6.5 Turingin koneiden pysähtymisongelma

Lause 6.9 Kieli

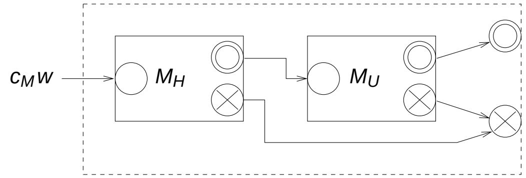
$$H = \{c_M w \mid M \text{ pysähtyy syötteellä } w\}$$

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. □

Seuraus 6.10 Kieli

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

ei ole rekursiivisesti numeroituva. □

6.7 Ricen lause

Ricen lauseen mukaan *kaikki* Turingin koneiden tunnistamia kieliä, t. niiden laskemia I/O-kuvauksia koskevat epätriviaalit kysymykset ovat ratkeamattomia.

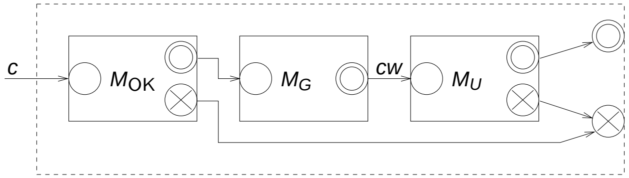
Johdantona lauseen todistukseen tarkastellaan ensin yhtä sen erikoistapausta, Turingin koneiden tunnistamien kielten *epätyhjyysongelmaa*: “Hyväksyykö annettu Turingin kone yhtään syötemerkkijonoa?” Ongelman esitys formaalina kielenä on

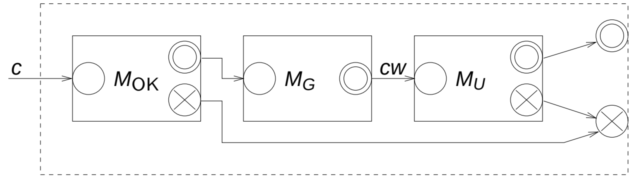
$$\mathrm{NE} = \{c \in \{0,1\}^* \mid L(M_c) \neq \emptyset\}.$$

Lause 6.11 Kieli NE on rekursiivisesti numeroituva, mutta ei rekursiivinen.

Todistus. Todetaan ensin, että kieli NE on rekursiivisesti numeroituva muodostamalla sille tunnistajakone M_{NE}. Kone M_{NE} on helpointa suunnitella epädeterministisenä.

Olkoon M_{OK} Turingin kone, joka testaa onko annettu syöte kelvollinen Turingin koneen koodi, ja olkoon M_G epädeterministinen Turingin kone, joka kirjoittaa nauhalla jo olevan merkkijonon perään mielivaltaisen binäärijonon w. Kone M_{NE} voidaan muodostaa yhdistämällä koneet M_{OK}, M_G ja universaalikone M_U seuraavasti:



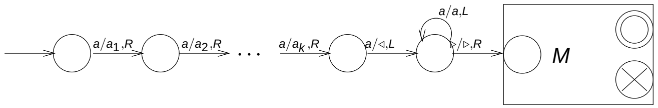


Selvästi on:

$$\begin{aligned}
& c \in L(M_{\text{NE}}) \\
\Leftrightarrow\quad & c \text{ on kelvollinen Tk-koodi ja } \exists\, w \text{ s.e. } cw \in U \\
\Leftrightarrow\quad & c \text{ on kelvollinen Tk-koodi ja } \exists\, w \text{ s.e. } w \in L(M_c) \\
\Leftrightarrow\quad & L(M_c) \neq \emptyset.
\end{aligned}$$

Osoitetaan, ettei kieli NE ole rekursiivinen. Oletetaan, että kielellä NE olisi totaalinen tunnistajakone M^T_{NE}, ja muodostetaan sitä käyttäen totaalinen tunnistajakone M^T_U kielelle U. (Ristiriita.)

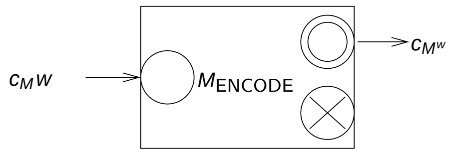
Konstruktio perustuu syötteiden koodaamiseen Turingin koneiden "ohjelmavakioiksi". Olkoon M mielivaltainen Turingin kone, jonka toimintaa syötteellä $w = a_1 a_2 \dots a_k$ halutaan tutkia. Merkitään M^w:llä konetta, joka aina korvaa "todellisen" syötteensä merkkijonolla w ja toimii sitten kuten M:



Koneen M^w toiminta ei siis riipu lainkaan sen todellisesta syötteestä, vaan se joko hyväksyy tai hylkää kaikki merkkijonot, sen mukaan miten M suhtautuu w:hen:

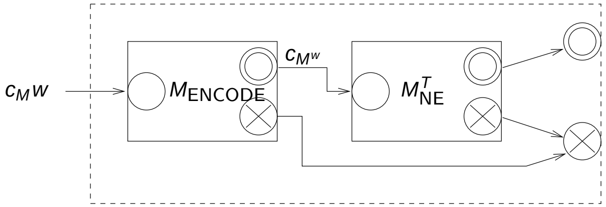
$$L(M^w) = \begin{cases} \{0,1\}^*, & \text{jos } w \in L(M); \\ \emptyset, & \text{jos } w \notin L(M). \end{cases}$$

Olkoon sitten M_{ENCODE} Turingin kone, joka saa syötteenään mielivaltaisen Turingin koneen M koodista c_M ja binäärijonosta w muodostuvan jonon $c_M w$ ja jättää tuloksenaan nauhalle edellä kuvatun koneen M^w koodin c_{M^w}:



(Jos syöte ei ole muotoa cw, missä c on kelvollinen Turingin koneen koodi, kone M_{ENCODE} päätyy hylkäävään lopputilaan.) Kone M_{ENCODE} operoi siis Turingin koneiden *koodeilla*. Annetun koneen M koodiin se lisää siirtymäviisikoita ("konekäskyjä") ja muuttaa tilojen numerointia siten, että koodi tulee koneen M sijaan esittämään konetta M^w.

Universaalikielelle U voitaisiin nyt koneet M_{ENCODE} ja hypoteettinen M^T_{NE} seuraavalla tavalla yhdistämällä muodostaa totaalinen tunnistajakone M^T_U:



Kone M^T_U on totaalinen, jos M^T_{NE} on, ja $L(M^T_U) = U$, koska:

$$c_M w \in L(M^T_U) \Leftrightarrow c_{M^w} \in L(M^T_{\text{NE}}) = \text{NE} \Leftrightarrow L(M^w) \neq \emptyset \Leftrightarrow w \in L(M).$$

Mutta kieli U ei ole rekursiivinen, joten tällainen totaalinen tunnistajakone M^T_U ei ole mahdollinen. Saadusta ristiriidasta päätellään, että myöskään kielellä NE ei voi olla totaalista tunnistajaa M^T_{NE}. □

Ricen lause

Turingin koneiden *semanttinen ominaisuus* $\mathcal{S}$ on mikä tahansa kokoelma rekursiivisesti numeroituvia aakkoston $\{0,1\}$ kieliä; *koneella* M *on ominaisuus* $\mathcal{S}$, jos $L(M) \in \mathcal{S}$. *Triviaalit ominaisuudet* ovat $\mathcal{S} = \emptyset$ (ominaisuus, jota ei ole millään koneella) ja $\mathcal{S} = RE$ (ominaisuus, joka on kaikilla koneilla).

Ominaisuus $\mathcal{S}$ on *ratkeava*, jos joukko

$$\text{codes}(\mathcal{S}) = \{c \mid L(M_c) \in \mathcal{S}\}$$

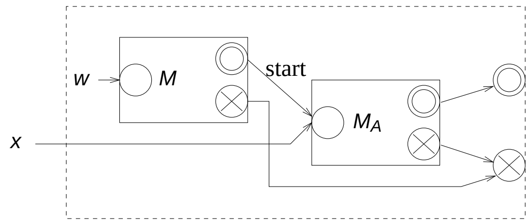
on rekursiivinen. Toisin sanoen: ominaisuus on ratkeava, jos annetusta Turingin koneen koodista voidaan algoritmisesti päätellä, onko koneella kysytty semanttinen ominaisuus.

Lause 6.12 [Rice 1953] Kaikki Turingin koneiden epätriviaalit semanttiset ominaisuudet ovat ratkeamattomia.

Todistus. Olkoon $\mathcal{S}$ mielivaltainen epätriviaali semanttinen ominaisuus. Voidaan olettaa, että $\emptyset \notin \mathcal{S}$: toisin sanoen, että tyhjän joukon tunnistavilla Turingin koneilla ei ole tarkasteltavaa ominaisuutta. Jos nimittäin $\emptyset \in \mathcal{S}$, voidaan ensin osoittaa, että ominaisuus $\bar{\mathcal{S}} = \text{RE} - \mathcal{S}$ on ratkeamaton, ja päätellä edelleen tästä että myös ominaisuus $\mathcal{S}$ on ratkeamaton. (Koska $\text{codes}(\bar{\mathcal{S}}) = \overline{\text{codes}(\mathcal{S})}$.)

Koska $\mathcal{S}$ on epätriviaali, on olemassa jokin Turingin kone M_A, jolla on ominaisuus $\mathcal{S}$ — jolla siis $L(M_A) \neq \emptyset \in \mathcal{S}$.

Olkoon nyt M_{ENCODE} Turingin kone, joka muodostaa syötteenä annetusta merkkijonosta $c_M w$ seuraavanlaisen Turingin koneen M^w koodin.



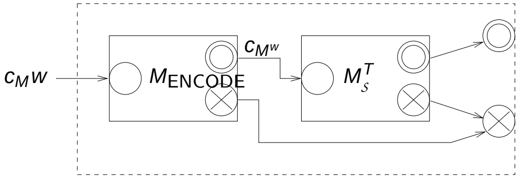
Jos syöte ei ole vaadittua muotoa, M_{ENCODE} päätyy hylkäävään lopputilaan.

Syötteellä x kone M^w toimii ensin kuten M syötteellä w. Jos M hyväksyy w:n, M^w toimii kuten kone M_A syötteellä x. Jos M hylkää w:n, myös M^w hylkää x:n. Kone M^w tunnistaa siis kielen

$$L(M^w) = \begin{cases} L(M_A), & \text{jos } w \in L(M); \\ \emptyset, & \text{jos } w \notin L(M). \end{cases}$$

Koska oletuksen mukaan $L(M_A) \in \mathcal{S}$ ja $\emptyset \notin \mathcal{S}$, on koneella M^w ominaisuus $\mathcal{S}$, jos ja vain jos $w \in L(M)$.

Oletetaan sitten, että ominaisuus $\mathcal{S}$ olisi ratkeava, so. että kielellä codes($\mathcal{S}$) olisi totaalinen tunnistajakone $M_{\mathcal{S}}^T$. Tällöin saataisiin edellisen todistuksen tapaan totaalinen tunnistajakone kielelle U yhdistämällä koneet M_{ENCODE} ja $M_{\mathcal{S}}^T$ seuraavasti:



Selvästi kone M_U^T on totaalinen, jos $M_{\mathcal{S}}^T$ on, ja

$$c_M w \in L(M_U^T) \Leftrightarrow c_{M^w} \in L(M_{\mathcal{S}}^T) = \text{codes}(\mathcal{S}) \Leftrightarrow L(M^w) \in \mathcal{S} \Leftrightarrow w \in L(M).$$

Koska kieli U ei ole rekursiivinen, tämä on mahdotonta, mistä päätellään, ettei ominaisuus $\mathcal{S}$ voi olla ratkeava. □

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

6.9 Rekursiiviset funktiot

Turingin koneen $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ *laskema osittaiskuvaus* (t. *-funktio*)

$$f_M : \Sigma^* \to \Gamma^*$$

määritellään:

$$f_M(x) = \begin{cases} u, \text{ jos } (q_0, \underline{x}) \underset{M}{\vdash}^* (q, u\underline{a}v) \text{ jollakin } q \in \{q_{\text{acc}}, q_{\text{rej}}\}, av \in \Gamma^*; \\ \text{määrittelemätön, muuten.} \end{cases}$$

Osittaisfunktio $f : \Sigma^* \to A$ on *osittaisrekursiivinen* jos se voidaan laskea jollakin Turingin koneella ja *(kokonais-)rekursiivinen*, jos se voidaan laskea jollakin totaalisella Turingin koneella. Ekvivalentisti voitaisiin määritellä, että osittaisrekursiivifunktio f on rekursiivinen, jos sen arvo $f(x)$ on määritelty kaikilla x.

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

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

Todistus. HT. $\square$