IN2080 Beregninger og kompleksitet

Løsningsforslag, eksamen våren 2022

Del 1: Automater (60 poeng)

Oppgave 1 La $\Sigma = \{a, b\}$ være alfabetet og la $|y|_x$ være antall forekomster av substrengen x i strengen y.

a) Lag en automat (DFA eller NFA) som gjenkjenner

1. a^*ba^*
2. $\{w \in \Sigma^* \mid |w|_a = 1 \text{ eller } |w|_b = 1\}$

b) Lag et regulært uttrykk som definerer

$$\{w \in \Sigma^* \mid \text{hvert tredje symbol i } w \text{ er en } b\}$$

Oppgave 2 La L være følgende språk over alfabetet Σ:

$$L = \{(ab)^n\, a\, (ab)^n \mid n \geq 0\}$$

Lag en PDA som gjenkjenner L.

Oppgave 3 Den kontekstfrie grammatikken G er gitt ved reglene

$$\begin{aligned}
&S \to XY \mid W \\
&X \to aXb \mid \varepsilon \\
&Y \to cY \mid \varepsilon \\
&W \to aWc \mid Z \\
&Z \to bZ \mid \varepsilon
\end{aligned}$$

a) Hvilke av følgende ord er i $\mathcal{L}(G)$: ε, ac, $aabc$, $aabbc$? Begrunn svaret.

b) Oversett G til Chomsky normalform.

c) Lag en PDA som gjenkjenner språket $\mathcal{L}(G)$.

d) Hvilke ord på formen $a^i\, b^j\, c^k$ kan genereres fra G? Begrunn svaret.

Oppgave 4 La $\Sigma = \{a, b, c\}$ være et alfabet. Språkene L_1, L_2 og $L_3 \subseteq \Sigma^*$ er gitt ved

- $L_1 = \{a^i\, b^j\, c^k \mid i + k = j\}$
- $L_2 = \{ww \mid w \in \Sigma^*\}$
- $L_3 = \{z \in \Sigma^* \mid b \text{ kan ikke forekomme før } a \text{ i } z\}$

Et av språkene L_1, L_2, L_3 er regulært, et er kontekstfritt men ikke regulært, og et er ikke kontekstfritt.

a) Hvilket språk er regulært? Begrunn svaret.

b) Hvilket språk er kontekstfritt men ikke regulært? Begrunn svaret.

c) Hvilket språk er ikke kontekstfritt? Begrunn svaret.

Del 2: Beregnbarhetsteori (10 poeng)

Språket

$$A_{\text{TM}} \;=\; \{\langle M, w\rangle \mid M \text{ er en TM og } M \text{ aksepterer } w\}$$

er kjent fra Sipsers bok. Språket A_{TM} er gjenkjennbart[1], men ikke avgjørbart[2].

Oppgave 5 La L_1 og L_2 være språk slik at $A_{\text{TM}} \leq_m L_1$ og $L_2 \leq_m A_{\text{TM}}$. Forklar hvorfor L_1 ikke er et avgjørbart språk. Forklar hvorfor L_2 er et gjenkjennbart språk.

Språket B_{TM} er gitt ved

$$\begin{aligned} B_{\text{TM}} \;=\; \{\langle M, w\rangle \mid\ & M \text{ er en TM og } M \text{ aksepterer } w \\ & \text{og } M \text{ aksepterer ingen input som er kortere enn } w\ \}\,. \end{aligned}$$

Oppgave 6 Vis at språket B_{TM} ikke er avgjørbart.

[1] Sipser bruker ordet *recognizable*
[2] Sipser bruker ordet *decidable*

Del 3: Kompleksitetsteori (30 poeng)

Intuitivt så kan en urettet graf *2-fargelegges* hvis det er mulig å fargelegge enhver node med en av to farger, la oss si rød eller blå, slik at to noder får forskjellig farge dersom en kant forbinder dem. Nedenfor finner du en mer formell definisjon av hva det betyr at en graf kan 2-fargelegges.

La G være en urettet graf, og la $n_1, n_2, \ldots, n_k$ være nodene i G. Vi sier at G kan *2-fargelegges* hvis det finnes to mengder av noder A og B slik at

1. $A \cup B = \{n_1, n_2, \ldots, n_k\}$
2. $A \cap B = \emptyset$
3. hvis $x, y \in A$, så finnes det ikke en kant i G som forbinder x og y
4. hvis $x, y \in B$, så finnes det ikke en kant i G som forbinder x og y.

Vi definerer språket *2FARG* ved

$$\mathit{2FARG} \;=\; \{\langle G\rangle \mid G \text{ er en urettet graf som kan 2-fargelegges}\} .$$

Oppgave 7 Vis at *2FARG* er i NP.

Oppgave 8 Er *2FARG* i SPACE(n)? Begrunn svaret. Er *2FARG* i NSPACE(n)? Begrunn svaret.

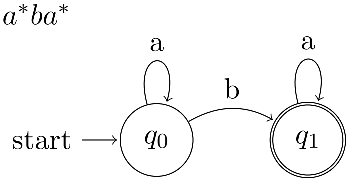
Oppgave 9 Vis at *2FARG* er i P.

Oppgave 10 Anta at NP $\neq$ P. Vis at *2FARG* ikke er NP-komplett.

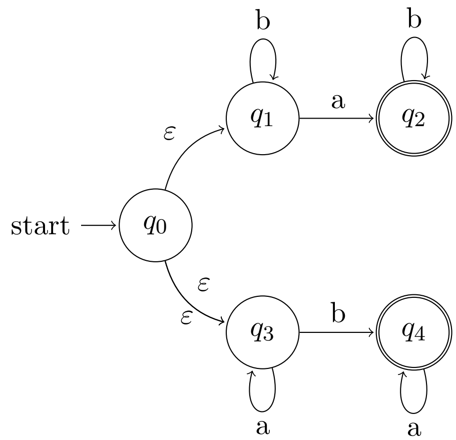
Løsningsforslag

Oppgave 1a

a^*ba^*



$\{w \in \Sigma^* \mid |w|_a = 1 \text{ eller } |w|_b = 1\}$

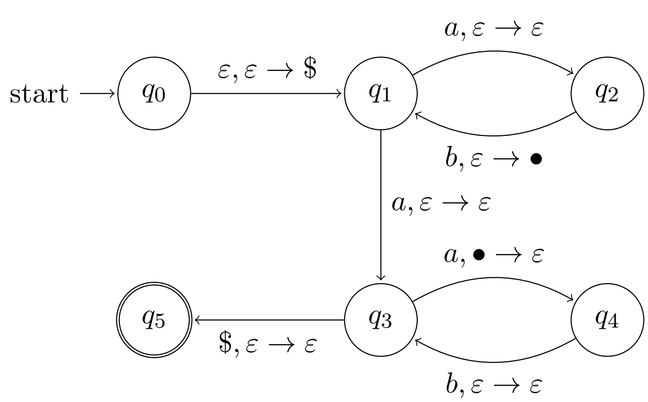


Oppgave 1b

$((a \cup b)\,(a \cup b)\,b)^* \cup$
$((a \cup b)\,(a \cup b)\,b)^*\,(a \cup b) \cup$
$((a \cup b)\,(a \cup b)\,b)^*\,(a \cup b)\,(a \cup b)$

Oppgave 2

La stack-alfabetet $\Gamma = \{\$, \bullet\}$.



Oppgave 3a

Følgende ord er i $\mathcal{L}(G)$. Begrunnelser kan gis ved derivasjoner i grammatikken.

- ε. Vi har $S \to W \to Z \to \varepsilon$
- ac. Vi har $S \to W \to aWc \to ac$
- $aabbc$. Vi har $S \to XY \to aXbY \to aaXbbY \to aaXbbc \to aabbc$

Følgende ord er *ikke* i $\mathcal{L}(G)$:

- $aabc$. Det fins ingen derivasjon for denne strengen. Det er lett å se, fordi alle derivasjoner enten har like mange a'er som b'er eller like mange a'er som c'er.

Oppgave 3b

Vi kan anvende algoritmen fra pensum, og får:

$$\begin{aligned}
&S \to XY \mid W\\
&X \to aXb \mid \varepsilon\\
&Y \to cY \mid \varepsilon\\
&W \to aWc \mid Z\\
&Z \to bZ \mid \varepsilon
\end{aligned}$$

Legg til ny startvariabel S_0:

$$\begin{aligned}
&\color{red}{S_0 \to S}\\
&S \to XY \mid W\\
&X \to aXb \mid \varepsilon\\
&Y \to cY \mid \varepsilon\\
&W \to aWc \mid Z\\
&Z \to bZ \mid \varepsilon
\end{aligned}$$

Fjern alle regler på formen $Z \to \varepsilon$:

$$\begin{aligned}
&S_0 \to S \mid \color{red}{\varepsilon}\\
&S \to X \mid XY \mid W\\
&X \to ab \mid aXb\\
&Y \to cY \mid c\\
&W \to ac \mid aWc \mid Z\\
&Z \to bZ \mid b
\end{aligned}$$

Del høyresiden med mer enn to elementer:

$$
\begin{aligned}
&S_0 \to S \mid \varepsilon \\
&S \to X \mid XY \mid W \\
&X \to ab \mid aX_1 \\
&X_1 \to Xb \\
&Y \to cY \mid c \\
&W \to ac \mid aW_1 \mid Z \\
&W_1 \to Wc \\
&Z \to bZ \mid b
\end{aligned}
$$

Lag produksjoner for konstantene:

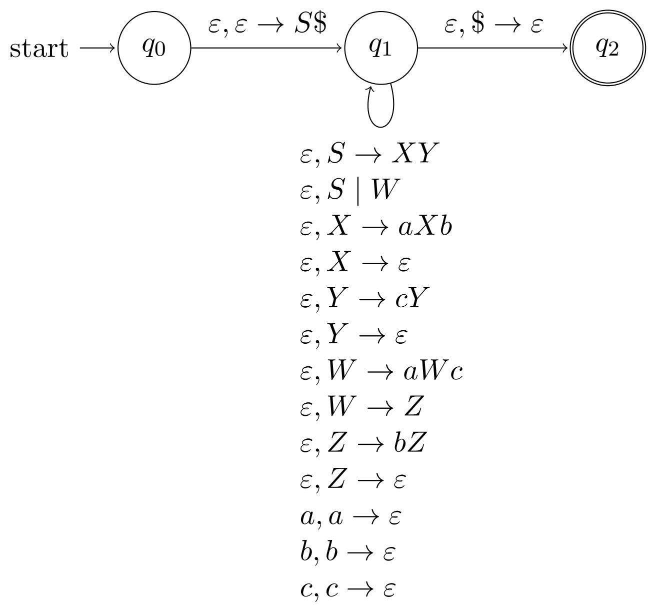
$$
\begin{aligned}
&S_0 \to S \mid \varepsilon \\
&S \to X \mid XY \mid W \\
&X \to AB \mid AX_1 \\
&X_1 \to XB \\
&Y \to CY \mid c \\
&W \to AC \mid AW_1 \mid Z \\
&W_1 \to WC \\
&Z \to BZ \mid b \\
&A \to a \\
&B \to b \\
&C \to c
\end{aligned}
$$

Fjern produksjoner med enkelt-variable på høyresiden:

$$
\begin{aligned}
&S_0 \to AB \mid AX_1 \mid XY \mid AC \mid AW_1 \mid BZ \mid b \mid \varepsilon \\
&X \to AB \mid AX_1 \\
&X_1 \to XB \\
&Y \to CY \mid c \\
&W \to AC \mid AW_1 \mid BZ \mid b \\
&W_1 \to WC \\
&Z \to BZ \mid b \\
&A \to a \\
&B \to b \\
&C \to c
\end{aligned}
$$

Oppgave 3c

Vi følger oppskriften fra forelesning:

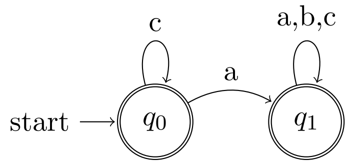


Oppgave 3d

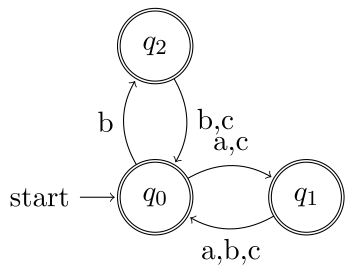
Grammatikken har to "hovedproduksjoner" fra S. I tilfelle 1 får vi XY, der x produserer et ord på formen $a^n b^n$ og Y produserer vilkårlig manger c'er. I tilfelle 2 får vi W som produserer ord på formen $a^n Z b^n$ der Z produserer vilkårlig mange b'er. Dermed ser vi at grammatikken produserer ord med enten like mange a'er og b'er eller like mange a'er og c'er. Dvs at $\mathcal{L}(G) = \{a^i\, b^j\, c^k \mid i = j \;\; or\; j = k\}$.

Oppgave 4a

L_3 er regulært. Vi kan vise dette ved å lage en DFA som gjenkjenner L_3.



Merk at en annen tolkning av oppgaven er at an b-transisjon aldri kan forekomme før en a-transisjon. Dette kan vises ved følgende DFA:



Oppgave 4b

L_1 er kontekstfritt. Dette ser vi ved å se på konkatineringen av språkene $L_1' = \{a^i b^i \mid i \geq 0\}$ og $L_1'' = \{b^j c^j \mid j \geq 0\}$. Disse språkene kan gjenkjennes ved PDAer som teller antall av første symbol. Opplagt at kontekstfrie språk er lukket under konkatinering (dvs. vi lar transisjonen $\varepsilon, \$ \to \varepsilon$ til aksepterende tilstand i automaten for L_1 gå til start-tilstanden for automaten for L_2 i stedet).

Vi viser at L_1 ikke er regulært ved å anta at vi kan bruke pumpelemmaet for regulære språk og få en motsigelse. Anta derfor at L_1 er regulært. Da fins pumpelengde n slik at alle ord w lengre enn n kan pumpes. La $w = a^n\, b^{2n}\, c^n$. Vi må dele opp w i tre deler x, y, z slik at $|y| > 0$, $|xy| \leq n$ og vise at $xy^i z \in L_1$ for alle i. Men da kan x og y bare bestå av a'er. Dermed har vi at $xy^0 z = a^m\, b^{2n}\, c^n$ der $m < n$, så $xy^0 z \notin L_1$. Vi får en motsigelse, og L_1 kan ikke være et regulært språk.

Oppgave 4c

L_2 er ikke kontekstfritt. Vi viser dette ved å anta at vi kan bruke pumpelemmaet for kontekstfrie språk og få en motsigelse. Anta derfor at L_2 er kontekstfritt. Da fins pumpelengde n slik at alle ord γ lengre enn n kan pumpes. La $w = a^n\, b^n$. Da er $\gamma = ww \in L_2$ og $|\gamma| = 4n > n$, så vi kan pumpe γ. Ved pumpelemmaet finnes det en inndeling $\gamma = uvxyz$ slik at $|vy| > 0$, $|vxy| \leq n$ og $uv^ixy^iz \in L_2$ for all i. La oss se på ordet uv^0xy^0z. Siden $uv^0xy^0z \in L_2$ fins det en w_1 slik at $uv^0xy^0z = w_1w_1$. Vi vet at u må starte med en a og z må slutte med en b. I så fall må w_1w_1 starte med en a og slutte med en b, og det samme gjelder for w_1.

La oss nå se på hvor vxy kan befinne seg i γ. Vi har tre tilfeller:

1. vxy er i første halvdel av γ. Men da vet vi at midten av γ har flyttet seg mot høyre i uv^0xy^0z. I så fall må w_1 slutte med en a. Motsigelse.
2. vxy er i siste halvdel av γ. Da har midten av γ har flyttet seg mot venstre i uv^0xy^0z, og w_1 må starte med en b. Motsigelse.
3. vxy krysser midten av delordet b^na^n, slik at v inneholder minst en b og y inneholder minst en a. I så fall vil det enten være færre a'er i andre halvdel enn i første halvdel av w_1w_1, eller det vil være færre b'er i første halvdel enn i andre halvdel av w_1w_1. Motsigelse.

Vi har vist at L_2 ikke kan pumpes, og dermed kan det ikke være kontekstfritt.

Oppgave 6

Vi viser at $A_{\text{TM}} \leq_m B_{\text{TM}}$.

Definisjonen av $\leq_m$ sier at $A_{\text{TM}} \leq_m B_{\text{TM}}$ hvis det finnes en beregnbar f slik at

$$\langle M, w\rangle \in A_{\text{TM}} \quad \Leftrightarrow \quad f(\langle M, w\rangle) \in B_{\text{TM}}\ . \tag{*}$$

Vi må lage en beregnbar f slik at (*) holder.

Vi lar $f(\langle M, w\rangle) = \langle M', w\rangle$ der M' er turingmaskinen gitt ved

- $M' =$ input x
  1. sjekk om $|x| < |w|$ (om x er kortere enn w)
  2. hvis $|x| < |w|$, så AVVIS (ellers fortsett)

3. kjør M på input x
4. AKSEPTER hvis M aksepterer x; AVVIS hvis M avviser x.

Vi har nå gitt f. Det er opplagt at f er en beregnbar funksjon. Vi skal nå argumentere for at f tilferdstiller (*).

Anta $\langle M, w\rangle \in A_{\mathrm{TM}}$ (vi viser $f(\langle M, w\rangle) = \langle M', w\rangle \in B_{\mathrm{TM}}$). Det betyr at M aksepterer w. Da vil også M' akseptere w (siden w ikke er kortere enn w). Videre så vil ikke M' akseptere noen input som er kortere enn w (alle slike input avvises i trinn 2). Dermed har vi $\langle M', w\rangle \in B_{\mathrm{TM}}$.

Anta $f(\langle M, w\rangle) = \langle M', w\rangle \in B_{\mathrm{TM}}$ (vi viser $\langle M, w\rangle \in A_{\mathrm{TM}}$). Det betyr at M' aksepterer w (det betyr også at at M' ikke aksepterer noen input kortere enn w, men det er irrelevant nå). Se på konstruksonen av M'. Da ser du at hvis M' aksepterer w så må det skyldes at M aksepterer w. Dermed konkulderer vi at M aksepterer w. Dermed har vi $\langle M, w\rangle \in A_{\mathrm{TM}}$.

Vi har nå vist at $A_{\mathrm{TM}} \leq_m B_{\mathrm{TM}}$. Det følger fra standard teoremer i læreboka at språket B_{TM} ikke er avgjørbart.

Oppgave 7

Vi skal argumentere for at *2FARG* kan gjenkjennes av en nondeterministisk turingmaskin som arbeider i polynom tid. (Vi kunne også ha argumentert for at *2FARG* kan gjenkjennes av en polynom-tid verifikator.)

Vi antar antar at input G er gitt på formen

$$n_1, n_2, \ldots, n_k \; (\alpha_1, \beta_1)\,(\alpha_2, \beta_2)\; \ldots \;(\alpha_\ell, \beta_\ell)$$

der $n_1, \ldots, n_k$ er nodene og $(\alpha_1, \beta_1), \ldots, (\alpha_\ell, \beta_\ell)$ er kantene (vi har altså $\alpha_i, \beta_i \in \{n_1, n_2, \ldots, n_k\}$).

En nondeterministisk turingmaskin kan gjenkjenne *2FARG* ved å utføre prosedyren nedenfor.

1. For enhver node $n_i \in \{n_1, \ldots, n_k\}$ utføres følgende
   (a) Velg nondeterministisk en farve $F \in \{0, 1\}$.
   (b) Merk samtlige kanter som som inneholder n_i med F. Hvis samme kant merkes både 0 og 1, så AVVIS.
2. Hvis trinn 1 terminerer, så AKSEPTER.

La n være lengden av input. Trinn 1 (b) kan utføres i tid $O(n^2)$. Det er maksimalt n noder i G. Dermed jobber turingmaskinen vi har skissert over i tid $O(n^3)$. Det er opplagt at turingmaskinen gjenkjenner *2FARG*. Dermed er *2FARG* i NP. (Med tanke på neste oppgave så bør man merke seg at turingmaskinen arbeider i rom $O(n)$. Den trenger ikke å bruke noe plass på teipen utover plassen som inneholder input.)

Oppgave 8

Ja, *2FARG* er i SPACE(n). En deterministisk turingmaskin M kan telle antall noder k i G, og den kan systematisk generere alle bitstrenger $b_0, b_1, \ldots b_m$ av lengde k (så $m = 2^k$). For $j = 1, \ldots, m$ kan M kjøre en deterministisk versjon av turingmaskinen over (se løsning til oppgave 7) på en kopi av input (som kan ødelegges). Denne deterministiske versjonen setter fargen F på noden n_i til bit nummer i i strengen b_j. Turingmaskinen M vil ikke trenge mer enn $2n + log_2 n + c$ teipseller der n er lengden av input og c er en konstant. Vi har $2n + log_2 n + c = O(n)$. (Det holder med $2n + c$ teipseller, men uasett blir det av orden $O(n)$.) Dermed er *2FARG* et språk i kompleksitetsklassen SPACE(n).

Det følger rett frem fra defininisjonene av SPACE(n) og NSPACE(n) at SPACE(n) $\subseteq$ NSPACE(n). Dermed besvarer vi det andre spørsmålet slik: Ja, *2FARG* er i NSPACE(n).

Oppgave 9

Vi gir en turingmaskinvennlig algoritme som avgjør *2FARG*. *Litt terminologi:* Vi sier at to noder α, β i G er *naboer* hvis G inneholder en kanten (α, β) eller kanten (β, α). En *hvit* node er en node som ikke har fått farge (hvit er ingen farge).

Algoritmen er gitt ved trinnene I, II og III nedfor. Input er en urettet graf G. Når algoritmen starter er alle noder i G hvite.

Trinn I: Finn en (vilkårlig) hvit node i G og farg den blå. Hvis G ikke inneholder en hvit node, så AKSEPTER. Gå trinn II.

Trinn II: For enhver hvit node α i G gjør følgende

1. løp gjennom α's naboer og finn ut om punkt 2, 3 eller 4 (nedenfor) skal utføres.

2. hvis minst en av α's naboer er røde og ingen er er blå, så farg α blå
3. hvis minst av α's naboer er blå og ingen er er røde, så farg α rød
4. hvis minst en av α's naboer er blå og minst en er rød, så AVVIS
5. (hvis alle α's naboer er hvite, så skal det ikke gjøres noe som helst)

Gå til trinn III.

Trinn III: Hvis en hvit node ble farget i tinn II, så gjenta trinn II. Hvis ingen hvite noder ble farget, så gjenta trinn I.

La n være lengden av input, dvs. $n = |G|$. Da inneholder G maksimalt n noder (og maksimalt n kanter). Trinn II vil bli utført maksimalt $2n$ ganger. Det ytre løkka i trinn II løper gjennom O(n) noder. Den indre løkka i trinn II (punkt 1) løper gjennom input (av lenge n) for å finne nabonoder til en gitt node. Dermed ser det jo ut som at algoritmen over kan utføres av en turingmaskin som arbeider i tid $O(n^3)$, og den kan helt sikkert utføres av en turingmaskin som arbeider i tid $O(n^4)$, og den kan helt helt sikkert utføres av en turingmaskin som arbeider i polynom tid. Algoritmen er deterministisk. Det betyr at *2FARG* er i P.

Kommentar: Denne deterministiske polynomtid-turingmaskinen arbeider i rom $O(n)$ (dette må man selvsagt argumentere for). Man kan altså besvare oppgave 7 og 8 ved å kun gi turingmaskinen i løsningsforslaget til oppgave 9. Hvis man gjør noe slikt bør man være sikker på at bevarelsen av oppgave 9 er korrekt. Hvor opplagt er det forøvrig at maskinen gjør det vi påstår den gjør, altså at den gjenkjenner *2FARG*? Krever ikke dette en begrunnelse? Eller er det greit nok å si at det er opplagt? (Jeg spørger kun, mit kald er ej at svare.)

Oppgave 10

Anta NP $\neq$ P. Anta *2FARG* er NP-komplett (vi viser at dette gir en selvmotsigelse)

Vi skal bruke følgende påstand (Th. 7.31 på side 303 i Sipsers bok):

$$A \leq_p B \;\text{ og }\; B \in \mathrm{P} \quad \Rightarrow \quad A \in \mathrm{P}\,. \qquad \text{(påstand)}$$

La L være et vilkårlig språk NP. Siden *2FARG* er NP-komplett, så vet vi at $L \leq_p$ *2FARG* (se definisjonen av et NP-komplett språk) Ved oppgave 9 har vi $\mathit{2FARG} \in \mathrm{P}$. Vi har altså at

$$L \leq_p \mathit{2FARG} \quad \text{og} \quad \mathit{2FARG} \in \mathrm{P}\ .$$

Nå kan vi bruke (påstand) og slutte at $L \in \mathrm{P}$. Dette viser at et vilkårlig språk i NP også er i P. Sagt med andre ord så har vi $\mathrm{NP} \subseteq \mathrm{P}$.

Det følger rett frem fra definisjonene av P og NP at $\mathrm{P} \subseteq \mathrm{NP}$.

Dermed har vi kommet frem til at $\mathrm{NP} \subseteq \mathrm{P}$ og $\mathrm{P} \subseteq \mathrm{NP}$, og det betyr jo at $\mathrm{NP} = \mathrm{P}$, noe som sier i mot vår antagelse om at $\mathrm{NP} \neq \mathrm{P}$.