582206 Laskennan mallit (syksy 2006)

2. kurssikoe, ratkaisuja

1. (a) Väite on *epätosi*. Olkoon C mikä tahansa yhteydetön ei-säännöllinen kieli, esim. $C = \{\, \mathrm{a}^n\mathrm{b}^n \mid n \in \mathbf{N} \,\}$, ja $R = \Sigma^*$. Nyt $R \cap C = C$, joka ei ole säännöllinen.

   (b) Väite on *tosi*. Jos R on säännöllinen ja C yhteydetön, on olemassa äärellinen automaatti $M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ ja pinoautomaatti $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, F_P)$, joilla $R = L(M)$ ja $C = L(P)$. Kieli $R \cap C$ voidaan tunnistaa pinoautomaatilla, jonka tilajoukkona on $Q_P \times Q_M$. Siirtymäfunktio δ määritellään siten, että jos $\delta_P(r_1, a, x) = (s_1, y)$ ja $\delta_M(r_2, a) = s_2$ jollain $a \in \Sigma$, niin $\delta((r_1, r_2), a, x) = ((s_1, s_2), y)$. Lisäksi jos $\delta_P(r_1, \varepsilon, x) = (s_1, y)$, niin $\delta((r_1, r_2), \varepsilon, x) = ((s_1, r_2), y)$ kaikilla $r_2 \in Q_M$. Hyväksyvien tilojen joukoksi valitaan $F = Q_P \times Q_M$.

   (c) Väite on *epätosi*. Tiedämme, että yhteydettömien kielten joukko ei ole suljettu komplementin suhteen. Siis on olemassa yhteydetön kieli C, jolla $\overline{C}$ ei ole yhteydetön. Kun valitaan $R = \Sigma^*$, niin $R - C = \overline{C}$, joka ei ole yhteydetön.

   (d) Väite on *tosi*. Jos R on säännöllinen, niin myös $\overline{R}$ on. Siis $C - R = C \cap \overline{R}$ on yhteydetön kohdan (b) nojalla.

2. (a) Ideana on muodostaa joukko $P \subseteq V$, joka sisältää ne muuttujat B, joilla $B \overset{+}{\Rightarrow} uAv$ joillain $u, v \in (V \cup \Sigma)^*$. Tämä voidaan tehdä seuraavasti:

   - Alusta joukkoon P kaikki ne muuttujat B, joilla $(B \to uAv) \in R$ joillain u, v.
   - Toista, kunnes P ei enää muutu:

     Käy läpi kaikki säännöt $(B \to w) \in R$. Jos voidaan kirjoittaa $w = uCv$, missä $C \in P$, niin aseta $P := P \cup \{\, B \,\}$.

   Nyt A uusii itsensä, jos ja vain jos $A \in P$.

   (b) Jos jossain jäsennyspuussa muuttuja A esiintyy kaksi kertaa jollain juuresta lehteen johtavalla polulla, niin A uusii itsensä. Siis jos itsensä uusivia muuttujia ei ole, niin juuresta lehteen johtavalla polulla on korkeintaan $|V| + 1$ solmua ($|V|$ eri muuttujaa ja yksi päätesymboli). Tällöin puun lehtien määrä on korkeintaan $b^{|V|}$, missä b on solmun suurin mahdollinen lasten lukumäärä eli suurin minkään säännön oikeana puolena esiintyvän merkkijonon pituus. Koska siis kielen sisältämien merkkijonojen pituudella on äärellinen yläraja, kieli on äärellinen.

   (c) Kieli voi olla äärellinen, vaikka siinä olisi itsensä uusivia muuttujia. Tämä voi johtua kahdesta eri syystä:

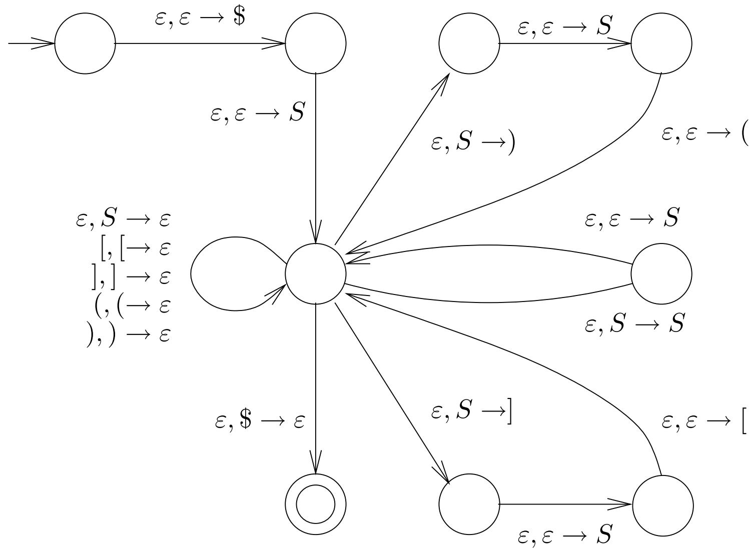
      i. Itsensä uusiva muuttuja A on sellainen, että $S \overset{*}{\Rightarrow} uAv$ ei päde millään $u, v \in (V \cup \Sigma)^*$. Esimerkki:

$$\begin{aligned} S &\to \mathrm{a} \\ A &\to \mathrm{a}A\mathrm{a} \mid \varepsilon. \end{aligned}$$

      ii. Itsensä uusiva muuttuja A on sellainen, että $A \overset{*}{\Rightarrow} uAv$ vain sellaisilla $u, v \in (V \cup \Sigma)^*$, että ei päde $u \overset{*}{\Rightarrow} x$ ja $v \overset{*}{\Rightarrow} y$ joillain $x, y \in \Sigma^*$. Esimerkki:

$$\begin{aligned} S &\to \mathrm{a} \mid A \\ A &\to BAB \mid \varepsilon. \end{aligned}$$

3. Seuraava automaatti on saatu soveltamalla luennoilla esitettyä muunnosta:



4. **Lause:** Kieli A on ratkeava, jos ja vain jos sekä A että $\overline{A}$ ovat Turing-tunnistettavia.

   Todistus: "$\Rightarrow$": Olkoon A ratkeava. Tällöin myös $\overline{A}$ on ratkeava. Siis A ja $\overline{A}$ ovat Turing-tunnistettavia.

   "$\Leftarrow$": Olkoot A ja $\overline{A}$ Turing-tunnistettavia. Oletetaan $A = L(M_1)$ ja $\overline{A} = L(M_2)$ joillain Turingin koneilla M_1 ja M_2. Nyt A voidaan ratkaista Turingin koneella M, joka syötteellä w toimii seuraavasti:

   > Simuloi koneita M_1 ja M_2 rinnakkain syötteellä w, kunnes toinen niistä pysähtyy. Jos M_1 hyväksyi tai M_2 hylkäsi, niin *hyväksy.* Jos M_1 hylkäsi tai M_2 hyväksyi, niin *hylkää.*

   Jos $w \in A$, niin M_1 hyväksyy w:n mutta M_2 ei hyväksy, joten M hyväksyy. Jos $w \notin A$, niin M_2 hyväksyy w:n mutta M_1 ei hyväksy, joten M hylkää. $\square$

   (Vaihtoehtoinen todistus on esitetty harjoituksen 10 tehtävässä 8.)

5.

   (a) Ongelmaa ei voi ratkaista.

   Annettu ongelma on oleellisesti "Java-kielen hyväksymisongelma". Churchin-Turingin teesin perusteella uskomme, ja tunnettujen laskentamallien ekvivalenssitulosten avulla voidaan tarkemmin perustella, että Java-kielellä voidaan esittää tasan samat algoritmit kuin Turingin koneella. Koska Turingin koneen hyväksymisongelma on ratkeamaton, myös Java-kielen hyväksymisongelma on ratkeamaton, eikä sitä erityisesti voida ratkaista Java-kielellä

   Esitetään edellinen päättely hieman täsmällisemmin. Tehdään vastaoletus, että haluttu Java-kielinen ohjelma on olemassa; merkitään sitä P:llä.

   Jos on annettu Turingin kone M, voidaan helposti muodostaa Java-funktio p_M, jolla kaikilla merkkijonoilla w pätee

   - $p_M(w)$ palauttaa 1, jos M hyväksyy syötteen w,
   - $p_M(w)$ palauttaa 0, jos M hylkää syötteen w ja
   - $p_M(w)$ jää silmukkaan, jos M jää silmukkaan syötteellä w.

   Merkitään Q:lla Turingin konetta, joka syötteellä $\langle M, w \rangle$ toimii seuraavasti:

   i. Muodosta Turingin koneesta M edellä esitetty Java-funktio p_M.

ii. Simuloi Java-ohjelmaa P syötteellä (w, p_M). Jos P tulostaisi "`kyllä`", niin *hyväksy*. Jos P tulostaisi "`ei`", niin *hylkää*.

Ohjelmasta P ja funktiosta p_M tehtyjen oletusten perusteella Q hyväksyy syötteen $\langle M, w \rangle$, jos $w \in L(M)$, ja hylkää, jos $w \notin L(M)$. Siis Q ratkaisee Turingin koneen hyväksymisongelman, mikä on mahdotonta. Vastaoletuksen, että ohjelma P on olemassa, täytyy olla väärä.

(Käytännössä Java-tulkin esittäminen Turingin koneena voisi olla melko työlästä. Esitetty argumentti kuitenkin toivottavasti vakuuttaa lukijan, että ehdotettu ohjelmointitehtävä ei ole realistinen.)

(b) Tehtävää ei ole mahdollista ratkaista.

Esitetty algoritminen ongelma on ekvivalenttia sen kanssa, että pitäisi ratkaista Turingin koneella kieli

$$A = \{ \langle R, M \rangle \mid R \text{ on säännöllinen lauseke ja } M \text{ Turingin kone, joilla } L(R) = L(M) \} .$$

(Siirtymät Turingin koneiden ja Java-ohjelmien välillä voidaan tehdä kuten kohdassa (a).) Tehdään vastaoletus, että kieli A on ratkeava. Nyt Turingin koneen hyväksymisongelma voidaan ratkaista Turingin koneella Q, joka syötteellä $\langle M, w \rangle$ toimii seuraavasti:

i. Muodosta kone M_1, joka syötteellä x toimii seuraavasti:

   Jos $x \neq w$, niin hylkää. Muuten simuloi konetta M syötteellä w ja hyväksy ja hylkää simulaation mukaan.

ii. Olkoon R_w säännöllinen lauseke, joka esittää yhdestä merkkijonosta koostuvaa kieltä $\{ w \}$.

iii. Jos $\langle R_w, M_1 \rangle \in A$, niin *hyväksy*; muuten *hylkää*.

Siis $L(R_w) = \{ w \}$. Toisaalta $L(M_1) = \{ w \}$, jos $w \in L(M)$, ja $L(M_1) = \emptyset$, jos $w \notin L(M)$. Siis $\langle R_w, M_1 \rangle \in A$, jos ja vain jos $w \in L(M)$, joten Q ratkaisee Turingin koneen hyväksymisongelman, kuten väitettiin. Tämä on kuitenkin mahdotonta, joten vastaoletuksen kielen A ratkeavuudesta pitää olla väärä.