Języki, Automaty i Obliczenia. Odcinek 2

Optymalizacja automatu skończonego. Własności języków regularnych.

Damian Niwiński

Październik 1996

1 Optymalizacja automatu skończonego

1.1 Determinizacja

O automatach A i B powiemy, że są *równoważne*, jeśli akceptują ten sam język, tzn. $L(A) = L(B)$.

Automat $A = (\Sigma, Q, I, \delta, F)$ nazwiemy *deterministycznym*, jeśli ma on dokładnie jeden stan początkowy, a relacja δ jest dobrze określoną *funkcją* z $Q \times \Sigma$ w Q, tzn., dla każdych q, σ, istnieje *dokładnie jedno* q' takie, że $(q, \sigma, q') \in \delta$. W takim przypadku możemy pisać $\delta(q, \sigma) = q'$ zamiast $(q, \sigma, q') \in \delta$. Podobnie, jeśli zbiór stanów początkowych jest singletonem, powiedzmy, $I = \{q_I\}$, to zwykle opuszczamy nawiasy $\{\}$, przedstawiając automat $A = (\Sigma, Q, q_I, \delta, F)$.

Zauważmy, że w automacie deterministycznym, dla każdego stanu q i każdego słowa w, istnieje dokładnie jeden przebieg automatu na słowie w startujący ze stanu q; w szczególności istnieje dokładnie jedno obliczenie automatu na w (tj. przebieg startujący ze stanu początkowego). Tak więc, również relacja $\hat{\delta}$ (okrełona w Odcinku 1, w punkcie 2.2) jest wtedy funkcją, $\hat{\delta} : Q \times \Sigma^* \to Q$, która ponadto spełnia następujące równania rekurencyjne

$$\hat{\delta}(q_I, \epsilon) = q_I \quad (1)$$

$$\hat{\delta}(q_I, w\sigma) = \delta(\hat{\delta}(q_I, w), \sigma) \quad (2)$$

O deterministycznym automacie skończonym możemy myśleć jak o maszynie realizującej pewien algorytm, mianowicie algorytm rozpoznawania słów języka $L(A)$.

Przykład 1 Niech $\Sigma = \{0, 1, \ldots, 9\}$ i niech L będzie zbiorem słów stanowiących przedstawienia w systemie dziesiętnym liczb podzielnych przez 3. Język ten jest rozpoznawany przez następujący automat $A = (\Sigma, Q, s, \delta, F)$.

- $Q = \{s, q_0, q_1, q_2\}$
- $q_I = s$

- $\delta(s,d) = q_{d \bmod 3}$, $\delta(q_i,d) = q_{(i+d) \bmod 3}$, *dla* $d \in \Sigma, i = 0,1,2$
- $F = \{q_0\}$

ćwiczenie. Uogólnić powyższy przykład, konstruując automat dla języka $L_{k,m}$, złożonego ze słów, które w systemie o podstawie k reprezentują liczby naturalne podzielne przez m.

Twierdzenie 1 *Dla każdego automatu skończonego o n stanach, można skonstruować równoważny mu automat deterministyczny o nie więcej niż 2^n stanach.*

Dowód. Niech $A = (\Sigma, Q, I, \delta, F)$ będzie dowolnym niedeterministycznym automatem skończonym. Idea jest następująca: deterministyczny automat A' symulujący A powinien, w miarę czytania kolejnych liter słowa w mieć informację o wszystkich stanach, do jakich mógłby dojść w tym miejscu automat A. Zauważmy, że jeśli już mamy taką informację dla jakiegoś słowa w, to nietrudno ją "zaktualizować" dla przedłużenia w o literę $\sigma \in \Sigma$, $w\sigma$, itd.

Dokładniej, określamy deterministyczny automat $A' = (\Sigma, P(Q), I, \delta', F')$, którego stanami są zbiory stanów automatu A, stanem początkowym jest zbiór stanów początkowych automatu A, oraz

- $\delta'(X,\sigma) = \{q : (\exists p \in X)(p,\sigma,q) \in \delta\}$, dla $X \in P(Q)$, $\sigma \in \Sigma$.
- $F' = \{X : X \cap F \neq \emptyset\}$

Dla dowodu, że $L(A') = L(A)$, wystarczy pokazać, że

$$\text{dla każdego } w \in \Sigma^*,\ \hat{\delta}'(I,w) = \{q \in Q : (\exists p \in I) p \xrightarrow{w} q\}$$

Stosując dla funkcji $\hat{\delta}'$ równania rekurencyjne (1) i (2) ze str. 1, otrzymujemy

$$\begin{aligned} \hat{\delta}'(I,\epsilon) &= I \\ &= \{q \in Q : (\exists p \in I) p \xrightarrow{\epsilon} q\} \end{aligned}$$

oraz

$$\begin{aligned} \hat{\delta}'(I,w\sigma) &= \delta'(\hat{\delta}'(I,w),\sigma) \\ &= \{q : (\exists q' \in \hat{\delta}'(I,w))(q',\sigma,q) \in \delta\} \\ &\quad \text{(z definicji } \delta'\text{)} \\ &= \{q : (\exists q' \in Q, \exists p \in I) p \xrightarrow{w} q' \xrightarrow{\sigma} q\} \\ &\quad \text{(z założenia indukcyjnego)} \\ &= \{q : (\exists p \in I) p \xrightarrow{w\sigma} q\} \end{aligned}$$

Ta uwaga kończy dowód. □

Udowodnione przed chwilą twierdzenie zastosujemy do wykazania ważnej własności klasy języków regularnych.

Twierdzenie 2 *Rodzina języków regularnych nad alfabetem Σ jest zamknięta na operacje binarnej sumy, przecięcia (tzn. części wspólnej) i różnicy.*

Dowód. Zamkniętość na sumę wynika wprost z definicji. Ze względu na zależności pomiędzy pozostałymi operacjami, wystarczy wykazać, że uzupełnienie $\Sigma^* - L$ języka regularnego L jest językiem regularnym. Niech $L = L(A)$ dla deterministycznego automatu $A = (\Sigma, Q, I, \delta, F)$. Tworzymy automat A', który różni się od A tylko tym, że rolę stanów akceptujących przejmują teraz stany nie-akceptujące A, tzn. $F' = Q - F$. Nietrudno sprawdzić, że $L(A') = \Sigma^* - L$. $\Box$

1.2 Twierdzenie Myhilla-Nerode'a

Widzieliśmy już do tej pory, że język regularny może być określony na różne sposoby: poprzez wyrażenie regularne, poprzez niedeterministyczny automat skończony i wreszcie, poprzez deterministyczny automat skończony. Udowodnimy teraz bardzo ważne twierdzenie, ukazujące, że *regularność* języka może być scharakteryzowana poprzez cechy strukturalne samego języka, bez odwoływania się do "obiektów z zewnątrz", takich jak wyrażenia regularne lub automaty. Niejako produktem ubocznym tego twierdzenia będzie konstrukcja automatu deterministycznego o minimalnej liczbie stanów.

Dla dowolnego języka $L \subseteq \Sigma^*$, określamy binarną relację $\sim_L$ na Σ^* następująco: dla każdych $u, w \in \Sigma^*$

$$\begin{aligned} u \sim_L w \quad &\Leftrightarrow_{def} \quad \{u\}^{-1}L = \{w\}^{-1}L \\ &\Leftrightarrow \quad (\forall v \in \Sigma^*)(uv \in L \Leftrightarrow wv \in L) \end{aligned}$$

ćwiczenie. Sprawdzić, że $\sim_L$ jest relacją równoważności.

Klasę abstrakcji słowa w w relacji $\sim_L$ będziemy oznaczać $[w]_L$. Odnotujmy dwie użyteczne własności relacji $\sim_L$.

Lemat 1 *1.* $(\forall u, w \in \Sigma^*)(\forall \sigma \in \Sigma) u \sim_L w \Rightarrow u\sigma \sim_L w\sigma$

2. $L = \bigcup_{w \in L} [w]_L$ $\Box$

Dowód. Punkt pierwszy wynika łatwo z definicji relacji $\sim_L$. W punkcie 2, inkluzja "$\subseteq$" jest oczywista. Dla dowodu inkluzji przeciwnej, wystarczy wykazać, że $w \sim u$ i $w \in L$ implikuje $u \in L$, co wynika z definicji $\sim_L$ przy $v = \epsilon$. $\Box$

Indeksem relacji równoważności nazywamy liczbę jej klas abstrakcji.

Przykład 2 Niech $\Sigma = \{0, 1\}$. Niech $\#_x(w)$ oznacza liczbę wystąpień symbolu x w słowie w. *(a)* Niech L będzie zbiorem ciągów o parzystej ilości jedynek. Wówczas $u \sim_L w$ wtedy i tylko wtedy, gdy $\#_1(u) \equiv \#_1(w) \pmod 2$. Indeks relacji $\sim_L$ jest równy 2. *(b)* Niech M będzie zbiorem słów v, w których $\#_0(v) = \#_1(v)$. Wówczas $u \sim_M w$ wtedy i tylko wtedy, gdy $\#_1(u) - \#_0(u) = \#_1(w) - \#_0(w)$. Indeks relacji $\sim_M$ jest nieskończony.

Twierdzenie 3 (Myhill-Nerode) *Niech $L \subseteq \Sigma^*$. Następujące warunki są równoważne:*

- *relacja $\sim_L$ ma skończony indeks;*
- *język L jest regularny.*

Ponadto, jeżeli warunki te zachodzą, to indeks relacji $\sim_L$ jest równy minimalnej liczbie stanów, jaką musi mieć automat deterministyczny rozpoznający język L.

Dowód.

"wtedy" Niech $L = L(A)$, gdzie $A = (\Sigma, Q, q_I, \delta, F)$ jest deterministycznym automatem skończonym. Określamy relację $\sim_A$ na Σ^*,

$$u \sim_A w \Leftrightarrow \hat{\delta}(q_I, u) = \hat{\delta}(q_I, w)$$

Oczywiście, ta relacja jest równoważnością o indeksie nie większym niż $|Q|$. Z drugiej strony, pokażemy, że $\sim_A \subseteq \sim_L$. Istotnie, jeśli $\hat{\delta}(q_I, u) = \hat{\delta}(q_I, w)$ i $x \in \Sigma^*$ jest dowolnym słowem, to $\hat{\delta}(q_I, ux) = \hat{\delta}(q_I, wx)$, a zatem $ux \in L \Leftrightarrow wx \in L$.

Z inkluzji $\sim_A \subseteq \sim_L$ wynika, że indeks $\sim_L$ jest niewiększy[1] niż indeks $\sim_A$, a zatem, tym bardziej niewiększy od liczby stanów automatu A.

"tylko wtedy" Przypomnijmy, że $[u]_L$ oznacza klasę abstrakcji słowa u w relacji $\sim_L$. Określamy deterministyczny automat $A^L = (\Sigma, Q^L, q_I^L, \delta^L, F^L)$ następująco:

- $Q^L = \{[u]_L : u \in \Sigma^*\}$
- $q_I^L = [\epsilon]_L$
- $\delta^L([u]_L, \sigma) = [u\sigma]_L$
- $F^L = \{[u]_L : u \in L\}$

Z lematu o relacji $\sim_L$ wynika, że funkcja δ^L jest dobrze określona oraz, że $[w]_L \in F \Leftrightarrow w \in L$. Zauważmy, że liczba stanów automatu A^L jest równa indeksowi relacji $\sim_L$.

Dla dowodu, że $L(A^L) = L$, wystarczy więc wykazać, że dla każdego $w \in \Sigma^*$,

$$\hat{\delta^L}(q_I, w) = [w]_L$$

Argument przebiega przez indukcję ze względu na długość w. Stosując równania rekurencyjne (1) i (2) ze str. 1 dla $\hat{\delta^L}$, otrzymujemy

- $\hat{\delta^L}(q_I, \epsilon) = q_I = [\epsilon]_L$,
- $\hat{\delta^L}(q_I, w\sigma) = \delta^L(\hat{\delta^L}(q_I, w), \sigma) = \delta^L([w]_L, \sigma) = [w\sigma]_L$

□

[1] Jeżeli r, s są równoważnościami na tym samym zbiorze i $r \subseteq s$, to klasy abstrakcji relacji s są sumami klas abstrakcji relacji r.

Pytanie. Czy rozumowanie z powyższego dowodu można rozszerzyć na przypadek dowolnego automatu *niedeterministycznego* i tym samym udowodnić, że indeks $\sim_L$ jest niewiększy od liczby stanów dowolnego automatu skończonego rozpoznającego L?

1.3 Automat minimalny

W poprzednim punkcie wykazaliśmy, że spośród wszystkich automatów deterministycznych akceptujących język regularny L, automat A^L opisany w dowodzie Twierdzenia Myhilla-Nerode'a ma ma minimalną możliwą liczbę stanów.

Obecnie zastanowimy się nad możliwością *skonstruowania* automatu A^L. Zauważmy, że automat ten zdefiniowaliśmy odołując się do relacji $\sim_L$, o której wiemy, że istnieje, ale która *a priori* nie jest nam znana, nawet jeśli mamy dane wyrażenie regularne lub automat rozpoznający język L. W konkretnych przypadkach, możemy często tę relację "odgadnąć" i przy jej pomocy zbudować automat A^L (tak będzie w większości zadań). Jednakże nie jest to metoda dostatecznie ogólna. Zobaczymy wszakże, że automat A^L można zawsze otrzymać z dowolnego automatu deterministycznego rozpoznającego L poprzez "sklejenie" niektórych jego stanów.

Potrzebne nam będzie uściślenie pojęcia "sklejania" a także jasne kryterium, kiedy można utożsamiać dwa automaty. Poniższe pojęcia można wprowadzić w przypadku ogólnym, jednak dla naszych potrzeb wystarczy rozważać automaty deterministyczne.

Izomorfizm. Intuicyjnie, powinno być oczywiste, że możemy utożsamić automaty, które różnią się jedynie *nazwami* stanów. Bardziej dokładnie, niech $A = (\Sigma, Q, q_I, \delta, F)$ i $B = (\Sigma, Q', q_I', \delta', F')$ będą automatami deterministycznymi nad tym samym alfabetem Σ, takimi, że $|Q| = |Q'|$. Funkcję *bijekcję* $h : Q \to Q'$ nazywamy *izomorfizmem* z A w B, jeśli

- $h(q_I) = q_I'$
- $(\forall q, q' \in Q)(\forall \sigma \in \Sigma)\, h(\delta(q, \sigma)) = \delta'(h(q), \sigma)$
- $(\forall q \in Q)\, q \in F \Leftrightarrow h(q) \in F'$

Zauważmy, że jeśli h jest izomorfizmem z A w B, to funkcja odwrotna $h^{-1} : Q' \to Q$ jest izomorfizmem z B w A.

Jeśli istnieje izomorfizm z A w B, to o automatach A i B mówimy, że są *izomorficzne* lub, że są *równe z dokładnością do izomorfizmu.*

Lemat 2 *Jeśli automat deterministyczny A rozpoznający język L ma tyle samo stanów, co automat A^L, to jest z nim izomorficzny.*

Dowód. Niech $L = L(A)$, gdzie $A = (\Sigma, Q, q_I, \delta, F)$ jest automatem deterministycznym i niech A^L będzie automatem skonstruowanym w dowodzie Twierdzenia Myhilla-Nerode'a, przy czym zakładamy, że $|Q| = |Q^L|$.

Przypomnijmy pojęcie relacji $\sim_A$ jakie zdefiniowaliśmy w tymże dowodzie (str. 4) :

$$u \sim_A w \Leftrightarrow \hat{\delta}(q_I, u) = \hat{\delta}(q_I, w)$$

Jak zauważyliśmy wówczas, zachodzi inkluzja $\sim_A \subseteq \sim_L$, przy czym indeks relacji $\sim_A$ jest większy lub równy od indeksu relacji $\sim_L$ a mniejszy lub równy od liczby stanów automatu A. Jeśli jednak, jak zakładamy, te skrajne wartości są równe, to relacje $\sim_A$ i $\sim_L$ mają ten sam indeks i, wobec jednej inkluzji, muszą być równe, tzn.

$$(*)\; \hat{\delta}(q_I, u) = \hat{\delta}(q_I, w) \;\Leftrightarrow\; u \sim_L w$$

Wykażemy teraz, że funkcja $h : Q^L \to Q$, określona

$$h \;:\; [w]_L \;\mapsto\; \hat{\delta}(q_I, w)$$

jest izomorfizmem automatów w sensie określonym powyżej.

Zauważmy najpierw, że z równoważności (*) wynika, iż

1. funkcja h jest dobrze określona, tzn. nie zależy od wyboru reprezentanta $w \in [w]_L$;
2. funkcja h jest różnowartościowa.

Wobec założonej równoliczności zbiorów Q i Q^L dowodzi to, że funkcja ta jest bijekcją.

Oczywiście $h([\epsilon]_L) = \hat{\delta}(q_I, \epsilon) = q_I$.

Z kolei, mamy

$$\begin{aligned} h(\delta^L([w]_L, \sigma)) &= h([w\sigma]_L) \\ &= \hat{\delta}(q_I, w\sigma) \\ &= \delta(\hat{\delta}(q_I, w), \sigma) \\ &= \delta(h([w]_L), \sigma) \end{aligned}$$

Wreszcie, dla każdego $w \in \Sigma^*$,

$$\begin{aligned} [w]_L \in F^L &\Leftrightarrow w \in L \\ &\Leftrightarrow \hat{\delta}(q_I, w) \in F \\ &\Leftrightarrow h([w]_L) \in F \end{aligned}$$

Wykazaliśmy, że funkcja h jest izomorfizmem automatów A i A^L. □

Konstrukcja automatu ilorazowego. Niech $A = (\Sigma, Q, q_I, \delta, F)$ będzie automatem deterministycznym i niech r będzie dowolną relacją równoważności na zbiorze jego stanów Q. Klasy abstrakcji relacji r będziemy oznaczać $[q]_r$. Intuicyjnie, relacja r "skleja" między sobą niektóre stany automatu A, co prowadzi do konstrukcji nowego automatu, który jednak na ogół nie musi być deterministyczny ani równoważny wyjściowemu. Podamy jednak warunki, które gwarantują poprawność operacji sklejania.

Powiemy mianowicie, że relacja równoważności $r \subseteq Q \times Q$ jest *kongruencją*[2], jeśli

1. $(\forall q, q')\, (q, q') \in r \wedge q \in F \Rightarrow q' \in F$

[2]Termin jest wzięty z algebry. Jak wiadomo relacja na uniwersum algebry jest nazywana kongruencją, jeśli jest niezmiennicza ze względu na operacje lub, co jest równoważne, jeśli jest jądrem pewnego homomorfizmu.

2. $(\forall q, q')(\forall \sigma)\,(q, q') \in r \Rightarrow (\delta(q, \sigma), \delta(q', \sigma)) \in r$

Jeśli relacja r jest kongruencją, to określamy *automat ilorazowy*

$$A/r = (\Sigma, Q_r, [q_I]_r, \delta_r, F_r)$$

gdzie $Q_r = \{[q]_r : q \in Q\}$, $F_r = \{[q]_r : q \in F\}$, a $\delta_r : Q_r \times \Sigma \to Q_r$ jest określona przez

$$(\forall q \in Q)(\forall \sigma \in \Sigma)\,\delta_r([q]_r, \sigma) = [\delta(q, \sigma)]_r$$

Zauważmy, że z warunku 2 wynika, że δ_r jest istotnie dobrze określoną funkcją, tzn. jej definicja nie zależy od wyboru reprezentanta klasy abstrakcji $[q]_r$. Tak więc, A/r jest również automatem deterministycznym.

Lemat 3 *$L(A/r) = L(A)$.*

Dowód. Wykażemy przez indukcję ze względu na długość słowa $w \in \Sigma^*$, że

$$\hat{\delta}_r([q_I]_r, w) \;=\; [\hat{\delta}(q_I, w)]_r$$

Ponieważ, z definicji zbioru F_r oraz z pierwszego warunku definicyjnego kongruencji mamy $[\hat{\delta}(q_I, w)]_r \in F_r$ wtedy i tylko wtedy, gdy $\hat{\delta}(q_I, w) \in F$, powyższa równość wystarczy do dowodu równości $L(A/r) = L(A)$.

Z równań rekurencyjnych (1) i (2) ze str. 1 zastosowanych dla funkcji $\hat{\delta}_r$, otrzymujemy najpierw

$$\hat{\delta}_r([q_I]_r, \epsilon) \;=\; [q_I]_r$$

a następnie

$$\begin{aligned}
\hat{\delta}_r([q_I]_r, w\sigma) &= \delta_r(\hat{\delta}_r([q_I]_r, w), \sigma) \\
&= \delta_r([\hat{\delta}(q_I, w)]_r, \sigma) \\
&\ \text{(z założenia indukcyjnego)} \\
&= [\delta(\hat{\delta}(q_I, w), \sigma)]_r \\
&\ \text{(z definicji } \delta_r\text{)} \\
&= [\hat{\delta}(q_I, w\sigma)]_r
\end{aligned}$$

Ta uwaga kończy dowód. □

Aby z dowolnego automatu deterministycznego rozpoznającego język L otrzymać metodą konstrukcji ilorazowej automat A^L, potrzebny jest jeszcze drobny zabieg techniczny: eliminacja stanów, które nie mogą być osiągnięte ze stanu początkowego.

Niech $A = (\Sigma, Q, q_I, \delta, F)$ będzie automatem deterministycznym. *Stanem osiągalnym automatu* nazwiemy każdy stan osiągalny ze stanu początkowego q_I, tzn. taki, że $q_I \xrightarrow{w} q$, dla pewnego w (por. Odcinek 1, punkt 2.5). Zauważmy, że stany nieosiągalne nie występują w żadnym obliczeniu automatu A. Mówiąc z grubsza, możemy je bez szkody "wyrzucić" z automatu.

Bardziej dokładnie, niech Q_o będzie zbiorem stanów osiągalnych automatu A. Określamy automat A_{red} (od ang. *reduced*) następująco:

$$A_{red} = (\Sigma, Q_o, q_I, \delta_{red}, F \cap Q_o)$$

gdzie $\delta_{red} = \delta \cap Q_o \times \Sigma \times Q_o$.

Następujący fakt pozostawiamy jako łatwe ćwiczenie.

Lemat 4 *A_{red} jest automatem deterministycznym i $L(A_{red}) = L(A)$.* □

Twierdzenie 4 *Niech $L \subseteq \Sigma^*$ będzie językiem regularnym, niech A^L będzie automatem skonstruowanym w dowodzie twierdzenia Myhilla-Nerode'a i niech A będzie dowolnym automatem deterministycznym rozpoznającym język L. Niech A_{red} będzie automatem otrzymanym z A przez eliminację stanów nieosiągalnych, tak jak to opisaliśmy powyżej. Wówczas istnieje kongruencja r na zbiorze stanów automatu A_{red}, taka, że automat ilorazowy A_{red}/r jest izomorficzny z A^L.*

Dowód. Dla uproszczenia notacji, załóżmy, że A jest już w postaci zredukowanej, tzn. $A_{red} = A = (\Sigma, Q, q_I, \delta, F)$. Określmy relację $\sim \subseteq Q \times Q$ przez

$$q \sim q' \;\Leftrightarrow\; L(A, q) = L(A, q')$$

(Przypomnijmy, Odcinek 1, punkt 2.3, że $L(A, q) = \{w : (\exists p \in F)\, q \stackrel{w}{\to} p\}$.) Oczywiście $\sim$ jest relacją równoważności. Aby wykazać, że jest także kongruencją, trzeba sprawdzić iż spełnia warunki definicyjne (por. str. 6), a mianowicie

1. $q \sim q' \;\Rightarrow\; (q \in F \Leftrightarrow q' \in F)$
2. $q \sim q' \;\Rightarrow\; (\forall \sigma \in \Sigma)\delta(q, \sigma) \sim \delta(q', \sigma)$

Pierwsza własność wynika z faktu, że $q \in F \Leftrightarrow \epsilon \in L(A, q)$. Dla dowodu drugiej, przypuśćmy, że $q \sim q'$ i niech $\sigma \in \Sigma$. Niech $\delta(q, \sigma) = p$ i $\delta(q', \sigma) = p'$. Potrzebujemy udowodnić

$$L(A, p) \;=\; L(A, p')$$

Przypuśćmy, że $w \in L(A, p)$, zatem $\sigma w \in L(A, q) = L(A, q')$. Odwołując się do pojęcia obliczenia jako ciągu przejść (por. Odcinek 1, Fakt 4), otrzymujemy $q' \stackrel{\sigma}{\to} q'' \stackrel{w}{\to} q_f$, dla pewnego $q'' \in Q$ i $q_f \in F$, ale ponieważ A jest automatem deterministycznym, więc $q'' = p'$, czyli $w \in L(A, p')$. W ten sposób stwierdziliśmy, że $L(A, p) \subseteq L(A, p')$, z symetrii wynika, że $L(A, p) = L(A, p')$, a zatem $p \sim p'$.

Tak więc, relacja $\sim$ jest kongruencją i z Lematu 3 wynika, że $A/\sim$ jest deterministycznym automatem rozpoznającym język L. Pozostaje udowodnić, że $A/\sim$ jest izomorficzny z A^L. Na podstawie Lematu 2, wystarczy udowodnić, że liczba stanów automatu $A/\sim$ jest równa indeksowi relacji $\sim_L$.

Zanim przystąpimy do właściwego dowodu, zrobimy pewną obserwację, jaka wynika wprost z przyjętych definicji.

(*) Jeśli $q_I \stackrel{w}{\to} q$, to $L(A, q) = \{w\}^{-1}L$

Określmy teraz funkcję $f : Q^L \to Q_\sim$ następująco:

$$(\forall w \in \Sigma^*)\, f([w]_L) \;=\; [\hat{\delta}(q_I, w)]_\sim$$

Odnotujmy najpierw, że z (*) wynika, że funkcja f jest dobrze określona, tzn. $w \sim_L v \Rightarrow \hat{\delta}(q_I, w) \sim \hat{\delta}(q_I, v)$. (Istotnie, $w \sim_L v$ implikuje, że $\{w\}^{-1}L = \{v\}^{-1}L$s, co, wobec (*) implikuje $L(A, \hat{\delta}(q_I, w)) = L(A, \hat{\delta}(q_I, v)$, skąd $\hat{\delta}(q_I, w) \sim \hat{\delta}(q_I, v)$.)

Z kolei wykażemy, że funkcja f jest różnowartościowa. Istotnie, $h([w]_L) = h([v]_L)$ oznacza, że $\hat{\delta}(q_I, w) \sim \hat{\delta}(q_I, v)$, czyli $L(A, \hat{\delta}(q_I, w)) = L(A, \hat{\delta}(q_I, v)$, co wobec (*) implikuje że $\{w\}^{-1}L = \{v\}^{-1}L$, a więc $[w]_L = [v]_L$.

Wreszcie, z faktu, że wszystkie stany automatu A są osiągalne, wynika, że funkcja f jest *na* zbiór $Q_\sim$.

Ta uwaga kończy dowód twierdzenia. □

Wniosek 1 *Spośród wszystkich automatów deterministycznych akceptujących język regularny L, automat A^L skonstruowany w dowodzie Twierdzenia Myhilla–Nerode'a ma* minimalną *ilość stanów, równą indeksowi relacji $\sim_L$. Ponadto, jeśli jakiś automat deterministyczny rozpoznający język L ma* tyle samo *stanów co A^L, to jest z nim izomorficzny.* □

Jest więc w pełni uzasadnione, że automat A^L będziemy nazywali *mimimalnym* automatem deterministycznym rozpoznającym język L, lub krótko: *automatem minimalnym* dla języka L.

W następującym przykładzie wykażemy, że eksponencjalne oszacowanie na wzrost liczby stanów przy determinizacji automatu niedeterministycznego (Twierdzenie 1) jest w ogólności optymalne.

Przykład 3 Niech $\Sigma = \{0, 1\}$ *i niech* $k \geq 1$. Niech L_k będzie zbiorem słów, które na k-tej od końca pozycji mają 1, tzn.

$$L_k = \{w \,:\, |w| \geq k \;\, i \;\, w_{|w|+1-k} = 1\}$$

Nietrudno podać automat *niedeterministyczny* o $k + 1$ stanach rozpoznający język L_k; pozostawiamy to jako ćwiczenie. Dla określenia liczby stanów minimalnego automatu *deterministycznego*, rozważmy relację $\sim_{L_k}$. Zauważmy najpierw, że żadne dwa różne słowa o długości k nie są równoważne. Istotnie, jeśli $|w| = |u| = k$ i np. $w_{k-i} = 0$ a $u_{k-i} = 1$, to $u0^i \in L_k$, podczas gdy $w0^i \notin L_k$. Z drugiej strony, nietrudno sprawdzić, że każde słowo jest równoważne, w sensie relacji $\sim_{L_k}$, pewnemu słowu długości k: gdy $|w| > k$, to w jest równoważne swojemu sufiksowi długości k, a gdy $0 \leq |w| < k$, to $w \sim_{L_k} 0^{k-|w|}w$. A zatem, indeks relacji $\sim_{L_k}$ *ergo* liczba stanów automatu minimalnego jest 2^k.

Pytanie. Co powyższy przykład mówi nam na temat operacji lustrzanego odbicia?

1.4 Algorytm minimalizacji automatu deterministycznego

W poprzednim punkcie opisaliśmy, jak z dowolnego automatu deterministycznego rozpoznającego język L można otrzymać automat minimalny A^L. Algorytmicznie, problem sprowadza się do dwóch zadań: eliminacji stanów nieosiągalnych i obliczenia relacji $\sim$. Kiedy już znamy relację $\sim$, konstrukcja automatu ilorazowego jest bezpośrednia.

Algorytm dla obliczenia stanów osiągalnych już rozważaliśmy w Odcinku 1 (Punkt 2.5, Algorytm 3). Obecnie podamy algorytm obliczania relacji $\sim$.

Niech więc $A = (\Sigma, Q, q_I, \delta, F)$ będzie automatem deterministycznym i $L = L(A)$. W związku z uwagą powyżej, możemy zakładać, że wszystkie stany automatu A są osiągalne.

Naszym zadaniem jest wyznaczenie relacji $\sim \subseteq Q \times Q$, która, jak pamiętamy, była zdefiniowana następująco:

$$q \sim q' \;\Leftrightarrow\; L(A,q) = L(A,q')$$

W rzeczywistości, wyznaczymy *uzupełnienie* relacji $\sim$, tj. zbiór par $\{(q,q') : q \not\sim q'\}$. W tym celu określimy najpierw pewien graf zorientowany, G_A. Wierzchołkami będą wszystkie pary nieuporządkowane $\{p,q\}$, $p \neq q$. Dalej, prowadzimy krawędź z wierzchołka $\{p,q\}$ do $\{p',q'\}$, o ile $\{p,q\} \neq \{p',q'\}$ i istnieje $\sigma \in \Sigma$ t.że $p \stackrel{\sigma}{\to} p'$ i $q \stackrel{\sigma}{\to} q'$. (Zauważmy w tym miejscu, że, z powodu determinizmu automatu A, z każdego wierzchołka grafu G_A wychodzi $|\Sigma|$ krawędzi. Zatem liczba krawędzi grafu G_A jest $O(n^2)$.) W grafie G_A zaznaczamy najpierw wszystkie wierzchołki postaci $\{p,q\}$, takie że $p \in F$ i $q \notin F$. Zauważmy, że, dla każdej zaznaczonej pary $\{p,q\}$, zachodzi na pewno $p \not\sim q$. Niech Z będzie zbiorem wszystkich zaznaczonych wierzchołków. Twierdzimy, że

> Dla każdych $p,q \in Q$, $p \not\sim q$ wtedy i tylko wtedy, gdy istnieje ścieżka z wierzchołka $\{p,q\}$ do jakiegoś wierzchołka w zbiorze Z.

Istotnie, jeśli $p \not\sim q$, to mamy jakieś słowo $v = v_1 \dots v_m$, takie, że, np. $\hat{\delta}(p,v) \in F$ a $\hat{\delta}(q,v) \notin F$. Mamy więc

$$p \stackrel{v_1}{\to} p_1 \stackrel{v_2}{\to} p_2 \dots p_{m-1} \stackrel{v_m}{\to} p_m \in F$$

i

$$q \stackrel{v_1}{\to} q_1 \stackrel{v_2}{\to} q_2 \dots q_{m-1} \stackrel{v_m}{\to} q_m \notin F$$

Nietrudno widzieć, że

$$\{p,q\} \to \{p_1,q_1\} \to \dots \{p_{m-1},q_{m-1}\} \to \{p_m,q_m\}$$

jest wówczas żądaną ścieżką w grafie, przy czym $p_m \in F$ a $q_m \notin F$, a więc wierzchołek $\{p_m,q_m\}$ należy do Z.

Dla udowodnienia implikacji odwrotnej, zauważmy najpierw, że jeśli $p \not\sim q$ i istnieje strzałka z $\{p',q'\}$ do $\{p,q\}$, to również $p' \not\sim q'$. Zatem teza powyższa wynika łatwo przez indukcję ze względu na długość ścieżki, jaka prowadzi z wierzchołka $\{p,q\}$ do zbioru Z.

Wyznaczenie relacji $\not\sim$ sprowadza się więc do obliczenia zbioru wierzchołków grafu G_A, dla których istnieje ścieżka prowadząca do jakiegoś wierzchołka w zbiorze Z. Łatwo widzieć, że interesujący nas zbiór jest zbiorem wierzchołków *osiągalnych* z Z w grafie dualnym G_A^d (tj., jeśli odwrócimy kierunek każdej krawędzi). Możemy więc dla jego obliczenia zastosować Algorytm 3 z Odcinka 1 (dokładniej, graf G_A^d, w którym krawędzie nie są etykietowane, możemy traktować jak graf automatu nad jednoliterowym alfabetem). Czas działania tego algorytmu jest proporcjonalny do liczby krawędzi w grafie, która w wypadku grafu G_A jest $O(n^2)$, takie jest też więc oszacowanie czasu działania przedstawionego algorytmu. Algorytm ten znany jest jako Algorytm Moore'a.

Nietrudno pokazać, że również wyznaczenie grafu G_A i zbioru Z może być wykonane w czasie $O(n^2)$, co pozwala nam podsumować powyższe rozważania w następującym twierdzeniu.

Twierdzenie 5 (Algorytm Moore'a) *Istnieje algorytm, który dla danego automatu deterministycznego o n stanach konstruuje w czasie $O(n^2)$* minimalny *automat deterministyczny rozpoznający ten sam język.*

W ćwiczeniach przedstawimy Algorytm Hopcrofta, którego czas działania jest lepszy, mianowicie $O(n \log n)$.

Zauważmy jeszcze na koniec, że łącząc opisany wyżej algorytm z algorytmem otrzymywania automatu niedeterministycznego z wyrażenia regularnego i algorytmem determinizacji automatu niedeterministycznego, otrzymujemy metodę konstrukcji automatu minimalnego dla języka zadanego wyrażeniem regularnym. Metoda ta nie jest jednak w każdym przypadku efektywna z uwagi na eksponencjalny wzrost rozmiaru automatu w procesie determinizacji.

Zagadnienie efektywnej metody znajdywania automatu *niedeterministycznego* o najmniejszej możliwej liczbie stanów, jest problemem otwartym.

ćwiczenia

Algorytm Hopcrofta

Przedstawimy teraz inny algorytm minimalizacji automatu. Niech $A = (\Sigma, Q, q_I, \delta, F)$ będzie, jak wyżej, automatem deterministycznym. Tym razem będziemy dążyć do obliczenia relacji $\sim$ (zdefiniowanej ja wyżej) poprzez obliczenie podziału, jaki ta relacja równoważności wyznacza na zbiorze Q. Poszukiwany podział będzie znaleziony "metodą kolejnych przybliżeń", przy czym każdy kolejny podział będzie rozdrobnieniem poprzedniego.

Niech $X, Y \subseteq Q$, $a \in \Sigma$. Określamy $X' = \{q \in X : \delta(q,a) \in Y\}$, $X'' = \{q \in X : \delta(q,a) \notin Y\}$. Oczywiście, $X' \cup X'' = X$, $X' \cap X'' = \emptyset$. Jeżeli oba zbiory X', X'' są niepuste, to powiemy, że para (Y,a) *łamie* zbiór X, a parę $\{X', X''\}$ nazwiemy *złamaniem* zbioru X przez (Y,a). Jeżeli P jest podziałem zbioru Q, to *złamaniem* P przez (Y,a) nazywamy podział otrzymany z podziału P przez zastąpienie każdego $X \in P$, który jest łamany przez (Y,a), przez jego złamanie X', X''.

W algorytmie będziemy używać dwóch typów struktur: z jednej strony, podziału (zmienna P), a z drugiej, listy par (Y,a) przeznaczonych do łamania (zmienna $\mathcal{L}$). Przyjmujemy oznaczenie $\bar{X} = Q - X$.

procedura **Hopcroft**

$P := \{F, \bar{F}\}$;
jeśli $|F| < |\bar{F}|$ **to** $\mathcal{L} := ((F, \sigma) : \sigma \in \Sigma)$
w przeciwnym przypadku $\mathcal{L} := ((\bar{F}, \sigma) : \sigma \in \Sigma)$
dopóki $\mathcal{L} \neq \emptyset$ **wykonuj**

1. zdejmij parę (Y, a) z listy $\mathcal{L}$;
2. oblicz zbiór $\mathcal{B}$ tych $X \in P$, że para (Y, a) łamie X;
3. **dla** $X \in \mathcal{B}$, $b \in \Sigma$ **wykonuj**
   (a) oblicz złamanie $\{X', X''\}$ przez (Y, a);
   (b) **jeśli** $(X, b) \in \mathcal{L}$ **to**
   zastąp (X, b) w $\mathcal{L}$ przez $(X', b), (X'', b)$;
   (c) **w przeciwnym przypadku**
   jeśli $|X'| < |X''|$ **to** dołącz (X', b) do $\mathcal{L}$
   w przeciwnym przypadku dołącz (X'', b) do $\mathcal{L}$

Należy dowieść, że algorytm działa poprawnie, tzn. zawsze zatrzymuje się i końcową wartością zmiennej P jest podział wyznaczony przez relację $\sim$ z wykładu. Ponadto, należy wykazać, że czas działania algorytmu jest $O(n \log n)$.

2 Własności domknięcia klasy języków regularnych

Rodzina języków regularnych ma wiele dobrych własności, w szczególności jest zamknięta na większość możliwych do pomyślenia operacji. W poniższym twierdzeniu podsumowujemy te własności. Poszczególne punkty zostały już udowodnione w trakcie wykładu lub jako ćwiczenia.

Twierdzenie 6 *Klasa języków regularnych jest zamknięta na*

- *operacje Boole'owskie, tzn. jeżeli* $L, M \subseteq \Sigma^*$ *są językami regularnymi, to również* $L \cup M$, $L \cap M$, $\Sigma^* - L$, $L - M$, *są językami regularnymi;*
- *konkatenację, tzn. jeżeli* $L, M \subseteq \Sigma^*$ *są językami regularnymi, to również* LM *jest językiem regularnym;*
- *operację gwiazdki, tzn. jeżeli* L *jest językiem regularnym, to* L^* *jest również językiem regularnym;*
- *ilorazy przez dowolne zbiory, tzn. jeżeli* L *jest językiem regularnym i* $X \subseteq \Sigma^*$ *jest dowolnym językiem, to* $X^{-1}L$ *i* LX^{-1} *są językami regularnymi;*
- *operację lustrzanego odbicia, tzn. jeżeli* L *jest językiem regularnym, to* $L^R = \{w^R : w \in L\}$ *jest językiem regularnym;*
- *podstawienie, tzn. jeśli* $f : \Sigma \to P(\Gamma^*)$ *jest funkcją taką, że, dla każdego* $\sigma \in \Sigma$, $f(\sigma)$ *jest językiem regularnym i* $L \subseteq \Sigma^*$ *jest językiem regularnym, to* $\hat{f}(L)$ *jest również językiem regularnym;*

- *obrazy homomorficzne, tzn. jeśli $L \subseteq \Sigma^*$ jest językiem regularnym, i $h : \Sigma \to \Gamma^*$ jest homomorfizmem, to $h(L)$ jest językiem regularnym;*
- *przeciwobrazy homomorficzne, tzn. jeśli $M \subseteq \Gamma^*$ jest językiem regularnym, i $h : \Sigma \to \Gamma^*$ jest homomorfizmem, to $h^{-1}(M)$ jest językiem regularnym.*

Większość powyższych własności domknięcia ma charakter efektywny, tzn. mając dane automaty lub wyrażenia regularne dla języków L, M, ..., potrafimy znaleźć automat (wyrażenie) dla języka otrzymanego w wyniku stosowanej operacji. Nie dotyczy to jedynie operacji ilorazu przez *dowolny* zbiór, który nie musi być zadany efektywnie, jeśli jednak ograniczymy się do ilorazów przez języki regularne, to operacja jest, oczywiście, efektywna.

Własności te pozwalają więc na dość elastyczne konstruowanie języków regularnych. Z drugiej strony mogą też być przydatne przy dowodzie nieregularności jakiegoś języka. Rozważmy np. język $bb^*\{a^n b^n : n \in N\}$. Język ten nie jest regularny, choć spełnia tezę lematu o pompowaniu (przynajmniej w tej formie jaką przyjęliśmy w Lemacie 4 w Odcinku 1). Jednakże stosując lewostronny iloraz przez bb^* otrzymujemy język $\{a^n b^n : n \in N\}$, którego nieregularność potrafimy wykazać. Stąd, z własności domknięcia na ilorazy, możemy wnioskować o nieregularności pierwszego języka.

Zadania o językach regularnych

1. Dowieść, że jeśli L jest językiem regularnym, to zbiór prefiksów słów z L, zbiór sufiksów słów z L, a także zbiór podsłów, tzn.

$$\{w : (\exists u, v \in \Sigma^*) uwv \in L\}$$

   są językami regularnymi.

2. Skonstruować minimalny automat deterministyczny rozpoznający język $L \subseteq \{a, b\}^*$, gdzie
   - $L = \{\epsilon\}$,
   - $L = \{a, b, aba\}$,
   - L jest zbiorem słów w których a występuje dokładnie 1993 razy,
   - L jest zbiorem słów w których liczba wystąpień a daje resztę 3 z dzielenia przez 5.

3. Ile stanów ma minimalny automat deterministyczny rozpoznający, że ciąg zer i jedynek jest rozwinięciem binarnym liczby podzielnej przez 3?

4. W hotelu jest K pięter. Przejazd windy bez zatrzymania z piętra i na piętro j ($i \neq j$) uważamy za jednostkową akcję. Z każdą taką akcją wiążemy literę: G jeśli $i < j$, D jeśli $i > j$. Rozważamy możliwe trasy windy, które rozpoczynają się i kończą na parterze. Każda taka trasa wyznacza słowo nad alfabetem $\{G, D\}$. Niech $L \subseteq \{G, D\}^*$ będzie zbiorem wszystkich tak otrzymanych słów.

   (a) Dowieść, że L jest regularny.

   (b) Ile stanów ma minimalny automat deterministyczny rozpoznający L?

5. Niech $L = \{a^{2^n} : n \geq 1\}$. Dowieść , że jedyne podzbiory L będące językami regularnymi, to zbiory skończone.

6. Σ jest dowolnym alfabetem. Ustalamy słowo $w \in \Sigma^*$ o długości n. Skonstruować automat deterministyczny o $n+1$ stanach rozpoznający wszystkie słowa $u \in \Sigma^*$ zawierające w jako podsłowo (tzn. $u = v_1 w v_2$; mówiąc bardziej obrazowo, konstruowany automat poszukuje wzorca w w tekście u).

7. Niech L będzie zbiorem słów $w \in \{a,b\}^*$, które zawierają przynajminiej jedno wystąpienie symbolu b, przy czym bezpośrednio przed *ostatnim* wystąpieniem symbolu b w słowie w musi wystąpić symbol a. Dowieść, że L jest językiem regularnym. Podać liczbę stanów minimalnego automatu deterministycznego rozpoznającego język L. Skonstruować wyrażenia regularne opisujące każdą z klas abstrakcji relacji $\sim_L$ z twierdzenia Myhill'a–Nerode'a wyznaczonej przez L.

8. Pan X zakupił pakiety trzech różnych akcji: A, B i C. Każdego dnia bada wzajemne położenie wartości swoich akcji, które może być reprezentowane jako *uporządkowanie* zbioru $\{A,B,C\}$, w kierunku od najmniej do najbardziej wartościowej. (Przyjmujemy założenie, że dwie różne akcje mają zawsze *różne* wartości.) Pan X postanowił, że jeśli w ciągu dwóch kolejnych dni któraś z akcji dwukrotnie spadnie na niższą pozycję, to sprzeda tę akcję. Np. jeśli kolejne notowania (w tym wypadku: uporządkowania) są $(A,B,C),(B,C,A),(C,B,A)$ to pan X sprzeda pakiet akcji C. Rozważamy zbiór G wszystkich uporządkowań liniowych zbioru $\{A,B,C\}$. Dowieść, że zbiór tych ciągów nad alfabetem G, które opisują takie wyniki notowań giełdowych, przy których pan X nie pozbędzie się żadnego pakietu akcji, jest językiem regularnym. Znaleźć liczbę stanów automatu minimalnego akceptującego ten język.

9. (*) Pan X postanowił, że każdego dnia będzie pracował lub nie, przestrzegając przy tym zasady, by na żadne siedem kolejnych dni nie przypadało więcej niż cztery dni pracy. Możliwy rozkład dni pracy pana X w ciągu n kolejnych dni możemy przedstawić jako n-bitowe słowo, gdzie 1 odpowiada dniowi pracy, a 0 dniowi odpoczynku. Dowieść, że zbiór wszystkich tak otrzymanych słów jest językiem regularnym (nad alfabetem $\{0,1\}$). Znaleźć minimalny automat deterministyczny.

10. (*) Dowieść, że dla dowolnego języka regularnego L, język

$$\{w : w^{|w|} \in L\}$$

jest również regularny.

Rozwiązania niektórych zadań

Rozwiązanie zadania 9.

Niech L będzie opisanym językiem. Rozważmy relację $\sim_L$ z twierdzenia Myhill'a Nerode'a. Twierdzimy, że każde słowo jest równoważne pewnemu słowu długości 7. Istotnie, zauważmy najpierw, że wszystkie słowa $w \notin L$ są równoważne między sobą, a więc równoważne w szczególności słowu 1111111. Dalej, nietrudno widzieć, że każde słowo, które należy do L i jest dłuższe niż 7, jest równoważne swojemu sufiksowi długości 7. Natomiast słowo w o długości $|w| = i < 7$ jest równoważne słowu $0^{7-i}w$. A zatem relacja $\sim_L$ ma nie więcej niż 2^7 klas abstrakcji, język L jest więc regularny.

Powyższy argument prowadzi też w prosty sposób do wyznaczenia automatu, którego stanami są słowa zero-jedynkowe długości 7, a którego przejścia są postaci

$$a_1a_2\ldots a_7, i \ \rightarrow\ a_2a_3\ldots a_7 i$$

gdzie $i \in \{0,1\}$, o ile ciąg $a_1a_2\ldots a_7$ zawiera nie więcej niż 4 jedynki, w przeciwnym razie odpowiednie przejście jest

$$a_1a_2\ldots a_7, i \ \rightarrow\ a_1a_2\ldots a_7$$

dla $i = 0,1$. Stanem początkowym jest ciąg 0000000, a stanami akceptującymi są oczywiście te ciągi, które zawierają nie więcej niż 4 jedynki.

Nie jest to jednak automat minimalny.

Poniżej wyznaczymy liczbę klas abstrakcji relacji $\sim_L$, konstrukcja samego automatu wynika z twierdzenia Myhill'a-Nerode'a. Na mocy wcześniejszej uwagi, wystarczy odpowiedzieć na pytanie, które słowa długości 7 są równoważne.

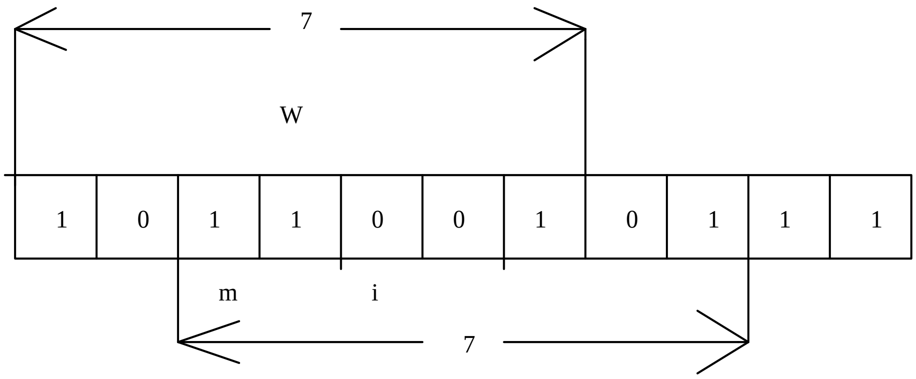
Słowo $w \in \{0,1\}^7$ nazwiemy *pełnym*, jeśli zawiera *dokładnie* 4 jedynki.

Lemat 1. Dwa różne słowa pełne są nierównoważne.

Istotnie, niech $w = w_1\ldots w_7$, $v = v_1\ldots v_7$ będą pełnymi słowami i niech i będzie najwyższą pozycją, gdzie słowa w i v się różnią, powiedzmy $w_i = 0$ i $v_i = 1$. Niech k będzie liczbą jedynek leżących w słowie w na lewo od pozycji i,

$$k = |\{j \,:\, j < i \wedge w_j = 1\}|$$

Zauważmy, że $k \geq 1$, bo w przeciwnym razie w miałoby mniej jedynek niż v, oraz, że, oczywiście, $k < i$ (w szczególności więc $i > 1$). Niech $u = 0^j1^k$, gdzie $j + k = i - 1$ ($j \geq 0$). Twierdzimy, że $wu \in L$, podczas gdy $vu \notin L$. Ten ostatni fakt wynika łatwo z konstrukcji, gdyż słowo $v_i\ldots v_7u$ jest długości 7 i zawiera 5 jedynek. Dla sprawdzenia, że $wu \in L$, wystarczy sprawdzić wszystkie podsłowa długości 7 postaci $w_m\ldots w_70^j1^\ell$, gdzie $m < i$ i $\ell > 0$ (pozostałe podsłowa długości 7 słowa wu nie mogą zawierać więcej jedynek niż w).



$k = 3,\ \ell = 1$

Zauważmy jednak, że $i - m = k - \ell$ (rysunek), a zatem słowo $w_m w_{m+1} \dots w_{i-1}$ zawiera co najwyżej $k - \ell$ jedynek, podczas gdy słowo $w_i w_{i+1} \dots w_7 0^j$ zawiera $7 - k$ jedynek, całe słowo $w_m \dots w_7 0^j 1^\ell$ zawiera więc niewięcej niż $(k - \ell) + (7 - k) + \ell = 7$ jedynek. Słowa w i v nie są zatem równoważne, co kończy dowód Lematu 1.

Lemat 2. Każde słowo $w \in \{0, 1\}^7$ zawierające $k < 4$ jedynki jest równoważne słowu pełnemu jakie otrzymamy z w pzez zastąpienie $4-k$ najwcześniej występujących zer przez jedynki.

Np., 0100100 jest równoważne 1110100. Istotnie, niech w będzie słowem w którym występują $k < 4$ jedynki i niech v będzie słowem otrzymanym z w jak w lemacie. Oczywiście, $(\forall u) vu \in L \Rightarrow wu \in L$. Przypuśćmy, że dla pewnego u implikacja odwrotna nie zachodzi, tj. $wu \in L$, ale $vu \notin L$. Wówczas możemy znaleźć i, takie że $v_i v_{i+1} \dots v_7 u_1 \dots u_{i-1}$ jest podsłowem vu długości 7 zawierającym więcej niż 4 jedynki. Wybierzmy najmniejsze i o tej własności. Pamiętajmy, że odpowiednie słowo $w_i w_{i+1} \dots w_7 u_1 \dots u_{i-1}$ zawiera nie więcej niż 4 jedynki. Zatem $v_i v_{i+1} \dots v_7$ różni się od $w_i w_{i+1} \dots w_7$, tzn. dla pewnego $i \leq i_0$, $w_{i_0} = 0$ a $v_{i_0} = 1$. Twierdzimy dalej, że $v_{i-1} = 1$. Istotnie, gdyby $v_{i-1} = 0$, to także $w_{i-1} = 0$ i zgodnie z definicją v, na pozycji $i - 1$ w słowie w zero powinno zostać zastąpione przez 1 (wiemy, że po transformacji $w_1 \dots w_{i-2}$ w $v_1 \dots v_{i-2}$ pozostawało jeszcze jakieś zero do zastąpienia, skoro zero na pozycji $i_0 \geq i$ zostało zastąpione przez 1 !). A zatem $v_{i-1} = 1$. Wówczas jednak słowo [3] $v_{i-1} v_i \dots v_7 u_1 \dots u_{i-2}$ zawiera co najmniej tyle samo jedynek, co rozważane wcześniej słowo $v_i v_{i+1} \dots v_7 u_1 \dots u_{i-1}$, co przeczy minimalności i. Ta uwaga kończy dowód Lematu 2.

Pamiętając o tym, że wszystkie słowa nie należące do L są równoważne, oraz że każde słowo jest równoważne pewnemu słowu o długości 7, z Lematów 1 i 2, otrzymujemy liczbę klas abstrakcji relacji $\sim_L$, która stanowi zarazem liczbę stanów minimalnego automatu deterministycznego

$$\binom{7}{4} + 1 \;=\; 36$$

Uwaga. Powyższa liczba może być otrzymana jeszcze na kilka innych sposobów. Na przykład, nietrudno zauważyć, że każde dwa słowa długości 7 zakończone sufiksem 000 są równoważne. Obserwacja ta może być uogólniona przez ciąg następujących stwierdzeń: każde dwa słowa długości 7 zakończone sufiksem długości 4 zawierającym dokładnie jedną jedynkę, są równoważne, podobnie każde dwa słowa zakończone sufiksem długości 5 zawierającym dokładnie dwie jedynki, itd. (ostatnie stwierdzenie dotyczy sufiksów tożsamych z całym słowem i jest trywialne). Powyższe uwagi prowadzą do wyznaczenia liczby nierównoważnych słów długości 7, które należą do L:

$$\binom{3}{0} + \left(\binom{4}{1} - \binom{0}{0}\right) + \left(\binom{5}{2} - \binom{4}{1}\right) + \left(\binom{6}{3} - \binom{5}{2}\right) + \left(\binom{7}{4} - \binom{6}{3}\right) = \binom{7}{4}$$

[3] Gdy $i - 2 = 0$, rozumiemy przez to, że sufiks $u_1 \dots u_{i-2}$ jest pusty.

Rozwiązanie zadania 10. Niech $A = (\Sigma, Q, q_1, \delta, F)$ będzie deterministycznym automatem, takim że $L(A) = L$. Niech $Q = \{q_1, \ldots, q_n\}$. W dalszym ciągu rozważań będziemy dla wygody notacyjnej utożsamiać funkcję $f : Q \to Q$ z ciągiem $(f(q_1), \ldots, f(q_n))$. Rozpocznijmy od uwagi, że każde słowo $w \in \Sigma^*$ wyznacza funkcję $f_w : Q \to Q$ określoną

$$f_w = (\hat{\delta}(q_1, w), \ldots, \hat{\delta}(q_n, w))$$

Skonstruujemy najpierw automat A', który, dla każdego w, "oblicza" funkcję f_w w następującym sensie. Zbiorem stanów automatu A' jest właśnie Q^Q (zbiór funkcji z Q w Q), stanem początkowym jest funkcja identycznościowa $id = (q_1, \ldots, q_n)$, a funkcja przejścia δ' jest określona przez

$$\delta'((p_1, \ldots, p_n), \sigma)(q) = (\delta(p_1, \sigma), \ldots, \delta(p_n, \sigma))$$

Nietrudno wykazać przez indukcję, że dla dowolnego w

$$\hat{\delta}'(id, w) = f_w$$

w tym właśnie sensie automat A' "oblicza" funkcję f_w (ewentualne stany akceptujące nie mają tu znaczenia).

Przypuśćmy na chwilę, że dla pewnego w znamy funkcję $f = f_w$. Możemy wówczas skonstruować automat, powiedzmy A_f, który czytając słowo w potrafi "obliczyć" stan, do którego automat A dochodzi po przeczytaniu słowa $w^{|w|}$. Zbiór stanów automatu A_f pokrywa się ze zbiorem stanów automatu A, stanem początkowym jest q_1, a funkcja przejścia, powiedzmy δ'', jest określona przez

$$\delta''(q, \sigma) = f(q)$$

Nietrudno widzieć, że istotnie

$$\hat{\delta}''(q_1, w) = \hat{\delta}(q_1, w^{|w|})$$

Oczywiście, automat skończony nie może "z góry" znać funkcji f_w. Jednakże, korzystając z faktu, że możliwych funkcji f_w jest skończenie wiele, możemy zastosować produkt automatów A_f po wszystkich możliwych funkcjach f, jednocześnie obliczając "właściwą" funkcję f_w przy pomocy automatu A'. Po przeczytaniu słowa w pozostaje sprawdzić, czy na współrzędnej automatu produktowego odpowiadającej obliczonej funkcji f_w pojawił się stan akceptujący.

Poniżej podamy formalną konstrukcję automatu B, takiego że

$$L(B) = \{w \,:\, w^{|w|} \in L\}$$

Ustawmy zbiór Q^Q w ciąg: $Q^Q = \{g_1, \ldots, g_{n^n}\}$ (pamiętamy $n = |Q|$). Określamy

- zbiór stanów automatu B: Q^{n+n^n},
- stan początkowy: $(q_1, q_1, \ldots, q_1)$,
- funkcja przejścia:

$$(p_1, \ldots, p_n, s_1, \ldots, s_{n^n}), \sigma \to (\delta(p_1, \sigma), \ldots, \delta(p_n, \sigma), g_1(s_1), \ldots, g_{n^n}(s_{n^n}))$$

- stany akceptujące:

$$\{(p_1, \ldots, p_n, s_1, \ldots, s_{n^n}) \,:\, (p_1, \ldots, p_n) = g_j \wedge s_j \in F \text{ dla pewngo } 1 \leq j \leq n^n\}$$

Literatura

- A.V.Aho, J.E.Hopcroft, J.D.Ullman, *Projektowanie i analiza algorytmów komputerowych*, w serii: *Biblioteka Informatyki*, Państwowe Wydawnictwo Naukowe, Warszawa 1983.
- J.E.Hopcroft, J.D.Ullman, *Wprowadzenie do teorii automatów, języków i obliczeń*, Wydawnictwo Naukowe PWN, Warszawa 1994.
- A.Blikle, *Automaty i gramatyki. Wstęp do lingwistyki matematycznej*, Państwowe Wydawnictwo Naukowe, Warszawa 1971.