6. LASKETTAVUUSTEORIAA

Churchin–Turingin teesi: Mielivaltainen (riittävän vahva) laskulaitte Ξ Turingin kone.

Laskettavuusteoria: Tarkastellaan mitä Turingin koneilla voi ja erityisesti mitä ei voi laskea.

Tärkeä sana: Pysähtyvä ja ei-pysähtyvä Turingin koneet.

Määritelmä 6.1 Turingin kone

\[ M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, q_f) \]

on totaalin, jos se pysähtyy kaikilla syötteillä. Formaali kieli \( A \) on rekursiivisesti numeroitava, jos se voidaan tunnistaa jollakin Turingin koneella, ja rekursiivinen, jos se voidaan tunnistaa jollakin totaaliseella Turingin koneella.

Vaihtoehtoinen termistö: Palautetaan mielillä päätösongelmien (binaärivasteisten I/O-kuvausten) ja formaalien kielen vastaavuus: päätösongelmaa \( \Pi \) vastaa formaali kieli \( A_\Pi \) koostuu niistä syotteistä \( x_i \), joille ongelman \( \Pi \) vastaus on "kylä" (so. toivottuaste \( 1 \)).

Päätösongelma \( \Pi \) on ratkeava, jos sitä vastaava formaali kieli \( A_\Pi \) on rekursiivinen, ja osittain ratkeava, jos \( A_\Pi \) on rekursiivisesti numeroitava. Ongelma, joka ei ole ratkeava, on ratkematon. (Huom.: ratkematon ongelma voi siis olla osittain ratkeava.)

Toisin sanoen; päätösongelma on ratkeava, jos sillä on totaalin, kaikilla syötteillä pysähtyvä ratkaisualgoritmi, ja osittain ratkeava, jos sillä on ratkaisualgoritmi joka "kylä"-tapauksissa vastaa aina olkein, mutta "ei"-tapauksissa voi jättää pysähtymättä.

6.2 Rekursiivisten ja rek. num. kielen perusominaisuuksia

Lause 6.1 Olkoot \( A, B \subseteq \Sigma^* \) rekursiivisia. Tällöin myös \( \overline{A} = \Sigma^* - A \), \( A \cup B \) ja \( A \cap B \) ovat rekursiivisia.

Todistus.

(i) Olkoon \( M_A \) totaalin Turingin kone, jolla \( L(M_A) = A \). Kielen \( \overline{A} \) tunnistava totaalin Turingin kone saadaan vaihtamalla \( M_A \) n hyväksyvää ja hyväksymättä oppuutta keskenään.

(ii) Olkoot \( M_A \) ja \( M_B \) totaaliset Turingin koneet, joilla \( L(M_A) = A \), \( L(M_B) = B \). Kielen \( A \cup B \) tunnistava totaalin Turingin kone \( M \) saadaan yhdistämällä \( M_A \) ja \( M_B \) toimimaan perakkain; jos \( M_A \) hyväksyy syötteen, myös \( M \) hyväksyy; jos \( M_A \) päätyy hyväksymättä, \( M \) simoi vielä \( M_B \)tä.

(iii) \( A \cap B = \overline{A} \cup \overline{B}. \)\]
Lause 6.2 Olkoot $A, B \subseteq \Sigma^*$ rekursiivisesti numeroituvia. Tällöin myös $A \cup B$ ja $A \cap B$ ovat rekursiivisesti numeroituvia.

Todistus. $A \cap B$ kuten Lause 6.1 ja $A \cup B$ kuten Lause 6.3, (HT)

Lause 6.3 Kieli $A \subseteq \Sigma^*$ on rekursiivinen, jos ja vain jos kielet $A$ ja $\overline{A}$ ovat rekursiivisesti numeroituvia.


Seuraus 6.4 Olkoon $A \subseteq \Sigma^*$ rekursiivisesti numeroituvan kieli, joka ei ole rekursiivinen. Tällöin kieli $\overline{A}$ ei ole rekursiivisesti numeroituvaa.

6.3 Turingin koneiden koodaus
Tarkastellaan standardimalliia Turingin koneita, joiden syöteakosto on $\Sigma = \{0, 1\}$. Jokainen tällainen kone

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$$

voidaan esittää binäärijonoona:

Oletetaan, että $Q = \{q_0, q_1, \ldots, q_\ell\}$, missä $q_{acc} = q_{\ell-1}$ ja $q_{rej} = q_{\ell}$, ja että $\Gamma = \{\$, <\} = \{a_0, a_1, \ldots, a_m\}$, missä $a_0 = 0$, $a_1 = 1$, $a_2 = \$, ja $a_3 = \$. Merkitään lisäksi $\Delta_0 = L$ ja $\Delta_1 = R$.

Siirtymäfunktion $\delta$ arvojen koodaus: säännön

$$\delta(q_i, a_j) = (q_k, a_s, \Delta_t)$$

koodi on

$$c_{ij} = 0^{i+1}10^{i+1}10^{j+1}10^{i+1}10^{j+1}.$$

Koko koneen $M$ koodi on

$$c_M = 111c_{00}11c_{01}11\ldots11c_{0n}11c_{10}11\ldots11c_{1m}11\ldots11c_{2n}211\ldots11c_{m-2}m111.$$
Eräs ei rekursiivisesti numeroitu kieli

**Lemma 6.5 Kieli**

\[ D = \{ c \in \{0, 1\}^* \mid c \notin L(M_c) \} \]

ei ole rekursiivisesti numeroitu.

**Todistus.** Oletetaan, että olisi \( D = L(M) \) jollakin standardimalliolla Turingin koneella \( M \). Olkuon \( d \) koneen \( M \) binäärikoodi, so. \( D = L(M_d) \). Tällöin on

\[ d \in D \iff d \notin L(M_d) = D. \]

Ristiriidasta seuraa, että kieli \( D \) ei voi olla rekursiivisesti numeroituva. □

---

### 6.4 Universaalit Turingin koneet

Aakkosten \( \{0, 1\} \) **universaalikielit** \( U \) määritellään:

\[ U = \{ c_M w \mid w \in L(M) \}. \]

Olkuon A jokin aakkoston \( \{0, 1\} \) rekursiivisesti numeroituva kieli, ja olkuon \( M \) kielen A tunnistava standardimalliinen Turingin kone. Tällöin on

\[ A = \{ w \in \{0, 1\}^* \mid c_M w \in U \}. \]

Myös kieli \( U \) on rekursiivisesti numeroituva. Kielen \( U \) tunnistavia Turingin koneita sanotaan **universaaliksi Turingin koneiksi**.
Lause 6.6 Kiele $U$ on rekursiivisesti numeroituva.

Todistus. Kielten $U$ tunnistava universaalikone $M_U$ on helpointa kuvata kolmenauhaiseen mallina. (Standardointi tavalliseen tapaan.) Laskennan aluksi tarkastettaan syöte sijoitetaan koneen $M_U$ ykkösauhan alkuun. Tämän jälkeen kone toimii seuraavasti:

1. Aluksi $M_U$ tarkastaa, että syöte on muotoa $cw$, missä $c$ on kelvollinen Turingin koneen koodi. Jos syöte ei ole kelvollista muotoa, $M_U$ hylkää sen; muuten se kopii merkijonon $w = a_1 a_2 ... a_k \in \{0, 1\}^*$ kakkosauhalle muodossa $00010^k+110^k+1 ... 10^k+110000$.

2. Jos syöte on muotoa $cw$, missä $c = a_M$ jollakin koneella $M$, $M_U:n$ on selvitetään, hyväksyisikö kone $M$ syötteen $w$. Tässä tarkoituksessa $M_U$ säilyttää ykkösauhalla $M:n$ kuvausta $c$, kakkosauhalla simuloi $M:n$ nauhaa, ja kolmosauhalla säilyttää tietoa $M:n$ simuloidusta tilasta muodossa $q_0 \sim 0^i+1$ (aluksi siis $M_U$ kirjoittaa kolmosauhalle tilan $q_0$ koodin $0$).

3. Alkutoimien jälkeen $M_U$ toimii vaihteuttaen, simuloiden kussakin vaiheessa yhden koneen $M$ siirtymän. Vaiheen aluksi $M_U$ etsii ykkösauhalla $M:n$ kuvauksesta kohdan, joka vastaa $M:n$ simulointua tilaa $q_i$ ja merkidiä $a_j$. Oliko ykkösauhalla koodinkohdta $0^i+110^i+110^i+110^i+1$.

Tällöin $M_U$ korvaa kolmosauhalla merkijonon $0^i+1$ merkijonolla, $0^i+1$, kakkosauhalla merkijonon $0^i+1$ merkijonolla $0^i+1$, ja siirtyä kakkosauhan nauhapäänä yhden simuloidun merkin vasemmalle, jos $t = 0$, ja oikealle, jos $t = 1$. Jos ykkösauhalla ei ole yhtään simulointuun tilaan $q_j$ ityvää koodia, simuloiu $M$ on tullut hyväksyvää tai hyväksyzään lopputilaan; tällöin $i = k + 1$ tai $i = k + 2$, missä $q_k$ on viimeinen ykkösauhalla kuvattu tila. Kone $M_U$ siirtyy vastaavasti lopputilaan $q_{sqc}$ tai $q_{sqj}$.