Curs 11

December 22, 2021

Ierarhii de clase de complexitate

Teorema. *Daca $L \in TIME_k(f(n))$, atunci $L \in TIME_2(f(n)\log f(n))$.*
Fara demonstratie.
Lema. *Daca L este acceptat de o masina Turing (determinista) in spatiu $S(n) \geq \log n$, atunci L este acceptat de o masina Turing care se opreste pe fiecare intrare in spatiu $S(n)$.*
Demonstratie. Fie M o masina Turing care accepta L in spatiu $S(n)$. Atunci M accepta L printr-un calcul de lungime cel mult $s(n+2)S(n)t^{S(n)}$, orice calcul mai lung avand configuratii care se repeta. In formula de mai sus, s este numarul de stari iar t este numarul de simboluri pe banda auxiliara. Observam ca $s(n+2)S(n)t^{S(n)} \leq (4st)^{S(n)}$. Construim M' care simuleaza M si care va folosi o banda pentru a numara tranzitiile in baza $4st$. Astfel M' se opreste fie cand M se opreste (si ia aceeasi decizie de acceptare) fie cand numarul de pasi depaseste $(4st)^{S(n)}$ (si respinge intrarea). Evident $space_{M'}(n) \leq S(n)$.

Ierarhii de clase de complexitate

Definitie. O functie $f(n)$ este *spatiu construibila* (sc) daca exista o masina Turing M a.i. $space_M(n) \leq f(n)$, pentru orice n si $\forall n \; \exists w_n$ a.i. $|w_n| = n$ si $space_M(w) = f(n)$.
Definitie. O functie $f(n)$ este *spatiu construibila complet* (scc) daca exista o masina Turing M a.i. pentru orice n si orice w a.i. $|w| = n$ avem $space_M(w) = f(n)$.
Definitii. Functiile *timp construibile* (tc) si *timp construibile complet* (tcc) se definesc analog.
Teorema. *Daca* $S_1(n), S_2(n) \geq \log n$, $S_2(n)$ *este scc si* $\lim \frac{S_1(n)}{S_2(n)} = 0$, *atunci* $DSPACE(S_2(n)) \setminus DSPACE(S_1(n)) \neq \emptyset$.
Dem. Idee: Construim o masina Turing M a.i. $space_M(n) \leq S_2(n)$ si limbajul sau difera prin cel putin un cuvant de orice limbaj acceptat de o masina Turing in spatiu $S_1(n)$.

Ierarhii de clase de complexitate

Continuare demonstratie. Modul de calcul al lui M:

1. Primeste la intrare $w \in \{0,1\}^*$, $|w| = n$;
2. Marcheaza pe o banda auxiliara $S_2(n)$ celule si se va opri ori de cate ori va trebui sa foloseasca un spatiu mai mare. Deci $space_M(n) \leq S_2(n)$.
3. Identifica masina Turing M_{0y} codificata cu $0y$, unde $w = x0y$ si $x \in \{1\}^*$;
4. M simuleaza M_{0y} pe intrarea w. Putem presupune ca M se opreste intotdeauna (vezi lema anterioara).
5. M respinge w ddaca simularea s-a incheiat cu acceptarea lui w de M_{0y}.

Fie M' o masina Turing cu spatiul marginit de $S_1(n)$ si codificarea $z \in 0\{0,1\}^*$. Putem presupune ca M' are t simboluri de banda si o singura banda auxiliara. Exista un cuvant $w' = 1^m z$, cu m suficient de mare a.i. $\lceil t \rceil S_1(|w'|) \leq S_2(|w'|)$. Rezulta ca $w' \in L(M)$ ddaca $w' \notin L(M')$. QED

Ierarhii de clase de complexitate

TEOREMA. *Daca* $T_2(n)$ *este tcc si* $\lim \frac{T_1(n)\log T_1(n)}{T_2(n)} = 0$, *atunci* $DTIME(T_2(n)) \setminus DTIME(T_1(n)) \neq \emptyset$.

DEM. Analog demonstratiei anerioare cu urmatoarele observatii:

- Deoarece $T_2(n)$ este tcc, exista o mT M_0 care executa exact $T_2(n)$ pasi pe orice intrare de lungime n. M_0 se poate folosi ca sa marcam $T_2(n)$ celule pe o banda sau poate fi rulata alternativ cu masina M construita.
- Masina M_{0y} se poate reduce la o masina cu doua crescand complexitatea sa timp $T(n)$ cu un factor $\log T(n$.

Relatii intre clase de complexitate

Teorema.

1. $(N)(D)TIME(f(n)) \subseteq (N)(D)SPACE(f(n))$.
2. Daca $L \in DSPACE(f(n))$ cu $f(n) \geq \log n$, atunci exista c_L a.i. $L \in DTIME(c_L^{f(n)})$.
3. Daca $L \in NTIME(f(n))$, atunci exista c_L a.i. $L \in DTIME(c_L^{f(n)})$.

Dem.

1. Trivial.
2. Fie M o masina Turing determinista care accepta L in spatiu $f(n)$. Atunci M accepta L printr-un calcul de lungime cel mult $s(n+2)f(n)t^{f(n)}$, orice calcul mai lung avand configuratii care se repeta. In formula de mai sus, s este numarul de stari iar t este numarul de simboluri pe banda auxiliara. Observam ca $s(n+2)f(n)t^{f(n)} \leq (4st)^{f(n)}$. Construim M' care simuleaza M si care va folosi o banda pentru a numara tranzitiile. Accepta ddaca M accepta si M' nu a efectuat mai mult de $(4st)^{f(n)}$ pasi.

Relatii intre clase de complexitate

Continuarea demonstratiei.

3 Fie M o masina Turing nedeterminista cu s stari, k benzi si t simboluri, care accepta L in timp $f(n)$. Numarul maxim de configuratii al lui M este $s(f(n)+1)^k t^{f(n)} \leq d^{f(n)}$, unde $d = s(t+1)^{3k}$. O mT determinista M' accepta limbajul lui M prin constructia unei liste cu toate configuratiile accesibile din configuratia initiala pe fiecare intrare. Aceasta lista are lungimea cel mult $d^{f(n)}$ si necesita un timp $(d^{f(n)})^2$. Fiecare configuratie are lungimea $1 + k(f(n)+1)$ deci timpul total necesar lui M' este cel mult $(d^{f(n)})^2(1 + k(f(n)+1))$ care se poate margini cu $c^{f(n)}$, pentru o constanta c convenabil aleasa.

Teorema lui Savitch

Teorema. Daca $S(n) \geq \log n$ si $S(n)$ este scc, atunci $NSPACE(S(n)) \subseteq DSPACE(S^2(n))$.

Dem. Fie M o mT nedeterminista cu spatiul marginit de $S(n)$. Pentru un input de lungime n, M accepta acest input printr-o secventa de cel mult $c^{S(n)}$ pasi.

Notam $C_1 \vdash_i C_2$ procesul prin care configuratia C_2 este atinsa din C_1 printr-o secventa de cel mult 2^i pasi. Urmatorul pseudocod decide daca w, cu $|w| = n$, este acceptat de M.

$m \leftarrow \lceil \log c \rceil$
C_0 este configuratia initiala pe intrarea w
for fiecare configuratie finala C_f de lungime cel mult $S(n)$ **do**
  if $TEST(C_0, C_f, mS(n))$ **then** accept
  end if
end for

Teorema lui Savitch

Continuare demonstratie.

Function $TEST(C_1, C_2, i)$;
if $(i = 0)$ si $((C_1 = C_2)$ sau $(C_1 \vdash C_2))$ **then** return true
end if
if $i \geq 1$ **then**
    for fiecare configuratie C' de lungime cel mult $S(n)$ **do**
        if $TEST(C_1, C', i-1)$ si $TEST(C', C_2, i-1)$ **then** return true
        end if
    end for
end if
return false

Teorema lui Savitch

Continuare demonstratie.

Acest algoritm poate fi implementat pe o masina determinista M' astfel:

- Foloseste o banda sau 3 benzi) ca o stiva pentru a salva argumentele functiei $TEST$, (C_1, C_2, i).
- Fiecare configuratie este de lungime cel mult $S(n)$.
- Parametrul i se poate memora pe cel mult $mS(n)$ celule. (In binar chiar mai putin).
- Pozitia pe banda de intrare se salveaza in binar pe cel mult $\log n$ celule.
- Numarul de apelari este de cel mult $mS(n)$ (la fiecare apelare indicele i scade.)
- Spatiul pentru testarea $C_1 \vdash_i C'$ poate fi refolosit pentru testarea $C' \vdash_i C_2$.

Rezulta ca spatiul total al lui M' este $\mathcal{O}(S^2(n))$ si aplicand teorema de eliminare a constantelor demosntratia este completa.