

Keerukusteooria alused

Ahto Buldas

Arvutatavus

Funktsiooni $A \xrightarrow{f} B$ **arvutatavus**: hulkade A ja B elemendid on sobivalt kodeeritud ja leidub arvutiprogramm (lõplik käskude jada), mis iga elemendi $a \in A$ koodist $\text{Code}(a)$ arvutab lõpliku aja jooksul välja elemendi $f(a) = b \in B$ koodi $\text{Code}(b)$.

Koodideks võetakse kõigi lõplike 0, 1-jadade hulk $\{0, 1\}^*$.

Kõigi k -elemendiliste 0, 1-jadade hulka tähistame $\{0, 1\}^k$. Arvestame ka 0 pikkusega jada, mida tähistame tavaliselt ϵ . Seega

$$\{0, 1\}^* = \bigcup_{k \in \mathbb{N}} \{0, 1\}^k,$$

kus $\mathbb{N} = \{0, 1, 2, \dots\}$ on kõigi naturaalarvude hulk.

Mittearvutatavad funktsioonid

Kaugeltki mitte kõik funktsioonid $\mathbb{N} \xrightarrow{f} \mathbb{N}$ ei ole arvutatavad.

See tuleneb juba ainuüksi faktist, et kõigi selliste funktsioonide hulk $\mathbb{N}^{\mathbb{N}}$ on mitteloenduv, samal ajal kui lõplikke programme on ainult loenduv hulk.

Cantori diagonaalne tõestus hulgateooriast näitab, et kõigi funktsioonide $\mathbb{N} \rightarrow \{0, 1\}$ hulka ei ole võimalik nummerdada naturaalarvudega.

On konkreetseid (ja vajalikke) funktsioone, mis ei ole arvutatavad. Näiteks programmi peatumise probleemi lahendav funktsioon.

Turingi masin

Arvuti enimkasutatav matemaatiline mudel on nn *Turingi masin*—teatud liiki lõplik automaat M koos lõpmatu järjestikmäliga (nn. *lint*), millele ligipääs on võimalik “kursori” (või ka “pea”) kaudu.

- Lint — jada $L = (\ell_0, \ell_1, \ell_2, \dots)$, mille iga element $\ell_i \in \{0, 1, \epsilon\}$. Igal arvutussammul võib muuta ainult seda pesa, millel on kursor, st pesa ℓ_k .
- Kursor k on naturaalarv, mis näitab, millise pesaga masin parasjagu tegeleb. Igal arvutussammul saab kursorit nihutada paremale (st $k := k + 1$), vasakule $k := k - 1$ või jätta paigale (k jääb muutumatuks). Arvutuse esimesel sammul $k = 0$.
- Igal sammul on masin mingis olekus $s \in S$, kus S on mingi lõplik hulk. Algolekut tähistame s_0 ja lõppolekut h .

Turingi masina programm

Järgmise sammu olek s' , lindi seis l'_k ja kursori asend k' arvutatakse funktsioonidega

$$s' := \delta_s(s, l_k) \in S$$

$$l'_k := \delta_l(s, l_k) \in \{0, 1, \epsilon\}$$

$$k' := k + \delta_k(s, l_k) \in \{k - 1, k, k + 1\}, \text{ st. } \delta_k(s, l_k) \in \{-1, 0, +1\}.$$

Lindi algseisu loetakse masina sisendiks ja lõppseisu väljundiks. Näiteks funktsiooni $\mathbb{N} \xrightarrow{f} \mathbb{N}$ arvutatavus tähendab seda, et leidub Turingi masin M , mis teisendab lindile L salvestatud arvu $x \in \mathbb{N}$ koodi arvu $y = f(x)$ koodiks, mis on lindil sel hetkel, kui masin jõuab olekusse h .

Nullfunktsiooni arvutatavus

Näiteks nullfunktsioon $f(x) = 0, \forall x \in \mathbb{N}$ on arvutatav, sest leidub teda arvutav kahe-olekuline Turingi masin:

s	ℓ_k	s'	ℓ'_k	$(k' - k)$
s_0	0	s_1	0	+1
	1	s_1	0	+1
	ϵ	h	0	0
s_1	0	s_1	ϵ	+1
	1	s_1	ϵ	+1
	ϵ	h	ϵ	0

Siin on eeldatud, et lindil L on esialgu arvu x kood, mis lõpeb tühja pesaga. Lindi lõpupoole võib olla veel mittetühje pesasid, kuid need ei tule kodeerimise/dekodeerimise juures arvesse.

Kaks harjutust

Harjutus 1: Leida Turingi masin, mis arvutab funktsiooni $y = 2x + 1$, eeldades, et arv $x = b_0 2^0 + b_1 2^1 + \dots + b_n 2^n$ (kus $b_i \in \{0, 1\}$) kodeeritakse lindi seisuga $L = (b_n, b_{n-1}, \dots, b_1, b_0, \epsilon, \epsilon, \dots)$.

Harjutus 2: Sama, mis eelmises ülesandes, kasutades kodeeringut vastupidises bittide järjestuses, et $L = (b_0, b_1, \dots, b_{n-1}, b_n, \epsilon, \epsilon, \dots)$.

Turingi tees

Ehkki Turingi masin võib näida ülilihtsa arvutusseadmena, usutakse, et tema abil saab arvutada absoluutselt kõike, mis on kuidagi arvutatav.

Sellist uskumust nimetatakse *Turingi teesiks*.

Et see tees ise ei ole matemaatiline lause (“kuidagi arvutatav” ei ole defineeritud), siis ei saa ka seda teesi matemaatiliselt tõestada.

Edaspidi me lihtsalt usume seda teesi ja enamikul juhtudest ei süvene Turingi masinate “siseellu”.

Pseudokood

Programmide kirjeldamiseks kasutame programmeerimiskeelt meenutavat pseudokoodi. Näiteks võiks nullfunktsiooni arvutavat Turingi masinat esitada järgmise pseudokoodina:

$k := 0$

s0: **IF** $L[k] = \epsilon$ **THEN** $L[k] := 0$, **HALT**

ELSE $k := k + 1$, **GOTO** s1

s1: **IF** $L[k] = \epsilon$ **THEN** **HALT**

ELSE $L[k] := \epsilon$, $k := k + 1$, **GOTO** s1

Keeled ja ülesanded

Olgu $L \subseteq \{0, 1\}^*$ mingi keel, st suvaline 0, 1-jadade hulk. Ütleme, et Turingi masin M *tuvastab keele* L , kui iga $x \in \{0, 1\}^*$ korral

$$M(x) = 1 \Leftrightarrow x \in L,$$

kus tähise $M(x)$ all mõeldakse Turingi masina M väljundit, eeldades, et sisend on x . Eeldame, sisendi ja väljundi lõpuks loetakse esimest (minimaalse indeksiga) tühja pesa. Esimesele tühikule järgnev lindi sisu ei ole oluline.

Kombinatorikaülesannete lahendamist saab enamasti formuleerida keele tuvastamise ülesandena. Selleks tuleb eelnevalt kokku leppida, kuidas kodeeritakse kombinatorikaülesanne keele sõneks.

Arvutusaeg ja keerukus

Olgu M Turingi masin ja x mingi sisend. Turingi masina M arvutusajaks $T(M(x))$ kohal x nimetatakse masina M poolt sooritatud arvutussammude arvu, kuni masina peatumiseni (jõudmiseni olekusse h).

Turingi masina M asümptootiliseks (ajaliseks) keerukuseks nimetatakse funktsiooni $\mathbb{N} \xrightarrow{T} \mathbb{N}$, nii et iga $n \in \mathbb{N}$ korral

$$T_M(n) = \max\{T(M(x)) : x \in \{0, 1\}^n\}.$$

Sel viisil defineeritud keerukust nimetatakse ka *halvima juhu* keerukuseks (*worst case complexity*), sest iga n korral läheb siin arvesse raskeim juht. Alternatiivne lähenemine on nn. *keskmise keerukus* (*average case complexity*).

O -tähistused ja klass P

Olgu $f(n)$ ja $g(n)$ mingid funktsioonid tüüpi $\mathbb{N} \rightarrow \mathbb{N}$. Võtame kasutusele järgmised tähistused:

$$f(n) = O(g(n)) \equiv \exists c, n_0 \in \mathbb{N}: \forall n \geq n_0: f(n) \leq c \cdot g(n)$$

$$f(n) = \Omega(g(n)) \equiv g(n) = O(f(n))$$

$$f(n) = \Theta(g(n)) \equiv f(n) = O(g(n)) = \Omega(g(n))$$

$$f(n) = \omega(g(n)) \equiv \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0 .$$

Klass P : Keel $L \subseteq \{0, 1\}^*$ loetakse kuuluvaks klassi P , kui leidub keelt L tuvastav Turingi masin M asümptootilise keerukusega $T_M(n) = n^{O(1)}$.

Mittedeterministlik Turingi masin

Matemaatiline mudel, mis esitab teatud ebarealistlikult võimasat arvutit, millel on piiramatu võime “kahestuda” ja jätkata arvutusi paralleelselt kahes erinevas harus (mis omakorda võivad ise hiljem “kahestuda”).

Järgmist olekut arvutav funktsioon δ annab järgmise oleku asemel vastuseks *olekute paari*, mis langevad kokku, kui hargnemist ei toimu.

Turingi masin siirdub mõlemasse uude olekusse samaaegselt ja jätkab sõltumatult tööd mõlemas olekus, st kui valida hargnemisel alati ainult üks haru, siis muutub masina olek ja lindi seis täpselt nii nagu teisi harusid ei olekski olemas.

Mittedeterministlik Turingi masin *tuvastab keelt* L , kui $x \in L$ parajasti siis, kui kõik harud lõpetavad töö ja leidub haru, milles väljastatakse 1.

Nullkoha leidmine mittedeterministliku masinaga

Mittedeterministlik programm funktsiooni $\{0, 1\}^* \xrightarrow{f} \{0, 1\}$ nullkoha $x \in \{0, 1\}^n$ olemasolu kindlakstegemiseks:

$x := \epsilon$

L2: **IF** $n > 0$ **GOTO** L0, L1

ELSE GOTO L3

L0: $x := x||'0'$ $n := n - 1$

GOTO L2

L1: $x := x||'1'$ $n := n - 1$

GOTO L2

L3: **IF** $f(x) = 0$ **RETURN** 1

HALT

Mittedeterministliku masina tööaeg

Mittedeterministliku Turingi masina N tööajaks $T(N(x))$ kohal x nimetatakse maksimaalse arvutussammudega haru pikkust, eeldusel et sisendiks on x . Turingi masina N asümptootiliseks ajaliseks keerukuseks (või ka lihtsalt *tööajaks*) $T_N(n)$ nimetatakse funktsiooni $\mathbb{N} \xrightarrow{T} \mathbb{N}$, nii et iga $n \in \mathbb{N}$ korral

$$T_N(n) = \max\{T(N(x)) : x \in \{0, 1\}^n\}.$$

Klass NP

Klass NP: Keel L loetakse kuuluvaks klassi NP, kui leidub keelt L tuvas-
tav mittedeterministlik Turingi masin N tööajaga $T_N(n) = n^{\mathcal{O}(1)}$.

Klassi NP saab defineerida ka mittedeterministliku Turingi masina mõistet kasutamata.

Paneme tähele, et kui masin N teeb pikimas harus ℓ sammu, siis saab iga haru kodeerida bitimassiiviga $a \in \{0, 1\}^\ell$.

Kui iga n korral $T_N(n) \leq p(n)$, siis mis tahes sisendi x korral saab iga haru esitada bitimassiivina $a \in \{0, 1\}^{p(|x|)}$.

Jada a mingi bitt ütleb kumb haru kahest võimalikust tuleb valida.

Mittedeterministliku masina haru simuleerimine

Teoreem (Haru simuleerimine): Igale polünoomiaalse tööajaga mittedeterministlikule Turingi masinale N leidub (tavaline) Turingi masin $M(x, a)$ tööajaga $|x|^{O(1)}$, nii et $M(x, a) = 1$ parajasti siis kui $N(x)$ haru koodiga a aktsepteerib x .

Tõestus: Masin M konstrueeritakse masinast N järgmiselt. Iga hargnemiskäsu (järjenumbriga i) asemel valitakse vaid üks haru — see, millele osutab bitt $a[i]$. Et masina N tööaeg (pikima haru pikkus) on polünoomiaalne, siis ka masina M tööaeg on polünoomiaalne.

Mittedeterministliku Masina Dekompositsioon

”Aadressi” a fikseerimine muudab masina tavaliseks Turingi masinaks M .

Seda võib ette kujutada ka nii, et iga mittedeterministliku Turingi masina programm on kaheosaline:

- Esimeses (mittedeterministlikus) osas genereeritakse kõikvõimalikud a väärtused. Seda saab teha analoogilise programmilõigu abil, mida kasutati funktsiooni nullkoha otsimisel.
- Teises (determineeritud) osas arvutatakse tavalise Turingi masina M abil välja $y = M(a, x)$.

Klassi NP alternatiivne definitsioon: I

Klass NP: $L \in \text{NP}$ parajasti siis kui leidub (tavaline) Turingi masin $M(x, a)$ tööajaga $|x|^{O(1)}$, nii et iga x korral:

$$x \in L \Leftrightarrow \exists a: M(x, a) = 1 .$$

Ekvivalentsus esialgse definitsiooniga: Kui $L \in \text{NP}$, siis leidub polünomi-aalses ajas töötav mittedeterministlik Turingi masin N , nii et $x \in L$ parajasti siis, kui $1 \leftarrow N(x)$. Järelikult tänu haru simuleerimise teoreemile leidub Turingi masin $M(x, a)$ tööajaga $|x|^{O(1)}$, nii et $M(x, a)$ väljund langeb kokku $N(x)$ selle haru väljundiga, mille kood on a .

Kui aga leidub M (uue definitsiooni järgi), siis koostame mittedeterministliku masina N nii, et N esmalt hargneb seni, kuni harudes on olemas kõikvõimalikud a väärtused ja seejärel arvutab $M(x, a)$ kasutades M koodi (tabelit).

Klassi NP alternatiivne definitsioon: II

Klass NP: $L \in \text{NP}$ parajasti siis kui leidub (tavaline) Turingi masin $V(x, a)$ (verifitseerija) tööajaga $|x|^{O(1)}$ ja piiramatu tööajaga Turingi masin P (tõestaja), nii et nii et iga x korral:

$$x \in L \Leftrightarrow V(x, P(x)) = 1 .$$

kus bitijada $a = P(x)$ nimetatakse **sertifikaadiks**.

Põhjendus: Kui $L \in \text{NP}$ vanas mõttes, siis leidub polünomiaalne M , nii et $x \in L$ parajasti siis kui $M(x, a) = 1$ mingi a korral. Võtame $V \equiv M$ ja P olgu algoritm, mis vaatab läbi kõikvõimalikud a -d ja väljastab "õige" a (kui see on olemas!).

Kui aga $L \in \text{NP}$ uues mõttes, siis piisab tähelepanekust, et P -d saab simuleerida mittedeterministliku Turingi masinaga.

P versus NP

Tänapäevani ei teata, kas klassid P ja NP on võrdsed või mitte.

Et paremini mõista lauset $P = NP$, esitame selle lause kaks versiooni, mis tegelikult on ekvivalentsed:

- **Otsustusversioon:** Igale Turingi masinale $F(x, a)$ tööajaga $|x|^{O(1)}$ leidub Turingi masin $D(x)$ tööajaga $|x|^{O(1)}$, nii et iga x korral

$$D(x) = 1 \Leftrightarrow \exists a: F(x, a) = 1 .$$

- **Otsinguversioon:** Igale Turingi masinale $F(x, a)$ tööajaga $|x|^{O(1)}$ leidub Turingi masin $S(x)$ tööajaga $|x|^{O(1)}$, nii et iga x korral

$$F(x, S(x)) = 1 \Leftrightarrow \exists a: F(x, a) = 1 .$$

Kaks ülesannet

Ülesanne 1: Tõesta, et $P = NP$ otsustus- ja otsinguversioonid on samaväärsed laused, st üks järeldeb teisest.

Ülesanne 2: Tõesta, et kui $P = NP$, kui F on polünoomiaalne Turingi masin, siis leidub polünoomiaalne Turingi masin G , nii et iga x korral

$$F(x) = F(G(F(x))) .$$

See tulemus tähendab sisuliselt seda, et kui $P = NP$, siis on iga efektiivselt arvutatav funktsioon ka kergesti pööratav ja tänapäeva krüptograafia jaoks nii olulisi ühesuunalisi funktsioone ei oleks olemas.

Stohhastiline Turingi masin

Stohhastiline Turingi masin on abstraktne arvutusseade, mis on lähedane mittedeterministlikule Turingi masinale, kuid kus hargnemise tähendus on erinev — selle asemel et paralleelselt täita mõlemat haru, valib masin juhuslikult ühe harudest tõenäosusega $\frac{1}{2}$.

Motivatsioon: Pea igas reaalses programmeerimiskeeles on võimalik kasutada juhuarve ja tundub, et vahel teeb juhuarvude kasutamine programmid efektiivsemaks.

Stohhastilise masina tööaeg

Stohhastilise masina tööajaks nimetatakse vastava mittedeterministliku masina tööaega $t(n)$ (kus mündivisked on hargnemisega asendatud).

Kui $t(n) = n^{\mathcal{O}(1)}$, siis masinat N nimetatakse *polünomiaalseks stohhastiliseks Turingi masinaks*.

Stohhastiline Turingi masin N tööajaga $t(n)$ võib ka defineerida kui "arvutusseadet", mis saades sisendiks x , valib juhuslikult $a \leftarrow \{0, 1\}^{t(|x|)}$ ja arvutab väljundi $N(x) = M(a, x)$.

Seega, $y \leftarrow N(x)$ ei ole funktsioon, vaid juhuslik suurus jaotusega

$$P(y|x) = \Pr[y \leftarrow N(x)] = \Pr_a[M(a, x) = y] ,$$

kus tõenäosus arvutatakse üle kõikvõimalike $a \in \{0, 1\}^{t(n)}$.

Klass RP

Def: Keel $L \subseteq \{0, 1\}^*$ kuulub klassi **RP** kui leidub polünoomiaalne stohhastiline Turingi masin N , nii et iga $x \in \{0, 1\}^*$ korral:

- $x \in L \Rightarrow \Pr[1 \leftarrow N(x)] > \frac{1}{2}$
- $x \notin L \Rightarrow \Pr[1 \leftarrow N(x)] = 0$.

Sellise omadusega N nimetatakse ka **Monte Carlo** algoritmiks.

Monte Carlo algoritmi iseärasus seisneb järgnevas:

- Kui $1 \leftarrow N(x)$, siis on kindlalt teada, et $x \in L$.
- Kui $0 \leftarrow N(x)$, siis on usaldusväärsusega $\epsilon > \frac{1}{2}$ teada, et $x \notin L$.

Kasutades algoritmi k korda (iga kord sõltumatute juhuarvudega) on vea tõenäosus $(1 - \epsilon)^k$.

Klassid coRP ja ZPP

Def: Keele $L \subseteq \{0, 1\}^*$ kuulub klassi coRP kui leidub polünoomiaalne stohhastiline Turingi masin N , nii et iga $x \in \{0, 1\}^*$ korral:

- $x \in L \Rightarrow \Pr[1 \leftarrow N(x)] = 1$;
- $x \notin L \Rightarrow \Pr[1 \leftarrow N(x)] < \frac{1}{2}$.

Def: Keele $L \subseteq \{0, 1\}^*$ täiendiks \bar{L} nimetatakse kõigi selliste lõplike $0, 1$ -jadage hulka, mis ei ole L elemendid.

$L \in \text{coRP}$ parajasti siis kui $\bar{L} \in \text{RP}$.

Def: $\text{ZPP} = \text{RP} \cap \text{coRP}$. Kui leiduvad Monte Carlo masinad N_1 keele L jaoks ja N_0 keele \bar{L} jaoks, siis itereerides neid (näiteks vaheldumisi) k korda, saame algoritmi, mis tõenäosusega $1 - 2^{-k}$ annab kindla vastuse ühele lauseist $x \in L$ ja $x \notin L$. Sellist algoritmi nimetatakse *Las Vegase algoritmiks*.

Klass PP

Def: Keel $L \subseteq \{0, 1\}^*$ kuulub klassi **PP** kui leidub polünoomiaalne stohhastiline Turingi masin N , nii et:

$$x \in L \Leftrightarrow \Pr[1 \leftarrow N(x)] > \frac{1}{2}.$$

Kui klassid **RP**, **coRP** ja **ZPP** esitasid ka praktikas hästi töötavaid stohhastilisi algoritme, siis klassi **PP** definitsioon ei anna praktilist algoritmi. Aktsepteerimise ja mitteaktsepteerimise tõenäosuste vahe võib olla liiga väike. Statistika ei tööta!

NB! $L \in \mathbf{NP}$ parajasti siis, kui leidub polünoomiaalne stohhastiline Turingi masin N , nii et iga $x \in \{0, 1\}^n$ korral $x \in L \Leftrightarrow \Pr[1 \leftarrow N(x)] > 0$.

Ülesanne: Tõesta, et $\mathbf{NP} \subseteq \mathbf{PP}$.

Klass BPP

Klasside **RP** ja **coRP** definitsioonis esinev stohhastiline algoritm N on nn. *ühepoolse veaga*:

- **RP** definitsioonis töötab $N(x)$ alati korrektselt kui $x \notin L$;
- **coRP** definitsioonis töötab $N(x)$ korrektselt kui $x \in L$.

Järgnevalt kirjeldatav keerukusklass **BPP** on aga defineeritud stohhastilise Turingi masina N abil, mis mõlemal juhul ($x \in L$ ja $x \notin L$) töötab korrektselt tõenäosusega $\frac{3}{4}$.

Def: Keel $L \subseteq \{0, 1\}^*$ kuulub klassi **BPP** kui leidub polünoomiaalne stohhastiline Turingi masin N , nii et iga $x \in \{0, 1\}^n$ korral:

- $x \in L \Rightarrow \Pr[1 \leftarrow N(x)] > \frac{3}{4}$;
- $x \notin L \Rightarrow \Pr[1 \leftarrow N(x)] < \frac{1}{4}$.

Hääletamine

Olgu $L \in \text{BPP}$ ja N vastav polünoomiaalne stohhastiline Turingi masin, st iga $x \in \{0, 1\}^n$ korral, tõenäosusega vähemalt $\frac{3}{4}$ on $N(x)$ korrektne vastus küsimusele "Kas $x \in L$?"

Algoritmi korrektsust saab parandada *hääletusalgoritmi* abil:

- arvutame $N(x)$ paaritu arv m korda ja salvestame tulemused:

$$b_1 \leftarrow N(x), b_2 \leftarrow N(x), \dots, b_m \leftarrow N(x) .$$

- Kui üle poolte b_i -dest on 1-d, st kui $\sum_{i=1}^m b_i > \frac{m}{2}$ siis väljastame 1;
- Vastasel korral väljastame 0.

Küsimus: Kui suur tuleb valida m , et saavutada etteantud usaldusväärsust?

Hääletusalgoritmi analüüs

Teoreem (Tšernovi tõke) (Chernoff bound) Olgu X_1, \dots, X_m sõltumatud juhuslikud suurused, mille väärtus on kas 1 või 0 vastavalt tõenäosustega p ja $1 - p$. Siis iga $0 \leq \Theta \leq 1$ korral

$$\Pr \left[\sum_{i=1}^m X_i \geq (1 + \Theta)pm \right] \leq e^{-\frac{\Theta^2}{3}pm} .$$

Hääletusalgoritmi analüüs: Olgu $X_i \in \{0, 1\}$ juhuslik suurus, mis = 1 parajasti siis kui i -ndal katsel $N(x)$ vastab küsimusele "Kas $x \in L$?" valesti, st kui $N(x) \neq [x \in L]$. Vastavalt BPP def-ile $p = \Pr[x_i = 1] < \frac{1}{4}$. Võttes $\Theta = 1$, saame

$$\Pr \left[\sum_{i=1}^m X_i \geq \frac{m}{2} \right] \leq e^{-\frac{m}{12}} .$$

Seega eksib hääletusalgoritm tõenäosusega $< e^{-\frac{m}{12}}$.

Klass BPP_ϵ

Olgu $\epsilon: \mathbb{N} \rightarrow [0 \dots 1]$ mingi funktsioon.

Def: Keel $L \subseteq \{0, 1\}^*$ kuulub klassi BPP_ϵ kui leidub polünoomiaalne stohastiline Turingi masin N , nii et iga x korral:

- $x \in L \Rightarrow \Pr[1 \leftarrow N(x)] > 1 - \epsilon(|x|)$;
- $x \notin L \Rightarrow \Pr[1 \leftarrow N(x)] < \epsilon(|x|)$.

Ülesanne: Tõesta järgmised laused:

- Kui $\epsilon(n) = 2^{-n^{O(1)}}$, siis $\text{BPP}_\epsilon = \text{BPP}$.
- Kui $\epsilon(n) = n^{-O(1)}$, siis $\text{BPP}_{\frac{1}{2}-\epsilon} = \text{BPP}$.

Klass P/Poly: arvutused nõuannetega

Keel $L \subseteq \{0, 1\}^*$ kuulub klassi P/Poly, kui leidub Turingi masin $M(x, a)$ tööajaga $|x|^{O(1)}$ ja jada $A = (a_1, a_2, \dots, a_n, \dots)$, nii et iga x korral:

$$x \in L \Leftrightarrow M(x, a_{|x|}) = 1 .$$

Jada A elemente nimetatakse nõuanneteks (ingl. *advice*).

NB! Klass P/Poly sisaldab keeli, mis ei ole äratuntavad tavalise Turingi masinaga. Olgu L mingi keel, mis ei ole tuvastatav Turingi masinaga. Olgu $\text{Code}: \mathbb{N} \rightarrow \{0, 1\}^*$ mingi kood, nii et $\text{Code}(\mathbb{N}) = \{0, 1\}^*$. Siis ka keel

$$U = \{1^n : \text{Code}(n) \in L\}$$

ei ole tuvastatav Turingi masinaga (näita seda!). Defineerime jada A , nii et $a_n \in \{0, 1\}$ ja $a_n = 1$ parajasti siis, kui $\text{Code}(n) \in L$.

Defineerime masina $M(x, a)$, mis väljastab 1 kui $x = 1^n$ (mingi n korral) ja $a = 1$. On selge, et $x \in U \Leftrightarrow M(x, a_{|x|}) = 1$ ja seega $U \in \text{P/Poly}$.

BPP \subseteq P/Poly

Teoreem: BPP \subseteq P/Poly.

Tõestus: Olgu $L \in \text{BPP}$. Et BPP = BPP $_{2^{-n}}$, siis leidub Turingi masin $M(x, a)$ tööajaga $|x|^{O(1)}$, nii et iga $x \in \{0, 1\}^n$ korral on vea tõenäosus:

$$\Pr_a [M(x, a) \neq [x \in L]] < 2^{-n} .$$

Seega,

$$\Pr_a [\exists x \in \{0, 1\}^n : M(x, a) \neq [x \in L]] < \sum_{x \in \{0, 1\}^n} 2^{-n} = 1 .$$

Järelikult leidub a_n , nii et iga $x \in \{0, 1\}^n$ korral

$$M(x, a_n) = [x \in L] .$$

Seega, masin M nõuannete jadaga $A = (a_0, a_1, \dots)$ tuvastab keele L ja seetõttu $L \in \text{P/Poly}$.

Tõenäosusteooria: Markovi võrratus

Markovi võrratus: Olgu X positiivsete reaalarvuliste väärtustega juhuslik suurus, mille erinevate väärtuste hulk on loenduv. Siis iga $k > 0$ korral

$$\Pr[X \geq k \cdot \mathbf{E}[X]] \leq \frac{1}{k} .$$

Tõestus:

$$\begin{aligned} \mathbf{E}[X] &= \sum_x x \cdot \Pr[x] = \sum_{x < k \cdot \mathbf{E}[X]} x \cdot \Pr[x] + \sum_{x \geq k \cdot \mathbf{E}[X]} x \cdot \Pr[x] \geq \\ &\geq k \cdot \mathbf{E}[X] \cdot \Pr[X \geq k \cdot \mathbf{E}[X]] . \end{aligned}$$

Järeldus: Olgu $X \geq 0$ mingi mittenegatiivne lõpliku keskväärtusega $\mathbf{E}[X]$ juhuslik suurus. Siis iga $\alpha > 0$ korral

$$\Pr[X \geq \alpha] \leq \frac{1}{\alpha} \mathbf{E}[X] .$$

Tõenäosusteooria: Kaks abivõrratust

Lemma 1: Iga $0 \leq \Theta \leq 1$ korral: $-\frac{\Theta^2}{2} \leq \Theta - (1 + \Theta) \ln(1 + \Theta) \leq -\frac{\Theta^2}{3}$.

Tõestus: Esmalt paneme tähele, et:

$$\begin{aligned} \Theta - (1 + \Theta) \ln(1 + \Theta) &= \Theta - (1 + \Theta) \cdot \left(\frac{\Theta}{1} - \frac{\Theta^2}{2} + \frac{\Theta^3}{3} - \frac{\Theta^4}{4} + \dots \right) \\ &= -\frac{\Theta^2}{1 \cdot 2} + \frac{\Theta^3}{2 \cdot 3} - \frac{\Theta^4}{3 \cdot 4} + \frac{\Theta^5}{4 \cdot 5} \dots \\ &= \sum_{n=2}^{\infty} (-1)^{n-1} \frac{\Theta^n}{n(n-1)}. \end{aligned}$$

Et rida $r = \frac{\Theta^3}{2 \cdot 3} - \frac{\Theta^4}{3 \cdot 4} + \frac{\Theta^5}{4 \cdot 5} \dots$ on vahelduvate märkidega, tema liikmete absoluutväärtused on rangelt kahanevad * ja esimene liige on positiivne,

*Võrratus $\frac{\Theta^n}{(n-1)n} > \frac{\Theta^{n+1}}{n(n+1)}$ tuleneb otseselt võrratusest $\frac{n-1}{n+1}\Theta < 1$.

siis see rida koondub positiivseks reaalarvuks. Järelikult:

$$\Theta - (1 + \Theta) \ln(1 + \Theta) = -\frac{\Theta^2}{2} + r \geq -\frac{\Theta^2}{2} .$$

Samal põhjusel võib väita, et rea $s = \frac{\Theta^4}{3 \cdot 4} - \frac{\Theta^5}{4 \cdot 5} + \frac{\Theta^6}{4 \cdot 5} - \dots$ summa on positiivne, mistõttu:

$$\begin{aligned} \Theta - (1 + \Theta) \ln(1 + \Theta) &= -\frac{\Theta^2}{2} + \frac{\Theta^3}{6} - s \leq -\frac{\Theta^2}{2} + \frac{\Theta^3}{6} \\ &\leq -\frac{\Theta^2}{2} + \frac{\Theta^2}{6} = -\frac{\Theta^2}{3} . \end{aligned}$$

Lemma 2: Iga $0 \leq \Theta \leq 1$ korral: $\Theta - (1 - \Theta) \ln(1 - \Theta) \geq \frac{\Theta^2}{2}$.

Tõestus Võrratus järeldub vahetult järgmisest tähelepanekust:

$$\Theta - (1 - \Theta) \ln(1 - \Theta) = \frac{\Theta^2}{2 \cdot 1} + \frac{\Theta^3}{3 \cdot 2} + \frac{\Theta^4}{4 \cdot 3} + \dots = \sum_{n=2}^{\infty} \frac{\Theta^n}{n(n-1)} .$$

Tõenäosusteooria: Tšernovi tõkked

Teoreem: Olgu X_1, \dots, X_m sõltumatud juhuslikud suurused, mille väärtus on kas 1 või 0 vastavalt tõenäosustega p ja $1 - p$. Olgu $X = \sum_{i=1}^m X_i$. Siis iga $0 \leq \Theta \leq 1$ korral kehtivad võrratused:

$$\Pr[X \geq (1 + \Theta)pm] \leq e^{-\frac{\Theta^2}{3}pm} \quad (1)$$

$$\Pr[X \leq (1 - \Theta)pm] \leq e^{-\frac{\Theta^2}{2}pm} . \quad (2)$$

Tõestus: (1): Kui $t \in \mathbb{R}^+$, siis $\Pr[X \geq (1 + \Theta)pm] = \Pr[e^{tX} \geq e^{t(1+\Theta)pm}]$. Markovi võrratusest saame, et $\Pr[e^{tX} \geq k \cdot \mathbf{E}(e^{tX})] \leq 1/k$ iga $k > 0$ korral. Võtame $k = e^{t(1+\Theta)pm} (\mathbf{E}(e^{tX}))^{-1}$. Siis

$$\Pr[X \geq (1 + \Theta)pm] \leq e^{-t(1+\Theta)pm} \mathbf{E}(e^{tX}) .$$

Et $\mathbf{E}(e^{tX}) = (\mathbf{E}(e^{tX_1}))^m = (1 + p(e^t - 1))^m$, siis tehes asenduse, saame

$$\begin{aligned} \Pr[X \geq (1 + \Theta)pm] &\leq e^{-t(1+\Theta)pm} (1 + p(e^t - 1))^m \\ &\leq e^{-t(1+\Theta)pm} \cdot e^{pm(e^t - 1)}. \end{aligned}$$

Siin kasutatakse asjaolu, et $1 + a \leq e^a$ iga $a > 0$ korral (siin $a = p(e^t - 1)$). Lõpuks võttes $t = \ln(1 + \Theta)$, saame **Lemma 1** põhjal, et

$$\Pr[X \geq (1 + \Theta)pm] \leq e^{pm[\Theta - (1+\Theta)\ln(1+\Theta)]} \leq e^{-\frac{\Theta^2}{3}pm}.$$

(2): Kui $t \in \mathbb{R}^+$, siis $\Pr[X \leq (1 - \Theta)pm] = \Pr[pm - X \geq \Theta pm] = \Pr[e^{t(pm-X)} \geq e^{t\Theta pm}]$. Markovi võrratusest saame, et

$$\Pr[e^{t(pm-X)} \geq k \cdot \mathbf{E}(e^{t(pm-X)})] \leq 1/k$$

iga $k > 0$ korral. Võttes $k = e^{t\Theta pm} (\mathbf{E}(e^{t(pm-X)}))^{-1}$ saame

$$\Pr[X \leq (1 - \Theta)pm] \leq e^{-t\Theta pm} \cdot \mathbf{E}(e^{t(pm-X)}) = e^{t(1-\Theta)pm} \cdot \mathbf{E}(e^{-tX}).$$

Et $\mathbf{E}(e^{-tX}) = (\mathbf{E}(e^{-tX_1}))^m = (1 - p(1 - e^{-t}))^m$, siis tehes asenduse, saame

$$\begin{aligned} \Pr[X \leq (1 - \Theta)pm] &\leq e^{-t(1-\Theta)pm} (1 - p(1 - e^{-t}))^m \\ &\leq e^{-t(1-\Theta)pm} \cdot e^{-pm(1-e^{-t})} \\ &= e^{-pm[t(1-\Theta)+1-e^{-t}]} . \end{aligned}$$

Lõpuks võttes $t = -\ln(1 - \Theta)$, saame **Lemma 2** põhjal, et

$$\Pr[X \leq (1 - \Theta)pm] \leq e^{-pm[\Theta - (1-\Theta)\ln(1-\Theta)]} \leq e^{-\frac{\Theta^2}{2}pm} .$$