

## 6. LASKETTAVUUSTEORIAA

*Churchin–Turingin teesi:* Mielivaltainen (riittävän vahva) laskulaite  $\equiv$  Turingin kone.

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

*Tärkeä erottelu:* Pysähtyvät ja ei-pysähtyvät Turingin koneet.

### Määritelmä 6.1 Turingin kone

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

on *totaalinen*, jos se pysähtyy kaikilla syötteillä. Formaali kieli  $A$  on *rekursiivisesti numeroituva*, jos se voidaan tunnistaa jollakin Turingin koneella, ja *rekursiivinen*, jos se voidaan tunnistaa jollakin totaalisella Turingin koneella.

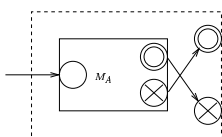
1

### 6.2 Rekursiivisten ja rek. num. kielten perusominaisuuksia

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

*Todistus.*

(i) Olkoon  $M_A$  totaalinen Turingin kone, jolla  $L(M_A) = A$ . Kielen  $\bar{A}$  tunnistava totaalinen Turingin kone saadaan vaihtamalla  $M_A$ :n hyväksyvä ja hylkäävä lopputila keskenään.



3

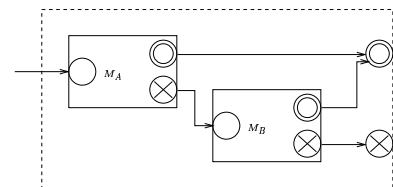
*Vaihtoehtoinen termistö:* Palautetaan mieliin päätösongelmien (binäärivasteisten I/O-kuvausten) ja formaalien kielten vastaavuus: päätösongelmaa  $\Pi$  vastaava formaali kieli  $A_\Pi$  koostuu niistä syötteistä  $x$ , joille ongelman  $\Pi$  vastaus on "kyllä" (so. toivottu vaste = 1).

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

Toisin sanoen: päätösongelma on ratkeava, jos sillä on totaalinen, kaikilla syötteillä pysähtyvä ratkaisualgoritmi, ja osittain ratkeava, jos sillä on ratkaisualgoritmi joka "kyllä"-tapauksissa vastaa aina oikein, mutta "ei"-tapauksissa voi jäädä pysähtymättä.

2

(ii) Olkoot  $M_A$  ja  $M_B$  totaaliset Turingin koneet, joilla  $L(M_A) = A$ ,  $L(M_B) = B$ . Kielen  $A \cup B$  tunnistava totaalinen Turingin kone  $M$  saadaan yhdistämällä  $M_A$  ja  $M_B$  toimimaan peräkkäin: jos  $M_A$  hyväksyy syötteen, myös  $M$  hyväksyy; jos  $M_A$  päätyy hylkäämiseen,  $M$  simuloi vielä  $M_B$ :tä.



(iii)  $A \cap B = \overline{\bar{A} \cup \bar{B}}$ .  $\square$

4

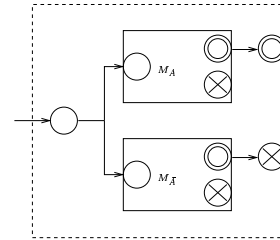
**Lause 6.2** Olkoot  $A, B \subseteq \Sigma^*$  rekursiivisesti numeroituvia. Tällöin myös  $A \cup B$  ja  $A \cap B$  ovat rekursiivisesti numeroituvia.

*Todistus.* Kuten Lause 6.1 (HT)  $\square$

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

*Todistus.* Väite vasemmalta oikealle seuraa lauseesta 6.1(i). Todistetaan väite oikealta vasemmalta.

Olkoot  $M_A$  ja  $M_{\bar{A}}$  Turingin koneet kielten  $A$  ja  $\bar{A}$  tunnistamiseen. Kaikilla  $x \in \Sigma^*$  joko  $M_A$  tai  $M_{\bar{A}}$  pysähtyy ja hyväksyy  $x$ :n. Muodostetaan  $M_A$  ja  $M_{\bar{A}}$  "rinnakkain" yhdistämällä totaalinen kaksinauhainen tunnistajakone  $M$ :  $M$  simuloi ykkösnauhallaan konetta  $M_A$  ja kakkosnauhallaan konetta  $M_{\bar{A}}$ . Jos ykkössimulaatio pysähtyy hyväksyvään lopputilaan,  $M$  hyväksyy syötteen; jos taas kakkössimulaatio hyväksyy,  $M$  hylkää syötteen.  $\square$



**Seuraus 6.4** Olkoon  $A \subseteq \Sigma^*$  rekursiivisesti numeroituva kieli, joka ei ole rekursiivinen. Tällöin kieli  $\bar{A}$  ei ole rekursiivisesti numeroituva.  $\square$

5

6

### 6.3 Turingin koneiden koodaus

Tarkastellaan standardimallisia Turingin koneita, joiden syöteaakkosto on  $\Sigma = \{0, 1\}$ . Jokainen tällainen kone

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

voidaan esittää binäärijonona:

Oletetaan, että  $Q = \{q_0, q_1, \dots, q_n\}$ , missä  $q_{acc} = q_{n-1}$  ja  $q_{rej} = q_n$ ; ja että  $\Gamma \cup \{>, <\} = \{a_0, a_1, \dots, a_m\}$ , 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  $\delta(q_i, a_j) = (q_r, a_s, \Delta_t)$  koodi on

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

Koko koneen  $M$  koodi on

$$c_M = 111c_{00}11c_{01}11\dots11c_{0m}11c_{10}11\dots11c_{1m}11\dots11c_{n-2,0}11\dots11c_{n-2,m}111.$$

7

Kääntäen voidaan jokaiseen binäärijonoon  $c$  liittää jokin Turingin kone  $M_c$ . Binäärijonoihin, jotka eivät ole edellisen koodauksen mukaisia Turingin koneiden koodeja, liitetään jokin triviaali, kaikki syötteen hylkäävä kone  $M_{triv}$ .

Määritellään siis:

$$M_c = \begin{cases} \text{kone } M, \text{ jolla } c_M = c, \text{ jos } c \text{ on kelvollinen} \\ \text{konekoodi;} \\ \text{kone } M_{triv}, \text{ muuten.} \end{cases}$$

Saadaan luettelo kaikista aakkoston  $\{0, 1\}$  Turingin koneista, ja epäsuorasti myös kaikista aakkoston  $\{0, 1\}$  rekursiivisesti numeroituvista kielistä. Koneet ovat

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, \dots,$$

kielet vastaavasti

$$L(M_\epsilon), L(M_0), L(M_1), L(M_{00}), L(M_{01}), \dots$$

(indeksit kanonisessa järjestyksessä). Kukin kieli voi esiintyä luettelossa monta kertaa.

8

## Eräs ei rekursiivisesti numeroituva kieli

**Lemma 6.5** Kieli

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

ei ole rekursiivisesti numeroituva.

*Todistus.* Oletetaan, että olisi  $D = L(M)$  jollakin standardimallisella Turingin koneella  $M$ . Olkoon  $d$  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.  $\square$

Kieltä  $D$  vastaava päätösongelma: "Hylkääkö annetun koodin  $c$  esittämä Turingin kone syöteen  $c$ ?" Luontevampia esimerkkejä seuraa jatkossa.

Kielen  $D$  muodostaminen kuvallisesti: jos kielten  $L(M_\varepsilon), L(M_0), L(M_1), \dots$  karakteristiset funktiot esitetään taulukkona, niin kieli  $D$  poikkeaa kustakin kielestä taulukon diagonaalilla:

$D$	$L(M_\varepsilon)$	$L(M_0)$	$L(M_1)$	$L(M_{00})$	$\dots$
$\varepsilon$	1	0	0	0	$\dots$
0	0	1	0	0	$\dots$
1	0	0	1	0	$\dots$
00	0	0	0	1	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$

9

10

## 6.4 Universaalit Turingin koneet

Aakkoston  $\{0, 1\}$  *universaalikieli*  $U$  määritellään:

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

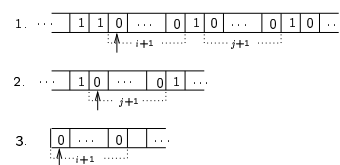
Olkoon  $A$  jokin aakkoston  $\{0, 1\}$  rekursiivisesti numeroituva kieli, ja olkoon  $M$  kielen  $A$  tunnistava standardimallinen Turingin kone. Tällöin on

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

Myös kieli  $U$  on rekursiivisesti numeroituva. Kielen  $U$  tunnistavia Turingin koneita sanotaan *universaaleiksi Turingin koneiksi*.

**Lause 6.6** Kieli  $U$  on rekursiivisesti numeroituva.

*Todistus.* Kielen  $U$  tunnistava universaalikone  $M_U$  on helpointa kuvata kolmenauhaisena mallina. (Standardointi tavalliseen tapaan.) Laskennan aluksi tarkastettava syöte sijoitetaan koneen  $M_U$  ykkösnauhan alkuun.



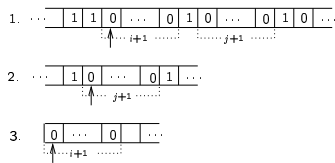
Tämän jälkeen kone toimii seuraavasti:

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 kopioi merkkijonon  $w = a_1 a_2 \dots a_k \in \{0, 1\}^*$  kakkosnauhalle muodossa

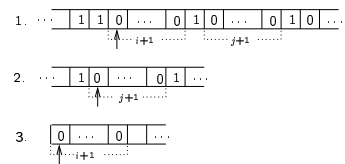
$$00010^{a_1+1}10^{a_2+1}1 \dots 10^{a_k+1}10000.$$

11

12



2. Jos syöte on muotoa  $cw$ , missä  $c = c_M$  jollakin koneella  $M$ ,  $M_U$ :n on selvitettävä, hyväksyisikö kone  $M$  syötteen  $w$ . Tässä tarkoituksessa  $M_U$  säilyttää ykkösnauhalla  $M$ :n kuvausta  $c$ , kakkosnauhalla simuloi  $M$ :n nauhaa, ja kolmosnauhalla säilyttää tietoa  $M$ :n simuloidusta tilasta muodossa  $q_i \sim 0^{i+1}$  (aluksi siis  $M_U$  kirjoittaa kolmosnauhalle tilan  $q_0$  koodin 0).

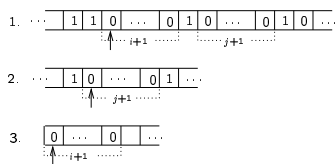


3. Alkutoimien jälkeen  $M_U$  toimii vaiheittain, simuloiden kussakin vaiheessa yhden koneen  $M$  siirtymän. Vaiheen aluksi  $M_U$  etsii ykkösnauhalla  $M$ :n kuvauksesta kohdan, joka vastaa  $M$ :n simuloitua tilaa  $q_i$  ja merkkiä  $a_j$ .

Olkoon ykkösnauhalla oleva koodinkohta

$$0^{i+1}10^{j+1}10^{r+1}10^{s+1}10^{t+1}.$$

Tällöin  $M_U$  korvaa kolmosnauhalla merkkijonon  $0^{i+1}$  merkkijonolla  $0^{r+1}$ , kakkosnauhalla merkkijonon  $0^{j+1}$  merkkijonolla  $0^{s+1}$ , ja siirtää kakkosnauhan nauhapäätä yhden simuloitun merkin vasemmalle, jos  $t = 0$ , ja oikealle, jos  $t = 1$ .

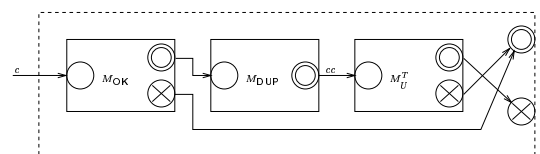


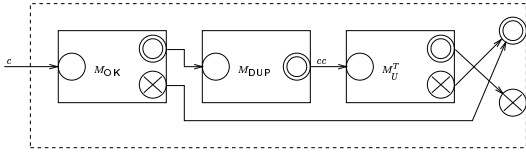
Jos ykkösnauhalla ei ole yhtään simuloitua tilaa  $q_i$  liittyvää koodia, simuloitu kone  $M$  on tullut hyväksyvään tai hylkäävään lopputilaan; tällöin  $i = k + 1$  tai  $i = k + 2$ , missä  $q_k$  on viimeinen ykkösnauhalla kuvattu tila. Kone  $M_U$  siirtyy vastaavasti lopputilaan  $q_{acc}$  tai  $q_{rej}$ . □

**Lause 6.7** Kieli  $U$  ei ole rekursiivinen.

*Todistus.* Oletetaan, että kielellä  $U$  olisi totaalinen tunnistajakone  $M_U^T$ . Tällöin voitaisiin Lemman 6.5 kielelle  $D$  muodostaa totaalinen tunnistajakone  $M_D$  seuraavasti.

Olkoon  $M_{OK}$  totaalinen Turingin kone, joka testaa, onko syötteenä annettu merkkijono kelvollinen Turingin koneen koodi, ja olkoon  $M_{DUP}$  totaalinen Turingin kone, joka muuntaa syötejonon  $c$  muotoon  $cc$ . Kone  $M_D$  muodostetaan koneista  $M_U^T$ ,  $M_{OK}$  ja  $M_{DUP}$  yhdistämällä seuraavan kaavion esittämällä tavalla:





Selvästi kone  $M_D$  on totaalinen, jos kone  $M_U^T$  on, ja

$$\begin{aligned}
 c \in L(M_D) &\Leftrightarrow c \notin L(M_{OK}) \text{ tai } cc \notin L(M_U^T) \\
 &\Leftrightarrow c \notin L(M_c) \\
 &\Leftrightarrow c \in D.
 \end{aligned}$$

Mutta lemmän 6.5 mukaan kieli  $D$  ei ole rekursiivinen; ristiriita.  $\square$ .

### Seuraus 6.8 Kieli

$$\tilde{U} = \{c_M w \mid w \notin L(M)\}$$

ei ole rekursiivisesti numeroituva.

*Todistus.* Kieli  $\tilde{U}$  on oleellisesti sama kuin uni-versaalikielen  $U$  komplementti  $\bar{U}$ ; tarkasti ot-taen on  $\bar{U} = \tilde{U} \cup \text{ERR}$ , missä ERR on helposti tunnistettava rekursiivinen kieli

$$\text{ERR} = \{x \in \{0,1\}^* \mid x \text{ ei sisällä alkuosanaan kelvollista Turingin koneen koodia}\}.$$

Jos siis kieli  $\tilde{U}$  olisi rekursiivisesti numeroituva, olisi samoin myös kieli  $\bar{U}$ . Koska kieli  $U$  on re-kursiivisesti numeroituva, seuraisi tästä, että  $U$  on peräti rekursiivinen. Mutta tämä on vastoin edellisen lauseen tulosta, mistä päätellään, et-tä kieli  $\tilde{U}$  ei voi olla rekursiivisesti numeroituva.

$\square$