
Keerukusteooria elemente

Algoritmide keerukus

Definitsioon:

Keerukus on funktsioon f , mis seab andmete mahule n vastavusse algoritmi sammude arvu (ajaline keerukus) või kasutatava mälu mahu (mahuline keerukus)

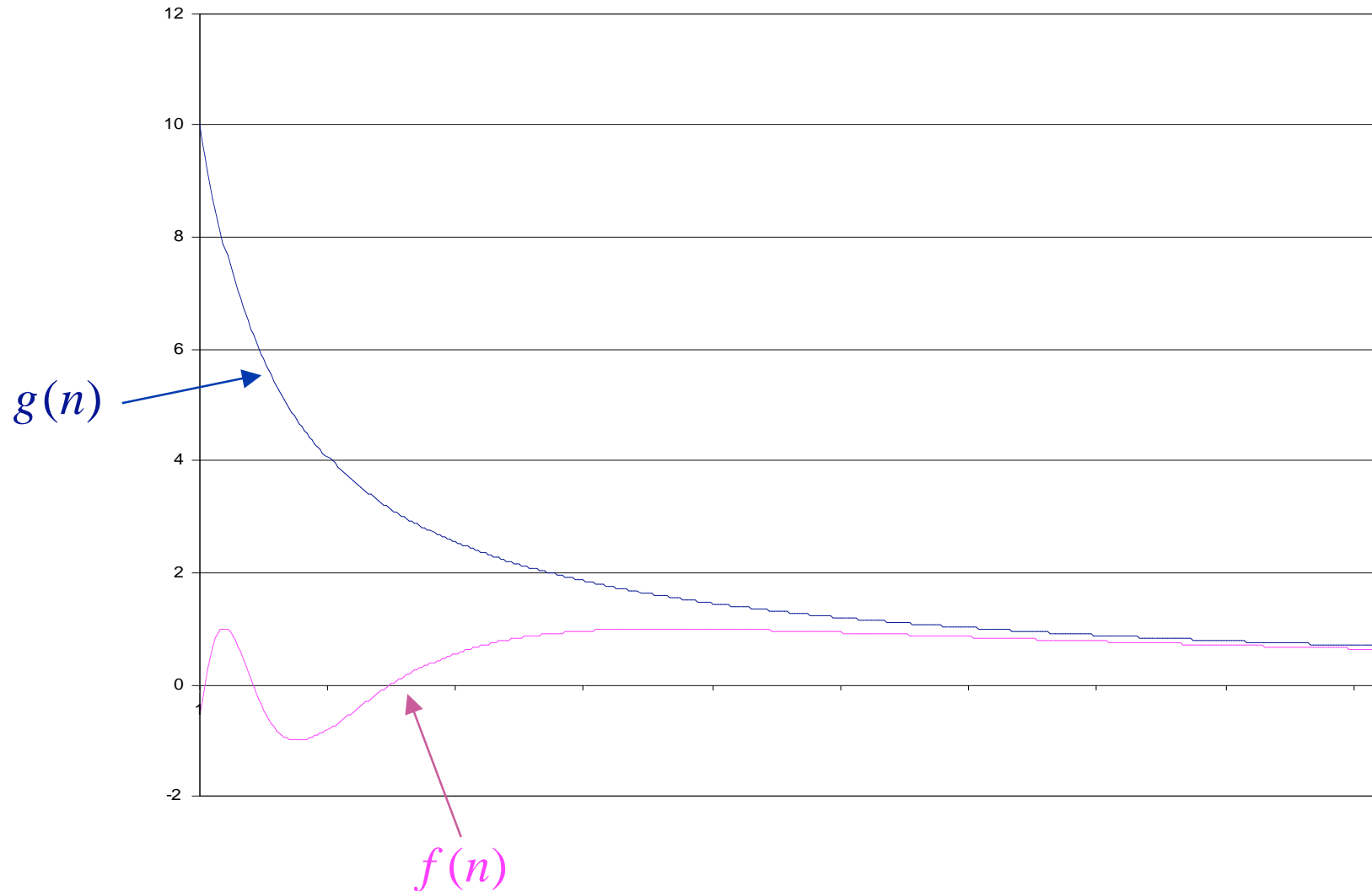
Turingi masina korral on

- ajaline keerukus - konfiguratsioonide arv;
- mahuline keerukus - kasutatud lindipositsioonide arv.

Kasutatakse asümptootilisi hinnanguid, mis iseloomustavad keerukusfunktsioonide käitumist andmete mahu n piiramatul kasvamisel.

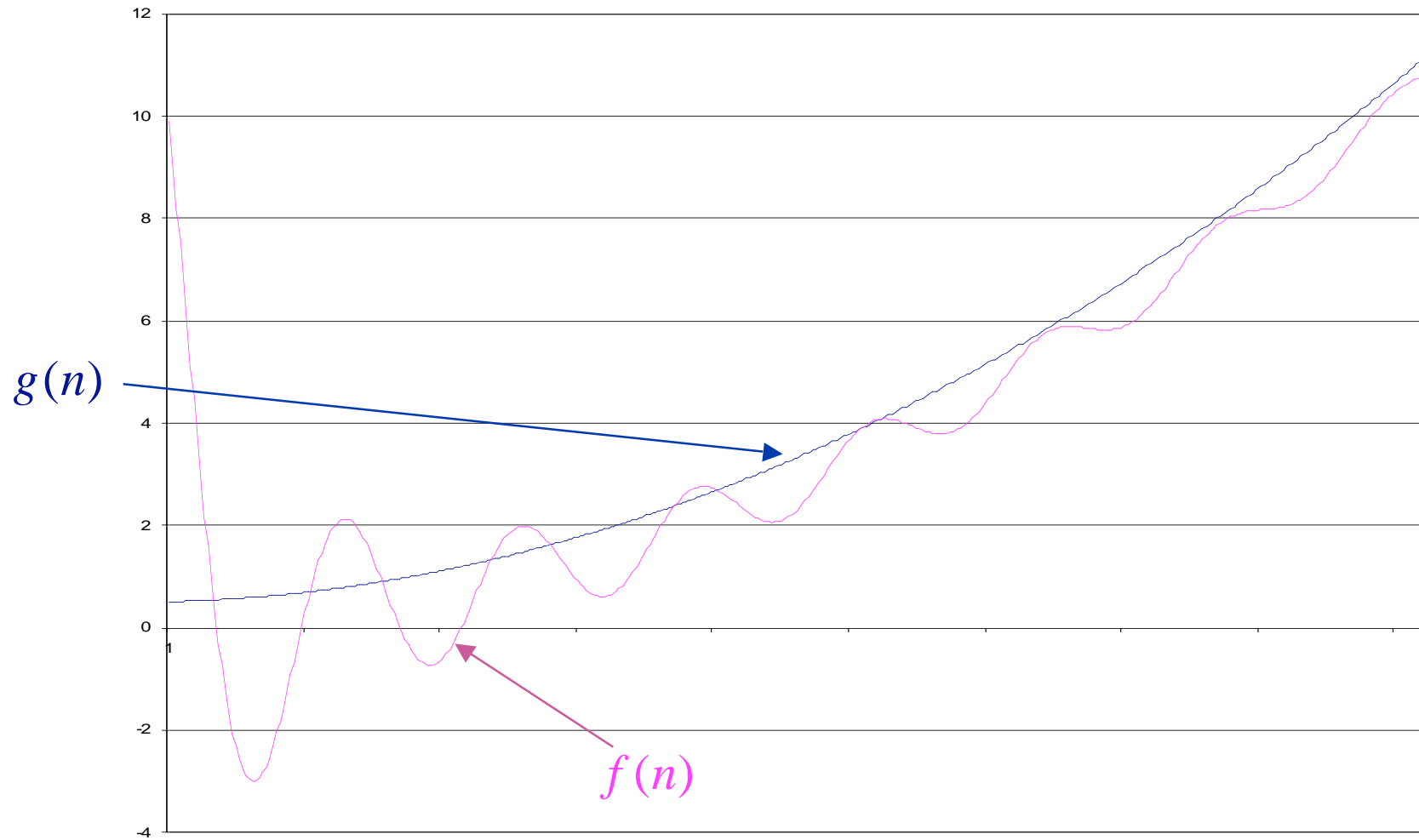
Asümptootiliselt võrdsed funktsioonid (näide 1)

$$f(n) = O(g(n))$$



Asümptootiliselt võrdsed funktsioonid (näide 2)

$$f(n) = O(g(n))$$



O- notatsioon

Definitsioon:

Olgu f ja g naturaalarvulised funktsioonid. Funktsiooni f asümptootiline (ülemine) hinnang

$$f(n) = O(g(n))$$

parajasti siis, kui leiduvad konstandid $c > 0$ ja $N > 0$, nii et iga $n > N$ korral $|f(n)| \leq c|g(n)|$.

Definitsioon:

g on funktsiooni f täpne asümptootiline hinnang, kui

$$f(n) = O(g(n)) \text{ ja } g(n) = O(f(n))$$

Tähistus: $f = \Theta(g)$.

Omadus:

Seosest $f = \Theta(g)$ järeldub, et $\frac{|f|}{|g|} \leq c$ ja $\frac{|g|}{|f|} \leq c'$

$$\text{ehk } \frac{1}{c'} \leq \frac{|f|}{|g|} \leq c.$$

O- notatsioon (2)

Omadusi:

- $\forall k > 0$ korral $kf = \mathcal{O}(f)$;
- kui $f = \mathcal{O}(g)$ ja $h = \mathcal{O}(g)$, siis $(f + h) = \mathcal{O}(g)$;
- kui $f = \mathcal{O}(g)$ ja $g = \mathcal{O}(h)$, siis $f = \mathcal{O}(h)$;
- $n^r = \mathcal{O}(n^s)$ kui $0 \leq r \leq s$;
- kui p on d -astme polünoom, siis $p = \mathcal{O}(n^d)$;
- kui $f = \mathcal{O}(g)$ ja $h = \mathcal{O}(r)$, siis $f \times h = \mathcal{O}(g \times r)$;
- $n^k = \mathcal{O}(b^n)$ kui $b > 1$ ja $k \geq 0$;
- $\log_k n = \mathcal{O}(n^k)$;
- $\log_b n = \mathcal{O}(\log_d n)$ iga $b, d > 1$ korral;
- $\sum_{k=1}^n k^r = \Theta(n^{r+1})$.

Keerukushinnangute praktiline tähendus

Programmi tööaeg $c f(n)$	Lahendamisaja suhteline suurenemine $f(25)/f(5)$
c_1	1
$c_2 \log n$	2
$c_3 n$	5
$c_4 n \log n$	10
$c_5 n^2$	25
$c_6 n^3$	125
$c_7 2^n$	1048576

Keerukushinnangute praktiline tähendus (2)

Programmi tööaeg (mikrosek.)	Suurim ülesanne, mille lahendamise aeg < 1 sek.	Suurim ülesanne, mille lahendamise aeg < 1 päev	Suurim ülesanne, mille lahendamise aeg < 1 aasta
n	$n = 1\,000\,000$	$n = 86\,400\,000\,000$	$n = 31\,530\,000\,000\,000$
$n \log_2 n$	$n = 62\,746$	$n = 2\,755\,147\,514$	$n = 798\,160\,978\,500$
n^2	$n = 1\,000$	$n = 293\,938$	$n = 561\,569$
n^3	$n = 100$	$n = 4\,421$	$n = 31\,593$
2^n	$n = 19$	$n = 36$	$n = 44$
$n!$	$n = 9$	$n = 14$	$n = 16$

Keerukushinnangute praktiline tähendus (3)

Näiteks:

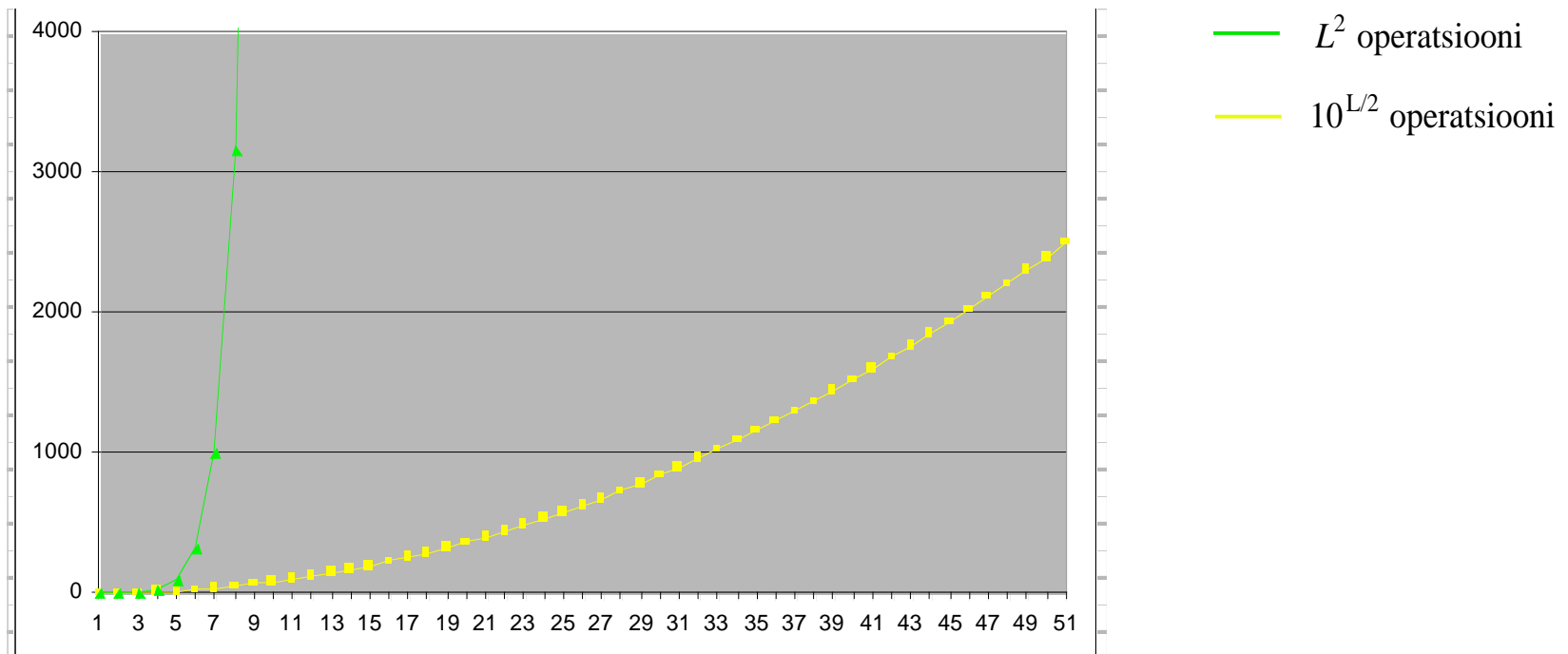
a) korrutamine - polünoomiaalne keerukus - $\sim L^2$ operatsiooni.

$$10433 * 16453 = ?$$

b) tegurdamine - eksponentsiaalne keerukus - $\sim 10^{L/2}$ operatsiooni.

$$? * ? = 171654149$$

Polünomiaalne- ja eksponentsiaalne keerukus



1 aasta = $31,5 \cdot 10^6$ s

Arvuti, mille kiirus on 10^6 op/s, teeks aastas $31,5 \cdot 10^{12}$ operatsiooni

Kui $L=50$, siis korrutamiseks kulub $0,0025$ s ja tegurdamiseks $0,3 \cdot 10^9$ aastat

Võrdluseks: suurest paugust olevat möödunud $14 \cdot 10^9$ aastat.

Algoritmide keerukus

- a) (ajaline) keerukus halvimal juhul - $W(n)$ maksimaalne operatsioonide arv, mida tuleb täita sisendi mahu n korral
- b) (ajaline) keerukus parimal juhul - $B(n)$ minimaalne operatsioonide arv, mida tuleb täita sisendi mahu n korral
- c) keskmise (ajaline) keerukus - $A(n)$ keskmine operatsioonide arv, mida tuleb täita sisendi mahu n korral

x - sisendi "omadusi" väljendav näitaja

n - sisendi maht

$p(x, n)$ - sagedus

$T(x, n)$ - algoritmi tööaeg

$$A(n) = \sum_x p(x, n)T(x, n) \quad \text{või} \quad A(n) = \int p(x, n)T(x, n)dx$$

Algoritmide keerukus ja arvutamise mudel

Teoreem 10.1

Olgu $t(n) > n$. Iga mitme lindiga Turingi masinal ajalise keerukusega $O(t(n))$ töötava programmi jaoks leidub ekvivalentse funktsionaalsusega programm ühe lindiga Turingi masinal, nii et tema ajaline keerukus on $O(t^2(n))$.

Teoreem 10.2

Olgu $t(n) > n$. Iga ühe lindiga mittedeterministlikul Turingi masinal ajalise keerukusega $O(t(n))$ töötava programmi jaoks leidub ekvivalentse funktsionaalsusega programm ühe lindiga deterministlikul Turingi masinal, nii et tema ajaline keerukus on $2^{O(t(n))}$.

Ülesannete keerukusklassid

Definitsioon

$$\text{TIME}(t(n)) = \left\{ L \mid \begin{array}{l} L \text{ on ülesanne, mille lahendamise aeg} \\ \text{Turingi masinal } W(t) = (O)(t(n)) \end{array} \right\}$$

Definitsioon

Polünoomiaalne keerukusklass P on nende ülesannete hulk, mis on lahenduvad *ühe lindiga deterministlikul Turingi masinal* polünoomiaalse ajaga :

$$P = \bigcup_k \text{TIME}(n^k).$$

Omadus

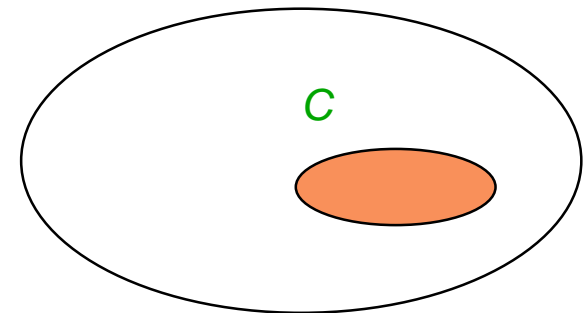
- Klass P on invariantne kõigil arvutamise mudelitel, mis on polünoomiaalses ajas modelleeritavad deterministlikul Turingi masinal
- “Klassi P ülesanded on reaalselt arvutil lahendatavad”

NP keerukus

Definitsioon

Omadus C on *verifitseeritav* (= *lahenduv*) hulgal A , kui leidub $\forall x \in A$ arvutatav predikaat

$$C(x) = \begin{cases} \mathbf{true}, & \text{kui } x \text{ evib omadust } C; \\ \mathbf{false}, & \text{vastasel korral.} \end{cases}$$



Definitsioon

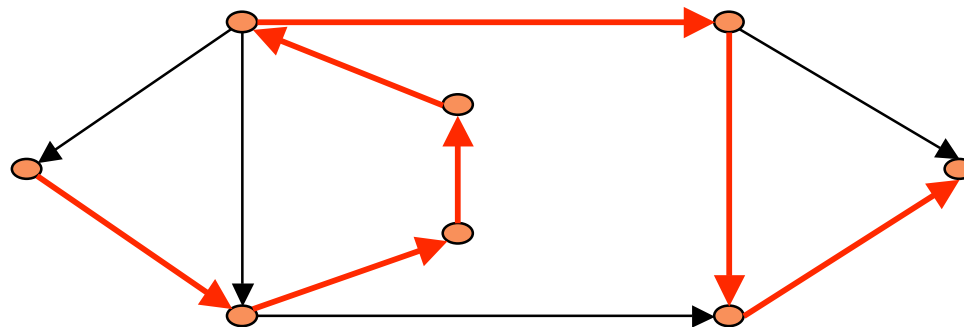
Omadus C on *tuvastatav* (= *genereeritav*) hulgal A , kui leidub $\exists x \in A$, nii et omadust verifitseeriv predikaat $C(x) = \mathbf{true}$.

Definitsioon

Omadus C on *NP-tuvastatav* hulgal A , kui ta polünoomiaalses ajas verifitseeritav ($C \in P$).

Hamiltoni ahel

Tingimus: graafi tipp tuleb läbida täpselt üks kord.



$$HAMPATH = \{G, s, t \mid G \text{ on } o\text{-graaf, milles on Hamiltoni ahel tipust } s \text{ tippu } t\}$$

- Ülesanne *HAMPATH* on polünoomiaalselt verifitseeritav
- Pole teada polünoomiaalset algoritmi, mis tuvastaks Hamiltoni ahela olemasolu
- Hamiltoni ahel on polünoomiaalselt tuvastatav mitedeterministlikul Turingi masinal
- Pole teada polünoomiaalset algoritmi ülesande *HAMPATH* verifitseerimiseks

NP keerukus (2)

Teoreem Omadus C on NP- tuvastatav parajasti siis, kui ta on polünoomiaalses ajas tuvastatav kasutades mittedeterministlikku Turingi masinat.

Definitsioon

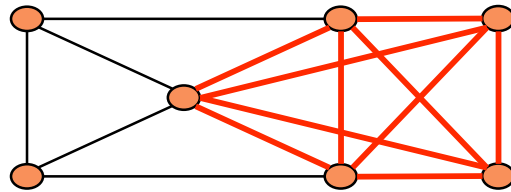
$$\text{NTIME}(t(n)) = \left\{ L \mid \begin{array}{l} L \text{ on ülesanne, mille lahendamise aeg} \\ \text{mitedeterministlikul Turingi masinal } B(t) = (O)(t(n)) \end{array} \right\}$$

Definitsioon (2)

$$\text{NP} = \bigcup_k \text{NTIME}(n^k).$$

NP keerukate ülesannete näited

Näide 1 $CLIQUE = \{ \langle G, k \rangle \mid G \text{ on mitteorienteeritud graaf, milles on } k\text{-klikk} \}$



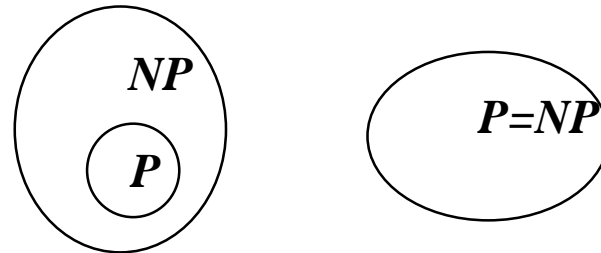
Näide 2 $SUBSET-SUM = \left\{ \langle S, t \rangle \mid \begin{array}{l} S = \{x_1, \dots, x_k\} \text{ mingi alamhulga } \{y_1, \dots, y_l\} \subseteq S \\ \text{korral } \sum y_i = t \end{array} \right\}$

- Ülesanded \overline{CLIQUE} ja $\overline{SUBSET-SUM}$ kuuluvad klassi $coNP$
- Pole teada, kas $NP=coNP$ või mitte.
- Parimad teadaolevad algoritmid NP ülesannete lahendamiseks on eksponentsiaalse keerukusega, st

$$NP \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$$

P = NP ?

Üks kahest võimalusest peab olema õige:



Kehtestatavuse ülesanne: $SAT = \{ \phi \mid \phi \text{ on kehtestatav Bool'i valem} \}$

$$\phi = (\bar{x} \wedge y) \vee (x \wedge \bar{z})$$

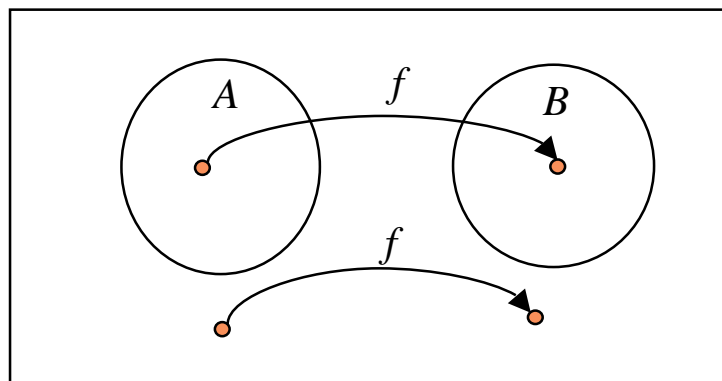
x	y	z	ϕ
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Cook-Levini teoreem:

$SAT \in P$ parajasti siis, kui $P = NP$.

Ülesannete redutseeritavus

Definitsioon Hulga $A \subseteq \Sigma^*$ lahenduvuse ülesanne on redutseeritav hulga $B \subseteq \Sigma^*$ lahenduvuse ülesandeks, kui leidub arvutatav funktsioon $f : \Sigma^* \rightarrow \Sigma^*$, nii et iga w korral kehtib

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


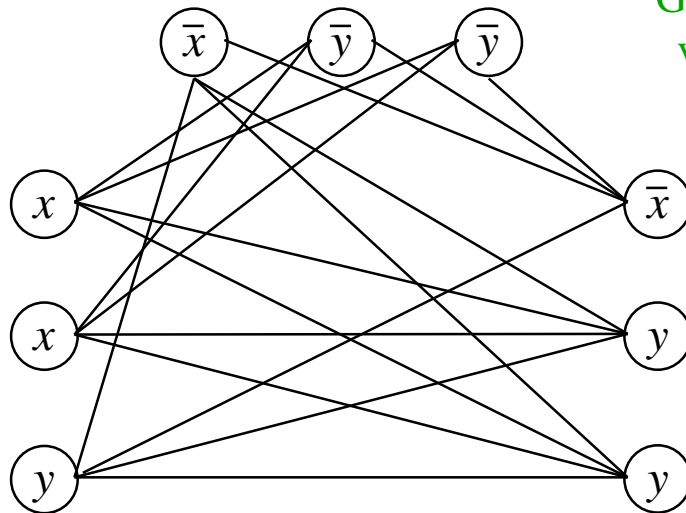
Definitsioon Kui redutseeritavuse funktsioon $f \in P$, siis A on polünoomiaalselt redutseeritav B-ks.

Teoreem Kui ülesanne A on polünoomiaalselt redutseeritav B-ks ning $B \in P$, siis $A \in P$

Ülesannete redutseeritavus (2)

Teoreem *SAT* on polünoomiaalselt redutseeritav ülesandeks *CLIQUE*.

$$\varphi = (x \vee x \vee y) \wedge (\bar{x} \vee \bar{y} \vee \bar{y}) \wedge (\bar{x} \vee y \vee y)$$



Graafis on kõik võimalikud kaared,
v.a. ühe diskunkti elementide vahel ning
vastandliteraalide vahel (näiteks x ja \bar{x} vahel)

NP täielikud ülesanded

Definitsioon Ülesanne B on NP täielik, kui ta rahuldab järgmisi tingimusi

- $B \in NP$
- iga NP ülesanne A on polünoomiaalselt redutseeritav B -ks

Teoreem Kui $B \in NP$ -täielik ja $B \in P$ siis $P = NP$.

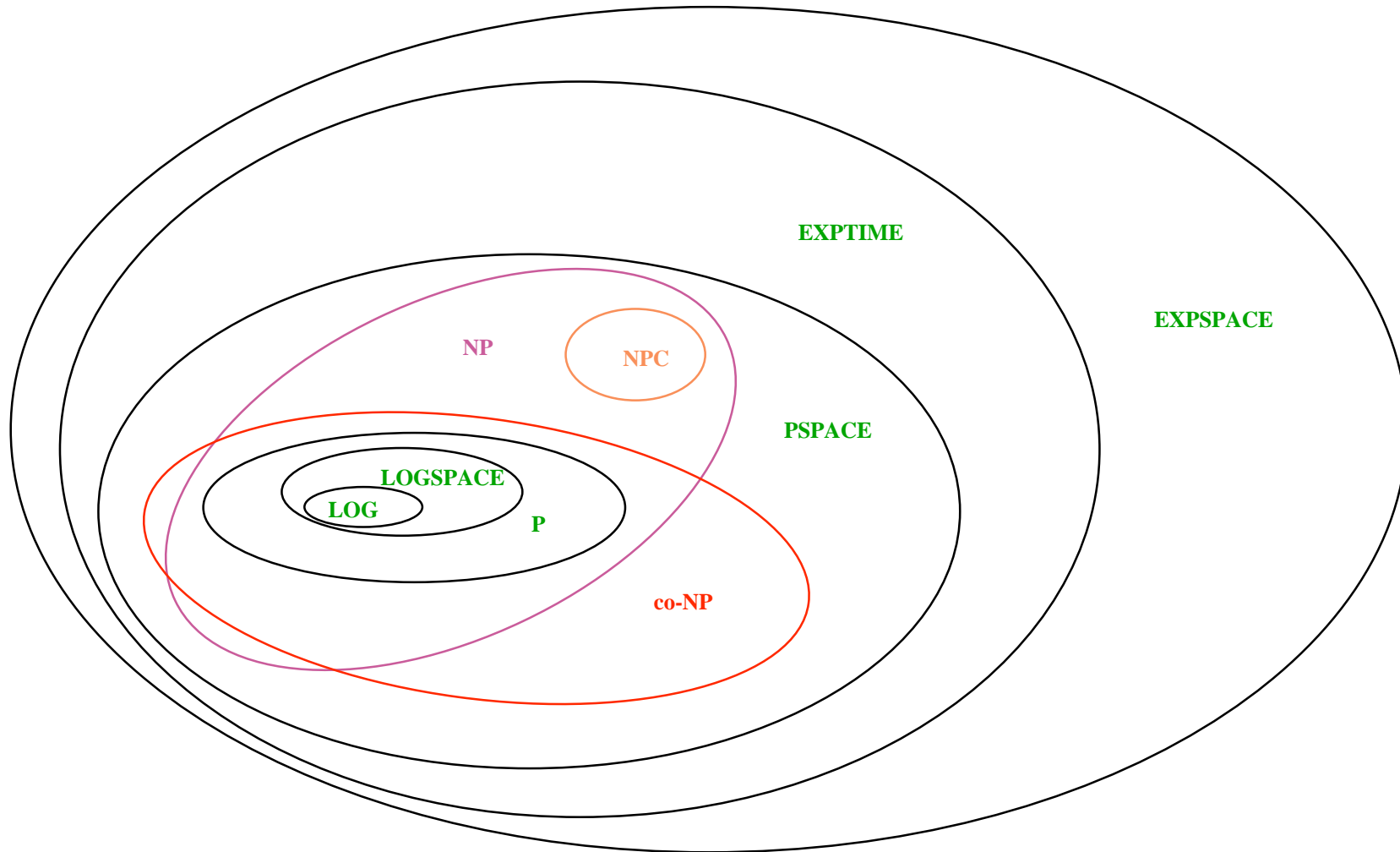
Teoreem Kui $B \in NP$ -täielik ja B on polünoomiaalselt redutseeritav C -ks, siis $C \in NP$ -täielik.

Teoreem $SAT \in NP$ -täielik.

Veel NP-täielikke ülesandeid:

- *HAMPATH*
- rändkaupmehe ülesanne
- *SUBSET-SUM*

Ülesannete keerukusklassid



Theoretical computer science

- Algorithmic information theory
- Computability theory
- Cryptography
- Formal semantics
- Theory of computation (or *theoretical computer science*)
 - ❑ analysis of algorithms and problem complexity
 - ❑ logics and meanings of programs
 - ❑ Mathematical logic and Formal languages
- Type theory