Teoreetiline informaatika

Kevad 2022

13. Keerukusteooria

Sellist tõestusmeetodit nimetatakse taandamiseks keelelt L_{TM}:

$$\underbrace{L_{\text{TM}}}_{\text{lahendab}} \quad \leq_{\text{M}} \quad \underbrace{\text{PEATUB}}_{\text{lahendab}}$$

$$\text{lahendab} \quad \Longleftarrow \quad \text{lahendab}$$

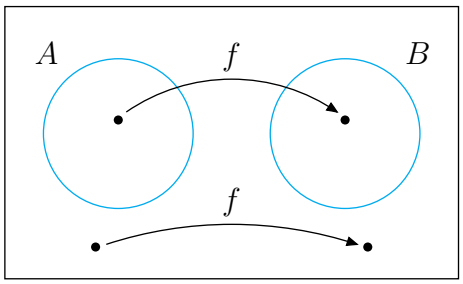
Kui leidub masin, mis lahendab keelt PEATUB, siis leidub masin, mis lahendab keelt L_{TM}. Keel PEATUB on vähemalt niisama raske kui keel L_{TM}.

Definition. *Funktsiooni $f\colon \Sigma^* \to \Sigma^*$ nimetatakse arvutatavaks, kui leidub Turingi masin, mis iga sisendsõne w puhul peatub ja jätab lindile sõne $f(w)$.*

Definition. *Keel A on kujutusega taandatav keelele B ehk $A \leq_M B$, kui leidub selline arvutatav funktsioon $f\colon \Sigma^* \to \Sigma^*$, et iga w puhul*

$$w \in A \Leftrightarrow f(w) \in B.$$

Funktsiooni f nimetatakse keele A reduktsiooniks keelele B.



Theorem. *Kui $L_A \leq_M L_B$ ja L_B on lahenduv, siis L_A on lahenduv.*

Tõestus. Olgu M_B Turingi masin, mis lahendab keelt L_B, ja f keele L_A reduktsioon keelele L_B. Kirjeldame masinat M_A, mis lahendab keelt L_A. Sisendsõnel w masin M_A:

1. arvutab $f(w)$;
2. teeb läbi masina M_B töökäigu sõnel $f(w)$ ja väljastab sama tulemuse nagu M_B.

Näitame, et masin M_A lahendab keelt L_A.

- Kui $w \in A$, siis $f(w) \in B$, sest f on reduktsioon. Seega M_B aktsepteerib sõnet $f(w)$. Järelikult M_A aktsepteerib sõnet w.
- Kui $w \notin A$, siis $f(w) \notin B$. Seega M_B lükkab $f(w)$ tagasi. Järelikult M_A lükkab w tagasi. □

Et Turingi masin töötab standardsete sammude kaupa, siis saab selle arvutusmudeli abil uurida algoritmide tööaega, lugedes aja mõõduks masina tehtavate sammude arvu.

Suur O ja väike o

Definition. *Olgu M determineeritud Turingi masin, mis peatub kõigil sisenditel. Masina M tööaeg ehk ajaline keerukus on funktsioon $f\colon \mathbb{N} \to \mathbb{N}$, kus $f(n)$ on maksimaalne sammude arv, mida M teeb sisendil pikkusega n.*

Seega leiame iga sisendi jaoks pikkusega n, mitu sammu Turingi masin sellel sisendil teeb, ning $f(n)$ on neist sammude arvudest maksimaalne. Mõnikord võib $f(n)$ avalduda ka lihtsa valemiga, näiteks võib ta olla $n+1$, $2n^2+3n+5$, n^3+n^2+101, 2^n+5n+1 jne.

Definition. *Olgu f, $g\colon \mathbb{N} \to \mathbb{R}^+$ funktsioonid. Ütleme, et $f(n) = O(g(n))$, kui leiduvad reaalarv $C > 0$ ja naturaalarv n_0 nii, et iga $n > n_0$ puhul*

$$f(n) < C\, g(n).$$

Example 1. Vaatleme järgmisi funktsioonide paare.

- Kui $f(n) = n + 10$, $g(n) = n^2$, siis $f(n) = O(g(n))$, sest kui $n > 3$, siis $n + 10 < n^2$. Seega võime võtta $C = 1$ ja $n_0 = 3$.
- Kui $f(n) = 5n^3 + 2n^2 + 7n + 10$ ja $g(n) = n^3$, siis $f(n) = O(g(n))$. Sobivad $C = 6$ ja $n_0 = 4$, aga ka näiteks $C = 100$ ja $n_0 = 0$ jne.
- Kui $f(n) = 10n^2 + 100n + 10$ ja $g(n) = 2^n$, siis $f(n) = O(g(n))$.
- Kui $f(n) = \log_2 n$ ja $g(n) = \sqrt{n}$, siis $f(n) = O(g(n))$.

Paneme tähele, et kahe funktsiooni võrdlemisel on $O(\cdot)$ tähistuse puhul olulised ainult funktsioonide kõige kiiremini kasvavad liikmed. Viimastes omakorda ei ole olulised liikmete kordajad, sest need on konstandid. Peale selle, kuna $\log_a n = \log_a b \cdot \log_b n$, siis ei ole oluline oluline ka logaritmi alus, mistõttu võime edaspidi kirjutada lihtsalt $O(\log n)$.

Definition. *Olgu f, $g\colon \mathbb{N} \to \mathbb{R}^+$ funktsioonid. Ütleme, et $f(n) = o(g(n))$, kui*

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Example 2. Kehtivad järgmised seosed.

- $\sqrt{n} = o(n)$
- $\log n = o(n^{\frac{1}{100}})$
- $n \log n = o(n^2)$
- $n^{100} = o(2^n)$

Üldiselt, logaritmfunktsioon kasvab aeglasemalt kui ükskõik millise positiivse astendajaga astmefunktsioon. Ükskõik milline astmefunktsioon kasvab aeglasemalt kui iga ühest suurema astendatavaga eksponentfunktsioon.

Klassid P ja NP

Järgnevas ei erista me üksteisest erinevaid polünomiaalseid ajalisi keerukusi, kuid eristame polünomiaalset keerukust ja eksponentsiaalset keerukust. See tuleb sellest, et paljud erinevad determineeritud arvutusmudelid on omavahel polünomiaalselt ekvivalentsed.

Definition. *Klass* P *koosneb kõigist sellistest keeltest, mille ajaline keerukus on $f(n) = O(p(n))$, kus p on mingi polünoom muutujast n.*

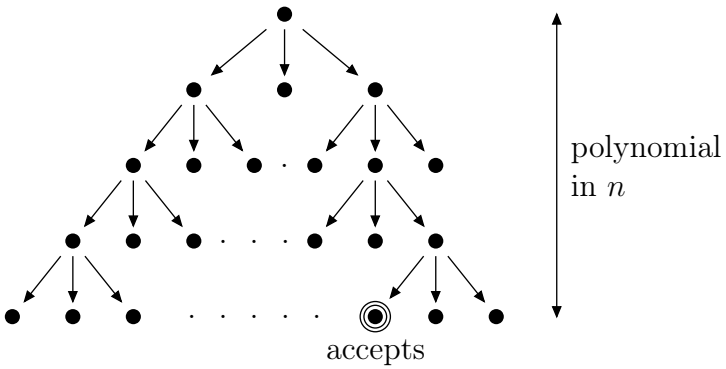
Keele ajaline keerukus on keelt lahendava Turingi masina ajaline keerukus. Siin defineeritud klass P on üks ja sama sama kõigi arvutusmudelite jaoks, mis on polünomiaalselt ekvivalentsed ühe lindiga determineeritud Turingi masinaga. Praktikas vastab klass P ligikaudu selliste ülesannete klassile, mis on arvutil polünomiaalse ajaga lahendatavad.

Example 3. Järgmised keeled kuuluvad klassi P.

- Iga regulaarne keel.
- Keel, mis koosneb kõigist monotoonselt kahanevatest järjendistest. Ülesanne kontrollida, kas antud järjend on monotoonselt kahanev, on polünomiaalse ajaga lahenduv.
- Antud on graaf G, tipud s ja t ning täisarv k. Kontrollida, kas lühim ahel graafis G tipust s tippu t on pikkusega k. Sellele vastab keel

$$L_G = \{\langle G, s, t, k\rangle \mid \text{lühim ahel graafi } G \text{ tipust } s \text{ tippu } t \text{ on pikkusega } k\}.$$

Definition. *Klass* NP *koosneb kõigist sellistest keeltest, mida suudab polünomiaalse ajaga lahendada mingi mittedetermineeritud Turingi masin.*



Example 4. Graafi klikk on selline alamgraaf, kus iga kaks tippu on servaga ühendatud. Vaatleme keelt

$$\text{KLIKK} = \{\langle G, k\rangle \mid G \text{ on suunamata graaf, milles leidub } k\text{-tipuline klikk}\}.$$

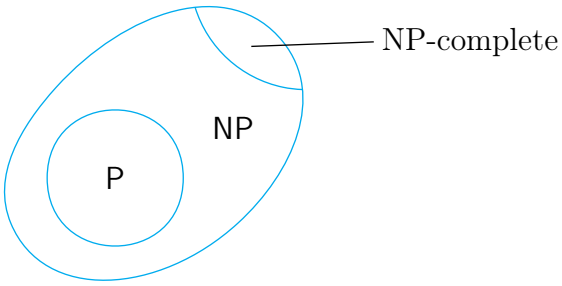
Tõestame, et keel KLIKK kuulub klassi NP.

Seda keelt lahendab mittedeterminеeritud Turingi masin, mis sisendil $\langle G, k\rangle$ töötab nii.

1. Mittedeterminеeritult valib graafi G tippude hulga alamhulga S suurusega k.
2. Kontrollib, kas G sisaldab kõiki servi hulga S tippude vahel.
3. Kui jah, siis aktsepteerib sisendit; kui ei, siis lükkab tagasi.

Triviaalselt $\mathsf{P} \subseteq \mathsf{NP}$. Ülesanne KLIKK on näide ülesandest, mis kuulub klassi NP, aga mille kohta pole teada, kas ta kuulub ka klassi P. Üldiselt polegi teada, kas $\mathsf{P} = \mathsf{NP}$. Teiste sõnadega, pole teada, kas leidub ülesanne, mis kuulub klassi NP, aga ei kuulu klassi P.

Klassis NP leidub ülesandeid, millel on omadus, et kui mõne sellise ülesande lahendamiseks leidub determineeritud polünomiaalne algoritm, siis klassi NP iga ülesane lahendamiseks leidub determineeritud polünomiaalne algoritm (siit järelduks $\mathsf{P} = \mathsf{NP}$). Sellised ülesandeid nimetatakse *NP-täielikeks*.



Praktikumiülesanded

1. Olgu

$$\text{REGULAAR} = \{\langle M \rangle \mid M \text{ on Turingi masin ja } L(M) \text{ on regulaarne keel}\}$$

Tõestada, et keel REGULAAR on mittelahenduv.

2. Kas lause on tõene või väär?

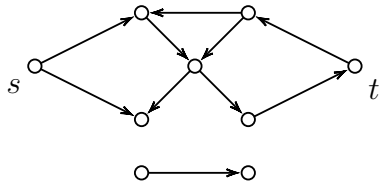
(a) $n = o(2n)$
(b) $3n^5 = O(10n^3 + 20n^2 + 100)$
(c) $2^n = o(3^n)$
(d) $n^2 = O(n \log n)$

3. Tõestada, et kui $t(n)$ funktsioon, mille puhul $t(n) \geq n$, siis iga mitme lindiga Turingi masina jaoks, mis töötab ajas $t(n)$, leidub temaga ekvivalentne ühe lindiga Turingi masin, mis töötab ajas $O(t^2(n))$.

4. Defineerime keele

$$\text{AHEL} = \{\langle G, s, t \rangle \mid G \text{ on suunatud graaf, milles leidub suunatud ahel tipust } s \text{ tippu } t\}.$$

Näide:



Tõestada, et AHEL $\in$ P.

Lahendused

1. *Solution.* Tõestame väite taandamisega

$$L_{\mathrm{TM}} \leq_{\mathrm{M}} \text{REGULAAR}.$$

Oletame, et REGULAAR on lahenduv, ja olgu M_{R} Turingi masin, mis keelt REGULAAR lahendab. Konstrueerime Turingi masina M_L, mis lahendab keelt L_{TM}. Masin M_L töötab sisendil $\langle M, w\rangle$ järgmiselt.

1. Konstrueerib masina M_0, millel on sisend x, kusjuures
   (a) kui x on kujul 0^n1^n, siis M_0 peatub aktsepteerivas olekus;
   (b) kui x ei ole kujul 0^n1^n, siis M_0 teeb läbi masina M töökäigu sisendil w ja peatub aktsepteerivas olekus parajasti siis, kui M peatub sisendil w aktsepteerivas olekus.
2. Teeb läbi masina M_{R} töökäigu sisendil $\langle M_0\rangle$.
3. Kui M_{R} lõpetab aktsepteerimisega, siis läheb aktsepteerivasse olekusse; kui M_{R} lõpetab tagasilükkamisega, siis läheb tagasilükkavasse olekusse.

Kõik sammud on Turingi masinate poolt teostatavad. Sealhulgas on masina M_0 konstrueerimine võimalik: kõigepealt kontrollib M_0, kas sisendil on kindel kuju, ning seejärel jäljendab masina M tööd sisendil w.

Mis on masina M_0 keel?

- Kui M aktsepteerib sõnet w, siis $L(M_0) = \Sigma^*$. See on regulaarne keel.
- Kui M ei aktsepteeri sõnet w, siis $L(M_0) = \{0^n1^n \mid n \geq 0\}$. See ei ole regulaarne keel.

Järelikult:

- kui M aktsepteerib sõnet w, siis sammul 2 lõpeb masina M_{R} töö sisendi $\langle M_0\rangle$ aktsepteerimisega ning M_L satub aktsepteerivasse olekusse;

- Kui M ei aktsepteeri sõnet w, siis sammul 2 lükkab M_{R} sisendi $\langle M_0 \rangle$ tagasi ja M_L lõpetab tagasilükkavas olekus.

See tähendab, et M_L aktsepteerib sisendit $\langle M, w \rangle$ parajasti siis, kui masin M aktsepteerib sisendit w.

Oleme saanud, et kui on olemas masin M_{R}, mis lahendab keelt REGULAAR, siis on olemas ka masin M_L, mis lahendab keelt L_{TM}. Viimast masinat aga pole. Vastuolu.

2. *Solution.*

(a) Väär: $\lim\limits_{n\to\infty} \frac{n}{2n} = \frac{1}{2} \neq 0$.

(b) Väär: iga n_0 ja C puhul võime valida n nii, et $n \geq \max\{n_0+1, 1000C\}$, siis $n > n_0$ ja $3n^5 > C(10n^3 + 20n^2 + 100)$.

(c) Tõene: $\lim\limits_{n\to\infty} \frac{2^n}{3^n} = \lim\limits_{n\to\infty} \left(\frac{2}{3}\right)^n = 0$.

(d) Väär: et $\lim\limits_{n\to\infty} \frac{n^2}{n \log n} = \lim\limits_{n\to\infty} \frac{n}{\log n} = \infty$, siis iga konstandi C puhul saame valida piisavalt suure indeksi n nii, et $f(n) > C\, g(n)$.

3. *Solution.* Varem oleme kursuses näinud, kuidas teisendada mitme lindiga Turingi masinat ühe lindiga Turingi masinaks. Tõestame, et mitme lindiga Turingi masina iga sammu jäljendamine võtab ühe lindiga masinal ülimalt $O(t(n))$ sammu.

Töö alguses kirjutab ühe lindiga masin M_S oma lindile mitme lindiga masina M_M kõigi lintide sisud. Ühe sammu sooritamiseks loeb M_S läbi kogu oma lindi sisu ja teeb kindaks masina M_M peade all olevad sümbolid. Seejärel läbib M_S lindi veelkord ning uuendab selle sisu. Kui masina M_M mõni pea liigub viimasest mittetühjast sümbolist paremale, siis nihutab M_S lindi sisu sellest kohast alates ühe lahtri võrra paremale.

Et M_M teeb kokku $O(t(n))$ sammu, siis on masina M_S lindi aktiivse osa pikkus $O(t(n))$. Seetõttu kulub masinal M_S igaks lindi läbilugemiseks aeg $O(t(n))$. Masina M_M iga sammu jäljendamiseks sooritab M_S kaks lindi läbivaatust ja vajadusel piiratud arvu paremale nihutamisi. Iga selline operatsioon (läbivaatus/nihutamine) võtab ülimalt $O(t(n))$ sammu.

Koguaeg, mis masinal M_S kulub masina M_M tegevuse jäljendamiseks, koosneb lindi algseadistamisest, mis võtab $O(n)$ sammu, ning masina M_M kõigi $t(n)$ sammu jäljendamisest, mis võtab $t(n) \cdot O(t(n)) = O(t^2(n))$ sammu. Koguaeg on seega $O(t^2(n)) + O(n)$. Et eelduse põhjal $t(n) \geq n$, siis saamegi, et kogu ajaline keerukus on $O(t^2(n))$.

4. *Solution.* Vaatleme keele AHEL puhul järgmist algoritmi, mille sisendiks on $\langle G, s, t\rangle$.

1. Märgistab tipu s.
2. Kordab järgmist, kuni veel märgistatakse mõni tipp:
   - vaatab läbi G kõik kaared (u, v); kui kaare tipp u on märgistatud ja tipp v ei ole, siis märgistab tipu v.
3. Kui t on märgistatud, siis läheb aktsepteerivasse oleksusse, vastasel juhul läheb tagasilükkavasse olekusse.

Tõestame, et see algoritm on korrektne. Kui graafis leidub mingi ahel $s = v_0 \to v_1 \to v_2 \to v_3 \to \ldots \to v_{l-1} \to v_l = t$, siis induktsiooniga i järgi saame, et selle ahela kõik tipud, sealhulgas tipp t, saavad märgistatud. Kui aga ahelat $s \rightsquigarrow t$ ei leidu, siis jääb tipp t märgistamata, sest igal sammul märgistatakse ainult sellised tipud, mis asuvad mingil tipust s lähtuval ahelal.

Tõestame, et see algoritm on polünomiaalne. Samm 1 võtab kindlasti polünomiaalse aja, samuti samm 3. Sammu 2 täidetakse ülimalt nii mitu korda, kui suur on graafi G tippude arv, sest igal sammul märgistatakse vähemalt üks tipp. Järelikult on kõik sammud 1, 2, 3 polünomiaalse ajalise keerukusega, seega ka algoritmi kogukeerukus on polünomiaalne.