TKT20005 Laskennan mallit Viikko7

Tehtävä 1 Esimerkkejä NP-täydellisistä ongelmista.

Tässä tehtävässä tutustutaan muutamaan esimerkkiin NP-täydellisistä ongelmista ja opetellaan määrittelemään ongelmia formaaleina kielinä.
Ositus (partition), *laatikonpakkaus* (bin packing) ja *verkon värittäminen* (graph coloring) ovat kolme tunnettua NP-täydellistä ongelmaa. Etsi nämä ongelmat oppikirjoista tai verkosta ja selitä, mistä niissä on kysymys. Sopiva vastaus on lyhyt sanallinen kuvaus ongelmalle ja sen täsmällinen esitys formaalina kielenä.

(1) **Ositus (Partition) https://en.wikipedia.org/wiki/Partition_problem**
Annettuna monijoukko kokonaislukuja $(a_1, \dots, a_n)$. Onko se jaettavissa kahteen osaan, joiden summat ovat samat?
Kielenä:

$$\text{PARTITION} = \Big\{ \langle a_1, \dots, a_n \rangle \;\Big|\; \exists I \subseteq \{1, \dots, n\} : \sum_{i \in I} a_i = \sum_{i \notin I} a_i \Big\}.$$

(2) **Laatikonpakkaus (Bin Packing) https://en.wikipedia.org/wiki/Bin_packing_problem**
Annettuna esineiden koot $s_1, \dots, s_n$, laatikon kapasiteetti B ja laatikkomäärä k. Voidaanko kaikki esineet jakaa enintään k laatikkoon siten, että kunkin laatikon kokonaiskoko $\leq B$?
Kielenä:

$$\text{BINPACK} = \Big\{ \langle s_1, \dots, s_n, B, k \rangle \;\Big|\; \exists f : \{1, \dots, n\} \to \{1, \dots, k\} \text{ siten, että } \forall j \in \{1, \dots, k\} : \sum_{i:\, f(i)=j} s_i \leq B \Big\}$$

(3) **Verkon värittäminen (3-väritys) https://en.wikipedia.org/wiki/Graph_coloring**
Annettuna suunnaton verkko $G = (V, E)$. Onko solmut väritettävissä kolmella värillä niin, että mikään kaaren päätepistepari ei saa samaa väriä?
Kielenä:

$$\text{3COLOR} = \Big\{ \langle G \rangle \;\Big|\; \exists c : V \to \{1, 2, 3\} \text{ siten, että } \forall \{u, v\} \in E : \; c(u) \neq c(v) \Big\}.$$

Tehtävä 2 Luokan NP ongelmat.

Tässä tehtävässä harjoitellaan, kuinka todistetaan, että jokin ongelma kuuluu luokkaan NP. *Repunpakkausongelmassa* (knapsack problem) on annettu joukko esineitä I, jokaiselle esineelle $i \in I$ sen arvo $v_i \in \mathbb{N}$ ja paino $w_i \in \mathbb{N}$, repun painoraja $W \in \mathbb{N}$ ja tavoiteltu kokonaisarvo $V \in \mathbb{N}$. Tehtävänä on ratkaista, onko mahdollista valita joukko $J \subseteq I$ esineitä reppuun pakattavaksi niin, että niiden yhteenlaskettu arvo on ainakin V, mutta painoraja W ei ylity. Formaalina kielenä tämän voi määritellä

$$\textit{KNAPSACK} = \{ \langle I, v, w, V, W \rangle \mid \text{jollain } J \subseteq I \text{ pätee } \textstyle\sum_{j \in J} w_j \leq W \text{ ja } \sum_{j \in J} v_j \geq V \}.$$

1. Osoita, että kieli *KNAPSACK* kuuluu luokkaan NP. Sopiva vastauksen tarkkuustaso on joko polynomisessa ajassa toimivan epädeterministisen Turingin koneen tai polynomisen tarkastajan kuvaus samaan tapaan kuin kurssimateriaalissa esim. *CLIQUE*-ongelmalle (luennot s. 259; Sipser s. 296).

2. **Vapaaehtoinen lisätehtävä pohdittavaksi:** Repunpakkausongelmalle tunnetaan ajassa $O(nW)$ toimiva algoritmi, missä $n = |I|$ on esineiden lukumäärä. Minkä takia tästä ei seuraa *KNAPSACK* $\in$ P?

Lisätieto: KNAPSACK on tunnettu NP-täydellinen ongelma.

https://en.wikipedia.org/wiki/Knapsack_problem

Given a set of items, each with a weight and a value, determine which items to include in the collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

(1) **KNAPSACK** $\in$ NP**.** Käytetään binäärikoodausta; instanssi $x = \langle (v_1, w_1), \ldots, (v_n, w_n), V, W \rangle$

Sertifikaatti: bittijono $b \in \{0,1\}^n$, jossa $b_i = 1$ iff esine i valitaan.

Verifioija: laskee

$$S_w = \sum_{i=1}^{n} b_i w_i, \qquad S_v = \sum_{i=1}^{n} b_i v_i$$

ja **hyväksyy** iff $S_w \leq W$ ja $S_v \geq V$.

Koska summat ja vertailut binääriluvuilla vievät aikaa poly$(|x|)$, tarkastus on polynominen.

Siis *KNAPSACK* $\in$ NP

Tehtävä 3 NP**-täydellisyys ja** P **vs** NP **-ongelma.**

Tässä tehtävässä tarkastellaan, miten NP-täydellisyys ja P vs NP -ongelma liittyvät toisiinsa. Kuten muistetaan, Hamiltonin polku -ongelma *HAMPATH* on tunnettu NP-täydellinen ongelma.

1. Oletetaan, että joku löytää polynomisessa ajassa toimivan ratkaisualgoritmin Hamiltonin polku -ongelmalle. Mitä tästä seuraa P vs. NP -ongelmalle?

2. Entä jos joku todistaa, että Hamiltonin polku -ongelmaa ei ole mahdollista ratkaista polynomisessa ajassa? Mitä tästä seuraa P vs. NP -ongelmalle?

3. Oletetaan, että joku todistaa, että P $=$ NP. Mitä tästä seuraa Hamiltonin polku -ongelmalle?

4. Entä jos joku todistaa, että P $\neq$ NP? Mitä tästä seuraa Hamiltonin polku -ongelmalle?

Kaikissa kohdassa perustele vastauksesi täsmällisesti lähtien luokan NP ja NP-täydellisyyden määritelmistä ja tiedosta, että *HAMPATH* on NP-täydellinen. Siis tarkoitus ei ole käyttää suoraan esim. lausetta 5.27, vaan esittää sen taustalla oleva päättely tähän tilanteeseen sovellettuna.

(1) **Jos** ***HAMPATH*** **ratkaistaan polynomiajassa, niin** $P = NP$
Koska *HAMPATH* on NP-täydellinen, pätee (i) *HAMPATH* $\in NP$ ja (ii) jokaiselle $L \in NP$ on polynominen monikoinen reduktio $L \leq_m^p$ *HAMPATH*. Oletuksesta *HAMPATH* $\in P$ ja (ii):sta seuraa, että *kaikki* $L \in NP$ ovat myös luokassa P (komponeeraamalla reduktion ja *HAMPATH*:n poli-algoritmin). Siis $NP \subseteq P$. Toisaalta aina $P \subseteq NP$, joten $P = NP$.

(2) **Jos** ***HAMPATH*****:lle ei ole polynomista algoritmia, niin** $P \neq NP$**.**
Koska *HAMPATH* $\in NP$ (NP-täydellisyyden ehto (i)), väite “ei ole polynomista algoritmia” tarkoittaa *HAMPATH* $\notin P$. Tällöin on olemassa NP-kieli, joka ei kuulu P:hen, joten $P \neq NP$ (koska aina $P \subseteq NP$).

(3) **Jos** $P = NP$**, niin** ***HAMPATH*** $\in P$**.**
NP-täydellisyyden ehto (i): *HAMPATH* $\in NP$. Yhtälöstä $P = NP$ seuraa suoraan, että *HAMPATH* $\in P$.

(4) **Jos** $P \neq NP$**, niin** ***HAMPATH*** $\notin P$**.**
Oletetaan vastaoletuksena *HAMPATH* $\in P$. Koska ehto (ii) sanoo, että jokainen $L \in NP$ redusoituu polynomiajassa *HAMPATH*:iin, tästä seuraisi $NP \subseteq P$. Yhdessä triviaalin $P \subseteq NP$ kanssa saataisiin $P = NP$, mikä on ristiriidassa oletuksen $P \neq NP$ kanssa. Siis *HAMPATH* $\notin P$.

Tehtävä 4 Polynomisen reduktion transitiivisuus.

Luentomateriaalissa lauseen 5.29 todistuksessa käytettiin tässä tehtävässä todistettavaa lemmaa: jos $A \leq_P B$ ja $B \leq_P C$, niin $A \leq_P C$. Tämä tehtävä siis täydentää luentomateriaalia tältä osin.
Oletetaan, että funktiot $f\colon \Sigma^* \to \Sigma^*$ ja $g\colon \Sigma^* \to \Sigma^*$ voidaan laskea polynomisessa ajassa. Osoita, että myös yhdistetty funktio $h\colon \Sigma^* \to \Sigma^*$, missä $h(x) = g(f(x))$, voidaan laskea polynomisessa ajassa. Päättele tästä edelleen, että jos $A \leq_P B$ ja $B \leq_P C$, niin $A \leq_P C$.
Huom. Muista, että aikavaativuus määritellään syötteen koon suhteen. Ota huomioon, minkä kokoisilla syötteillä funktion g arvoa lasketaan.

lemma
olkoon $f, g : \Sigma^* \to \Sigma^*$ polynomiajassa laskettavia. Tällöin myös $h(x) = g(f(x))$ on polynomiajassa laskettava.

Todistus. On olemassa polynomit p, s siten, että f voidaan laskea ajassa $p(n)$ syötteen pituudella n, ja g voidaan laskea ajassa $s(m)$ syötteen pituudella m.

Kun lasketaan $h(x)$ syötteellä x, tehdään ensin $f(x)$ ajassa $p(|x|)$. Samalla saadaan $|f(x)| \leq p(|x|)$, koska yhden merkin kirjoituskin vie yhden askeleen.
Seuraavaksi ajetaan g syötteellä $f(x)$; tämän aika on $s(|f(x)|) \leq s(p(|x|))$. Kokonaisaika on siis

$$T_h(|x|) \ \leq \ p(|x|) \ + \ s(p(|x|))$$

mikä on polynomi koska polynomin sijoitus polynomiin on polynomi

Siis h on polynomiajassa laskettava. $\square$

Seuraus. Jos $A \leq_P B$ funktiolla f ja $B \leq_P C$ funktiolla g, määritä $h = g \circ f$. Lemmasta h on polynominen Lisäksi kaikilla x:

$$x \in A \iff f(x) \in B \iff g(f(x)) \in C \iff h(x) \in C$$

Siksi $A \leq_P C$ $\square$

Huom $\Rightarrow$ aikavaativuus määritellään syötteen koon suhteen:

g:n aika riippuu sen oman syötteen pituudesta $|f(x)|$, joka on polynomisesti sidottu $|x|$:ään, koska f on polynominen ja ei voi tulostaa enempää symbooleja kuin mitä sen laskenta askeleita on