Analiza Algoritmilor - Calculabilitate CheatSheet

Fie $f_Q : \mathbb{N} \to \{1, 0\}$, specificația problemei de decizie Q. Fie $A = \{x \in \mathbb{N} \mid f_Q(x) = 1\}$, mulțimea de adevăr asociată.

Definiție - Mulțime recursivă (R)

Mulțimea A este **recursivă** dacă și numai dacă există un program care răspunde pentru **orice** element din $\mathbb{N}$, în timp **finit**, dacă aparține sau nu mulțimii.

$$A \in R \iff P(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Exemple de mulțimi - R

- $\{x \in \mathbb{N} \mid$ x este un număr prim$\}$
- $\{x \in \mathbb{N} \mid P_x(w)$ se termină în cel mult t unități de timp$\}$[1]

[1] Folosim indicele x pentru a identifica un program specific din mulțimea infinit numărabilă a tuturor programelor.

Strategii de demonstrație - R

- Construim un program care corespunde specificației mulțimii A și care se termină în timp finit pentru orice intrare validă.
- Arătăm că putem reformula Q ca o instanță mai simplă a unei alte probleme cunoscute (Q_r) a cărei mulțime de adevăr este recursivă. ($Q \leq_T Q_r$)

Observații

- Dacă putem reformula Q ca o problemă de căutare într-un spațiu **finit** de posibilități, atunci Q este calculabilă. (ex: căutăm divizorii lui X în mulțimea numerelor mai mici decât X pentru a verifica dacă este prim).
- Dacă A este recursivă, atunci și complementul său ($\mathbb{N} \setminus A$) este o mulțime recursivă.

Definiție - Mulțime recursiv-enumărabilă (RE)

- Există un program care calculează răspunsul în timp **finit** pentru un element $x \in \mathbb{N}$ dacă $x \in A$ (dar poate să cicleze la infinit pentru $x \notin A$).

$$P(x) = \begin{cases} 1, & \text{pentru } x \in A \\ \{0, \perp\}, & \text{pentru } x \notin A \end{cases}$$

- Există un program G (generator) care la fiecare apel generează succesiv un element din A (orice element $x \in A$ va fi returnat după un număr **finit** de apeluri ale lui G). Dacă A este finită, G poate să cicleze după generarea ultimului element.

Exemple de mulțimi - RE

- $\{x \in \mathbb{N} \mid$ Programul $P_x(w)$ se termină$\}$
- $\{x \in \mathbb{N} \mid \exists y_1, .., y_k \in \mathbb{N} \to p(x, y_1, .., y_k) = 0\}$, unde p este un polinom cu coeficienți numere întregi [1]

Strategii de demonstrație - RE

- Conform definiției, putem fie să construim programul P, care se termină în timp finit pentru orice $x \in A$, fie generatorul echivalent care returnează toate elementele din A.
- Arătăm că putem reformula Q ca o instanță mai simplă a unei alte probleme cunoscute (Q_{rn}) a cărei mulțime de adevăr este RE. ($Q \leq_T Q_{rn}$)

Observații

- O mulțime recursivă este și recursiv-enumărabilă dar reciproca nu este în general adevărată.
- Dacă A este recursivă-enumărabilă dar nu este recursivă, atunci complementul său nu este o mulțime recursiv-enumărabilă.
- Principala dificultate asociată provine din faptul că nu putem explora un spațiu **infinit** de posibilități fără a avea o informație suplimentară care să ne permită să restrângem spațiul de căutare.

Exemple de mulțimi - !RE

- $\{x \in \mathbb{N} \mid$ Programul $P_x(w)$ **NU** se termină$\}$
- $\{x, y \in \mathbb{N} \mid \forall w. P_x(w) = P_y(w)\}$

Strategii de demonstrație - Mulțimi nerecursive

- Demonstrăm direct, prin reducere la absurd, că dacă mulțimea ar fi recursivă am obține o contradicție.
- Demonstrăm că putem reformula o problema cunoscută, Q_{nr} (ex: problema opririi), a cărei mulțime de adevăr este nerecursivă, ca o instanță mai simplă a problemei Q ($Q_{nr} \leq_T Q$).
- Analog, pentru a demonstra că o problemă nu este RE, arătăm că putem reformula o problemă cunoscută, Q_{nre} (ex: complementul problemei opririi), a cărei mulțime de adevăr nu este nici RE, ca o instanță mai simplă a problemei Q ($Q_{nre} \leq_T Q$).

Tipuri de probleme [3]

- Q este **decidabilă** dacă A este **recursivă**.
- Q este **semi-decidabilă** dacă A este **nerecursivă** dar este **recursiv-enumărabilă**[2].

[2] Folosim această terminologie din rațiuni didactice, dar, în alte lucrări din domeniu, puteți întâlni termenul de semidecidabilitate folosit pentru a descrie toate problemele a căror mulțime de adevăr este RE [2].

Strategii de abordare - probleme nedecidabile

- Acceptăm o rezolvare parțială care poate să rateze unele soluții corecte dar acoperă cazurile importante.
- Dezvoltăm soluții specifice pentru cazuri particulare ale problemei, luând în considere probabilitatea ca acestea să fie întalnite în practică.

References

[1] Martin Davis. Hilbert's tenth problem is unsolvable. *The American Mathematical Monthly*, 80(3):233–269, 1973.

[2] Bernhard Reus. *Limits of Computation*. Springer International Publishing, 2016.

[3] Cristian A. Giumale. *Introducere în Analiza Algoritmilor*. Polirom, 2004.