T-79.1001 **Syksy 2006**
Tietojenkäsittelyteorian perusteet T
Harjoitus 11
Demonstraatiotehtävien ratkaisut

4. **Tehtävä**: Osoita, Ricen lausetta käyttämättä, että seuraava ongelma on ratkeamaton:

   Annettu Turingin kone M; hyväksyykö M tyhjän merkkijonon?

   Vastaus: Ongelma voidaan osoittaa ratkeamattoksi näyttämällä, että sen avulla voitaisiin ratkaista jokin ratkaisematon ongelma. Tätä kutsutaan rekursiiviseksi palautukseksi (*recursively reduce*).

   Ongelmassa tarkastellaan kieltä $L_\varepsilon = \{M \mid M \text{ hyväksyy syötteen } \varepsilon\}$. Oletetaan, että voidaan sille voidaan muodostaa Turingin kone M_ε. Kone M_ε saa syötteekseen mielivaltaisen Turingin koneen M binäärikoodauksen c_M, ja se hyväksyy sen, mikäli M hyväksyy tyhjän syötteen:

$$c_M \in L(M_\varepsilon) \Leftrightarrow \varepsilon \in L(M) \ .$$

   Seuraavaksi muodostetaan koneen M_ε avulla universaali Turingin kone M_U^ε. Universaalikone saa syötteekseen binäärikoodattuna jonkin Turingin koneen M ja sen syötteen x, ja se hyväksyy syötteen mikäli M hyväksyisi x:n:

$$c_M c_x \in L(M_U^\varepsilon) \Leftrightarrow x \in L(M) \ .$$

   Rakennettava universaalikone toimii kahdessa vaiheessa:

   (a) Muodostetaan koneen M ja merkkijonon x pohjalta Turingin kone M_x, joka aluksi kirjoittaa nauhalle jonon x ja sen jälkeen toimii kuten M.

   (b) Tarkistetaan koneella M_ε hyväksyykö kone M_x tyhjän syötteen.

   Ensimmäisessä vaiheessa M_U^ε muokkaa koodausta c_M lisäämällä sen alkuun $|x| + 1$ uutta tilaa. Ensimmäisessä $|x|$ tilassa kirjoitetaan kussakin yksi x:n merkeistä nauhalla ja siirretään lukupäätä oikealle. Viimeisessä uudessa tilassa kelataan lukupää takaisin alkuun ja siirrytään sittem M:n alkutilaan. Tämä vaihe voidaan toteuttaa totaalisella Turingin koneella sillä $|x|$ on määritelmän perusteella aina äärellinen, joten uusia tiloja tarvitaan vain äärellinen määrä.

   Jos kieli L_ε on ratkeava, niin koneesta M_ε voidaan tehdä totaalinen. Tällöin M_U^ε on myös totaalinen, sillä siinä ajetaan kaksi totaalista Turingin konetta peräjälkeen. Tästä seuraa kuitenkin ristiriita. Opetusmonisteen lauseen 6.7 perusteella tiedetään, että universaalikieli U ei ole rekursiivinen, joten sille ei ole mahdollista muodostaa totaalista Turingin konetta. Kieli L_ε ei siis voi olla rekursiivinen, joten tehtävänannossa annettu ongelma on ratkeamaton.

   Jos L_ε on ratkeava, niin U on ratkeava.
   U ei ole ratkeava.
   Johtopäätäs: L_ε ei ole ratkeava.

5. **Tehtävä**: Todista monisteen Lause 6.15:

   (i) Kieli $A \subseteq \Sigma^*$ on rekursiivinen, jos ja vain jos sen karakteristinen funktio

$$\chi_A : \Sigma^* \to \{0, 1\}, \qquad \chi_A(x) = \begin{cases} 1, & \text{jos } x \in A; \\ 0, & \text{jos } x \notin A \end{cases}$$

   on rekursiivinen funktio.

(ii) Kieli $A \subseteq \Sigma^*$ on rekursiivisesti numeroituva, jos ja vain jos on $A = \emptyset$ tai on olemassa rekursiivinen funktio $g : \{0,1\}^* \to \Sigma^*$, jolla

$$A = \{g(x) \mid x \in \{0,1\}^*\}.$$

Vastaus: Määritellään aluksi viisi yksinkertaista apukonetta:

- **1** kirjoittaa nauhalle merkin 1, siirtää lukupäätä askeleen oikealle ja pysähtyy.
- **0** kirjoittaa nauhalle merkin 0, siirtää lukupäätä askeleen oikealle ja pysähtyy ja pysähtyy.
- C tyhjentää koneen syötenauhan, palauttaa lukupään nauhan alkuun ja pysähtyy.
- NEXT lukee nauhalla olevan merkkijonon $x \in \Sigma^*$ ja korvaa sen leksikografisesti seuraavalla merkkijonolla (tehtävän 9.2 tapaan).
- $Cmp^{i,j}$ vertailee moninauhaisen Turingin koneen nauhoja i ja j keskenään ja hyväksyy syötteen mikäli niiden sisältö on sama.

Koneet ovat yksinkertaisia, joten niitä ei esitetä tässä.

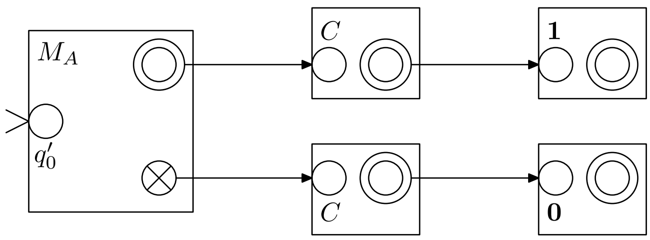
(i) [$\Rightarrow$] Olkoon kieli $A \subseteq \Sigma^*$ rekursiivinen. Tällöin on olemassa Turingin kone

$$M_A = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}} \rangle$$

siten, että:

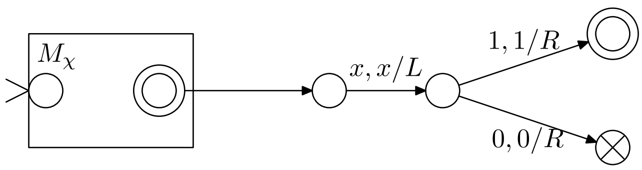
$$\begin{aligned} \forall w \in \Sigma^* : w \in L \Leftrightarrow (q_0, w) \vdash^*_{M_A} (q_{\mathrm{acc}}, \alpha) \quad &\text{ja} \\ w \notin L \Leftrightarrow (q_0, w) \vdash^*_{M_A} (q_{\mathrm{rej}}, \alpha) \end{aligned}$$

Muodostetaan Turingin kone M yhdistämällä kone M_A koneisiin **1**, **0** ja C seuraavaan tapaan:



Mikäli $w \in L$, niin kone M_A päätyy hyväksyvään tilaan, minkä jälkeen M tyhjentää nauhan ja kirjoittaa nauhalle luvun 1. Muussa tapauksessa nauhalle kirjoitetaan 0. Koska A on rekursiivinen, M_A pysähtyy aina, joten myös M pysähtyy aina, joten M laskee funktion $\chi(w) = \begin{cases} 1, w \in A \\ 0, w \notin A \end{cases}$, joka on kielen A karakteristinen funktio.

[$\Leftarrow$] Seuraavaksi oletetaan, että funktio $\chi(w)$ on rekursiivinen. Tällöin on olemassa Turingin kone M_χ, joka laskee sen arvon. Määritelmän mukaan M_χ pysähtyy aina, ja laskennan tulos kirjoitetaan nauhalle lukupään vasemmalle puolelle. Muodostetaan Turingin kone M seuraavaan tapaan:



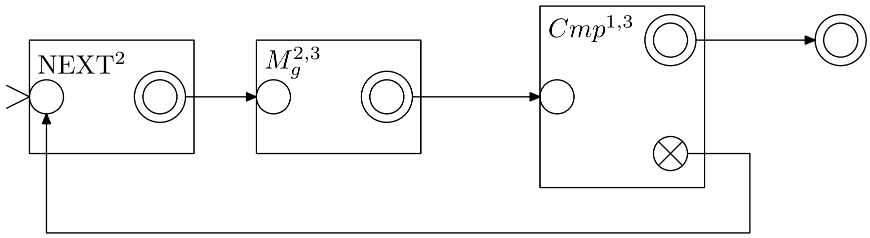
Nyt kone M hyväksyy sanan w aina, kun $\chi(w) = 1$, ja hylkää sen, kun $\chi(w) = 0$. Näin ollen M tunnistaa kielen A totaalisesti, joten A on rekursiivinen.

(ii) Halutaan todistaa väite:

> Kieli $A \subseteq \Sigma^*$ on rekursiivisesti numeroituva ($A \in RE$), jos ja vain jos $A = \emptyset$ tai on olemassa rekursiivinen funktio $g : \{0,1\}^* \to \Sigma^*$, jolla
>
> $$A = \{g(x) \mid x \in \{0,1\}^*\}$$

Mikäli $A = \emptyset$, niin triviaalisti $A \in RE$ ja $g(x) = 0$ vastaava rekursiivinen funktio.

Jos annetut ehdot täyttävä funktio g on olemassa, niin on olemassa Turingin kone M_g, joka laskee g:n. Tästä voidaan triviaalisti muodostaa kaksinauhainen kone $M_g^{1,2}$, joka laskee g:n mutta tallettaa tuloksen toiselle nauhalle ensimmäisen asemasta. Muodostetaan 3-nauhainen Turingin kone M_A seuraavasti:



Kone saa syötteensä ensimmäisellä nauhalla, ja sitä ei enää missään vaiheessa muuteta. Varsinaisessa laskentasilmukassa kone M_A korvaa 2. nauhalla olevan bittijonon x leksikografisesti seuraavalla bittijonolla y, minkä jälkeen lasketaan $g(y)$ ja talletetaan tulos 3. nauhalle. Lopuksi vertaillaan 1. ja 3. nauhojen sisältö keskenään. Mikäli nauhojen sisältö on sama, sana hyväksytään, muuten palataan alkuun.

[$\Leftarrow$] Tarkastellaan sanaa $w \in A$. Oletetaan, että annetut ehdot täyttävä rekursiivinen funktio g on olemassa. Tällöin $w = g(x)$ jollain $x = x_1 x_2 \cdots x_n$, missä n on äärellinen. Koska millä tahansa äärellisen mittaisella merkkijonolla on vain äärellinen määrä edeltäjiä leksikografisessa järjestyksessä, generoi NEXT lopulta x:n, jolloin $M_g^{2,3}$ laskee kolmannelle nauhalle sanan w, ja kone hyväksyy sanan. Näin ollen M_A tunnistaa kielen A, joten $A \in RE$.

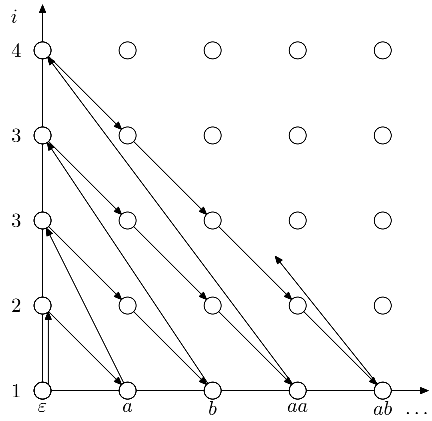
[$\Rightarrow$] Seuraavaksi oletetaan, että $A \in RE - \{\emptyset\}$. Tällöin on olemassa Turingin kone M_A, joka tunnistaa A:n. Määritellään apukone $M_{A,i}$, joka simuloi koneen M_A toimintaa i:n askeleen ajan. Kone $M_{A,i}$ hyväksyy syötteen x mikäli M_A hyväksyy sen käyttäen korkeintaan i askelta, muuten se hylkää sen. Huomataan, että kone $M_{A,i}$ on totaalinen, eli se pysähtyy aina.

Funktio g rakennetaan koneen $M_{A,i}$ avulla. Kaikki syötteet x ja rajat i koodataan funktion $c(x,y) = 0^x 1 0^y$ avulla bittijonoiksi pareittain[1], ja $g(c(x,y)) = x$, mikäli $M_{A,y}$ hyväksyy sanan x. Funktiota g määriteltäessä valitaan jokin kiinnitetty sana $x_0 \in A$, joka palautetaan aina kun $M_{A,y}$ ei pysähdy tai jos g:n syöte on väärää muotoa.

Konstruktion perusajatuksena on käydä läpi kaikki mahdolliset M_A:n syötteet lomitettuina: ensin ajetaan M_A:ta yhden askeleen verran ensimmäisellä syötteellä, sitten kaksi askelta ensimmäisellä syötteellä, yksi askel toisella syötteellä, jne. Alla olevassa kuvassa on tämä havainnollistettu aakkostolle $\Sigma = \{a, b\}$ (huomaa samankaltaisuus

[1] Tässä samaistetaan Σ^*:n sanat luonnollisten lukujen kanssa tavanomaiseen tapaan käyttäen leksikografista järjestystä.

tehtävän 2.5 kanssa):



Esitetään vielä lopuksi todistus formaalimmin: Määritellään funktio $g' : \{0,1\}^* \rightarrow \{0,1\}^*$ seuraavasti:

$$g'(w) = \begin{cases} x, & w = 0^x 10^y \text{ ja } M_{A,y}(x) \text{ pysähtyy} \\ x_0, & \text{muulloin} \end{cases},$$

missä $x_0 \in A$. Lopuksi määritellään funktio $g(x) = d(g'(x))$, missä d on funktio, joka kuvaa bittijonon 0^x joukon Σ^* x:nteen alkioon leksikografisessa järjestyksessä. Funktion $g'(x)$ arvo voidaan laskea äärellisessä ajassa, koska $M_{A,y}(x)$ ei voi jäädä ikuiseen silmukkaan. Näin ollen g' on rekursiivinen, joten myös g:kin on.

Tässä vaiheessa on huomattava, että vaikka g on aina olemassa, niin sitä ei ole välttämättä mahdollista löytää, sillä sopivan alkion $x_0 \in A$ etsiminen on yleisessä tapauksessa ratkeamaton ongelma.