Teoretyczne podstawy informatyki

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. letni 2025/26

inf.ug.edu.pl/~amb/TPI

Lemat o pompowaniu

- jak pokazać, że język nie jest regularny?
- lemat: niech L jest akceptowany przez automat $\mathcal{A}$ o n stanach, niech $z \in L$ ma długość większą od n. Wówczas istnieje przedstawienie $z = uvw$, $|uv| \leq n$, $|v| \geq 1$, t.ż. $uv^\ell w \in L$ dla wszystkich $\ell \in \mathbb{N}$
- dw.: ścieżka $z \in L$ nim dojdzie do stanu akceptującego musi powtórzyć jakiś stan
  u jest słowem pierwszego dotarcia do tego stanu, v jest pętlą wracającą do tego stanu
  z konstrukcji wynika zarówno $|uv| \leq n$, $|v| \geq 1$, jak i możliwość wielokrotnego przechodzenia pętli
- zastosowanie: $\{a^n b^n \mid n \in \mathbb{N}\}$ nie jest językiem regularnym
  dw.: wybieramy n tak duże, by pompowaniu podlegało podsłowo w a^n, wówczas do języka należałyby również słowa $a^k b^n$ dla $k > n$

Przykłady języków nieregularnych

- $\{wcw \mid w \in \{a,b\}^*\}$
- $\{wcw^R \mid w \in \{a,b\}^*\}$
- $\{a^m b^n \mid m < n\}$
  (można napompować początek słowa i naruszyć warunek $m < n$)
- $\{a^m b^n \mid m > n\}$
  dw.: $\{a^m b^n \mid \text{dowolne } m,n\} \setminus \{a^m b^n \mid m > n\} = \{a^m b^n \mid m \leq n\}$
  wynik różnicy nie jest regularny więc i $\{a^m b^n \mid m > n\}$ nie jest
- $\{w \mid |w|_a = |w|_b\}$
  (można rozważyć szczególne słowa, np. $w = a^n b^n$ i zastosować lemat o pompowaniu)
- $\{a^p \mid p \text{ jest liczbą pierwszą }\}$
  dw.: pompowanie fragmentu słowa wyprodukuje ciąg długości $p + p|v|$, nie jest to liczba pierwsza

Operacja przeplotu

- dane języki $L_1 \subseteq A_1^*$ oraz $L_2 \subseteq A^*$, przeplotem jest język

  $$L_1 \odot L_2 = \{u_1 v_1 \dots u_n v_n \mid u_1 \dots u_n \in L_1, v_1 \dots v_n \in L_2, u_i \in A_1^*, v_i \in A_2^*\}$$

  $$L_1 \odot L_2 \subseteq (A_1 \cup A_2)^*$$
- tw.: przeplot języków regularnych jest regularny
- dw.: niech $\mathcal{A}_1$ oraz $\mathcal{A}_2$ są automatami akceptującymi języki L_1 i L_2, budujemy produkt tych automatów z relacją przejścia
  $((q_1,q_2),a) \mapsto (q_1',q_2')$ jeśli $(q_1,a) \mapsto q_1'$ w $\mathcal{A}_1$ i $q_2' = q_2$ albo $(q_2,a) \mapsto q_2'$ w $\mathcal{A}_2$ i $q_1' = q_1$
  stanem początkowym jest para stanów początkowych, akceptują pary stanów końcowych

Wyrażenia regularne

Wyrażenia regularne

- służą do definiowania języków regularnych
- dany alfabet A,
  wyrażenia regularne to m.in. $\emptyset$, ε, a, dla każdego $a \in A$
  definiują one języki $\emptyset$, $\{\varepsilon\}$, $\{a\}$
- jeśli r i s są wyrażeniami regularnymi, to są nimi również $r + s$, rs, r^*, (r) (zamiast $r + s$ pisze się również $r|s$)
  definiują one języki $L(r) \cup L(s)$, $L(r) \cdot L(s)$, $L(r)^*$ oraz $L(r)$
- np. $(\varepsilon + a)(ba)^*(\varepsilon + b)$
  definiuje język, w którym a i b występują na przemian, nie narzucamy czy musi zaczynać się lub kończyć konkretną literą
  słowo puste również wchodzi do tego języka
- nawiasy nie zmieniają języka, służą jedynie do budowy wyrażenia regularnego
- *nie ma* konstrukcji dla przekroju, różnicy czy dopełnienia

Tożsamości dla wyrażeń regularnych

- piszemy $r = s$ rozumiejąc $L(r) = L(s)$
- przemienność dodawania: $r + s = s + r$
- ale nie dla mnożenia, na ogół $rs \neq sr$
- łączność dodawania i mnożenia: $(r+s)+t = r+(s+t)$, $(rs)t = r(st)$
- pochłanianie dodawania: $r + r = r$
- rozdzielczość z każdej strony: $r(s + t) = rs + rt$ i $(s + t)r = sr + tr$
- zero: $r + \emptyset = r$, $r\emptyset = \emptyset r = \emptyset$
- jedynka: $r\varepsilon = \varepsilon r = r$
- iteracje: $r^* = r^+ + \varepsilon$, gdzie oznaczamy $r^+ = rr^* = r^*r$
  $r^*r^* = (r^*)^* = (r^+)^* = (r^*)^+ = r^*$, $(r^+)^+ = r^+$
- rotacje dla iteracji: $r(sr)^* = (rs)^*r$
- iteracja a suma:
  $(r + s)^* = (r^*s^*)^* = (s^*r^*)^* = (\varepsilon + r^+)(s^+r^+)^*(\varepsilon + s^+)$

Twierdzenie Kleene'ego

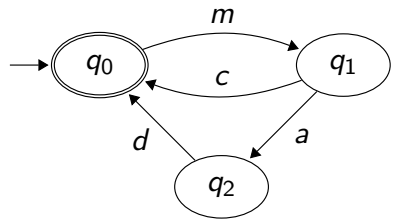
- wyrażenia regularne są równoważne automatom skończonym (tzn. generują te same języki)
- dw. cz.1: dla wyrażenia budujemy automat (niedeterministyczny, z cichymi przejściami) generujący ten sam język
  - dla $\emptyset$ jedyny stan początkowy nie jest akceptujący
  - dla ε jedyny stan początkowy jest akceptujący
  - dla $a \in A$ automat ma jedną strzałkę etykietowaną a ze stanu początkowego do końcowego
  - suma: ze stanu początkowego mamy dwa ciche przejścia do stanów początkowych obu automatów, akceptują oba automaty
  - konkatenacja: dodajemy ciche przejścia ze stanów akceptujących pierwszego automatu do początkowego drugiego
  - gwiazdka Kleene'ego: ciche przejścia ze stanów akceptujących z powrotem do początkowego, który też staje się akceptującym
  - nawiasy niczego nie zmieniają

Twierdzenie Kleene'ego, c.d.

- dw. cz.2: dla automatu deterministycznego budujemy wyrażenie regularne,
  niech $Q = \{q_0, \ldots, q_n\}$
  - niech $r^k_{i,j}$ zawiera słowa, które pozwalają przejść od q_i do q_j poprzez stany pośrednie $q_0, \ldots, q_{k-1}$
    pokażemy, że języki te są generowane przez wyrażenia regularne
  - $$r^0_{i,j} = \Sigma_{\delta(q_i,a)=q_j} a$$
    wyłącznie bezpośrednie przejścia z q_i do q_j
    pusta suma daje $\emptyset$, jeśli $i = j$, to dodatkowo $+\varepsilon$,
  - $$r^{k+1}_{i,j} = r^k_{i,j} + r^k_{i,k} \, (r^k_{k,k})^* \, r^k_{k,j}$$
    czyli użycie q_k oznacza przejście z q_i do q_k, być może pętlenie się w tym stanie, i przejście do q_j

Twierdzenie Kleene'ego, c.d.

- ostateczne wyrażenie regularne jest sumą $r^k_{i,j}$
  dla $i = 0$ (zaczynamy od stanu początkowego)
  dla wszystkich j takich, że $q_j \in F$ (kończymy w stanie akceptującym)
  dla $k = n + 1$ (wszystkie stany wolno odwiedzać)
  $$r = \Sigma_{q_j \in F} \; r^{n+1}_{0,j}$$
- przykład: $r = r^3_{0,0}$



- $r^3_{0,0} = r^2_{0,0} + r^2_{0,2}(r^2_{2,2})^* r^2_{2,0}$
- $r^2_{0,0} = r^1_{0,0} + r^1_{0,1}(r^1_{1,1})^* r^1_{1,0}$
  $= \varepsilon + m(cm)^* c = (mc)^*$
- $r^2_{0,2} = \emptyset + m(cm)^* a = (mc)^* ma$
- $r^2_{2,2} = \varepsilon + dm(cm)^* a = \varepsilon + d(mc)^* ma$
- $r^2_{2,0} = d + dm(cm)^* c = d(mc)^*$
- $r = (mc)^* + (mc)^* ma(d(mc)^* ma)^* d(mc)^*$
  $= (mc)^* + ((mc)^* mad)^+ (mc)^* = (mc + mad)^*$

Podstawienie

- dany alfabet A i język $L \subseteq A^*$
- dany być może inny alfabet Σ i dla każdej litery $a \in A$ język $L_a \subseteq \Sigma^*$
- dla słowa $u \in A^*$ definiujemy język $L_u = L_{u_1} \cdot \ldots \cdot L_{u_\ell}$
- dla całego języka L definiujemy $subs(L) = \bigcup_{u \in L} L_u$
- tw. jeśli L i wszystkie języki L_a są regularne, to $subs(L)$ jest językiem regularnym nad Σ
- dw.: możemy to wszystko opisać podstawiając wyrażenia regularne
  - dla każdej litery $a \in A$ dane jest wyrażenie regularne r_a nad alfabetem Σ, definiujące L_a
  - dane jest też wyrażenie regularne r nad A definiujące L
  - każde wystąpienie litery a w r zastępujemy wyrażeniem (r_a) i ew. usuwamy niepotrzebne nawiasy

Podstawienie, przykład

- niech $A = \{r, s, t\}$, rozważamy wyrażenia regularne np. $r(sr)^*$ albo $(rs)^* r$ albo $(r + s)^*$ albo $(r^* s^*)^*$ albo $(\varepsilon + r^+)(s^+ r^+)^*(\varepsilon + s^+)$
- dla alfabetu Σ rozważamy języki definiowane wyrażeniami R, S i T nad Σ, definiujemy podstawienie $h : A \to \mathcal{P}(\Sigma^*)$ jako $h(r) = L(R)$, $h(s) = L(S)$, $h(t) = L(T)$
- dla każdego ze wspomnianych wyrażeń nad trzema literami mamy podstawienie, w którym występują całe wyrażenia nad alfabetem Σ
- inny przykład, liczby są definiowane wyrażeniem $znak \; cyfra\_nie0 \, (cyfra)^* (\varepsilon | . \, cyfra \, (cyfra)^*)$ gdzie $znak = \varepsilon | + | -$, $cyfra\_nie0 = 1|2|3|4|5|6|7|8|9$, $cyfra = 0|1|2|3|4|5|6|7|8|9$
- pierwsze wyrażenie jest nad alfabetem $\{znak, cyfra\_nie0, cyfra, .\}$, kolejne równości określają homomorfizm z tego alfabetu do $\mathcal{P}((\{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\})^*)$, po wykonaniu podstawienie otrzymujemy wyrażenie regularne nad alfabetem $\{+, -, ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ definiujące zapis liczb dziesiętnych

Homomorfizm

- homomorfizm $h : A \to \Sigma^*$ będziemy automatycznie przedłużać na słowa, $h : A^* \to \Sigma^*$, $h(u) = h(u_1) \dots h(u_\ell)$
- tw.: obraz i przeciwobraz języka regularnego jest językiem regularnym
  - dw.: obraz: homomorfizm to szczególne podstawienie $L_a = \{h(a)\}$, było przed chwilą
  - przeciwobraz: niech $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ będzie automatem definiującym język $L \subseteq \Sigma^*$
    definiujemy przejścia w Q etykietowane symbolami z A następująco $\delta'(q, a) = \delta(q, h(a))$
    akceptowane są słowa z A^* prowadzące do stanów akceptujących

Homomorfizm, przykład

- niech $h : \{a, b\} \to \{0, 1\}^*$ będzie zdefiniowany jako $h(a) = 01$ i $h(b) = 10$, niech $r = (00 + 1)^*$, $L = L(r)$
- wówczas $h^{-1}(L) = L((ba)^*)$
- zawieranie $\supseteq$: niech $w \in L((ba)^*)$ czyli $w = (ba)^n$, wówczas $h(w) = (1001)^n \in L$
- zawieranie $\subseteq$: niech $w \in \{a, b\}^*$, $h(w) \in L$, czy jest możliwe, że $w \notin L((ba)^*)$? kilka przypadków
  - $w = a \dots$, wówczas $h(w) = 01 \dots \notin L$
  - $w = \dots b$, wówczas $h(w) = \dots 10 \notin L$
  - $w = \dots aa \dots$, wówczas $h(w) = \dots 0101 \dots \notin L$
  - $w = \dots bb \dots$, wówczas $h(w) = \dots 1010 \dots \notin L$

  niemożliwe, czyli $h(w) \in L \Leftrightarrow w \in L((ba)^*)$
- jeśli $s = (00 + 11)^*$ to przeciwobraz jest równy $\{\varepsilon\}$
- jeśli $s = (00 + 11)^+$ to przeciwobraz jest równy $\emptyset$

Własności homomorfizmów

- niech $h : A \to \Sigma^*$ będzie homomorfizmem przedłużonym do $h : A^* \to \Sigma^*$, $L_1, L_2 \subseteq A^*$ dowolnymi językami, wówczas
  - $h(L^*) = (h(L))^*$
  - $h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$
  - $h(L_1 \cdot L_2) = h(L_1) \cdot h(L_2)$
  - $h^{-1}(L_1) \cdot h^{-1}(L_2) \subseteq h^{-1}(L_1 \cdot L_2)$
- ale nie ma zawierania $h^{-1}(L_1 \cdot L_2) \subseteq h^{-1}(L_1) \cdot h^{-1}(L_2)$
  $\Sigma = \{0\}$, $L_1 = L_2 = \{0\}$, $L_1 \cdot L_2 = \{00\}$, $A = \{a\}$, $h(a) = 0$,
  $h(w) = 00$ ma jedno rozwiązanie $w = a$, $h^{-1}(L_1 \cdot L_2) = \{a\}$
  $h^{-1}(L_1) = \emptyset$, czyli $h^{-1}(L_1) \cdot h^{-1}(L_2) = \emptyset$
- podobnie $h^{-1}(L_1^*) = a^*$ ale $(h^{-1}(L_1))^* = \emptyset$