

INF2080

Oblig 3

Elsie Mestl

April 28, 2016

Oppgave 1: P

Det å være medlem i P vil si at det finnes en deterministisk TM som avgjør problemet i polynom tid.

DNFSAT = $\{\phi \mid \phi \text{ is on DNF, and } \phi \text{ is satisfiable}\}$

La $\phi \in \text{DNFSAT}$ for at ϕ skal være sant så holder det at en av klausulene er sann. For å sjekke om ϕ kan gjøres sann holder det altså å gå gjennom hver klausul og sette inn verdier til litteralene helt til en klausul er sann. Lager TM M som løser for DNFSAT som beskrevet.

M på input $\langle \phi \rangle$:

1. for hver klausul k_i i ϕ gjør 2
2. for hver litteral l_j^i finn tilhørende variabel.
3. Hvis denne variabelen ikke har blitt tildelt en verdi velg sannhetsverdien som gjør l_j^i sann, eller variabelen har blitt tildelt en verdi, men denne verdien gjør l_j^i sann, gå til steg4. Ellers gå til steg5.
4. Hvis det ikke er fler litteraler i klausulen aksepter, ellers gå til steg2.
5. Er det siste klausulen avvis, ellers nullstill variablene og gå til steg1.

Her er det beskrevet en deterministisk turing masking og analyserer vi den ser vi den vil kjøre i $O(N)$, hvor N er det totale antallet litteraler i ϕ . Altså er DNFSAT i P.

DNFUNSAT = $\{\phi \mid \phi \text{ is on DNF, and } \phi \text{ is unsatisfiable}\}$

Dette er den konjugerte av DNFSAT. Vi kan altså bruke akkurat samme TM som for DNFSAT men hvor vi reverserer output. Og siden DNFSAT er i P, og det å endre aksepter til avvis og avvis til aksepter er i $O(1)$ så er TM til DNFUNSAT også i P.

$$\text{CNF-TAUT} = \{\phi \mid \phi \text{ is on CNF, and } \phi \text{ is a tautology}\}$$

For at et uttrykk på CNF skal være en tautologi må hver klausul inneholde minst en litteral hvor komplementet til variabelen også er en litteral i samme klausul.

Lager vi en turingmaskin som beskrevet under

For hver klausul lager vi to mengder “invers” og “ikke invers”. For hver litteral l_i^j i klausul j henter vi ut variabelen v . Hvis $\bar{v} = l_i^j$ settes den i mengden “invers” ellers settes v i “ikke invers”. Når klausulen er ferdig oppdelt som beskrevet over går man gjennom og ser om det er en variabel som finnes i begge mengdene. Er det så vil klausulen alltid være sann og man går videre til neste klausul og gjør det samme. Hvis ikke avvis, for da er det mulig å velge en tilordning som gjør klausulen usann, og dermed også resten av uttrykket usann. Hvis man kommer gjennom alle klausulene og finner minst et element i hver av mengdene “invers” og “ikke invers” for alle klausulene så aksepterer man.

Denne TM vil kjøre i $O(N^2)$ på en deterministisk turingmaskin og CNF-TAUT ligger altså i P.

Oppgave 2: NP-komplett

For at et problem A skal være NP-komplett må følgende to være oppfylt:

1. $A \in \text{NP}$
2. $B \leq_p A$, for alle $B \in \text{NP}$

Men her kan steg2 erstattes med å vise at $B \leq_p A$ og $B \in \text{NP-komplett}$.

$$\text{CNF-SAT} = \{\phi \mid \phi \text{ is on CNF, and } \phi \text{ is satisfiable}\}$$

Viser først at CNF-SAT er med i NP La N være en TM som bestemmer SAT og la M være gitt ved følgende:

M på input $\langle w \rangle$:

1. Kjør N på ϕ .
2. Hvis N aksepterer (dvs det finnes en tilegning av verdier til litteralene som gjør ϕ sann) aksepter. Ellers avvis.

Siden CNF-SAT er polynomisk reduserbar til SAT, som er i NP, så er CNF-SAT i NP.

For å vise at CNF-SAT er NP-komplett holder det å vise at et anntet NP-komplett problem kan reduseres i polynom tid til CNF-SAT.

Vi har at 3-SAT er NP-komplett.

Kan skrive en TM S for 3-SAT som tar inn et input ϕ . Sjekker først om input er på lovlig format, hvis nei avvis ellers fortsett. Kjører så M på ϕ hvis M aksepterer finnes en valuasjon av litteralene som gjør uttrykket sant, og vi aksepterer. Aviser M, avis.

Oppgave 3: coNP-komplett

For at et problem A skal være NP-komplett må følgende to være oppfylt:

1. $A \in \text{coNP}$
2. $B \leq_p A$, for alle $B \in \text{coNP}$

Men her kan steg2 erstattes med å vise at $B \leq_p A$ og $B \in \text{coNP-komplett}$. Eller at hvis $\neg A \in \text{NP-komplett}$ så er $A \in \text{coNP-komplett}$ ¹.

$$\text{CNFUNSAT} = \{\phi \mid \phi \text{ is on CNF, and } \phi \text{ is unsatisfiable}\}$$

Siden CNFSAT $\in \text{NP-komplett}$ må $\overline{\text{CNFSAT}} \in \text{coNP-komplett}$. Og vi vet at $\overline{\text{CNFSAT}} = \text{CNFUNSAT}$ altså er CNFUNSAT $\in \text{coNP-komplett}$.

$$\text{DNFTAUT} = \{\phi \mid \phi \text{ is on DNF, and } \phi \text{ is a tautology}\}$$

La ϕ være gitt ved:

$$\begin{aligned} \phi &= (l_1^1 \wedge l_2^1 \wedge \dots \wedge l_{k_1}^1) \vee \dots \vee (l_1^n \wedge l_2^n \wedge \dots \wedge l_{k_n}^n) = \overline{\overline{(l_1^1 \wedge l_2^1 \wedge \dots \wedge l_{k_1}^1) \vee \dots \vee (l_1^n \wedge l_2^n \wedge \dots \wedge l_{k_n}^n)}} \\ &= \overline{(\neg l_1^1 \vee \neg l_2^1 \vee \dots \vee \neg l_{k_1}^1) \wedge \dots \wedge (\neg l_1^n \vee \neg l_2^n \vee \dots \vee \neg l_{k_n}^n)} \end{aligned}$$

Hvis vi kan vise at det under første negasjon alltid er usant så vil selve uttrykket alltid være sant (siden det er negert en gang til).

Vi ser at det som er under negasjonen er et uttrykk på CNF, og hvis dette uttrykket alltid skal være usant så blir det en oppgave for CNFUNSAT. Vi lager altså en maskin som negerer uttrykket, kjører det negerte uttrykket på CNFUNSAT. Aksepterer hvis CNFUNSAT avviser og avviser hvis CNFUNSAT aksepterer.

Dette viser at DNFTAUT er i coNP. For å vise at DNFTAUT er coNP-komplett så må vi vise at CNFUNSAT kan reduseres til DNFTAUT. Men det gjøres på akkurat samme måte som for å vise at DNFTAUT kan reduseres til CNFUNSAT. Neger uttrykket en gang, og kjør negasjonen på DNFTAUT og neg utfallet som maskinen gir.

DNFTAUT er altså i coNP-komplett

Oppgave 4

Et språk i P kan ikke være NP-komplett

Proof. Bevis ved selvmotsigelse:

Anta at $A \in P$ og at $A \in \text{NP-komplett}$. Altså kan alle $B \in \text{NP}$ reduseres til A. Men da er $B \in P$ Men da er $\text{NP} = P$ noe som ikke stemmer. Altså må bevisantagelsen være feil og vi har at et språk kan ikke både være i P og i NP-komplett samtidig.

□

Et språk kan ikke være både NP-komplett og coNP-komplett

Proof. Bevis ved selvmotsigelse:

Anta for selvmotsigelsens skyld at det finnes et språk A som er både NP-komplett og coNP-komplett. Dvs at $A \in \text{NP}$ og $B \leq_p A$ for alle språk $B \in \text{NP}$, og at $A \in \text{coNP}$ og $C \leq_p A$ for alle språk $C \in \text{coNP}$.

Siden $B \leq_p A$, hvor A er i coNP så er B i coNP. Det samme gjelder for at C er i NP. Men da er nødvendigvis $\text{NP} = \text{coNP}$, noe som ikke er sant. Altså er antagelsen feil, og A kan ikke både være NP-komplett og coNP-komplett.

□

Kilder:

[1] University of Cambridge, Complexity Theory (2009-2010):

<https://www.cl.cam.ac.uk/teaching/0910/Complexity/lecture9.pdf> Lastet ned 28.04.2016