

Teoretyczne podstawy informatyki

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. letni 2025/26

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

Andrzej M. Borzyszkowski (Instytut Informatyki) Teoretyczne podstawy informatyki sem. letni 2025/26 1 / 7

Wykład 10

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

Andrzej M. Borzyszkowski (Instytut Informatyki) Teoretyczne podstawy informatyki sem. letni 2025/26 3 / 7

Wykład 10

Maszyny Turinga – nierozstrzygalność

Andrzej M. Borzyszkowski (Instytut Informatyki) Teoretyczne podstawy informatyki sem. letni 2025/26 2 / 7

Wykład 10

Przykłady problemów rozstrzygalnych

- problem osiągalności w grafie
- problem słowa dla języka regularnego lub bezkontekstowego
- problem pustości języka regularnego lub bezkontekstowego
- problem nieskończoności języka regularnego lub bezkontekstowego
- problem zawierania języka regularnego lub bezkontekstowego w regularnym
- problem równości języków regularnych

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

Andrzej M. Borzyszkowski (Instytut Informatyki) Teoretyczne podstawy informatyki sem. letni 2025/26 4 / 7

Redukcja

- def.: język $L \subset \Sigma^*$ redukuje się do języka $K \subset \Delta^*$, oznaczenie

$$L \leq K$$

jeżeli istnieje obliczalna funkcja $f : \Sigma^* \rightarrow \Delta^*$ taka, że dla każdego $w \in \Sigma^*$

$$w \in L \Leftrightarrow f(w) \in K$$

- istnienie redukcji $L \leq K$ oznacza, że język L **nie** jest trudniejszy od języka K – jeżeli mamy algorytm rozpoznający K , to mamy także algorytm rozpoznający L

Twierdzenie

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

Wniosek

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

Problemy nierozstrzygalne dla gramatyk i języków bezkontekstowych

- niech G_1, G_2 gramatyki bezkontekstowe, R - język regularny, następujące problemy są nierozstrzygalne:
 - 1 $L(G_1) \subset L(G_2)$
 - 2 $L(G_1) = L(G_2)$
 - 3 $R \subset L(G_2)$
 - 4 $R = L(G_2)$
 - 5 $L(G_2) = \Sigma^*$
 - 6 problem niepustości $L(G_1) \cap L(G_2)$
- uwaga:** problem $L(G_1) \subset R$ jest rozstrzygalny, bo jest równoważny $L(G_1) \cap (\Sigma^* \setminus R) = \emptyset$, przekrój języka bezkontekstowego i regularnego jest bezkontekstowy, problem sprowadza się do pustości języka bezkontekstowego
- dowody nierozstrzygalności budowane są poprzez odpowiednie redukcje problemów

Język uniwersalny a problem stopu

Język uniwersalny

Tw.: język $L_u = \{\langle M, w \rangle \mid M \text{ akceptuje } w\}$ nie jest rekurencyjny

Język L_{stop}

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