Calculabilitate & Complexități
Subiectul 3

REC, RE, NRE

By diana, mai nou s3333fa automat3333lor!

Ce tre să știi?

Multimi recursive, recursiv enumerabile, nerecursiv enumerabile.

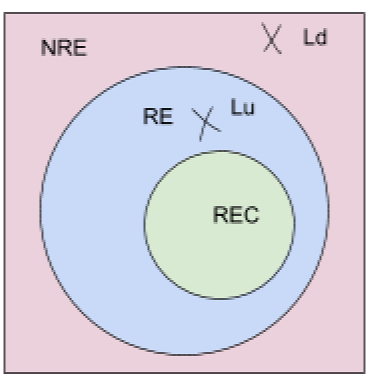
Nota 6:

- definitiile celor 3 tipuri de multimi

- relatiile intre ele (enunturi si exemple de limbaje, fara demonstratii)

Fiecare demonstratie, la alegere: 2p

Definiții



- Spunem că limbajul L este **recursiv (REC)** dacă una dintre următoarele condiții este îndeplinită:
  - Fie $f(x) = (x \in L) ? 1 : 0$. Atunci, avem trei opțiuni:
    - Funcția f este recursivă (total)

- Funcția f este calculabilă cu Programe Standard
- Există o mașină Turing M care se oprește pe fiecare intrare și acceptă L

- Spunem că limbajul L este **recursiv enumerabil (RE)** dacă există o mașină Turing care acceptă L (dar care poate să nu se oprească pe fiecare intrare)
- Spunem că limbajul L este **nerecursiv enumerabil (NRE)** dacă nu există nicio mașină Turing care acceptă L.

Atunci, putem defini cele 3 mulțimi din diagramă:

- $REC = \{L\ limbaj\ recursiv\}$
- $RE = REC \cup \{L\ limbaj\ recursiv\ enumerabil\}$
- $NRE = RE \cup \{L\ limbaj\ nerecursiv\ enumerabil\}$

Proprietăți și relații între ele

Notăm C(A) = complementul mulțimii A.

1. REC este închisă la $\cap,\ \cup,\ C$
2. RE este închisă la $\cap,\ \cup$
3. RE nu este închisă la C
4. $L,\ C(L)\ \in RE\ <=> L\ \in REC$
5. $REC\ \subset\ RE\ \subset\ NRE$

Demonstrații

REC este închisă la $\cap,\ \cup,\ C$

Fie $L1,\ L2\ \in\ REC$.

Atunci, există mașina Turing Mi cu L(Mi) = Li care se oprește pe fiecare input, unde $i\ \in \{1,\ 2\}$.

Construim mașina Turing M$\cap$:

- Input: w

- Simulează mașina M1 pe intrarea w:
    - M1 respinge => M∩ respinge
    - M1 acceptă => Simulează M2 pe intrarea w:
        - M2 acceptă => M∩ acceptă
        - M2 respinge => M∩ respinge

Fie w ∈ L(M∩) ⇔ w e acceptat și de M1, și de M2 ⇔ w ∈ L(M1)∩L(M2) ⇔ w ∈ L1∩L2.

Deci, L1∩L2 este limbaj recursiv.

Construim mașina Turing M∪:

- Input: w
- Simulează mașina M1 pe intrarea w:
    - M1 acceptă => M∪ acceptă
    - M1 respinge => Simulează M2 pe intrarea w:
        - M2 acceptă => M∪ acceptă
        - M2 respinge => M∪ respinge

Fie w ∈ L(M∪) ⇔ w e acceptat de M1 sau de M2 ⇔ w ∈ L(M1)∪L(M2) ⇔ w ∈ L1∪L2.

Deci, L1∪L2 este limbaj recursiv.

Construim mașina Turing M $c(L1)$:

- Input: w
- Simulează mașina M1 pe intrarea w:
    - M1 acceptă => M $c(L1)$ respinge
    - M1 respinge => M $c(L1)$ acceptă

Fie w ∈ L(M $c(L1)$) ⇔ w nu e acceptat de M1 ⇔ w ∉ L(M1) ⇔ w ∈ C(L1).

Deci, C(L1) este limbaj recursiv.

RE este închisă la ∩, ∪

Fie $L1, L2 \in RE$.

Atunci, există mașina Turing Mi cu L(Mi) = Li care se pot să nu se oprească pe input-urile respinse, unde $i \in \{1, 2\}$.

Construim mașina Turing M∩:

- Input: w
- Simulează mașina M1 pe intrarea w:
    - M1 se oprește și respinge => M∩ se oprește și respinge
    - M1 nu se oprește => M∩ nu se oprește
    - M1 se oprește și acceptă => Simulează M2 pe intrarea w:
        - M2 se oprește și acceptă acceptă => M∩ acceptă
        - M2 se oprește și respinge => M∩ se oprește și respinge
        - M2 nu se oprește => M∩ nu se oprește

Fie $w \in$ L(M∩) ⇔ w e acceptat și de M1, și de M2 ⇔ $w \in$ L(M1)∩L(M2) ⇔ $w \in$ L1∩L2.

Deci, L1∩L2 este limbaj recursiv enumerabil.

Construim mașina Turing M∪:

- Input: w
- Simulează alternativ mașinile M1 și M2 - efectuează un pas al mașinii M1, un pas al mașinii M2:
    - M1 sau M2 se oprește si acceptă => M∪ acceptă
    - M1 sau M2 se oprește și respinge => așteaptă rezultatul celeilalte
        - Și cealaltă mașină s-a oprit:
            - A acceptat => M∪ acceptă
            - A respins => M∪ respinge
        - Cealaltă mașină nu se oprește => M∪ nu se oprește
    - Nici M1, nici M2 nu se oprește => M∪ nu se oprește

Fie $w \in$ L(M∪) ⇔ w e acceptat de M1 sau de M2 ⇔ $w \in$ L(M1)∪L(M2) ⇔ $w \in$ L1∪L2.

$L,\ C(L)\ \in RE \iff L\ \in REC$

„=>”

L și C(L) sunt recursiv enumerabile, deci există mașinile Turing M1 și M2 cu L(M1) = L și L(M2) = C(L) care pot să nu se oprească pe intrările ce nu sunt acceptate.

Construim mașina Turing M L:

- Input: w
- Simulează alternativ mașinile M1 și M2 - efectuează un pas al mașinii M1, un pas al mașinii M2:
    - M1 se oprește:
        - Acceptă => M L acceptă
        - Respinge => M L respinge
    - M2 se oprește
        - Acceptă => M L respinge
        - Respinge => M L acceptă
    - M1 nu se oprește - o să se oprească M2 pentru că w ∈ C(L)
    - M2 nu se oprește - o să se oprească M1 pentru că w ∈ L.

Fie w ∈ L ⇔ w e acceptat de M1 sau respins de M2 ⇔ w ∈ L și $w \notin C(L)$.

Deci, există o mașină Turing care acceptă L și se oprește pe fiecare input => L limbaj recursiv.

„<=”

REC este închis la C, deci și C(L) aparține lui REC, iar REC este inclus în RE.

$REC \subset RE \subset NRE$

Pornim de la faptul că $REC \subseteq RE \subseteq NRE$ (reiese din definiții, din diagramă). Rămâne să demonstrăm că incluziunea e strictă.

Considerăm mulțimea {0, 1}* (toate șirurile de lungime finită formate din 0 și 1) și le ordonăm mai întâi după lungime și, în caz de lungime egală, lexicografic:

0, 1, 00, 01, 10, 11, 100 etc.

Notăm cu $\widehat{x}$ numărul de ordine al lui x în mulțimea ordonată. ($\widehat{11}$ = 6).

#(M) = codificarea mașinii M peste {0, 1, 2, (,), L, R} (e explicată în cursul cu mașina universală).

$\widehat{M}$ = numărul de ordine al mașinii M în enumerarea tuturor cuvintelor peste {0, 1, 2, (,), L, R}.

$NRE \setminus RE \neq \Phi$

Considerăm limbajul Ld.

Ld = {w $\in$ {0, 1}* | w $\notin$ L(M) unde $\widehat{M}$ = $\widehat{w}$} - exemplu de limbaj NRE.

Presupunem că Ld este recursiv enumerabil. Atunci, există o mașină Turing M care să îl accepte.

Alegem w cu $\widehat{M}$ **=** $\widehat{w}$.

- M acceptă w => w este acceptat de M cu $\widehat{M}$ **=** $\widehat{w}$ => w $\notin$ Ld = L(M)
- M respinge w => w nu este acceptat de M cu $\widehat{M}$ **=** $\widehat{w}$ => w $\in$ Ld = L(M)

Deci, avem contradicție.

=> Nu există o mașină M care să accepte Ld => Ld $\in NRE \setminus RE$.

$RE \setminus REC \neq \Phi$

Considerăm limbajul Lu.

Lu = {w#(M) |w $\in$ {0, 1}* și w $\in$ L(M) } - exemplu de limbaj RE.

(w concatenat cu #(M))

Presupunem că Lu este recursiv. Atunci, există o mașină Turing Mu care se oprește pe fiecare intrare cu L(Mu) = Lu.

Atunci, putem construi următoarea mașină M:

- Input: w din {0, 1}*
- Găsește $\widehat{w}$
- Găsește M cu $\widehat{M}$ **=** $\widehat{w}$
- Simulează Mu pe intrarea w#(M):
    - Mu acceptă => M respinge
    - Mu respinge => M acceptă

Din moment ce Mu se oprește pe fiecare intrare, și M se oprește pe fiecare intrare, deci L(M) este recursiv.

Dar L(M) = {w $\in$ {0, 1}* | w $\notin$ L(M) unde $\widehat{M}$ = $\widehat{w}$} = Ld, iar Ld $\in$ $NRE \setminus RE$ => contradicție.

Deci, nu putem construi mașina Mu, iar Lu $\in$ $RE \setminus REC$.