

Digitális számításelmélet

2019 / 1. vizsgaZH megoldás

Ekart Csaba

2019. június 4.

1. Adja meg a DVA és az NVA definícióját! Adjon meg állapotdiagrammal egy olyan DVA-t, amely azokat az $\{a, b, c\}$ feletti szavakat fogadja el, amelyek hossza páratlan és pontosan egy a -t tartalmaznak!

Solution:

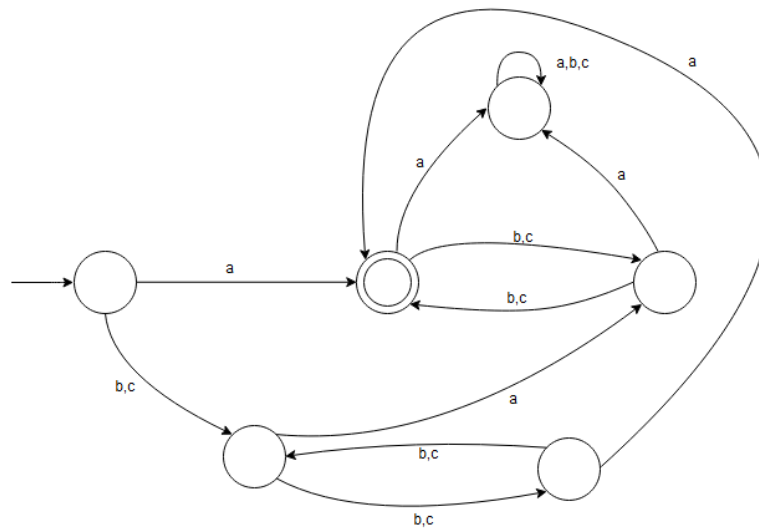
Determinisztikus véges automata (DVA) az $M = (Q, \Sigma, \delta, q_0, F)$, ahol

1. Q : az állapotok véges halmaza
2. Σ : az abc véges halmaza
3. $\delta: Q \times \Sigma \longrightarrow Q$
4. $q_0 \in Q$: a kezdeti állapot
5. $F \subseteq Q$: a végállapotok halmaza

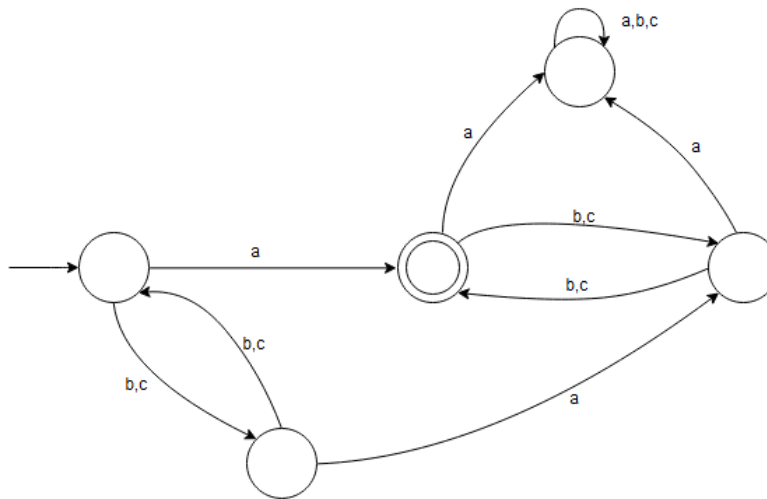
Nem determinisztikus véges automata (NVA) az $N = (Q, \Sigma, \delta, q_0, F)$, ahol

1. Q : az állapotok véges halmaza
2. Σ : az abc véges halmaza
3. $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \longrightarrow P(Q)$, ahol $P(Q)$ Q hatványhalmaza, ε az üres szó.
4. $q_0 \in Q$: a kezdeti állapot
5. $F \subseteq Q$: a végállapotok halmaza

A feladathoz két megoldás is itt van, az első tuti, hogy jó, a másodikról nem tudom, mert én csináltam, de szerintem jó.



1. ábra. Ez itt a megbízható megoldás



2. ábra. Ez itt az enyém

2. Mondja ki és bizonyítsa be a reguláris nyelvcsaládba nem tartozást tanúsító pumpálási lemmát! Reguláris-e az $L = \{ww \mid w \in \{a, b\}^*\}$ nyelv?

Solution:

Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor $\exists p \in \mathbb{N}$ szám, melyre $\forall s \in L$ esetén, ha s hossza legalább p , akkor s felbontható $s = xyz$ alakba úgy, hogy:

1. $xy^n z \in L \quad \forall n \geq 0$
2. $|y| > 0$

3. $|xy| \leq p$

Bizonyítás. • Legyen $p = |Q|$

- Legyen $s = s_1 s_2 s_3 \dots s_n \in L$ és az s -t elfogadó állapotok sorozata $r = r_1, r_2, \dots, r_{n+1}$, ekkor $p < n + 1$. Tehát r tagjai között kell lennie legalább két azonosnak, a skatulya elv miatt. Jelölje ezeket r_j és r_k . Feltehetjük, hogy $k \leq p + 1$.

- $x = s_1 s_2 \dots s_{j-1}$, $y = s_j s_{j+1} \dots s_{k-1}$, $z = s_k s_{k+1} \dots s_n$

Láthatjuk a fentiek alapján, hogy

1. $xy^n z \in L \quad \forall n \geq 0$ valóban
2. $|y| > 0$, mert $j \neq k$
3. $|xy| = k - 1 \leq p$, mert $k \leq p + 1$

□

Feladat megoldása:

Ütő Bence vizsga alatt készült rögtönzött bizonyítása:

Bizonyítás. Legyen w C hosszú, melyet felosztunk xyz -re úgy hogy, x az első felébe esik (tehát az első w -be). Ekkor, ha y -t elkezdjük pumpálni, annak meg kéne jelennie a szó végén is (mert ugye két w van egymás mögött), ami nem történik meg, tehát a nyelv nem reguláris. □

Interneten talált bizonyítás:

Bizonyítás. Legyen $ww = (a^p)(b^p)(a^p)(b^p)$. Ekkor a pumpáló lemma alapján $|xy| \leq p$ és $|y| \geq 1$, tehát a $w = xyz$ felosztásban az y rész a -kra esik. De ha elkezdjük pumpálni több a lesz az első felében a szónak, mint a másodikban, tehát a kettő nem fog megegyezni. Tehát a nyelv nem reguláris. □

3. Mikor mondjuk azt, hogy az L nyelv rekurzív? Mikor mondjuk azt, hogy az L nyelv rekurzíve felsorolható? Bizonyítsa be, hogy algoritmikusan nem eldönthető, hogy az M_1 és M_2 Turing gépek ugyanazt a nyelvet fogadják el! (Használhatja azt a tényt, hogy algoritmikusan nem eldönthető, hogy egy M Turing gép az üres nyelvet fogadja-e el.)

Solution: Egy $L \subseteq \Sigma^*$ nyelv Turing eldönthető, vagy rekurzív, ha van olyan M Turing gép, melyre $L(M) = L$ és $M \forall s \in \Sigma^*$ szóra megáll.

Egy $L \subseteq \Sigma^*$ nyelv Turing felismerhető, vagy rekurzívan felsorolható, ha van olyan M Turing gép, melyre $L(M) = L$.

Bizonyítás. A bizonyítás módja a következő: A hivatalos jegyzet bizonyítása, hogy algoritmikusan nem eldönthető, hogy két Turing gép ugyanazt a nyelvet fogadja el:

1. Csinálok egy Turing gépet, amely az üres nyelvet fogadja el: T_1
2. Beadok egy Turing gépet (T_3) T_2 -nek.
3. Ha T_2 el tudja dönteni, hogy $T_1 == T_3$ -mal, akkor T_2 el tudja dönteni az üres nyelvet is. Ez ellentmondás, mert tudjuk, hogy algoritmikusan nem eldönthető, hogy egy Turing gép elfogadja-e az üres nyelvet.

A feladatra konkretizált (megint csak felelősséget nem vállalok féle) bizonyítás:

1. A bizonyítás indirekt. Tegyük fel, hogy eldönthető, hogy M_1 és M_2 ugyanazt a nyelvet fogadja el.
2. Ha a probléma algoritmikusan eldönthető, létezik egy M_3 Turing gép, mely képes eldönteni a problémát.
3. Tegyük fel, hogy M_1 az üres nyelvet elfogadja el.
4. Amennyiben M_3 képes eldönteni, hogy M_1 és M_2 ugyanazt a nyelvet fogadja el, képes eldönteni az üres nyelvet is, erről viszont tudjuk, hogy nem lehetséges.
5. Ellentmondásra jutottunk, tehát az eredeti állítás helyes.

□

4. Adja meg a P és az NP osztályok definícióját! Mikor mondjuk azt, hogy az L_1 nyelv polinomiálisan visszavezethető L_2 -re? Mikor mondjuk azt, hogy egy nyelv NP teljes? Rekurzívak-e az NP-teljes nyelvek? Tegyük fel, hogy tudjuk, hogy NP teljes probléma annak az eldönthetősége, hogy egy gráf 3 színnel kiszínezhető-e. Lásza be, hogy NP-teljes probléma annak az eldönthetősége, hogy egy gráf 10 színnel kiszínezhető-e?

Solution:

Mese (nem kell leírni, csak kiegészítés): A P osztály a polinom időben megoldható problémákból áll, azaz olyan problémákból, melyekhez $\exists k$ konstans, melyre a probléma n hosszú bemenet esetén $\mathcal{O}(n^k)$ idő alatt megoldható.

Az NP osztály olyan problémákból áll, amelyek polinom időben ellenőrizhetők. Ez valami olyasmit jelent, hogy ha valaki egy "bizonyítékot" adna nekünk a megoldásról, akkor annak helyességét polinomiális időben letudnánk ellenőrizni. Pl ilyen a Hamilton-kör probléma.

Az NP teljes problémák azon problémák halmaza, melyek legalább olyan nehezek, mint bármely NP osztálybeli probléma.

Innentől a válasz:

P osztály: $P = \cup_{k \geq 1} \text{TIME}(n^k)$ a polinom időben egyszalagos Turing géppel eldönthető nyelvek osztálya.

NP osztály: $P = \cup_{k \geq 1} \text{NTIME}(n^k)$ a polinom időben egyszalagos Nem determinisztikus Turing géppel eldönthető nyelvek osztálya.

L_1 polinomiálisan visszavezethető L_2 -re:

L_1 és L_2 Σ^* feletti nyelvek. Azt mondjuk L_1 polinomiálisan visszavezethető L_2 -re, ha $\exists f : \Sigma^* \rightarrow \Sigma^*$ polinomiális időben kiszámítható függvény, melyre $w \in L_1 \leftrightarrow f(w) \in L_2$. Jelölés $L_1 \leq_p L_2$

NP teljes nyelv

Egy L NP-beli nyelv NP-teljes, ha abból a feltételezésből, hogy L P-beli is, következik, hogy az összes NP-beli nyelv P-beli, azaz $\text{NP}=\text{P}$.

Rekurzívak-e az NP-teljes nyelvek

Igen, az NP-teljes nyelvek rekurzívak.

A feladat megoldására, egy internetről halászott és egy a Csutakos megoldásból:

Bizonyítás. El akarom dönteni, hogy egy gráf kiszínezhető-e 3 színnel. Felveszek 7 új csúcsot, összekötöm őket egymással (K_7), majd minden új csúcsot összekötök minden eredeti csúccsal. Mivel a K_7 kiszínezéséhez 7 szín kell és a kapcsolatok miatt ezek egyike sem használható a régi csúcsokhoz, az új gráf pontosan akkor színezhető 10 színnel, ha az eredeti 3 színnel színezhető volt. Ha a 10-színűt el tudom dönteni P-ben, akkor a fent leírt módon - a 3 színt is el tudom. Ellentmondás, készen vagyunk. □

Bizonyítás. A 10 színnel kiszínezhető gráfok problémája visszavezethető a 3 színnel kiszínezhető gráfok problémájára. Képezzük úgy a G' gráfot, hogy annak egyik fele a K_7 (Ha valaki nem emlékezne DM-ből vettük: 7 csúcsú teljes gráf), a másik fele pedig legyen G gráf, melyről tudjuk, hogy a 3 színnel kiszínezhetősége NP teljes probléma. Kössük össze a két rész gráf összes csúcsát a másik összes csúcsával. A K_7 rész nyilván 7 színnel színezhető ki, és a 7 szín egyikét sem használhatjuk G -ben, tehát a 10-ből 3 színünk maradt, melyet G kiszínezéséhez használhatunk. Mivel ez egy NP teljes probléma, ebből látszik, hogy a 10 színes kiszínezés is NP teljes. $\square$

Ezt innen:

<https://math.stackexchange.com/questions/1752064/reducing-graph-3-coloring-to-10-coloring/1752130>