

Częściowa rozstrzygalność

W czasie poprzedniego wykładu poznaliśmy pojęcie rozstrzygalności. Aby rozważany język L (problem decyzyjny należenia do L) był rozstrzygalny, wymagamy istnienia algorytmu zatrzymującego się dla każdego danych wejściowych i dającego odpowiedź *TAK* dla słów $w \in L$ oraz *NIE* dla słów $w \notin L$. Problemy nie spełniające powyższego warunku nazywamy nierozstrzygalnymi. Możemy wśród nich znaleźć zarówno takie, dla których nie istnieją algorytmy dające odpowiedź w którymkolwiek przypadku ($w \in L$ lub $w \notin L$), jak i takie, dla których istnieją algorytmy dające odpowiedź wyłącznie w jednym przypadku: *TAK* dla $w \in L$ lub *NIE* dla $w \notin L$. Problemy, dla których istnieją algorytmy dające pozytywną odpowiedź w przypadku $w \in L$ nazywamy częściowo rozstrzygalnymi (lub rozpoznawalnymi).

Definicja 8.1

Język (problem) $L \subseteq \Sigma^*$ nazywamy rozpoznawalnym (częściowo rozstrzygalnym) jeśli istnieje maszyna Turinga, która go rozpoznaje (t.j. zatrzymuje się w stanie akceptującym dla wszystkich $x \in L$ i tylko dla nich).

Przykład 8.1

Problem stopu jest częściowo rozstrzygalny. Odpowiada mu język:

$$L_S = \left\{ (M, w) : \text{maszyna Turinga } M \text{ zatrzymuje się na } w \right\}.$$

Maszyna Turinga M_S rozpoznająca ten język symuluje działanie maszyny M na wejściu w . Jeśli maszyna M zatrzyma się, M_S zatrzyma się w stanie

akceptującym. Jeśli natomiast maszyna M się nie zatrzyma, M_S również się nie zatrzyma.

Zauważmy, że rozpoznawalność jest własnością słabszą niż rozstrzygalność. Wymaganie, aby maszyna Turinga zatrzymywała się dla każdego danych wejściowych ogranicza klasę języków, które może ona rozpoznać.

Zbiory rekurencyjnie przeliczalne

Przypomnijmy, że językom rozstrzygalnym odpowiadają zbiory rekurencyjne. Podobnie, językom rozpoznawalnym odpowiadają zbiory rekurencyjnie przeliczalne.

Definicja 8.2

Niech $A \subseteq \mathbb{N}^n$ będzie zbiorem, zaś $cc_A : \mathbb{N}^n \rightarrow \mathbb{N}$ jego częściową funkcją charakterystyczną określoną:

$$cc_A(\bar{x}) = \begin{cases} 1 & \bar{x} \in A \\ \infty & \bar{x} \notin A \end{cases}.$$

Zbiór A nazywamy rekurencyjnie przeliczalnym, jeśli jego częściowa funkcja charakterystyczna jest obliczalna ($cc_A \in C_n$).

Przykład 8.2

1. Każdy zbiór rekurencyjny jest rekurencyjnie przeliczalny.
2. Język $L_1 = \{(M, x) : \text{maszyna Turinga } M \text{ akceptuje słowo } x\}$ jest rekurencyjnie przeliczalny.
3. Język $L_2 = \{(M, x) : \text{maszyna Turinga } M \text{ nie akceptuje słowa } x\}$ nie jest rekurencyjnie przeliczalny. Dopełnienie L_2 jest językiem rekurencyjnie przeliczalnym.
4. Język $L_3 = \{x \in \mathbb{N} : \phi_x \text{ jest totalna}\}$ nie jest rekurencyjnie przeliczalny. Jego dopełnienie również nie jest językiem rekurencyjnie przeliczalnym.

Zauważmy, że jeśli zbiór A jest rekurencyjnie przeliczalny (jego częściowa funkcja charakterystyczna cc_A jest obliczalna), problem „ $\bar{x} \in A$ ” jest częściowo rozstrzygalny (rozpoznawalny). Z drugiej strony, jeśli problem „ $\bar{x} \in A$ ” jest częściowo rozstrzygalny, funkcja cc_A jest obliczalna, a zatem zbiór A

jest rekurencyjnie przeliczalny. Relację między rekurencyjną przeliczalnością zbiorów oraz rozpoznawalnością problemów możemy zapisać w skrócie:

A jest rek. przeliczalny $\iff cc_A \in C_n \iff „\bar{x} \in A”$ jest rozpoznawalny

Bezpośrednio z definicji otrzymujemy następujące twierdzenie:

Twierdzenie 8.1

Język L jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy istnieje maszyna Turinga M taka, że $L = L(M)$.

Równoważnie:

Twierdzenie 8.2

Zbiór $A \subseteq \mathbb{N}^n$ jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy jest dziedziną pewnej funkcji obliczalnej.

Podstawowe własności zbiorów rekurencyjnie przeliczalnych opisuje poniższe twierdzenie:

Twierdzenie 8.3

Dla ustalonego $n \geq 1$ zbiór wszystkich podzbiorów rekurencyjnie przeliczalnych $A \subseteq \mathbb{N}^n$ jest zamknięty ze względu na sumę i przekrój zbiorów.

Dowód

Niech $A, B \subseteq \mathbb{N}^n$ będą zbiorami rekurencyjnie przeliczalnymi, zaś M_A i M_B maszynami Turinga obliczającymi ich częściowe funkcje charakterystyczne. Konstruujemy trzytaśmowe maszyny Turinga $M_{A \cup B}$ oraz $M_{A \cap B}$ obliczające częściowe funkcje charakterystyczne zbiorów $A \cup B$ oraz $A \cap B$. W obu konstruowanych maszynach na pierwszej taśmie znajduje się słowo wejściowe x , druga taśma służy do symulacji działania maszyny M_A , zaś trzecia do symulacji działania maszyny M_B . W czasie symulacji kolejne instrukcje maszyn M_A oraz M_B wykonywane są naprzemiennie.

Schemat działania maszyny $M_{A \cup B}$ na wejściu x :

- Uruchom „współbieżnie” maszyny M_A oraz M_B na danych x .
- Zaakceptuj x jeśli jedna z maszyn M_A, M_B zatrzyma się w stanie akceptującym.

Schemat działania maszyny $M_{A \cap B}$ na wejściu x :

- Uruchom „współbieżnie” maszyny M_A oraz M_B na danych x .
- Zaakceptuj x jeśli obie maszyny M_A , M_B zatrzymają się w stanie akceptującym.

■

Uwaga

Zauważmy, że zbiory rekurencyjnie przeliczalne nie są domknięte ze względu na operację dopełnienia. Przykładowo, język

$$L = \{(M, x) : \text{maszyna Turinga } M \text{ zatrzymuje się na słowie } x\}$$

jest rekurencyjnie przeliczalny, natomiast jego dopełnienie

$$\bar{L} = \{(M, x) : \text{maszyna Turinga } M \text{ nie zatrzymuje się na słowie } x\}$$

nie jest językiem rekurencyjnie przeliczalnym.

Twierdzenie 8.4

Język L jest rekurencyjny wtedy i tylko wtedy, gdy zarówno język L jak i jego dopełnienie \bar{L} są rekurencyjnie przeliczalne.

Dowód

\Rightarrow :

Założmy, że L jest zbiorem rekurencyjnym. Wówczas również jego dopełnienie \bar{L} jest zbiorem rekurencyjnym. Ponieważ każdy zbiór rekurencyjny jest rekurencyjnie przeliczalny, L oraz \bar{L} są rekurencyjnie przeliczalne.

\Leftarrow :

Niech L będzie językiem rekurencyjnie przeliczalnym takim, że jego dopełnienie \bar{L} również jest językiem rekurencyjnie przeliczalnym. Niech M_1 oraz M_2 będą maszynami Turinga rozpoznającymi odpowiednio języki L oraz \bar{L} . Skonstruujemy trzytaśmową maszynę Turinga M_L rozstrzygającą język L . Na pierwszej taśmie M_L znajduje się słowo wejściowe x , druga taśma służy do symulacji działania maszyny M_1 , natomiast trzecia do symulacji działania maszyny M_2 . W czasie symulacji kolejne instrukcje maszyn M_1 oraz M_2 wykonywane są naprzemiennie.

Schemat działania maszyny M_L na wejściu x :

- Uruchom „współbieżnie” maszyny M_1 oraz M_2 na danych x .
- Zaakceptuj x jeśli maszyna M_1 zatrzymała się w stanie akceptującym.
- Odrzuć x jeśli maszyna M_2 zatrzymała się w stanie akceptującym.

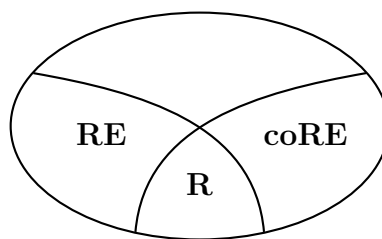
■

Zależności między klasami rozstrzygalności

Zaprezentowane powyżej rozważania dotyczące zbiorów rekurencyjnych oraz rekurencyjnie przeliczalnych pozwalają określić relację zależności między klasami rozstrzygalności problemów obliczeniowych.

Dla dowolnej pary wzajemnie się dopełniających języków L oraz \bar{L} zachodzi dokładnie jeden z warunków:

1. Oba języki L oraz \bar{L} są rekurencyjne.
2. Język L jest rekurencyjnie przeliczalny (ale nie rekurencyjny), zaś język \bar{L} nie jest rekurencyjnie przeliczalny.
3. Język \bar{L} jest rekurencyjnie przeliczalny (ale nie rekurencyjny), zaś język L nie jest rekurencyjnie przeliczalny.
4. Żaden z języków L oraz \bar{L} nie jest rekurencyjnie przeliczalny.



Rysunek 8.1: Relacje między zbiorami języków rekurencyjnych (**R**), rekurencyjnie przeliczalnych (**RE**) oraz dopełnień języków rekurencyjnie przeliczalnych (**coRE**).

Twierdzenie Rice'a-Shapiro

Twierdzenie Rice'a-Shapiro jest ważnym narzędziem pozwalającym badać częściową rozstrzygalność problemów obliczeniowych. Jest ono uogólnieniem twierdzenia Rice'a przedstawionego w czasie poprzedniego wykładu. Mówi ono, że jeśli pewna częściowo rozstrzygalna własność funkcji obliczalnych \mathcal{P} jest spełniana przez funkcję f , to jest ona spełniana również przez jej pewne skończone obcięcie. Równoważnie, jeśli pewna skończona funkcja g spełnia \mathcal{P} , muszą ją spełniać również wszystkie funkcje obliczalne, dla których g jest skończonym obcięciem.

Definicja 8.3

Niech $f, g : \mathbb{N}^n \rightarrow \mathbb{N}$ będą funkcjami. Powiemy, że funkcja g jest obcięciem funkcji f (ozn. $g \subseteq f$) jeśli:

$$\forall \bar{x} \in \mathbb{N}^n \quad \bar{x} \in D_g \Rightarrow \bar{x} \in D_f \wedge g(\bar{x}) = f(\bar{x}).$$

Twierdzenie 8.5 (Rice & Shapiro)

Niech \mathcal{A} będzie zbiorem funkcji obliczalnych takim, że zbiór ich indeksów $A = \{x \in \mathbb{N} : \phi_x \in \mathcal{A}\}$ jest rekurencyjnie przeliczalny. Wówczas $f \in \mathcal{A}$ wtedy i tylko wtedy, gdy istnieje skończona funkcja $\theta \in \mathcal{A}$ taka, że $\theta \subseteq f$.

Dowód

\Rightarrow :

Założmy, że zbiór A jest rekurencyjnie przeliczalny, zaś $f \in \mathcal{A}$ jest funkcją taką, że żadna skończona funkcja $\theta \subseteq f$ nie należy do \mathcal{A} . Ponadto, niech $P = P_e$ będzie programem obliczającym częściową funkcję charakterystyczną zbioru

$$K = \{x \in \mathbb{N} : x \in D_x\},$$

oraz

$$H(e, z, t) = \text{„}P_e \text{ zatrzymuje się na } z \text{ w co najwyżej } t \text{ krokach”}.$$

Zauważmy, że zbiór K jest rekurencyjnie przeliczalny, zaś problem $H(e, z, t)$ jest rozstrzygalny.

Rozważmy funkcję:

$$g(z, t) = \begin{cases} f(t) & \text{jeśli } H(e, z, t) \text{ nie jest prawdziwy} \\ \infty & \text{jeśli } H(e, z, t) \text{ jest prawdziwy} \end{cases}$$

Z rozstrzygalności $H(e, z, t)$ oraz obliczalności funkcji f i funkcji pustej wynika obliczalność funkcji g . Zatem, na mocy twierdzenia o parametryzacji, istnieje totalna i obliczalna funkcja s , taka że

$$\forall_{z,t} g(z, t) = \phi_{s(z)}(t).$$

Wówczas:

- Z określenia $\phi_{s(z)} \subseteq f$.
- Jeśli $z \in K \implies \phi_{s(z)}$ jest skończona, zatem $\phi_{s(z)} \notin \mathcal{A}$.
- Jeśli $z \notin K \implies \phi_{s(z)} = f \in \mathcal{A}$ (poza skończoną liczbą elementów).

Zatem $z \notin K \iff \phi_{s(z)} \in \mathcal{A}$, więc zbiór A nie może być rekurencyjnie przeliczalny, co jest sprzeczne z założeniem.

\Leftarrow :

Założmy, że zbiór A jest rekurencyjnie przeliczalny, oraz przypuśćmy, że istnieje obliczalna funkcja $f \notin \mathcal{A}$, dla której $\Theta \in \mathcal{A}$ dla pewnej skończonej funkcji $\Theta \subseteq f$. Rozważmy funkcję:

$$g(z, t) = \begin{cases} f(t) & \text{jeśli } t \in D_\Theta \vee z \in K \\ \infty & \text{w przeciwnym przypadku} \end{cases}$$

Zauważmy, że:

- D_Θ – dziedzina funkcji Θ jest zbiorem skończonym, więc jest zbiorem rekurencyjnym, a zatem również rekurencyjnie przeliczalnym.
- Zbiór $K = \{x \in \mathbb{N} : x \in D_x\}$ jest rekurencyjnie przeliczalny.
- Zbiór $B = D_\Theta \cup K$ również jest rekurencyjnie przeliczalny.
- Funkcja $g(z, t) = f(t) \cdot cc_B(z)$ jest obliczalna.

Zatem na mocy twierdzenia o parametryzacji istnieje totalna i obliczalna funkcja s , tż.

$$\forall_{z,t} g(z, t) = \phi_{s(z)}(t).$$

Wówczas:

- Jeśli $z \in K \implies \phi_{s(z)} = f \notin \mathcal{A}$.
- Jeśli $z \notin K \implies \phi_{s(z)} = \Theta \in \mathcal{A}$.

Zatem $z \notin K \iff \phi_{s(z)} \in \mathcal{A}$, więc zbiór \mathcal{A} nie może być rekurencyjnie przeliczalny, co daje sprzeczność z założeniem. ■

Przykład 8.3

Rozważmy zbiór $\mathcal{A} = \{\phi \in C_n : \phi \text{ jest totalna}\}$. Zauważmy, że do zbioru \mathcal{A} nie należy żadna funkcja skończona. Zatem na mocy twierdzenia Rice'a-Shapiro zbiór \mathcal{A} nie jest rekurencyjnie przeliczalny (problem „ ϕ_x jest totalna” nie jest częściowo rozstrzygalny).

Podobnie, zbiór $\overline{\mathcal{A}} = \{\phi \in C_n : \phi \text{ nie jest totalna}\}$ nie jest rekurencyjnie przeliczalny, ponieważ wszystkie funkcje skończone do niego należą (w szczególności funkcja pusta), ale zbiór ten jest różny od całego C_n .