Laskennan vaativuus **Ratkaisuehdotuksia**
Harjoitus 2, 27.3.2007

1. **Yhdiste:**

   Olkoot A ja B luokan **NP** kieliä. Niille on siis olemassa epädeterministiset Turingin koneet M_A ja M_B, joiden aikafunktiot f_A ja f_B ovat polynomisia. Voidaan muodostaa NTM $M_{A\cup B}$, joka syötteellä x valitsee epädeterministisesti, tutkiiko se $M_A(x)$ vai $M_B(x)$. Jos M_A tai M_B päätyy hyväksyvään tilaan, $M_{A\cup B}$ hyväksyy syötteen. Aikaa kone $M(x)$ kuluttaa nyt $\mathcal{O}(\max\{f_A(|x|), f_B(|x|)\}) \in \mathbf{NTIME}(\mathbf{P})$, ja siten $A \cup B \in \mathbf{NP}$.

   Peräkkäinen yhdistäminen:

   Tässä tapauksessa katenoituja kieliä tunnistava kone M_{AB} voi epädeterministisesti jakaa syötteensä x lineaarisessa ajassa kahtia osiksi y ja z. Seuraavaksi kone voi tutkia, päteekö $M_A(y)$ ja $M_B(z)$. Jos molemmat koneet hyväksyvät syötteensä, kone M_{AB} hyväksyy syötteen, muuten hylkää sen. Selvästi M_{AB} toimii polynomisessa ajassa, joten $AB \in \mathbf{NP}$.

2. Alla on algoritmi, joka tutkii, onko verkossa (V, E) kolmiota:

   for $a \in V$ **do**
       for $b \in V \setminus \{a\}$ **do**
           for $c \in V \setminus \{a, b\}$ **do**
               if $(a, b) \in E \wedge (a, c) \in E \wedge (b, c) \in E$ **then** **return** *Hyväksy*
   return *Hylkää*

   Tästä nähdään, että jopa huonoilla tietorakenteiden valinnoilla ongelma voidaan ratkaista ajassa $\mathcal{O}(|V|^3 \cdot 3|E|) \leq \mathcal{O}(|V|^5)$, mikä säilyy polynomisena.

3. Aloitetaan tutkimalla, miksi tyhjä kieli $L_\emptyset$ ja kieli $L_{\Sigma^*} = \{x | x \in \Sigma^*\}$ eivät voi olla **NP**-täydellisiä. On olemassa SAT:n hyväksymiä kaavoja ja merkkijonoja jotka eivät kuulu SAT:n hyväksymään kieleen. Palautus SAT $\leq_\mathbf{P} L_\emptyset$ ei ole mahdollista, sillä ei voida muodostaa sellaista palautusfunktiota f, jolla SAT:iin kuuluvat tai kuulumattomat merkkijonot φ voisi erottaa toisistaan testin $f(\varphi) \overset{?}{\in} L_\emptyset$ perusteella. Vastaava ongelma tulee vastaan, jos yritetään muodostaa palautus SAT $\leq_\mathbf{P} L_{\Sigma^*}$.

   Oletetaan nyt, että $\mathbf{P} = \mathbf{NP}$, $A \in \mathbf{P} \setminus \{L_\emptyset, L_{\Sigma^*}\}$, $B \in \mathbf{NP}$ ja M_B on kielen B hyväksyvä TM. Koska A on epätyhjä, on olemassa jokin $a \in A$. Toisaalta koska $A \neq L_{\Sigma^*}$, on olemassa $a' \notin A$. Nyt voidaan muodostaa sellainen palautusfunktio f, jolla syötteellä x

   $$f(x) = \begin{cases} a, & M_B(x) \text{ hyväksyy,} \\ a', & M_B(x) \text{ hylkää.} \end{cases}$$

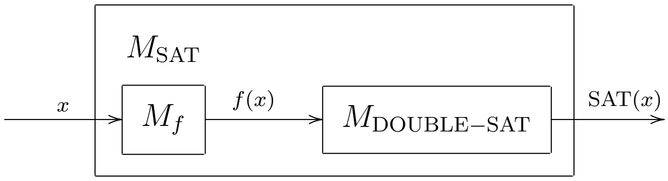
   Kone M_B toimii polynomisessa ajassa ja a ja a' ovat kooltaan polynomisia, joten funktio f on polynominen. Näin muodostetulla funktiolla pätee $x \in B \iff f(x) \in A$, joten se tosiaan soveltuu palautusfunktioksi. Koska $B \leq_\mathbf{P} A$ ja $B \in \mathbf{NP}$ on mielievaltainen, niin A on **NP**-täydellinen.

4. Kaava ei ole toteutuva. Ratkaisu nähdään esimerkiksi totuustaululla:

x	y	$x \vee y$	$x \vee \overline{y}$	$\overline{x} \vee y$	$\overline{x} \vee \overline{y}$	$(x \vee y) \wedge (x \vee \overline{y}) \wedge (\overline{x} \vee y) \wedge (\overline{x} \vee \overline{y})$
0	0	0	1	1	1	0
0	1	1	0	1	1	0
1	0	1	1	0	1	0
1	1	1	1	1	0	0

5. On helppo huomata, että DOUBLE-SAT $\in$ **NP**. Keksitään vain kaksi eri sijoitusta (sertifikaattia) annetulle kaavalle polynomisessa ajassa, ja edelleen tarkistetaan polynomisessa ajassa että kaava toteutuu näillä sijoituksilla. Keskitytäänkin tutkimaan polynomista palautusta.

   Tässä ratkaisussa käytetään hyväksi SAT:n **NP**-täydellisyyttä, ja osoitetaan, että SAT $\leq_{\mathbf{P}}$ DOUBLE-SAT. SAT:n ratkaisevalle koneelle M_{SAT} annettu syöte x muunnetaan polynomisella funktiolla f DOUBLE-SAT:n koneelle $M_{\text{DOUBLE-SAT}}$ ratkaistavaksi soveltuvaan muotoon. Todistuksen idean voi havainnollistaa kuvana:



   Syötteenä voidaan saada mitä tahansa, mutta jos hyväksymme esimerkiksi vain CNF-muotoisia kaavoja, niin tarkistuksen voi suorittaa polynomisessa ajassa. Olkoon siis $\varphi(x_0, x_1, \ldots, x_{n-1})$ syötteenä saatu kaava, jonka muuttujat ovat $x_0, \ldots, x_{n-1}$. Jos muuttujille on olemassa jokin sellainen sijoitus s, jolla φ toteutuu, niin $\varphi \in$ SAT. DOUBLE-SAT sisältää puolestaan kaavat, joilla on olemassa sellaiset sijoitukset s ja s' ($s \neq s'$), että kaava toteutuu. Jotta soveltuvia sijoituksia olisi aina joko ei yhtään tai vähintään kaksi, muodostetaan seuraavanlainen palautusfunktio f. Valitaan jokin vapaana oleva muuttuja x_n, joka ei esiinny kaavassa φ, ja muodostetaan uusi kaava $\varphi' = f(\varphi) = \varphi \wedge (x_n \vee \overline{x}_n)$ laajentamalla φ:tä tautologialla. Nyt kaava φ toteutuu sijoituksella s silloin ja vain silloin kun φ' toteutuu sijoituksilla $s(0/x_n)$ ja $s(1/x_n)$. Toisin sanoen $\varphi \in \text{SAT} \iff f(\varphi) \in \text{DOUBLE-SAT}$.

   Muuttujan x_n löytäminen ja kaavan φ' muodostaminen onnistuvat varmasti polynomisessa ajassa, sillä kaavassa esiintyvien eri muuttujien määrää rajoittaa kaavan φ merkkijonoesityksen pituus. Jos $\varphi' \in$ DOUBLE-SAT, niin on olemassa kaavan φ' toteuttavat sijoitukset $s(0/x_n)$ ja $s(1/x_n)$. Sijoituksen $s(0/x_n)$ olemassa olosta seuraa puolestaan, että kaava φ toteutuu sijoituksella s ja $\varphi \in$ SAT. Näin nähdään, että DOUBLE-SAT on vähintään yhtä vaikea ongelma kuin SAT. Koska DOUBLE-SAT $\in$ **NP** ja SAT on **NP**-täydellinen, täytyy DOUBLE-SAT:n olla myös **NP**-täydellinen.