Obliczalność i złożoność

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

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

Twierdzenie Rice'a

Teza Turinga-Churcha

Teza

Problem jest rozstrzygalny wtedy i tylko wtedy, gdy istnieje (deterministyczna) maszyna Turinga zatrzymująca się dla każdej instancji i akceptująca te instancje, dla których odpowiedź jest pozytywna

- co rozumiemy przez instancje problemu?
- uwaga: to nie jest twierdzenie! raczej definicja i przekonanie, że prawidłowo opisaliśmy możliwość obliczeń, w szczególności rozstrzygnięć

Przykłady problemów rozstrzygalnych

- problem równoważności deterministycznych automatów skończonych (czy akceptują ten sam język)
- problem osiągalności w grafie
- problem słowa dla gramatyki bezkontekstowej (algorytm CYK)
- problem nieskończoności języka regularnego (pętla w automacie)
- problem nieskończoności języka bezkontekstowego (pętla w grafie symboli terminalnych)
- problem pustości języka bezkontekstowego

terminologia:

- język rekurencyjny jest akceptowany przez maszynę Turinga, która zatrzymuje się dla każdego wejścia
- problem rozstrzygalny ma maszynę Turinga z własnością stopu
- język L jest rekurencyjny jeśli problem $w \in L$ jest rozstrzygalny

Redukcja

- niech $K \subset \Sigma^*$ oraz $L \subset \Delta^*$ będą dwoma językami, K **redukuje się** do L, oznaczenie
$$K \leq L$$
jeżeli istnieje obliczalna funkcja $f : \Sigma^* \to \Delta^*$ taka, że dla każdego $w \in \Sigma^*$
$$w \in K \Leftrightarrow f(w) \in L$$
- istnienie redukcji $K \leq L$ oznacza, że język K **nie** jest trudniejszy od języka L – jeżeli mamy algorytm rozpoznający L, to mamy także algorytm rozpoznający K

Twierdzenie

Jeżeli $K \leq L$ oraz L jest rozstrzygalny, to K jest rozstrzygalny

Wniosek

Jeżeli $K \leq L$ oraz K jest nierozstrzygalny, to L jest nierozstrzygalny

Język uniwersalny a problem stopu

- tw: język $L_u = \{\langle M, w\rangle \mid M \text{ akceptuje } w\}$ nie jest językiem rekurencyjnym
- tw.: Język $L_{stop} = \{\langle M, w\rangle \mid M \text{ zatrzymuje się dla } w\}$ nie jest językiem rekurencyjnym
dw.: język L_u redukuje się do języka L_{stop}: parze $\langle M, w\rangle$ przypisujemy parę $\langle M', w\rangle$ taką, że M' zatrzymuje się na w wtedy i tylko wtedy, gdy M akceptuje w; jeżeli M zatrzymuje się w stanie odrzucającym, to M' wpada w nieskończoną pętlę

Nierozstrzygalność problemu stopu

Problem stopu dla maszyny Turinga jest nierozstrzygalny

Przykład

- czy język jest rekurencyjny?
$$L_{100} = \{\langle M, w\rangle \mid M \text{ zatrzymuje się dla } w \text{ po 100 krokach}\}$$
- oczywiście, po wykonaniu najwyżej 100 kroków mamy rozstrzygnięcie

Twierdzenie Rice - potrzebne pojęcia

- niech Φ będzie pewną własnością języków rekurencyjnie przeliczalnych nad alfabetem $\{0, 1\}^*$, na przykład:
$$\Phi_1(L) \equiv L \text{ jest skończony}$$
$$\Phi_2(L) \equiv L \text{ nie zawiera słowa pustego}$$
każdej takiej własności odpowiada rodzina $\mathcal{S}$ języków rekurencyjnie przeliczalnych, które mają daną własność
- def.: własność jest **trywialna**, jeżeli $\mathcal{S}$ jest pusta lub $\mathcal{S}$ zawiera wszystkie języki rekurencyjnie przeliczalne nad $\{0, 1\}$
rodzinie języków $\mathcal{S}$ odpowiada zbiór kodów
$$L_{\mathcal{S}} = \{\langle M\rangle \mid L(M) \in \mathcal{S}\}$$

Twierdzenie Rice'a

Twierdzenie Rice'a

żadna nietrywialna własność języków rekurencyjnie przeliczalnych nie jest rozstrzygalna

- innymi słowy, dla żadnej nietrywialnej rodziny $\mathcal{S}$ zbiór $L_{\mathcal{S}}$ nie jest rekurencyjny
- czyli problem $\{\langle M \rangle \mid L(M) \in \mathcal{S}\}$ jest nierozstrzygalny
- jest to problem *semantyczny*, mówi o języku akceptowanym przez maszynę Turinga, a nie np. o jej strukturze
- problemy rodzaju "maszyna zatrzyma się po 100 krokach" są rozstrzygalne

Szkic dowodu twierdzenia

- Φ – nietrywialna własność języków rekurencyjnie przeliczalnych
- $\mathcal{S}$ – rodzina języków rekurencyjnie przeliczalnych, które mają własność Φ
- zakładamy, że $\emptyset$ nie należy do $\mathcal{S}$ (wp.p. rozpatrujemy własność przeciwną).
- niech $L \in \mathcal{S}$, niech M_L będzie maszyną Turinga akceptującą L
- definiujemy redukcję f języka uniwersalnego L_u do $L_{\mathcal{S}}$: $f(\langle M, w \rangle) = \langle M' \rangle$ gdzie

$$L(M') = \begin{cases} \emptyset & \text{jeżeli } M \text{ nie akceptuje } w \\ L & \text{jeżeli } M \text{ akceptuje } w \end{cases}$$

$f(\langle M, w \rangle)$ na słowie w' działa tak, że najpierw wykonujemy obliczenia M na w, a po ew. zaakceptowaniu wykonujemy obliczenia M_L na w'

Wnioski z twierdzenia Rice'a

- poniższe własności języków rekurencyjnie przeliczalnych są nierozstrzygalne:
  - pustość
  - skończoność
  - regularność
  - bezkontekstowość

Problemy nierozstrzygalne

Problemy nierozstrzygalne dla gramatyk i języków bezkontekstowych

- obliczenie **poprawne** deterministycznej maszyny Turinga M:

$$\beta_1 \# \beta_2^R \# \cdots \# \beta_n \# \quad \text{(lub } \beta_n^R \text{, gdy } n \text{ parzyste)}$$

  - β_1 jest konfiguracją początkową,
  - β_n jest konfiguracją akceptującą,
  - konfiguracja β_{i+1} jest bezpośrednim następnikiem β_i ($\beta_i \longrightarrow_M \beta_{i+1}$), dla $i = 1, \ldots, n-1$
- $Corr_M$ oznacza zbiór obliczeń poprawnych

Przekrój dwóch języków bezkontekstowych

- $$L_1 = \{x_1 \# x_2 \# \cdots \# x_n \# \ : \ x_i \longrightarrow_M x_{i+1}^R, i - \textit{nieparzyste}\}$$
$$L_2 = \{x_1 \# x_2 \# \cdots \# x_n \# \ : \ x_i \longrightarrow_M x_{i+1}^R, i - \textit{parzyste}\}$$
$$Corr_M = L_1 \cap L_2$$
- poniższe języki są bezkontekstowe (automat ze stosem!)

$$L_3 = \{y \# z^R \ : \ y \longrightarrow_M z\}$$

$$L_4 = \{y^R \# z \ : \ y \longrightarrow_M z\}$$

- również

$$L_1 = (L_3 \#)^* (\{\lambda\} \cup \Gamma^* F \Gamma^* \#)$$

$$L_2 = q_0 \Sigma^* \# (L_4 \#)^* (\{\lambda\} \cup \Gamma^* F \Gamma^* \#)$$

  są bezkontekstowe

Nierozstrzygalność niepustości przekroju jęz. bezkontekstowych

- $Corr_M$ jest niepusty wtedy i tylko wtedy, gdy język maszyny Turinga M jest niepusty, inaczej
- problem pustości języka maszyny Turinga sprowadza się do rozstrzygania czy $Corr_M$ niepusty

Problem przekroju języków bezkontekstowych

Problem niepustości $L(G_1) \cap L(G_2)$ dla gramatyk bezkontekstowych G_1, G_2 jest nierozstrzygalny

Uzupełnienie zbioru obliczeń poprawnych

- napis nad alfabetem $Q \cup \Gamma$ nie jest obliczeniem poprawnym, jeśli
  1. nie jest postaci $x_1 \# x_2 \cdots x_n \#$, gdzie x_i konfiguracje, lub
  2. x_1 nie jest konfiguracją początkową, lub
  3. x_n nie jest konfiguracją akceptującą, lub
  4. nie zachodzi $x_i \longrightarrow_M x_{i+1}^R$ dla pewnego i nieparzystego, lub
  5. nie zachodzi $x_i^R \longrightarrow_M x_{i+1}$ dla pewnego i parzystego,
- warunki 1, 2, 3 można sprawdzić przy pomocy automatu skończonego; pozostałe przy pomocy automatu ze stosem
- suma języka regularnego i bezkontekstowego jest językiem bezkontekstowym
- uzupełnienie zbioru obliczeń poprawnych maszyny Turinga jest generowane przez gramatykę bezkontekstową

Nierozstrzygalność problemu uniwersalności

Nierozstrzygalność problemu uniwersalności

dla gramatyki bezkontekstowej G nad alfabetem końcowym Σ problem $L(G) = \Sigma^*$ jest nie rozstrzygalny

- dw.: dla maszyny Turinga M zbiór obliczeń niepoprawnych (uzupełnienie zbioru obliczeń poprawnych) jest generowany przez gramatykę bezkontekstową G, czyli $L(G) = \Sigma^*$ wtedy i tylko wtedy gdy $L(M) = \emptyset$.

Problemy nierozstrzygalne dla gramatyk i języków bezkontekstowych

- niech G_1, G_2 gramatyki bezkontekstowe, R - język regularny, następujące problemy są nierozstrzygalne:
  1. $L(G_1) = L(G_2)$
  2. $L(G_1) = R$
  3. $L(G_2) \subset L(G_1)$
  4. $R \subset L(G_1)$
- dw.: wystarczy przyjąć $R = L(G_2) = \Sigma^*$
- **uwaga**: problem $L(G) \subset R$ jest rozstrzygalny, bo jest równoważny $L(G) \cap (\Sigma^* \setminus R) = \emptyset$, przekrój języka bezkontekstowego i regularnego jest bezkontekstowy, problem sprowadza się do pustości języka bezkontekstowego