Obliczalność i złożoność

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

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

Języki regularne, automaty skończone, c.d.

Iloczyn kartezjański automatów

- dane dwa automaty (deterministyczne) $\mathcal{A}_1$ i $\mathcal{A}_2$ nad tym samym językiem A, definiujące języki L_1 i L_2, funkcje przejścia całkowite (zawsze można tak założyć uzupełniając brak przejścia dodatkowym stanem pułapkowym)
- w iloczynie kartezjańskim $Q_1 \times Q_2$ definiujemy przejścia jako skoordynowane przejścia, tj. $\delta(\langle q_1, q_2\rangle, a) = \langle \delta_1(q_1, a), \delta_2(q_2, a)\rangle$
- czyli każdy automat działa niezależnie i rejestrujemy ich stany
- stan początkowy jest parą stanów początkowych
- $L_1 \cap L_2$ będzie definiowany przez automat, którego stan akceptujący jest parą dwóch stanów akceptujących
- $L_1 \cup L_2$ będzie definiowany przez automat, który zaakceptuje gdy jedna ze współrzędnych będzie stanem akceptującym
- można też określić automaty akceptujące obie różnice języków

Operacje na językach regularnych

- tw.: jeśli $K, L \subseteq A^*$ są językami regularnymi, to regularne są też
  suma $K \cup L$
  konkatenacja $K \cdot L$
  gwiazdka Kleene'ego K^*: odpowiednia konstrukcja nowego automatu
  przekrój $K \cap L$
  różnica $K \setminus L$, $L \setminus K$, $K \oplus L$
  dopełnienie $A^* \setminus K$: iloczyn kartezjański automatów

Podstawienie

- dany alfabet A i język $L \subseteq A^*$
- dany inny alfabet Σ, i dla każdej litery $a \in A$ język $L_a \subseteq \Sigma^*$
- czyli dla słowa $u \in A^*$ określony jest język $L_{u_1} \cdot \ldots \cdot L_{u_\ell}$
- tw. jeśli wszystkie te języki są regularne, to powstaje znowu język regularny nad Σ
- dw.: możemy to wszystko opisać podstawiając wyrażenia regularne
- def.: homomorfizm gdy $L_a = \{w_a\}$
- obraz i przeciwobraz przy homomorfiźmie jest językiem regularnym
- dw.: obraz, było przed chwilą
  przeciwobraz: niech $h : A \to \Sigma^*$ jest homomorfizmem,
  $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ automatem definiującym język $L \subseteq \Sigma^*$
  definiujemy przejścia etykietowane symbolami z A poprzez
  $\delta'(q, a) = \delta(q, h(a))$
  akceptowane są słowa z A^* prowadzące do stanów akceptujących

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 $z = uvw$, $|uv| \leqslant n$, $|v| \geqslant 1$, t.ż. $uv^\ell w \in L$ dla wszystkich ℓ
- 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 $|uv| \leqslant n$, $|v| \geqslant 1$ jak i możliwość wielokrotnego przechodzenia pętli
- zastosowanie: $\{a^n b^n \mid n \in \mathbb{N}\}$ nie jest językiem regularnym
  pętla albo leży całkowicie w części a lub b, wówczas nie zgadza się liczba wystąpień
  albo łączy fragment ab i po napompowaniu występuje a po b

Wyrażenia regularne w językach programowania

- oprócz gwiazdki $*$ – dowolnego powtórzenia, są jeszcze wersje
  $+$ – powtórzenie co najmniej jeden raz
  ? – element opcjonalny, wystąpienie 0 lub 1 raz
  różne wersje określające dokładniej liczbę wystąpień, minimalną, maksymalną, zakres
- oznaczenia na całe zbiory symboli
  [...] – wylicza możliwy zbiór symboli
  są oznaczenia na uzupełnienie zbioru, tylko litery, tylko cyfry, znaki białe, zakresy znaków
  poszukiwanie wzorca na początku lub na końcu wiersza
- przy operacji zamiany wzorca na inny można odwoływać się do znalezionych fragmentów wzorca
- zawsze trzeba przestudiować dokumentację, różne języki mają pewne możliwości a pewnych innych nie mają