Algoritmiteorian jatkokurssi, kevät 1999

Harjoitus 1: ratkaisuja (KP teht. 2–6, PO täydenn. & teht. 1)

1. Äärelliset automaatit ovat oleellisesti Turingin koneita ilman työnauhaa. Tarkemmin sanoen: annettu äärellinen automaatti voidaan helposti muuttaa Turingin koneeksi, joka käyttää työnauhaltaan vain aloituspaikan (jos nauhapään on konemallissa mahdollista pysyä siirtymän yhteydessä paikallaan), tai aloituspaikan ja sen viereisen paikan (jos nauhapäätä on pakko siirtää, sitä voidaan siirrellä edestakaisin näiden kahden paikan välillä). Jos säännöllisten kielten luokkaa merkitään REG, on siis REG $\subseteq$ DSPACE(1) tai REG $\subseteq$ DSPACE(2) valitun Turingin kone -mallin yksityiskohtien mukaan.

   Äärellisestä automaatista muokattu Turingin kone tekee n merkin mittaisella syötteellä $n+1$ siirtymää ennen pysähtymistään ($(n+1)$:s siirtymä tarvitaan loppumerkin havaitsemiseen). Aikavaativuuden suhteen on siis voimassa REG $\subseteq$ DTIME$(n+1)$.

   Lisätietoja: Itse asiassa voidaan osoittaa, että DSPACE(k) = REG kaikilla vakioilla $k \geq 1$: äärellinen automaatti voi pitää sisäisessä tilassaan kirjaa simuloimansa Turingin koneen työnauhan sisällöstä, jos nauha on rajoitetun pituinen, ja lisäksi tiedetään että mahdollisuus siirtää syötenauhapäätä myös vasemmalle ei lisää äärellisten automaattien kielentunnistusvoimaa. Mielenkiintoisempi tulos on, että DSPACE(s) = REG peräti kaikilla $s(n) = o(\log\log n)$. (Luentomuistiinpanojen Lause 1.8.)

   Sen sijaan luokka DTIME$(n+1)$ (ns. "reaaliaikakielet") sisältää muitakin kuin säännöllisiä kieliä: esimerkiksi palindromikieli $\{wcw^R \mid w \in \{a,b\}^*\}$ on helppo tunnistaa ajassa $n+1$, mutta on tunnetusti ei-säännöllinen.

2. Oletetaan, että luettelo $M_\lambda, M_0, M_1, M_{00}, M_{01}, \ldots$ sisältää kaikki *moninauhaiset* Turingin koneet.[1]

   Määritellään Turingin koneen M toiminta syötteellä c seuraavasti:

   – M laskee funktion arvon $t(|c|)$ ja asettaa aikarajoituslaskurille arvon $t(|c|)+1$.

   – M simuloi koneen M_c toimintaa syötteellä c vähentäen jokaisella laskenta-askelella aikarajoituslaskuria yhdellä. Jos M_c pysähtyy hyväksyvässä lopputilassa, M hylkää c:n. Jos M_c pysähtyy hylkäävässä lopputilassa tai aikalaskuri nollautuu, M hyväksyy c:n.

   Näin määritelty Turingin kone M tunnistaa kielen D^t. Koska M pysähtyy kaikilla syötteillä, se on totaalinen. Näin ollen D^t on rekursiivinen kieli.

   Oletetaan, että $D^t \in$ DTIME(t). Silloin on olemassa kielen D^t ajassa t tunnistava Turingin kone N, joka on sama kuin M_d jollakin $d \in \{0,1\}^*$.

   Jos $d \in D^t$, niin N hyväksyy syötteen d ajassa $t(|d|)$ ja silloin D^t:n määritelmän mukaan $d \notin D^t$. Jos $d \notin D^t$, niin N hylkää syötteen d ja silloin D^t:n määritelmän mukaan $d \in D^t$. Ristiriita johtuu oletuksesta, että $D^t \in$ DTIME(t). Siis $D^t \notin$ DTIME(t).

   [*Lisätieto (PO):* Jos funktio $t(n)$ on aikakonstruoituva, niin kone M saadaan toimimaan ajassa $t'(n)$ millä tahansa $t' = \omega(t \log t)$. Tällaisilla funktiopareilla on siis aina DTIME$(t') \setminus$ DTIME$(t) \neq \emptyset$. Lisätekijän $\log t$ aiheuttaa se, että koneen M täytyy simuloida koneen M_c mielivaltainen nauhamäärä kiinteällä nauhajoukolla, ja ω-luokituksen se, että tämän lisäksi tarvitsee vielä koneen M_c mielivaltainen nauha-aakkosto koodata M:n kiinteälle aakkostolle.]

[1] Jos luettelo rajoittuu yksinauhaisiin Turingin koneisiin, kuten "Automaatit ja kieliopit"-kurssilla, voidaan osoittaa vain että $D^t \notin \mathrm{DTIME}_1(t(n))$.

3. *Väite.* Olkoon $f : \Sigma^* \to \Gamma^*$ polynomisessa ajassa laskettava ja $x \in \Gamma^*$. Tällöin $f^{-1}(x) \in$ P.

   Todistus. Kieleen $f^{-1}(x)$ kuuluvat ne Σ^*:n merkkijonot y, joilla $f(y) = x$.

   Olkoon M Turingin kone, joka laskee funktion f polynomisessa ajassa t. Määritellään Turingin kone M_x siten, että se laskee syötteellä y M:n tavoin funktion arvon $f(y)$ ja vertaa sitä x:ään pysähtyen hyväksyvässä lopputilassa, jos $f(y) = x$, ja hylkäävässä lopputilassa, jos $f(y) \neq x$. Tämä M_x tunnistaa kielen $f^{-1}(x)$. On vielä osoitettava, että M_x:n laskenta tapahtuu polynomisessa ajassa.

   Syötteellä y M_x laskee $f(y)$:n polynomisessa ajassa ja, koska $|f(y)| \leq |y| + t(|y|)$, myös $f(y)$:n vertailu x:ään tapahtuu polynomisessa ajassa. Siten $f^{-1}(x) \in$ P. Λ

   Olkoon $A \subseteq \Gamma^*$ ja $A \in$ P. Laskekoon Turingin kone M_A syötteellä y M:n tavoin funktion arvon $f(y)$ ja testatkoon sen kuulumisen kieleen A. Laskenta tapahtuu polynomisessa ajassa, joten $f^{-1}(A) \in$ P.

4. *Väite.* Olkoot A, $B \subseteq \Sigma^*$ ja A, $B \in$ P. Tällöin

   (a) $\Sigma^* \setminus A \in$ P,

   (b) $A \cup B \in$ P ja

   (c) $A \cap B \in$ P.

   Todistus. Olkoot M ja N Turingin koneita, jotka tunnistavat kielet A ja B polynomisessa ajassa.

   (a) Turingin koneesta M saadaan vaihtamalla keskenään sen hyväksyvät ja hylkäävät lopputilat kielen $\Sigma^* \setminus A$ polynomisessa ajassa tunnistava Turingin kone. Siten $\Sigma^* \setminus A \in$ P.

   (b) Määritellään Turingin koneen H toiminta syötteellä x seuraavasti:

   – H testaa syötteen kuulumisen kieleen A kuten M. Myönteisessä tapauksessa H pysähtyy hyväksyvässä lopputilassa.

   – Muuten H testaa syötteen kuulumisen kieleen B kuten N. Myönteisessä tapauksessa H pysähtyy hyväksyvässä, kielteisessä hylkäävässä lopputilassa.

   Näin määritelty Turingin kone H tunnistaa kielen $A \cup B$. Tunnistus tapahtuu polynomisessa ajassa, sillä kumpikin kieli erikseen tunnistetaan polynomisessa ajassa. Siten $A \cup B \in$ P.

   (c) Väite seuraa edellisistä kohdista, sillä

   $$A \cap B = \Sigma^* \setminus ((\Sigma^* \setminus A) \cup (\Sigma^* \setminus B)). \quad \Lambda$$

5. Olkoot A, $B \subseteq \Sigma^*$ ja A, $B \in$ NP. Tällöin

   (a) $A \cup B \in$ NP ja

   (b) $A \cap B \in$ NP.

   Todistus. Olkoot M ja N epädeterministisiä Turingin koneita, jotka tunnistavat kielet A ja B epädeterministisesti polynomisessa ajassa.

   (a) Määritellään epädeterministisen Turingin koneen H toiminta syötteellä x seuraavasti:

   – H testaa syötteen kuulumisen kieleen A kuten M. Myönteisessä tapauksessa H pysähtyy hyväksyvässä lopputilassa.

– Muuten H testaa syötteen kuulumisen kieleen B kuten N. Myönteisessä tapauksessa H pysähtyy hyväksyvässä, kielteisessä hylkäävässä lopputilassa.

Näin määritelty epädeterministinen Turingin kone H tunnistaa kielen $A \cup B$ epädeterministisesti. Tunnistus tapahtuu polynomisessa ajassa, sillä kumpikin kieli erikseen tunnistetaan polynomisessa ajassa. Siten $A \cup B \in$ NP.

(b) Määritellään epädeterministisen Turingin koneen K toiminta syötteellä x seuraavasti:

– K testaa syötteen kuulumisen kieleen A kuten M. Kielteisessä tapauksessa K pysähtyy hylkäävässä lopputilassa.

– Muuten K testaa syötteen kuulumisen kieleen B kuten N. Myönteisessä tapauksessa K pysähtyy hyväksyvässä, kielteisessä hylkäävässä lopputilassa.

Näin määritelty epädeterministinen Turingin kone K tunnistaa kielen $A \cap B$ epädeterministisesti. Tunnistus tapahtuu polynomisessa ajassa, sillä kumpikin kieli erikseen tunnistetaan polynomisessa ajassa. Siten $A \cap B \in$ NP. Λ

Jos NP:n sulkeisuutta komplementoinnin suhteen yritetään todistaa samalla metodilla kuin tehtävässä 4, ongelmana on kielen A tunnistavan Turingin koneen M epädeterministisyys. Epädeterministinen Turingin kone hyväksyy syötteen, jos koneen laskenta päätyy jollakin laskentapolulla koneen pysähtymiseen hyväksyvässä lopputilassa. Ei ole kielletty, että samalla syötteellä voitaisiin päätyä toisella laskentapolulla koneen pysähtymiseen hylkäävässä lopputilassa. Koneen lopputila ei siis yksikäsitteisesti liity syötteen hyväksymiseen kuten deterministisellä Turingin koneella.

6. *Lemma.* co-P = P ja co-PSPACE = PSPACE.

Todistus. Ensimmäinen väite seuraa tehtävästä 4. Jälkimmäinen todistetaan seuraavasti:

Jos $A \in$ PSPACE, niin kielen A polynomisessa tilassa tunnistavasta Turingin koneesta saadaan vaihtamalla keskenään sen hyväksyvät ja hylkäävät lopputilat kielen $\Sigma^* \setminus A$ polynomisessa tilassa tunnistava Turingin kone. Siten $\Sigma^* \setminus A \in$ PSPACE. Λ

(a) Triviaalisti $P \subseteq NP$.

Jos $A \in$ P, niin $\Sigma^* \setminus A \in$ P $\subseteq$ NP, joten $A \in$ co-NP. Näin ollen P $\subseteq$ NP $\cap$ co-NP.

(b) Kun $t(n) \geq n$, lauseen 7.5[2] mukaan DTIME(t) $\subseteq$ DSPACE(t). Samalla tavalla samalla ehdolla voidaan päätellä, että NTIME(t) $\subseteq$ NSPACE(t). Niinpä NP $\subseteq$ NPSPACE = PSPACE (lauseen 7.10 mukaan).

Jos $A \in$ co-NP, niin $\Sigma^* \setminus A \in$ NP $\subseteq$ PSPACE, joten A $\in$ co-PSPACE = PSPACE. Näin ollen NP $\cup$ co-NP $\subseteq$ PSPACE.

(c) Jos P = NP, niin co-NP = co-P = P = NP. Jos NP = PSPACE, niin co-NP = co-PSPACE = PSPACE = NP. Näilllä päättelyillä väitteet todistuvat epäsuorasti.

[2]Monisteessa P. Orponen, “Laskennan teoria”.