6.5 Turingin koneiden pysähtymisongelma

Lause 6.9 Kieli

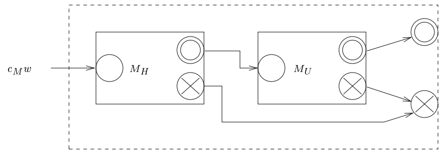
$$H = \{c_M w \mid M \text{ pysähtyy syötteellä } w\}$$

on rekursiivisesti numeroituva, mutta ei rekursiivinen.

Todistus. Todetaan ensin, että kieli H on rekursiivisesti numeroituva. Lauseen 6.6 todistuksessa esitetystä universaalikoneesta M_U on helppo muokata kone, joka syötteellä $c_M w$ simuloi koneen M laskentaa syötteellä w ja pysähtyy hyväksyvään lopputilaan, jos ja vain jos simuloitu laskenta ylipäätään pysähtyy.

Osoitetaan sitten, että kieli H ei ole rekursiivinen. Oletetaan nimittäin, että olisi $H = L(M_H)$ jollakin totaalisella Turingin koneella M_H. Oletetaan lisäksi, että kone M_H pysähtyessään jättää nauhalle alkuperäisen syötteensä, mahdollisesti tyhjämerkeillä jatkettuna. Olkoon M_U lauseen 6.6 todistuksessa konstruoitu universaalikone.

Kielelle U voitaisiin nyt muodostaa totaalinen tunnistaja yhdistämällä koneet M_H ja M_U seuraavasti:



Lauseen 6.7 mukaan tällaista kielen U tunnistajakonetta ei kuitenkaan voi olla olemassa. Saatu ristiriita osoittaa, että H ei voi olla rekursiivinen. $\square$

Seuraus 6.10 Kieli

$$\tilde{H} = \{c_M w \mid M \text{ ei pysähdy syötteellä } x\}$$

ei ole rekursiivisesti numeroituva. $\square$

6.7 Ricen lause

Hyvin monet tietojenkäsittelyongelmat ovat ratkeamattomia. Ns. Ricen lauseen mukaan itse asiassa jokseenkin *kaikki* ohjelmien toimintaa, tai tarkemmin sanoen niiden laskemia I/O-kuvauksia koskevat kysymykset ovat ratkeamattomia.

Sanotaan Turingin koneen M *semanttiseksi ominaisuudeksi* mitä tahansa sellaista ominaisuutta $\mathcal{S}$, joka riippuu vain koneen M tunnistamasta kielestä, ei sen syntaktisesta rakenteesta.

Esimerkkejä semanttisista ominaisuuksista: "M hyväksyy tyhjän syötejonon", "M hyväksyy jonkin syötejonon", "M hyväksyy äärettömän monta merkkijonoa", "M:n tunnistama kieli on säännöllinen" jne.

Jos kahdella Turingin koneella M_1 ja M_2 on $L(M_1) = L(M_2)$, niin koneilla M_1 ja M_2 on täsmälleen samat semanttiset ominaisuudet.

Ominaisuus $\mathcal{S}$ on *ratkeava*, jos annetusta Turingin koneen koodista voidaan algoritmisesti päätellä, onko koneella kysytty semanttinen ominaisuus.

Triviaalit ominaisuudet ovat ominaisuus $\mathcal{S}_\emptyset$, jota ei ole millään koneella ja ominaisuus $\mathcal{S}_{RE}$, joka on kaikilla koneilla.

Lause 6.12 (H. Rice 1953) Kaikki Turingin koneiden epätriviaalit semanttiset ominaisuudet ovat ratkeamattomia. □

6.8 Muita ratkeamattomuustuloksia

Lause 6.13 (Predikaattikalkyylin ratkeamattomuus; Church/Turing 1936)

Ei ole olemassa algoritmia, joka ratkaisisi, onko annettu ensimmäisen kertaluvun predikaattikalkyylin kaava ϕ validi ("loogisesti tosi", todistuva predikaattikalkyylin aksioomista). □

Lause 6.14 ("Hilbertin 10. ongelma"; Matijasevitsh/Davis/Robinson/Putnam 1953–70)

Ei ole olemassa algoritmia, joka ratkaisisi, onko annetulla kokonaislukukertoimisella polynomilla $P(x_1,\ldots,x_n)$ kokonaislukunollakohtia (so. jonoja $(m_1,\ldots,m_n) \in \mathbf{Z}^n$, joilla $P(m_1,\ldots,m_n) = 0$). Ongelma on ratkematon jo, kun $n = 15$ tai $\deg(P) = 4$. □

Eräiden kielioppiongelmien ratkeavuus, kun annettuna on kieliopit G ja G' Chomskyn hierarkian tietyltä tasolta i ja merkkijono w. Taulukossa $R \sim$ "ratkeava", $E \sim$ "ei ratkeava", $T \sim$ "aina totta".

	Taso i:			
Ongelma: onko	3	2	1	0
$w \in L(G)$?	R	R	R	E
$L(G) = \emptyset$?	R	R	E	E
$L(G) = \Sigma^*$?	R	E	E	E
$L(G) = L(G')$?	R	E	E	E
$L(G) \subseteq L(G')$?	R	E	E	E
$L(G) \cap L(G') = \emptyset$?	R	E	E	E
$L(G)$ säännollinen?	T	E	E	E
$L(G) \cap L(G')$ tyyppiä i?	T	E	T	T
$\overline{L(G)}$ tyyppiä i?	T	E	T	E

5. RAJOITTAMATTOMAT KIELIOPIT

Määritelmä 5.1 *Rajoittamaton kielioppi* t. *yleinen merkkijonomuunnossysteemi* on nelikko

$$G = (V, \Sigma, P, S),$$

missä

- V on kieliopin aakkosto;
- $\Sigma \subseteq V$ on kieliopin *päätemerkkien* joukko; $N = V - \Sigma$ on *välikemerkkien* t. *-symbolien* joukko;
- $P \subseteq V^+ \times V^*$ on kieliopin *sääntöjen* t. *produktioiden* joukko ($V^+ = V^* - \{\varepsilon\}$);
- $S \in N$ on kieliopin *lähtösymboli*.

Produktiota $(\omega, \omega') \in P$ merkitään tavallisesti $\omega \to \omega'$.

Merkkijono $\gamma \in V^*$ *tuottaa* t. *johtaa suoraan* merkkijonon $\gamma' \in V^*$ kieliopissa G, merkitään

$$\gamma \underset{G}{\Rightarrow} \gamma'$$

jos voidaan kirjoittaa $\gamma = \alpha\omega\beta$, $\gamma' = \alpha\omega'\beta$ ($\alpha, \beta, \omega' \in V^*$, $\omega \in V^+$), ja kieliopissa on produktio $\omega \to \omega'$.

Jos kielioppi G on yhteydestä selvä, merkitään yksinkertaisesti $\gamma \Rightarrow \gamma'$.

Merkkijono $\gamma \in V^*$ *tuottaa* t. *johtaa* merkkijonon $\gamma' \in V^*$ kieliopissa G, merkitään

$$\gamma \underset{G}{\Rightarrow}^* \gamma'$$

jos on olemassa jono V:n merkkijonoja $\gamma_0, \gamma_1, \ldots, \gamma_n$ ($n \geq 0$), siten että

$$\gamma = \gamma_0 \underset{G}{\Rightarrow} \gamma_1 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} \gamma_n = \gamma'.$$

Jälleen, jos G on yhteydestä selvä, merkitään yksinkertaisesti $\gamma \Rightarrow^* \gamma'$.

Merkkijono $\gamma \in V^*$ on kieliopin G *lausejohdos*, jos on $S \underset{G}{\Rightarrow}^* \gamma$. Pelkästään päätemerkeistä koostuva G:n lausejohdos $x \in \Sigma^*$ on G:n *lause*.

Kieliopin G *tuottama* t. *kuvaama kieli* $L(G)$ koostuu G:n lauseista, s.o.:

$$L(G) = \{x \in \Sigma^* \mid S \underset{G}{\Rightarrow}^* x\}.$$

Esimerkki. Rajoittamaton kielioppi ei-yhteydettömälle kielelle $\{a^k b^k c^k \mid k \geq 0\}$.

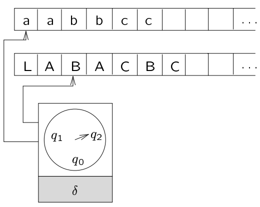
$$\begin{array}{rcl}
S & \to & LT \mid \varepsilon \\
T & \to & ABCT \mid ABC \\
BA & \to & AB \\
CB & \to & BC \\
CA & \to & AC \\
LA & \to & a \\
aA & \to & aa \\
aB & \to & ab \\
bB & \to & bb \\
bC & \to & bc \\
cC & \to & cc.
\end{array}$$

Esimerkiksi lauseen $aabbcc$ johto:

$$\begin{array}{lllllll}
\underline{S} & \Rightarrow & L\underline{T} \Rightarrow LABC\underline{T} & \Rightarrow & LAB\underline{CA}BC & \Rightarrow & LA\underline{BA}CB \\
 & \Rightarrow & LAAB\underline{CB}C & \Rightarrow & \underline{LA}ABBCC & \Rightarrow & \underline{aA}BBCC \\
 & \Rightarrow & a\underline{aB}BCC & \Rightarrow & aa\underline{bB}CC & \Rightarrow & aab\underline{bC}C \\
 & \Rightarrow & aabb\underline{cC} & \Rightarrow & aabbcc. & &
\end{array}$$

Lause 5.1 Jos formaali kieli L voidaan tuottaa rajoittamattomalla kieliopilla, se voidaan tunnistaa Turingin koneella.

Todistus. Olkoon $G = (V, \Sigma, P, S)$ kielen L tuottava rajoittamaton kielioppi. Muodostetaan kielen L tunnistava kaksinauhainen epädeterministinen Turingin kone M_G seuraavasti:



Nauhalla 1 kone säilyttää kopiota syötejonosta. Nauhalla 2 on kullakin hetkellä jokin G:n lausejohdos, jota kone pyrkii muuntamaan syötejonon muotoiseksi. Toimintansa aluksi M_G kirjoittaa kakkosnauhalle kieliopin lähtösymbolin S.

Koneen M_G laskenta koostuu vaiheista. Kussakin vaiheessa kone:

(i) vie kakkosnauhan nauhapään epädeterministisesti johonkin kohtaan nauhalla;

(ii) valitsee epädeterministisesti jonkin G:n produktion, jota yrittää soveltaa valittuun nauhankohtaan (produktiot on koodattu M_G:n siirtymäfunktioon;

(iii) jos produktion vasen puoli sopii yhteen nauhalla olevien merkkien kanssa, M_G korvaa ao. merkit produktion oikean puolen merkeillä;

(iv) vaiheen lopuksi M_G vertaa ykkös- ja kakkosnauhan merkkijonoja toisiinsa: jos jonot ovat samat, kone siirtyy hyväksyvään lopputilaan ja pysähtyy, muuten aloittaa uuden vaiheen (kohta (i)). $\square$

Lause 5.2 Jos formaali kieli L voidaan tunnistaa Turingin koneella, se voidaan tuottaa rajoittamattomalla kieliopilla.

Todistus. Olkoon $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ kielen L tunnistava standardimallinen Turingin kone. Muodostetaan kielen L tuottava rajoittamaton kielioppi G_M seuraavasti.

Idea: Kieliopin G_M välikkeiksi otetaan (muiden muassa) kaikkia M:n tiloja $q \in Q$ edustavat symbolit. Koneen M tilanne $(q, u\underline{a}v)$ esitetään merkkijonona $[uqav]$. M:n siirtymäfunktion perusteella G_M:ään muodostetaan produktiot, joiden ansiosta

$$[uqav] \underset{G_M}{\Rightarrow} [u'q'a'v'] \quad \text{joss} \quad (q, uav) \underset{M}{\vdash} (q', u'a'v').$$

Siten M hyväksyy syötteen x, jos ja vain jos

$$[q_0x] \underset{G_M}{\Rightarrow}^* [uq_{\text{acc}}v]$$

joillakin $u, v \in \Sigma^*$.

Kaikkiaan kielioppiin G_M tulee kolme ryhmää produktioita:

1. Produktiot, joilla lähtösymbolista S voidaan tuottaa mikä tahansa merkkijono muotoa $x[q_0x]$, missä $x \in \Sigma^*$ ja [, q_0 ja] ovat G_M:n välikkeitä.

2. Produktiot, joilla merkkijonosta $[q_0x]$ voidaan tuottaa merkkijono $[uq_{\text{acc}}v]$, jos ja vain jos M hyväksyy x:n.

3. Produktiot, joilla muotoa $[uq_{\text{acc}}v]$ oleva merkkijono muutetaan tyhjäksi merkkijonoksi.

Kieleen $L(M)$ kuuluvan merkkijonon x tuottaminen tapahtuu tällöin seuraavasti:

$$S \overset{(1)}{\Rightarrow^*} x[q_0x] \overset{(2)}{\Rightarrow^*} x[uq_{\text{acc}}v] \overset{(3)}{\Rightarrow^*} x.$$

Määritellään siis $G = (V, \Sigma, P, S)$, missä

$$V = \Gamma \cup Q \cup \{S, T, [,], E_L, E_R\} \cup \{A_a \mid a \in \Sigma\},$$

ja produktiot P muodostuvat seuraavista kolmesta ryhmästä:

1. Alkutilanteen tuottaminen:

$$\begin{array}{lll} S & \rightarrow T[q_0] & \\ T & \rightarrow \varepsilon & \\ T & \rightarrow aTA_a & (a \in \Sigma) \\ A_a[q_0 & \rightarrow [q_0A_a & (a \in \Sigma) \\ A_ab & \rightarrow bA_a & (a, b \in \Sigma) \\ A_a] & \rightarrow a] & (a \in \Sigma) \end{array}$$

2. M:n siirtymien simulointi ($a, b \in \Gamma$, $c \in \Gamma \cup \{[\}$) :

Siirtymät:	*Produktiot:*
$\delta(q, a) = (q', b, R)$	$qa \to bq'$
$\delta(q, a) = (q', b, L)$	$cqa \to q'cb$
$\delta(q, >) = (q', >, R)$	$q[\to [q'$
$\delta(q, <) = (q', b, R)$	$q] \to bq']$
$\delta(q, <) = (q', b, L)$	$cq] \to q'cb]$
$\delta(q, <) = (q', <, L)$	$cq] \to q'c]$

3. Lopputilanteen siivous:

$$\begin{array}{lll} q_{\text{acc}} & \to & E_L E_R \\ q_{\text{acc}}[& \to & E_R \\ aE_L & \to & E_L \qquad (a \in \Gamma) \\ [E_L & \to & \varepsilon \\ E_R a & \to & E_R \qquad (a \in \Gamma) \\ E_R] & \to & \varepsilon \end{array}$$

□

Yhteysherkät kieliopit

Rajoittamaton kielioppi on *yhteysherkkä*, jos sen kaikki produktiot ovat muotoa $\omega \to \omega'$, missä $|\omega'| \geq |\omega|$, tai mahdollisesti $S \to \varepsilon$, missä S on lähtösymboli.

Lisäksi vaaditaan, että jos kieliopissa on produktio $S \to \varepsilon$, niin lähtösymboli S ei esiinny minkään produktion oikealla puolella.

Formaali kieli L on *yhteysherkkä*, jos se voidaan tuottaa jollakin yhteysherkällä kieliopilla.

Normaalimuoto: Jokainen yhteysherkkä kieli voidaan tuottaa kieliopilla, jonka produktiot ovat muotoa $S \to \varepsilon$ ja $\alpha A \beta \to \alpha\omega\beta$, missä A on välike ja $\omega \neq \varepsilon$. (Säännön $A \to \omega$ sovellus "kontekstissa" $\alpha \_ \beta$.)

Lause 5.3 Formaali kieli L on yhteysherkkä, jos ja vain jos se voidaan tunnistaa epädeterministisellä Turingin koneella, joka ei tarvitse enempää työtilaa kuin syötejonon pituuden verran — siis koneella, jolla ei ole muotoa $\delta(q, <) = (q', b, \Delta)$ olevia siirtymiä, missä $b \neq$ '$<$'. □

Lauseen 5.3 koneita sanotaan *lineaarisesti rajoitetuiksi automaateiksi*.

Avoin ongelma ("LBA ?= DLBA"): onko epädeterminismi lauseessa 5.3 välttämätöntä?

Chomskyn hierarkia

Kielioppien, niillä tuotettavien kielten ja vastaavien tunnistusautomaattien ryhmittely:
Luokka 3: oikealle ja vasemmalle lineaariset (säännölliset) kieliopit / säännölliset kielet / äärelliset automaatit.
Luokka 2: yhteydettömät kieliopit / yhteydettömät kielet / pinoautomaatit.
Luokka 1: yhteysherkät kieliopit / yhteysherkät kielet / lineaarisesti rajoitetut automaatit.
Luokka 0: rajoittamattomat kieliopit / rekursiivisesti numeroituvat kielet / Turingin koneet.