatrzmy więc słowa $w1^{k-1}$ i $v1^{k-1}$ (słowa z dopisanymi $k-1$ jedynkami). Wtedy słowa różnią się na dziesiątej literze od końca (k'tej literze od początku) a są w tym samym stanie, co daje nam sprzeczność.

■

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

Rozwiązanie:
Rozważmy język L taki jak w treści zadania i $\Sigma = \{a, b, c\}$. Twierdzę, że automat rozpoznający L musi mieć przynajmniej 16 stanów. Żeby to pokazać, stworzę najpierw zbiór S w taki sposób:

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

Zauważmy, że $|S| = 16$. Teraz udowodnimy, że minimalna liczba stanów deterministycznego automatu skończonego rozpoznającego nasz język to właśnie 16.

Dowód. Załóżmy nie wprost, że istnieje automat $\mathcal{A}$ który rozpoznaje nasz język, a liczba jego stanów jest mniejsza niż 16. Wtedy z zasady szufladkowej istnieją takie dwa słowa $w, v \in S$ dla których automat skończy w tym samym stanie. Niech w będzie dłuższym słowem. Rozpatrzmy przypadki:

- $|w| = 3$ i $|v| = 3$. Niech x - dwie pierwsze litery słowa w. Wtedy słowo $wx \in L$ a słowo $vx \notin L$.
- $|v| < |w|$.Niech $c(w)$ – dopełnienie słowa w tzn najkrótsze słowo w' takie, że $ww' \in L$. $vc(w) \notin L$ (bo za krótkie), $wc(w) \in L$ (z definicji).
- $|w| = 2$ i $|v| = 2$. Niech x - litera z alfabetu której nie ma w w. Jeśli $x \in v$, to $wx \in L$ a $vx \notin L$. W przeciwnym przypadku, w i v różnią się na pierwszej literze. Niech pierwsza litera z $w = y$. Wtedy $wxy \in L$ a $vxy \notin L$.
- $|w| = 1$ i $|v| = 1$. Wtedy w i v są różnymi literami alfabetu. Niech x - trzecia litera ($w \neq v \neq x$ i $w, v, x \in \Sigma$). Wtedy $wvx \in L$ ale $vvx \notin L$.

Oznacza to, że do każdych dwóch słów należących do S możemy dopisać skończony ciąg liter alfabetu tak, że oba słowa nadal będą w tym samym stanie, ale jedno będzie należeć do języka a drugie nie, co prowadzi do sprzeczności i kończy dowód.

■

Zadanie 3. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający zbiór tych wszystkich słów nad alfabetem $\{a, b, c\}$, które mają przynajmniej 4 symbole i których ostatnie 4 symbole są jednakowe?

Rozwiązanie:
Analogicznie do zadania poprzedniego. Jakiś szkic dowodu:
Nasz zbiór to:

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

Minimalna liczba stanów to 13. Jeśli byłaby mniejsza to dla jakichś dwóch słów z powyższego zbioru, po ich przeczytaniu automat będzie w tym samym stanie, więc dopisanie takiego samego ciągu liter do obu sprawi, że nadal będą w tym samym stanie. Tym razem do dłuższego słowa (jeśli są równe to do dowolnego) dopisujemy tyle takich samych liter by znalazło się w stanie akceptującym, do drugiego słowa dopisujemy to samo, jedno należy do L drugie nie, a są w tym samym stanie, sprzeczność.

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

Rozwiązanie:
Zadanie rozwiązujemy korzystając z lematu o pompowaniu.

Dowód. Weźmy słowo $w = a^p b^{2p}$ gdzie p jest stałą z lematu o pompowaniu. Wiemy (z lematu o pompowaniu), że wtedy możemy podzielić słowo w na xyz, takie, że $|xy| \leqslant p$. W takim razie, w y będą same litery a. Skoro tak, to weźmy teraz słowo $xyyz$. Na mocy lematu, gdyby język był regularny, to takie słowo również należałoby do języka, ale widzimy, że liter a jest więcej niż p (bo $y \neq \epsilon$), więc słowo nie należy do języka, co kończy dowód.

■

Zadanie 5*. (za 2 punty) Niech L będzie dowolnym podzbiorem $L(0^*)$. Udowodnij, że L^* jest językiem regularnym.

Dowód. Niech L, L^* - języki takie jak w treści. Kilka przypadków:

1. L jest puste, lub $L = \{\epsilon\}$ - Wtedy język jest oczywiście regularny.
2. $|L| = n$. Załóżmy, że $L = L\backslash\{\epsilon\}$.
   Wiemy, że nasz język $L^* = \{0^{\alpha_1 l_1 + \ldots + \alpha_n l_n} | \alpha_1, \ldots, \alpha_n \in \mathbb{N}\}$. Twierdzę, że

   $$L^* = L_s \cup \{0^{x+yi} | i \in \mathbb{N}\}$$

   Gdzie:

   $$x = I(a_1 l_1 + \ldots + a_n l_n)$$
   $$y = nwd(l_1, l_2, \ldots, l_n)$$
   $$I = l_1 l_2 \cdots l_n$$
   $$a_1 l_1 + \ldots + a_n l_n \equiv_I y$$
   $$L_s = \{0^k \in L | k \leqslant x\}$$

   Pokażemy, że:

   - $L^* \subseteq L_s \cup \{0^{x+yi} | i \in \mathbb{N}\}$
     Weźmy dowolny element z L^*. Jeśli ten element jest mniejszy niż x, to z definicji należy do L_s. Załóżmy więc, że ten element jest większy od x. Ten element możemy zapisać jako kombinację $\alpha_1 l_1 + \ldots + \alpha_n l_n$. Chcemy pokazać, że tą kombinację możemy zapisać jako $x + yi$, czyli musimy znaleźć takie i.

     $$\alpha_1 l_1 + \ldots + \alpha_n l_n = x + yi$$
     $$\alpha_1 l_1 + \ldots + \alpha_n l_n - x = yi$$
     $$\frac{\alpha_1 l_1 + \ldots + \alpha_n l_n}{y} - \frac{x}{y} = i$$

     To jest całkowite, więc mamy takie i.

- $L^* \supseteq L_s \cup \{0^{x+yi} | i \in \mathbb{N}\}$ To, że $L_s \subseteq L^*$. Weźmy dowolny element z $\{0^{x+yi} | i \in \mathbb{N}\}$. Znajdziemy takie $\alpha_1, \ldots, \alpha_n$, że $x + yi = \alpha_1 l_1 + \ldots + \alpha_n l_n$ dla dowolnego i. Sprawdźmy jak wygląda $x + yi \ mod \ I$

  $$x + yi \equiv_I yi \equiv_I a_1 i l_1 + \ldots + a_n i l_n$$

  W takim razie

  $$x + yi = a_1 i l_1 + \ldots + a_n i l_n + dI$$

  Gdzie $d \geqslant 0$. Ale w takim razie:

  $$x + yi = a_1 i l_1 + \ldots + a_n i l_n + l_1(d l_2 l_3 \cdots l_n)$$

  $$x + yi = (a_1 i + d l_2 l_3 \cdots l_n) l_1 + \ldots + a_n i l_n$$

  Skoro $a_1, \ldots, a_n \geqslant 0$, to:

  $$x + yi = \beta_1 l_1 + \beta_2 l_2 + \ldots + \beta_n l_n$$

  Gdzie $\beta_1, \beta_2, \ldots, \beta_n \geqslant 0$.
- L nieskończone. Niech $nwd(l_1, l_2 \ldots) = y$. Wtedy wybieramy L_s na podstawie L^*. Wybieramy dowolny podzbiór L^* który daje takie samo nwd i robimy to samo co w poprzednim punkcie.

■

Zadanie 6*. Udowodnij, że język L tych słów nad alfabetem $\{0, 1\}$, które są zapisem binarnym liczby pierwszej, nie jest regularny.

Rozwiązanie:
Skorzystamy z lematu o pompowaniu i małego twierdzenia fermata.

Dowód. Załóżmy, że język L taki jak w zadaniu jest językiem regularnym. Skoro jest regularny, to spełnia lemat o pompowaniu. Niech n będzie stałą z lematu o pompowaniu. Weźmy słowo w będące reprezentacją liczby pierwszej $p > 2^n$ (wiemy, że taka istnieje, bo jest nieskończenie wiele liczb pierwszych). Na mocy lematu istnieją takie x, y, z, że $w = xyz$, $y \neq \epsilon$ i $\forall k xy^k z$ należy do języka, czyli jest reprezentacją pewnej liczby pierwszej. Weźmy $k = p$. Niech b_x, b_y, b_z oznaczają kolejno reprezentacje x, y, z jako liczb binarnych. Wtedy wartość w to $b_z + b_y 2^{|z|} + b_x 2^{|y|+|z|}$. Dodatkowo, wartość $q = xy^p z$ to $b_z + b_y 2^{|z|} + b_y 2^{|z|+|y|} + \ldots + b_y 2^{|z|+(p-1)|y|} + b_x 2^{p|y|+|z|}$. Wiemy więc, że:

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

Teraz, z małego twierdzenia fermata, wiemy, że

$$a^p \equiv_p a$$

Oznaczmy $s = (1 + 2^{|y|} + \ldots + 2^{(p-1)|y|})$ i sprawdźmy, czemu jest równe $q \ mod \ p$.

$$q = b_z + b_y 2^{|z|} s + b_x 2^{p|y|+|z|} \equiv_p b_z + b_y 2^{|z|} s + b_x 2^{|y|+|z|}$$

Czemu jest równe $s \ mod \ p$?

$$s = (1 + 2^{|y|} + \ldots + 2^{(p-1)|y|}) = \frac{1 \cdot (1 - 2^{p|y|})}{1 - 2^{|y|}} \equiv_p \frac{1 - 2^{|y|}}{1 - 2^{|y|}} \equiv_p 1$$

Mamy więc, że $s = 1 \ mod \ p$. Skoro tak, to

$$q \equiv_p b_z + b_y 2^{|z|} s + b_x 2^{|y|+|z|} \equiv_p b_z + b_y 2^{|z|} + b_x 2^{|y|+|z|} = p$$

Wyszło nam, że $q \equiv_p p$ więc $p | q$. Skoro tak, to q nie może być liczbą pierwszą, więc $q \notin L$, sprzeczność.

■

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

Zadanie 7*. Czy język $L = \{ww^R x : w, x \in \{0,1\}^* \text{ i } w, x \neq \epsilon\}$ jest regularny? Czy język $L = \{xwx : w, x \in \{0,1\}^* \text{ i } x \neq \epsilon\}$ jest regularny?

Rozwiązanie:
Twierdzę, że:

- $L = \{ww^R x : w, x \in \{0,1\}^* \text{ i } w, x \neq \epsilon\}$ nie jest regularny.

  Dowód. Skorzystajmy z lematu o indeksie (który udowodnimy w zadaniu 9), pokazując, że liczba klas abstrakcji relacji $\sim_L$ jest nieskończona. W tym celu pokażemy nieskończony ciąg wyrazów, z których żadne dwa nie są w relacji $\sim_L$.

  Niech $w_k = (10)^k$. Twierdzę, że dla dowolnych k i k' $k \neq k'$, $w_k \nsim_L w_{k'}$. Żeby to pokazać, załóżmy bez straty ogólności, że $k < k'$. Muszę pokazać, że istnieje takie v, że $w_k v \in L$ ale $w_{k'} v \notin L$. Weźmy $v = w_k^R 0$. W oczywisty sposób $w_k v = (10)^k (01)^k 0 \in L$. Zastanówmy się, czy $w_{k'} v = (10)^{k'} (01)^k 0$ rzeczywiście nie należy do języka. W tym celu spróbujmy znaleźć takie niepuste słowa w, x, że $(10)^{k'} (01)^k 0 = ww^R x$. Zauważmy, że w^R musi się zaczynać na tą samą literę, na którą kończy się w, czyli inaczej musimy znaleźć moment, w którym widzimy dwie takie same litery obok siebie. Jedyny taki moment jest po słowie $w_{k'}$, jednak $w_{k'}$ jest dłuższe niż $w_k^R 0$, więc nie znajdziemy słowa w^R. To oznacza, że v jest dobrym kontrprzykładem. Możemy więc stworzyć taki nieskończony ciąg $w_1, w_2 \ldots$, że $\forall_{i \neq j} w_i \nsim_L w_j$, więc liczba klas abstrakcji jest nieskończona, co z lematu o indeksie mówi, że L nie jest regularny.

  ■

- $L = \{xwx : w, x \in \{0,1\}^* \text{ i } x \neq \epsilon\}$ nie jest regularny.

  Dowód. Ponownie skorzystamy z lematu o indeksie.

  Niech $w_k = 10^k$. Twierdzę, że dla dowolnych k i k' $k \neq k'$, $w_k \nsim_L w_{k'}$. Żeby to pokazać, załóżmy bez straty ogólności, że $k' < k$. Muszę pokazać, że istnieje takie v, że $w_k v \in L$ ale $w_{k'} v \notin L$. Weźmy $v = w_k$. Słowo $w_k v = 10^k 10^k \in L$. Zastanówmy się, czy słowo $w_{k'} v = 10^{k'} 10^k$ rzeczywiście nie należy do L. Spróbujmy znaleźć takie x, w że $10^{k'} 10^k = xwx$. x musi zaczynać się na 1, a jedyne miejsce oprócz pierwszego gdzie występuje 1 to miejsce, w którym zaczyna się słowo w_k. Ale po tamtej 1 następuje k zer, a $k > k'$, więc $|10^{k'}| < |10^k|$, czyli nie możemy stworzyć takiego podziału, co oznacza, że v jest dobrym kontrprzykładem. Możemy więc stworzyć taki nieskończony ciąg $w_1, w_2 \ldots$, że $\forall_{i \neq j} w_i \nsim_L w_j$, więc liczba klas abstrakcji jest nieskończona, co z lematu o indeksie mówi, że L nie jest regularny.

  ■

Zadanie 8. Rozważmy alfabet A_n składający się z liter $a, b1, b2, ..., bn$. Niech język L_n^1 składa się z tych słów nad A_n, które mają parzystą liczbę wystąpień wzorca $b_1 b_2$. Niech język L_n^2 składa się z tych słów nad A_n, które maja parzysta liczbę wystąpień wzorca $b_2 b_3$, itd. Niech wreszcie język L_n^n składa się z tych słów nad A_n, które maja parzysta liczbę wystąpień wzorca $b_n b_1$. Zdefiniujmy język L_n jako przecięcie $L_n^1 \cap L_n^2 \cap \ldots \cap L_n^n$. Jaką minimalną liczbę stanów musi mieć deterministyczny automat skończony rozpoznający L_n?

2 Twierdzenie o indeksie

Zadanie 9*. (Twierdzenie o indeksie) Niech $L \subseteq \mathcal{A}^*$. Relację $\sim_L \subseteq \mathcal{A}^* \times \mathcal{A}^*$ definiujemy w następujący sposób: $w \sim_L w'$ w.t.w gdy $\forall v \in \mathcal{A}^*(wv \in L \Leftrightarrow w'v \in L)$. Udowodnij następujące *twierdzenie o indeksie*: L jest regularny wtedy i tylko wtedy gdy liczba klas abstrakcji relacji $\sim_L$ jest skończona. Minimalna liczba stanów DFA rozpoznającego L jest wtedy równa liczbie tych klas abstrakcji.

Rozwiązanie:
Udowodnijmy kilka rzeczy:

1. Liczba klas abstrakcji relacji $\sim_L$ jest skończona $\Rightarrow$ L jest regularny.

   Dowód. Po pierwsze, skonstruujmy automat skończony $\mathcal{A}$ i pokażmy, że rzeczywiście rozpoznaje on nasz język.

   Stanami naszego automatu będą wszystkie klasy abstrakcji. Będziemy więc utożsamiać klasy abstrakcji relacji $\sim_L$ ze stanami automatu. Skoro klas abstrakcji jest skończenie wiele, to nasz automat jest skończony. Stanem początkowym będzie klasa abstrakcji słowa pustego. Teraz, dla każdego stanu q, wybieramy przedstawiciela w z klasy abstrakcji którą z nim (ze stanem) utożsamiamy, i dla każdej litery alfabetu $a \in \Sigma$ sprawdzamy, jaka jest klasa abstrakcji słowa wa, i to jest stan do którego przechodzimy z q po wczytaniu a.

   Mamy już zbiór stanów, stan początkowy, alfabet i funkcję przejścia, zastanówmy się jeszcze jak wygląda zbiór stanów akceptujących.

   Obserwacja. Dla dowolnej klasy abstrakcji i dla dowolnych dwóch jej przedstawicieli w, u, oba albo są w języku albo ich nie ma.

   Jest to w miarę oczywiste, ponieważ tak właśnie zdefiniowaliśmy naszą relację.
   Skoro $\forall v \in \mathcal{A}^*(wv \in L \Leftrightarrow uv \in L)$, to tym bardziej to będzie prawda, jeśli weźmiemy $v = \epsilon$. Skoro tak, to wszystkie słowa które trafiły do danego stanu q albo są w języku albo ich nie ma (bo skoro słowo trafiło do tego stanu, to znaczy, że jest w tej klasie abstrakcji). W takim wypadku dla każdego stanu możemy wybrać dowolnego przedstawiciela klasy abstrakcji którą z tym stanem utożsamiamy, sprawdzić czy jest ono w języku, i jeśli tak to oznaczyć ten stan jako akceptujący.

   To, że automat rozpoznaje nasz język, wynika wprost z konstrukcji automatu.

   ■

2. Minimalna liczba stanów automatu rozpoznającego język L jest równa liczbie klas abstrakcji relacji $\sim_L$.

   Dowód. To, że automat o takiej liczbie stanów można skonstruować, pokazaliśmy w poprzednim punkcie konstruując go. Teraz zastanówmy się, czy jest to automat minimalny.

   Załóżmy nie wprost, że istnieje automat rozpoznający język, który ma mniej stanów. To oznacza, że istnieją dwa słowa w, w' takie, że trafią one do jednego stanu, a będą w dwóch różnych klasach abstrakcji (zasada szufladkowa). Skoro słowa są w tym samym stanie, to dopisanie do nich dowolnego słowa należącego do A^* sprawi, że w i w' nadal będą w tym samym stanie. Skoro tak, to $\forall v \in \mathcal{A}^*(wv \in L \Leftrightarrow w'v \in L)$. Ale to oznacza, że w i w' są w jednej klasie abstrakcji, co daje nam sprzeczność.

   ■

3. L jest regularny $\Rightarrow$ liczba klas abstrakcji relacji $\sim_L$ jest skończona.

   Dowód. Skoro wiemy, że L jest regularny, to istnieje skończony automat $\mathcal{A}$ rozpoznający L. Z poprzedniego punktu wiemy, że najmniejszy automat rozpoznający L ma tyle stanów, ile klas abstrakcji relacji $\sim_L$. To z kolei oznacza, że liczba tych klas abstrakcji musi być skończona (bo liczba stanów jest skończona), co automatycznie dowodzi naszej tezy.

   ■

Udowodnienie tych trzech punktów kończy zadanie.

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

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

Kolejne trzy exania dotyczą wzajemnych relacji między liczbami i_L a i_L^{inf}

Zadanie 10*. Udowodnij, że jeśli jedna z liczb i_L, i_L^{inf} jest skończona, to obie są skończone (z Twierdzenia o Indeksie wiemy, że ma to miejsce wtedy i tylko wtedy gdy L jest regularny). Dokładniej mówiąc:

a. udowodnij, że $i_L \leqslant i_L^{inf}$

b. udowodnij, że $i_L^{inf} \leqslant i_L^{i_L}$

Rozwiązanie:
Tak jak w zadaniu, pokażmy oba punkty:

a. $i_L \leqslant i_L^{inf}$

Dowód. Po pierwsze, dla dwóch słów w i v, jeśli były one w dwóch różnych klasach abstrakcji relacji $\sim_L$, to w oczywisty sposób będą w różnych klasach abstrakcji relacji $\sim_L^{inf}$ (bo skoro $w \nsim_L v$ to oznacza, że nie istnieje x takie, że $wx \in L \Leftrightarrow vx \in L$, więc tym bardziej nie mogą istnieć takie x, y że $xwy \in L \Leftrightarrow xvy \in L$). Z kolei, jeśli w i v były w tej samej klasie abstrakcji relacji $\sim_L$, to nie musi oznaczać, że będą w tej samej klasie abstrakcji relacji $\sim_L^{inf}$. W takim razie nierówność jest prawdziwa.

■

b. udowodnij, że $i_L^{inf} \leqslant i_L^{i_L}$

Dowód. Wiemy, że i_L jest skończona, więc z Twierdzenia o Indeksie istnieje skończony automat rozpoznający L. Weźmy najmniejszy taki automat $\mathcal{A}$. Niech jego stany to kolejno $q_0, q_1, \ldots, q_{n-1}$, gdzie $i_L = n$.

Zdefiniujmy funkcję F, która przyjmuje słowo, i zwraca krotkę $n$elementową. Niech F będzie zdefiniowana w następujący sposób:

$$F(w) = (\widehat{\delta}(q_0, w), \widehat{\delta}(q_1, w), \ldots, \widehat{\delta}(q_{n-1}, w))$$

Gdzie δ to funkcja przejścia automatu $\mathcal{A}$. Zdefiniujmy relację $\sim_F$ w następujący sposób: $w \sim_F v \Leftrightarrow F(w) = F(v)$.

Zauważmy, że $|\Sigma^*/\sim_F| \leqslant n^n = i_L^{i_L}$. Jeśli pokażemy, że $|\Sigma^*/\sim_L^{inf}| \leqslant |\Sigma^*/\sim_F|$, to pokażemy, że $|\Sigma^*/\sim_L^{inf}| = i_L^{inf} \leqslant i_L^{i_L}$, czyli to co chcemy pokazać. Weźmy dowolne dwa wyrazy w i v. Jeśli $w \sim_L^{inf} v$ to w jest w klasie abstrakcji $\sim_L^{inf}$ z v. To, czy są one w jednej klasie abstrakcji w $\sim_F$ nas nie interesuje, gdyż jeśli będą w innych, liczba klas abstrakcji $\sim_F$ się co najwyżej zwiększy. Musimy jedynie pokazać, że $w \nsim_L^{inf} v \Rightarrow w \nsim_F v$. To będzie oznaczać, że liczba klas abstrakcji relacji $\sim_L^{inf}$ będzie mniejsza bądź równa liczbie klas abstrakcji relacji $\sim_F$. Żeby to pokazać, pokażmy przez kontrapozycję, że $w \sim_F v \Rightarrow w \sim_L^{inf} v$.

Załóżmy, że $w \sim_F v$. To oznacza, że dla dowolnego stanu z którego zaczną, po ich wczytaniu automat zatrzyma się w tym samym stanie. To oznacza, że dla dowolnego słowa początkowego x, xw i xv zatrzymają pracę w tym samym stanie. Skoro tak, to dla dowolnego słowa x' xwx' i xvx' automat skończy działanie w tym samym stanie, więc w szczególności oba będą w języku lub oba nie będą w języku, więc $w \sim_L^{inf} v$.

Pokazaliśmy, że jeśli dwa słowa nie są w relacji $\sim_L^{inf}$, to nie są w relacji $\sim_F$. To oznacza, że liczba klas abstrakcji relacji $\sim_F$ jest większa (bądź równa) liczbie klas abstrakcji $\sim_L^{inf}$. To oznacza, że $i_L^{inf} \leqslant i_L^{i_L}$

■

Zadanie 11*. W zadaniu tym należy pokazać, że szacowanie z punktu **b.** z zadania 10 nie może zostać poprawione.

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

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

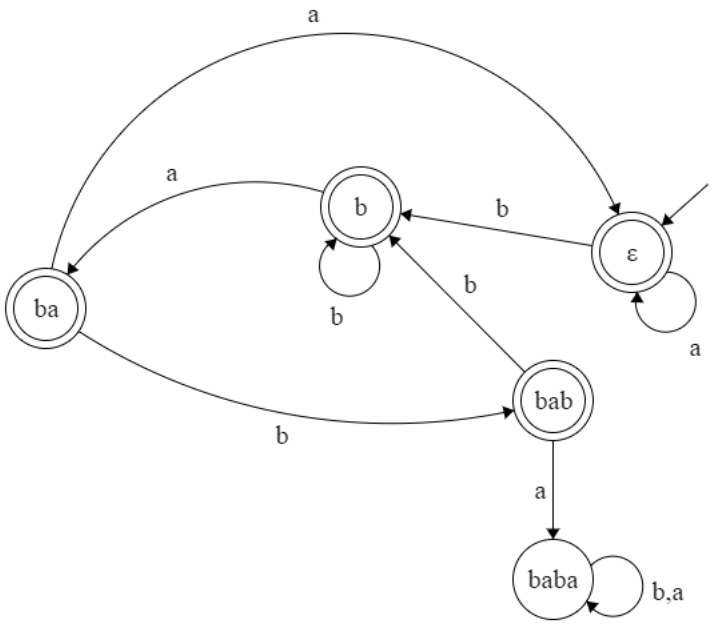
Zadanie 12*. Pokaż, że jeśli $|\Sigma| = 1$, to $i_L^{inf} = i_L$

Rozwiązanie:
Idea: niech $\Sigma = \{a\}$. Wtedy każdy wyraz nad alfabetem to a^k dla pewnego k. To czy zapiszemy ileś liter a na początku i na końcu, czy tylko na końcu nie zmieni nam wyrazu. Trzeba pokazać zawierania w dwie strony, pozostawiam jako proste ćwiczenie.

3 Wyrażenia regularne

Zadanie 14*. Skonstruuj automat skończony rozpoznający i wyrażenie regularne definiujące nad alfabetem $\{a, b\}$ język tych słów, które nie zawierają wzorca *baba*

Rozwiązanie:



W oczywisty sposób automat rozpoznaje nasz język. Teraz wyrażenie regularne tworzymy za pomocą eliminacji stanów.

Zadanie 15*. 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.

Rozwiązanie:
Weźmy języki L_n z jednym słowem: $((\ldots((a_0a_1)^2a_2)^2\ldots)^2a_n)^2$. Widać, że wyrażenie regularne opisujące te słowo ma długość $O(2^n)$. Zdefiniujmy:
$v_k = (a_1^*a_2^*\ldots a_{k-1}^*)^*$ - wszystkie słowa złożone z liter $a_0\ldots a_{k-1}$
$w_0 = a_0$
$w_k = (w_{k-1}a_k)^* \cap (v_k^*a_kv_k^*a_k)$

Obserwacja. w_n ma długość $O(n^2)$.
Niech $w(k)$ opisuje długość w_k. Widać, że $w(0) = 1$ oraz, że $w(k) = w(k-1) + 1 + 2k$.

Obserwacja. w_n opisuje język L_n.
Załóżmy że tak nie jest. Weźmy najmniejsze takie k, że w_k nie opisuje języka L_k. Zauważmy, że słowo $w_k = (w_{k-1}a_k)^2$. Skoro tak, to lewa część przekroju opisuje nam wszystkie i że $(w_{k-1}a_k)^i$. Z kolei z prawej strony mamy słowa które mają dokładnie dwie litery a_k. Z lewej strony taka sytuacja występuje tylko wtedy, kiedy $i = 2$, a to już jest nasze słowo.

Zadanie 17*. 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:
Załóżmy, że istnieje wyrażenie regularne ϕ takie, że $L_{a\phi} = L_{\phi b}$. ϕ musi zaczynać się od iluś liter a. Weźmy słowo z $L_{\phi b}$ które ma najmniej (k) liter a. To oznacza, że każde słowo wyprodukowane

z ϕ ma na początku k liter a. W takim razie słowo to nie należy do $L_{a\phi}$, bo każde słowo stamtąd ma przynajmniej $k+1$ liter a, sprzeczność.

Weźmy $\phi = a^*b^*$. Wtedy $a^*\phi = a^*a^*b^* = a^*b^* = a^*b^*b^* = \phi b^*$. To oznacza, że $L_{a^*\phi} = L_{\phi b^*}$.

5 Niedeterministyczne Automaty Skończone

Zadanie 21*. 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:

Zadanie 22*. Udowodnij, że jeśli dla pewnego języka L istnieje rozpoznający go NDFA, to również istnieje NDFA rozpoznający język $L^R = \{w : w^R \in L\}$.

Rozwiązanie:
Wiemy, że dla dowolnego NFA istnieje odpowiadający mu DFA. Odwróćmy DFA, to znaczy zmieńmy stany akceptujące z początkowymi (być może dodając stan początkowy p_0 z ϵ-przejściami) i odwróćmy funkcję przejścia (jeśli z q dochodziliśmy do q' wczytując a, to teraz z q' dochodzimy do q wczytując a). Teraz widać że automat rozpoznaje nasz język, bo skoro oryginalny rozpoznał słowo w, to znaczy, że istniała jakaś ścieżka od stanu początkowego do stanu akceptującego, to teraz wczytując słowo odwrotnie możemy ze stanu początkowego nowego automatu dotrzeć do stanu akceptującego.

Zadanie 23*. Wiadomo, że L jest językiem regularnym. Pokaż, że w takim razie język $\{w : \exists n \in \mathbb{N} \exists v \in L w^n = v\}$ jest też językiem regularnym. Przez w^n rozumiemy tu słowo w skonkatenowane ze sobą n razy.

Rozwiązanie:
Idea rozwiązania opiera się na założeniu, że nie musimy sprawdzać wszystkich n, tylko maksymalnie $|Q|$. Dlaczego? Możemy myśleć o tym tak, że wczytujemy słowo w, kończymy w jakimś stanie, a potem z tego stanu ponownie wczytujemy słowo w. Wczytując te słowo wielokrotnie, w szczególności więcej niż $|Q|$ razy, w pewnym momencie coś się nam powtórzy. Wtedy wpadniemy w cykl, i dla każdych n większych będziemy sprawdzać to samo. Jeśli więc do pierwszego powtórzenia jakiegoś stanu nie znaleźliśmy się w stanie akceptującym, to nie znajdziemy stanu akceptującego. To co zostało, to pokazać automat który rozpoznaje ten język. Niech $A = (\sigma, Q, q_0, F, \delta)$ będzie automatem rozpoznającym L. Skonstruujmy $B = (\sigma, Q', q_0', F', \delta')$. $Q' = Q^n$, $q_0' = (q_0, q_1, \ldots, q_{n-1})$, $\delta'((q_0, \ldots, q_{n-1}), a) = (\delta(q_0, a) \ldots, \delta(q_{n-1}, a))$. Teraz zostało tylko powiedzieć jak wyglądają stany akceptujące. Jednak widać, że

$$\overline{F} = \{(r_0, \ldots, r_{n-1}) : \exists k_0, \ldots k_{n-1} < n, i \in \mathbb{N}, i \leqslant n, k_0 = 0 \wedge \forall j < i, r_{k_j} = q_{k_{j+1}} \wedge q_{k_i} \in F\}$$

Zadanie 27*. Minimalny DFA rozpoznający język L ma zawsze tyle samo stanów co minimalny DFA rozpoznający dopełnienie L. Stwierdzenie to przestaje być prawdziwe, jeśli rozważamy automaty niedeterministyczne. Udowodnij, że istnieje język L, który daje się rozpoznać za pomocą NDF $\mathcal{A}$ o mniej niż 20 stanach, ale którego dopełnienie nie daje się rozpoznać żadnym NDF $\mathcal{A}$ o mniej niż 200 stanach.

Rozwiązanie:
Niech $L = \{0^n : n \nmid 210\}$. Oczywiście, $210 = 2 * 3 * 5 * 7$. Potrafimy stworzyć NFA o 18 stanach rozpoznający ten język. Załóżmy, że istnieje NFA o mniej niż 200 stanach rozpoznający dopełnienie L to znaczy $L' = \{0^n : n \mid 210\}$. Pierwszym słowem akceptowanym przez automat jest 0^{210}. Skoro automat ma mniej niż 200 stanów, to na ścieżce akceptującej słowo 0^{210} jakiś stan powtarza się przynajmniej dwa razy. W takim razie mamy ciąg stanów $q_0 \to \ldots \to q_k \to \ldots \to q_k \to \ldots \to q_f$. Ale wtedy istnieje też ścieżka $q_0 \to \ldots \to q_k \to \ldots \to q_f$ która akceptuje słowo krótsze, sprzeczność.

6 Zadania o hipotezie Černego

Zadanie 37*. Język $L \subseteq \Sigma^*$ nazywany jest regularnym ideałem, jeśli jest regularny i jeśli dla każdego słowa $w \in L$ i każdy słów $v, v' \in \Sigma^*$ zachodzi $v, w, v' \in L$.

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

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

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

Zadanie 38*.

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

b. Udowodnij, że jeśli zbiór $sync(Q)$ jest niepusty, to zawiera on jakieś słowo o długości nie większej niż $|Q|^3$.

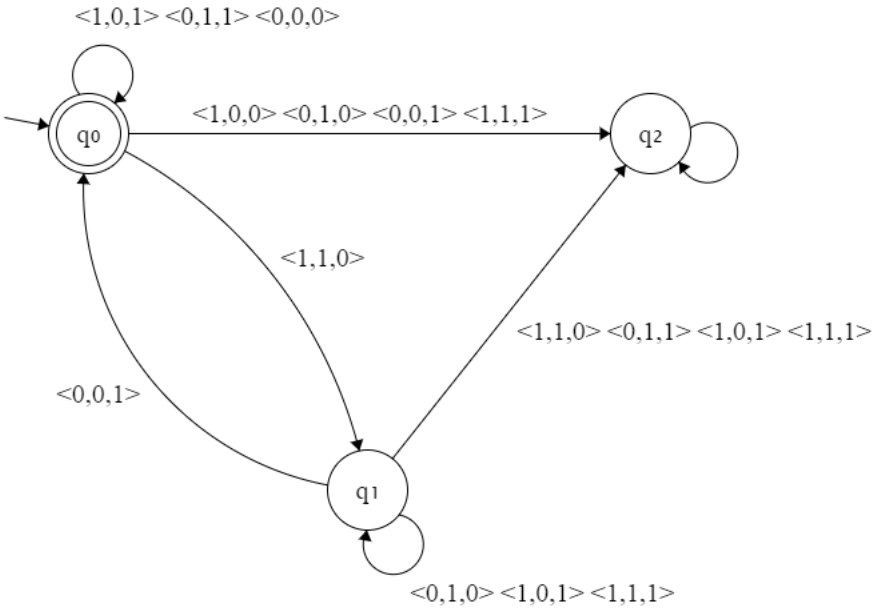
Zadanie 39*. Udowodnij, że dla każdego dostatecznie dużego n naturalnego istnieje automat $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$, gdzie $\Sigma = \{a, b\}$, $|Q| = n$, i dwuelementowy zbiór $S \subseteq Q$, takie że zbiór $sync(S)$ jest niepusty, ale nie zawiera słowa o długości mniejszej niż $n^2/4$.

7 Relacje automatyczne

Definicja. Tu będzie definicja.

Zadanie 43*. Czy relacja dodawania jest automatyczna? Przez relację dodawania rozumiemy tu $\{< a, b, c >\in \mathbb{N} : a + b = c\}$

Rozwiązanie:
Tak. Zauważmy, że Π_k^j "wyciąga" nam j'ty element krotki, natomiast l zmienia nam liczbę binarną zapisaną od tyłu na liczbę dziesiętną. W takim razie nasz automat będzie implementować dodawanie. Stan q_0 to stan akceptujący, z przeniesieniem 0, stan q_1 to stan z przeniesieniem 1, a stan q_2 to stan nieakceptujący, z którego już nie można wyjść.



Zadanie 44*. Czy relacja mnożenia jest automatyczna? Przez relację mnożenia rozumiemy tu $\{< a, b, c >\in \mathbb{N} : ab = c\}$

Rozwiązanie:
Nie. Załóżmy nie wprost, że tak. Skoro tak to istnieje DFA rozpoznający język taki jak w zadaniu. Weźmy najmniejszy taki DFA i załóżmy, że ma n stanów. Rozpatrzmy słowa w postaci $< 0, 0, 0 >^k$ gdzie $k = 0, \ldots, n$. Oczywiście, wszystkie słowa w tej postaci należą do σ_3^*. Skoro jest $n + 1$ słów a tylko n stanów, to jakieś dwa słowa trafią do jednego stanu. Weźmy te słowa. Niech w - krótsze z nich, v - te drugie. Do tego niech $w =< 0, 0, 0 >^l$ Teraz, dopisanie dowolnego sufiksu sprawi, że słowa nadal będą w tym samym stanie. Weźmy sufiks $s =< 10^l, 10^l, 0^l 1 >$. Z właściwości mnożenia binarnego (pamiętajmy, że liczby są odwrócone) widzimy, że $ws \in L$ a $vs \notin L$, co kończy dowód.

10 Zbiory i Funkcje Rekurencyjne

Zadanie 81*. Rozszerz definicje zbioru rekurencyjnego tak, aby można było rozważać rekurencyjne zbiory par liczb naturalnych i udowodnij, ze jeśli zbiór $A \in \mathbb{N}^2$ jest rekurencyjny, to zbiór $\{n : \exists_m [n, m] \in A\}$, czyli rzut A na pierwsza oś, jest zbiorem rekurencyjnie przeliczalnym.

Rozwiązanie:
Zbiór $A \in \mathbb{N}^2$ jest rekurencyjny, gdy istnieje program w MUJP ϕ_A taki, że

$$(\phi_A(n, m) = 1 \Leftrightarrow [n, m] \in A) \wedge (\phi_A(n, m) = 0 \Leftrightarrow [n, m] \notin A)$$

Teraz weźmy $A \in \mathbb{N}^2$ rekurencyjny i ϕ_A w MUJP który rozstrzyga przynależność do A. Niech zbiór B - rzut A na pierwszą oś. Napiszmy program w MUJP który zwraca 1 wtedy i tylko wtedy gdy $n \in B$. To będzie oznaczać, że B jest rekurencyjnie przeliczalny.

Dane: Liczba n.
Wynik: $1 \Leftrightarrow \exists_m [n, m] \in A$

```
Wczytaj n;
for m ← 1 to ∞ do
    if φ_A(n, m) = 1 then
        return 1
    end
end
```

Program w MUJP: ϕ_B

Zadanie 82*. Pokaz, że każdy zbiór rekurencyjnie przeliczalny jest rzutem pewnego zbioru rekurencyjnego, to znaczy jeśli B jest r.e. to istnieje taki rekurencyjny $A \in \mathbb{N}^2$, że $B = \{n : \exists_m [n, m] \in A\}$.

Rozwiązanie:
Niech ϕ_B będzie programem w MUJP który rozstrzyga przynależność do zbioru B.
Weźmy $A = \{[n, m] | \phi_B(n) = 1 \wedge \phi_B(n)$ zatrzymał się co najwyżej po m krokach $\}$

Twierdzenie. A jest rekurencyjny.

Dowód. Skonstruujmy algorytm:

Dane: Liczby n, m.
Wynik: $(1 \Leftrightarrow [n, m] \in A) \vee (0 \Leftrightarrow [n, m] \notin A)$

```
Wczytaj n, m
Uruchom φ_B(n) na m kroków
if φ_B(n) = 1 (zatrzymał się po co najwyżej m krokach i zwrócił 1) then
    return 1
else
    return 0
end
```

Program w MUJP: ϕ_A

■

Zadanie 84*. Pokaż, że $\{n : |Dom(\phi_n)| \geqslant 7\}$ jest rekurencyjnie przeliczalny.

Rozwiązanie:
Niech A - zbiór z zadania.

Dowód. Skonstruujemy program w MUJP który semi-rozstrzyga przynależność do tego zbioru.

Dane: Liczba n.
Wynik: $1 \Leftrightarrow n \in A$

Wczytaj n
$arg\_set = \emptyset$, $m = 0$
while 1 **do**
  for $arg \leftarrow 0$ **to** m **do**
    Uruchom $\phi_n(arg)$ *na* m *kroków*
    if $\phi_n(arg)$ *zatrzymał się po* m *krokach* **then**
      $arg\_set = arg\_set \cup \{arg\}$
    end
  end
  if $|arg\_set| \geqslant 7$ **then**
    return 1
  else
    $m++$
  end
end

Program w MUJP: ϕ_A

■

Zadanie 85*. Niech A, B, C, D będą zbiorami rekurencyjnie przeliczalnymi, takimi że każda liczba naturalna należy do dokładnie dwóch z nich. Udowodnij, że w takim razie wszystkie cztery zbiory są rekurencyjne.

Dowód. Pokażemy to konstruując algorytm. W tym przypadku wszystkie cztery będą wyglądać analogicznie, wiec pokażę konstrukcję jednego z nich. Niech ψ_A, psi_B, ψ_C, ψ_D będą programami smemi-rozstrzygającymi przynależność do zbiorów.

Dane: Liczba n.
Wynik: $(1 \Leftrightarrow n \in A) \wedge (0 \Leftrightarrow n \notin A)$

Wczytaj n
for $m \leftarrow 0$ **to** ∞ **do**
  Odpal wszystkie programy $\psi_x(n)$ *na* m *kroków*
  if $\psi_A(n) = 1$ **then**
    return 1
  else
    if $\psi_x(n) = 1$ *dla co najmniej dwóch programów (oprócz* ψ_A*)* **then**
      return 0
    end
  end
end

Program w MUJP: ϕ_A

■

Zadanie 86*. Udowodnij, ze jeśli ϕ jest niemalejącą całkowitą funkcją rekurencyjną, to zbiór jej wartości jest rekurencyjny. Czy pozostaje to prawdą bez założenia o całkowitości ϕ?

Rozwiązanie:
Najpierw, zbiór wartości $\phi(\mathbb{N})$ jest rekurencyjny.

Dowód. Jeśli ϕ jest od pewnego miejsca stała, to zbiór jej wartości jest skończony, zatem obliczalny.

W przeciwnym wypadku:

Dane: Liczba n.
Wynik: $(1 \Leftrightarrow n \in \phi(\mathbb{N})) \wedge (0 \Leftrightarrow n \notin \phi(\mathbb{N}))$

Wczytaj n;
for $m \leftarrow 0$ **to** ∞ **do**
  if $\phi(m) = n$ **then**
    return 1
  end
  if $\phi(m) > n$ **then**
    return 0
  end
end

Program w MUJP: ψ

Ta funkcja jest niemalejąca i nie jest stała od pewnego momentu, więc program się zatrzyma. ■

Teraz, jeśli funkcja ϕ nie jest całkowita, wtedy nie jest rekurencyjna.

Dowód. Zdefiniujmy $\phi(n) = \begin{cases} n & \text{gdy } \varphi_n(n) \in \mathbb{N} \\ \bot & \text{wpp} \end{cases}$
Wtedy zbiorem wartości ϕ jest zbiór K, który jak wiemy nie jest zbiorem rekurencyjnym. ■