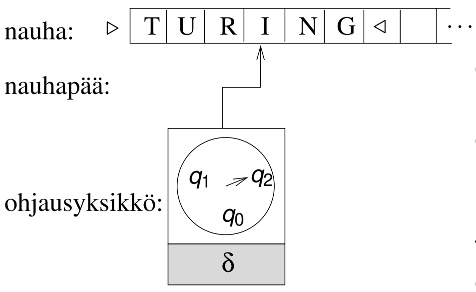
4. TURINGIN KONEET

Alan Turing 1935–36.



Turingin kone on kuin äärellinen automaatti, jolla on käytössään nauha. Kone voi siirtää nauhapäätä vasemmalle tai oikealle; se voi myös lukea tai kirjoittaa nauhapään kohdalla olevan merkin. Nauha on oikealle rajaton.

Churchin–Turingin teesi: Mikä tahansa mekaanisesti ratkeava ongelma voidaan ratkaista Turingin koneella.

T–79.1001/1002 Tietojenkäsittelyteorian perusteet

Turingin koneen kanssa ekvivalentteja laskentamalleja:

- Gödelin–Kleenen rekursiivisesti määritellyt funktiot (1936),
- Churchin λ-kalkyyli (1936),
- Postin (1936) ja Markovin (1951) merkkijonomuunnossysteemit, kaikki nykyiset ohjelmointikielet.

Turingin koneet $\equiv$ ohjelmointikieli.

Pekka Orponen syksy 2005

T–79.1001/1002 Tietojenkäsittelyteorian perusteet

Määritelmä 4.1 *Turingin kone* on seitsikko

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

missä:

- Q on koneen *tilojen* äärellinen joukko;
- Σ on koneen *syöteaakkosto*;
- $\Gamma \supseteq \Sigma$ on koneen *nauha-aakkosto* (ol. että $\triangleright, \triangleleft \notin \Gamma$);
- $\delta : (Q - \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}) \times (\Gamma \cup \{\triangleright, \triangleleft\}) \to Q \times (\Gamma \cup \{\triangleright, \triangleleft\}) \times \{L, R\}$ on koneen *siirtymäfunktio*;
- $q_0 \in Q$ on koneen *alkutila*;
- $q_{\mathrm{acc}} \in Q$ on koneen *hyväksyvä* ja
- $q_{\mathrm{rej}} \in Q$ sen *hylkäävä lopputila*.

Pekka Orponen syksy 2005

T–79.1001/1002 Tietojenkäsittelyteorian perusteet

Siirtymäfunktion arvon

$$\delta(q, a) = (q', b, \Delta)$$

tulkinta:

Ollessaan tilassa q ja lukiessaan nauhamerkin (tai alku- tai loppumerkin) a, kone siirtyy tilaan q', kirjoittaa lukemaansa paikkaan merkin b, ja siirtää nauhapäätä yhden merkkipaikan verran suuntaan Δ ($L \sim$ “left”, $R \sim$ “right”).

Sallittuja kirjoitettavia merkkejä ja siirtosuuntia on rajoitettu, mikäli a = ‘$\triangleright$’ tai ‘$\triangleleft$’, ja siirtymäfunktion arvo on aina määrittelemätön, kun $q = q_{\mathrm{acc}}$ tai $q = q_{\mathrm{rej}}$. Joutuessaan jompaan kumpaan näistä tiloista kone pysähtyy heti.

Pekka Orponen syksy 2005

Siis: siirtymäfunktion arvoilta

$$\delta(q,a) = (q', b, \Delta)$$

vaaditaan:

(i) jos $b = \triangleright$, niin $a = \triangleright$;

(ii) jos $a = \triangleright$, niin $b = \triangleright$ ja $\Delta = R$;

(iii) jos $b = \triangleleft$, niin $a = \triangleleft$ ja $\Delta = L$.

Koneen *tilanne* on nelikko

$$(q, u, a, v) \in Q \times \Gamma^* \times (\Gamma \cup \{\varepsilon\}) \times \Gamma^*,$$

missä voi olla $a = \varepsilon$, mikäli myös $u = \varepsilon$ tai $v = \varepsilon$.

Tulkinta: kone on tilassa q, nauhan sisältö sen alusta nauhapään vasemmalle puolelle on u, nauhapään kohdalla on merkki a ja nauhan sisältö nauhapään oikealta puolelta käytetyn osan loppuun on v.

Mahdollisesti on $a = \varepsilon$, jos nauhapää sijaitsee aivan nauhan alussa tai sen käytetyn osan lopussa. Ensimmäisessä tapauksessa ajatellaan, että kone "havaitsee" merkin '$\triangleright$' ja toisessa tapauksessa merkin '$\triangleleft$'.

Alkutilanne syötteellä $x = a_1 a_2 \dots a_n$ on nelikko

$$(q_0, \varepsilon, a_1, a_2 \dots a_n).$$

Tilannetta (q, u, a, v) merkitään yleensä yksinkertaisemmin $(q, u\underline{a}v)$, ja alkutilannetta syötteellä x yksinkertaisesti $(q_0, \underline{x})$.

Tilanne (q, w) *johtaa suoraan* tilanteeseen (q', w'), merkitään

$$(q, w) \underset{M}{\vdash} (q', w'),$$

jos jokin seuraavista ehdoista täyttyy: kaikilla $q, q' \in Q$, $u, v \in \Gamma^*$, $a, b \in \Gamma$ ja $c \in \Gamma \cup \{\varepsilon\}$:

jos $\delta(q,a) = (q', b, R)$, niin $(q, u\underline{a}cv) \underset{M}{\vdash} (q', ub\underline{c}v)$;

jos $\delta(q,a) = (q', b, L)$, niin $(q, uc\underline{a}v) \underset{M}{\vdash} (q', u\underline{c}bv)$;

jos $\delta(q,\triangleright) = (q', \triangleright, R)$, niin $(q, \underline{\varepsilon}cv) \underset{M}{\vdash} (q', \underline{c}v)$;

jos $\delta(q,\triangleleft) = (q', b, R)$, niin $(q, u\underline{\varepsilon}) \underset{M}{\vdash} (q', ub\underline{\varepsilon})$;

jos $\delta(q,\triangleleft) = (q', b, L)$, niin $(q, uc\underline{\varepsilon}) \underset{M}{\vdash} (q', u\underline{c}b)$;

jos $\delta(q,\triangleleft) = (q', \triangleleft, L)$, niin $(q, uc\underline{\varepsilon}) \underset{M}{\vdash} (q', u\underline{c})$.

Tilanteet, jotka ovat muotoa (q_{acc}, w) tai (q_{rej}, w) eivät johda mihinkään muuhun tilanteeseen. Näissä tilanteissa kone *pysähtyy*.

Tilanne (q, w) *johtaa tilanteeseen* (q', w'), merkitään

$$(q, w) \underset{M}{\vdash}^* (q', w'),$$

jos on olemassa tilannejono (q_0, w_0), (q_1, w_1), ..., (q_n, w_n), $n \geq 0$, siten että

$$(q, w) = (q_0, w_0) \underset{M}{\vdash} (q_1, w_1) \underset{M}{\vdash} \cdots \underset{M}{\vdash} (q_n, w_n) = (q', w').$$

Turingin kone M *hyväksyy* merkkijonon $x \in \Sigma^*$, jos

$$(q_0, \underline{x}) \underset{M}{\vdash}^* (q_{\text{acc}}, w) \qquad \text{jollakin } w \in \Gamma^*;$$

muuten M *hylkää* x:n.

Koneen M *tunnistama kieli* on:

$$L(M) = \{x \in \Sigma^* \mid (q_0, \underline{x}) \underset{M}{\vdash}^* (q_{\text{acc}}, w) \text{ jollakin } w \in \Gamma^*\}.$$

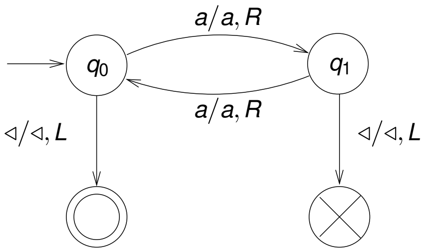
Esimerkki 1. Kieli $\{a^{2k} \mid k \geq 0\}$ voidaan tunnistaa Turingin koneella

$$M = (\{q_0, q_1, q_{acc}, q_{rej}\}, \{a\}, \{a\}, \delta, q_0, q_{acc}, q_{rej}),$$

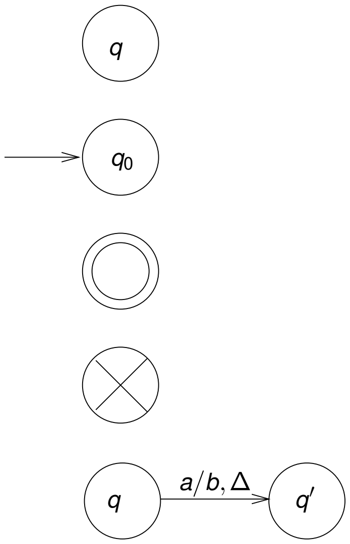
missä

$$\begin{aligned}
\delta(q_0, a) &= (q_1, a, R),\\
\delta(q_1, a) &= (q_0, a, R),\\
\delta(q_0, \triangleleft) &= (q_{acc}, \triangleleft, L),\\
\delta(q_1, \triangleleft) &= (q_{rej}, \triangleleft, L).
\end{aligned}$$

Kaavioesitys:



Kaavioesityksessä käytetyt merkinnät:



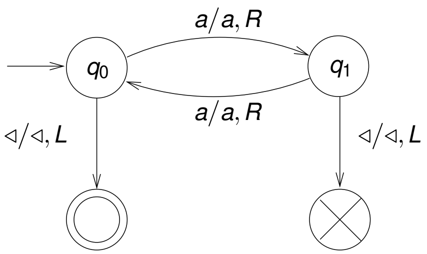
Tila q

Alkutila

Hyväksyvä lopputila (q_{acc})

Hylkäävä lopputila (q_{rej})

Tilasiirtymä $\delta(q, a) = (q', b, \Delta)$

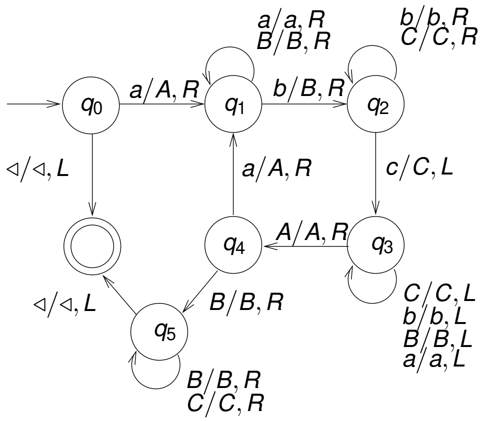


Koneen M laskenta esimerkiksi syötteellä aaa etenee seuraavasti:

$$\begin{aligned}
(q_0, \underline{a}aa) \underset{M}{\vdash} & (q_1, a\underline{a}a) \underset{M}{\vdash} (q_0, aa\underline{a})\\
& \underset{M}{\vdash} (q_1, aaa\underline{\varepsilon}) \underset{M}{\vdash} (q_{rej}, aa\underline{a}).
\end{aligned}$$

Kone pysähtyy tilassa q_{rej}, joten $aaa \notin L(M)$.

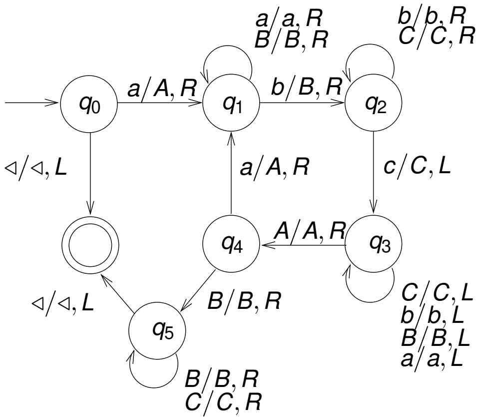
Esimerkki 2. Kielen $\{a^k b^k c^k \mid k \geq 0\}$ tunnistava Turingin kone:



Selkeyden vuoksi ei koneen hylkäävää lopputilaa q_{rej} ole tässä esitetty eksplisiittisesti. Tulkinta on tällöin, että kaikki kaaviosta "puuttuvat" kaaret johtavat tähän tilaan.

Koneen laskenta syötteellä *aabbcc*:

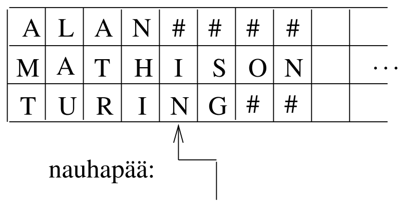
$(q_0, \underline{a}abbcc) \vdash (q_1, A\underline{a}bbcc) \vdash$
$(q_1, Aa\underline{b}bcc) \vdash (q_2, AaB\underline{b}cc) \vdash$
$(q_2, AaBb\underline{c}c) \vdash (q_3, AaB\underline{b}Cc) \vdash$
$(q_3, Aa\underline{B}bCc) \vdash (q_3, A\underline{a}BbCc) \vdash$
$(q_3, \underline{A}aBbCc) \vdash (q_4, A\underline{a}BbCc) \vdash$
$(q_1, AA\underline{B}bCc) \vdash (q_1, AAB\underline{b}Cc) \vdash$
$(q_2, AABB\underline{C}c) \vdash (q_2, AABBC\underline{c}) \vdash$
$(q_3, AABB\underline{C}C) \vdash (q_3, AAB\underline{B}CC) \vdash$
$(q_3, AA\underline{B}BCC) \vdash (q_3, A\underline{A}BBCC) \vdash$
$(q_4, AA\underline{B}BCC) \vdash (q_5, AAB\underline{B}CC) \vdash$
$(q_5, AABB\underline{C}C) \vdash (q_5, AABBC\underline{C}) \vdash$
$(q_5, AABBCC\underline{\varepsilon}) \vdash (q_{\mathrm{acc}}, AABBC\underline{C})$.



4.2 Turingin koneiden laajennuksia

1. Moniuraiset koneet

Sallitaan, että Turingin koneen nauha koostuu k:sta rinnakkaisesta urasta, jotka kaikki kone lukee ja kirjoittaa yhdessä laskenta-askelessa: Koneen siirtymäfunktion arvot ovat tällöin muotoa:



$$\delta(q, (a_1, \dots, a_k)) = (q', (b_1, \dots, b_k), \Delta),$$

missä $a_1, \dots, a_k$ ovat urilta $1, \dots, k$ luetut merkit, $b_1, \dots, b_k$ niiden tilalle kirjoitettavat merkit, ja $\Delta \in \{L, R\}$ on nauhapään siirtosuunta.

Laskennan aluksi tutkittava syöte sijoitetaan ykkösuran vasempaan laitaan; muille urille tulee sen kohdalle erityisiä tyhjämerkkejä #.

Formaalisti *k-urainen Turingin kone* on seitsikko

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

missä muut komponentit ovat kuten standardimallissa, paitsi siirtymäfunktio:

$$\delta : (Q - \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}) \times (\Gamma^k \cup \{\rhd, \lhd\}) \to Q \times (\Gamma^k \cup \{\rhd, \lhd\}) \times \{L, R\}.$$

Seuraajatilannerelaation $\underset{M}{\vdash}$, alkutilan jne. määritelmät ovat pieniä muutoksia lukuunottamatta samanlaiset kuin standardimallissa.

Lause 4.1. Jos formaali kieli L voidaan tunnistaa k-uraisella Turingin koneella, se voidaan tunnistaa myös standardimallisella Turingin koneella.

Todistus. Olkoon $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}})$ k-urainen Turingin kone, joka tunnistaa kielen L. Vastaava standardimallinen kone $\widehat{M}$ muodostetaan seuraavasti:

$$\widehat{M} = (\hat{Q}, \Sigma, \hat{\Gamma}, \hat{\delta}, \hat{q}_0, q_{\mathrm{acc}}, q_{\mathrm{rej}}),$$

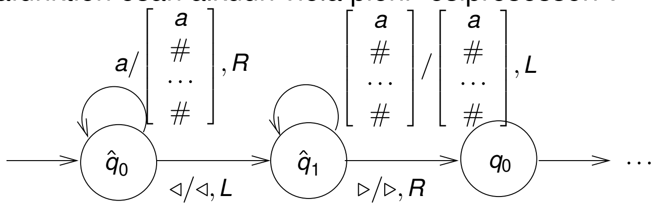
missä $\hat{Q} = Q \cup \{\hat{q}_0, \hat{q}_1, \hat{q}_2\}$, $\hat{\Gamma} = \Sigma \cup \Gamma^k$ ja kaikilla $q \in Q$ on

$$\hat{\delta}\left(q, \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}\right) = \left(q', \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}, \Delta\right),$$
$$\text{kun} \quad \delta(q, (a_1, \dots, a_k)) = (q', (b_1, \dots, b_k), \Delta).$$

Koneen $\widehat{M}$ laskennan aluksi täytyy syötejono “nostaa” ykkösuralle, so. korvata nauhalla merkkijono $a_1 a_2 \dots a_n$ merkkijonolla

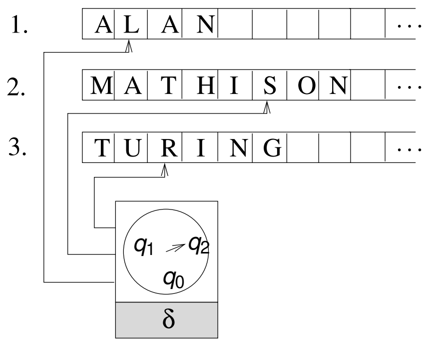
$$\begin{bmatrix} a_1 \\ \# \\ \vdots \\ \# \end{bmatrix} \begin{bmatrix} a_2 \\ \# \\ \vdots \\ \# \end{bmatrix} \cdots \begin{bmatrix} a_n \\ \# \\ \vdots \\ \# \end{bmatrix}.$$

Tätä operaatiota varten liitetään M:stä kopioidun siirtymäfunktion osan alkuun vielä pieni “esiprosessori”:



□

2. Moninauhaiset koneet



Sallitaan, että Turingin koneella on k toisistaan riippumatonta nauhaa, joilla on kullakin oma nauhapäänsä. Kone lukee ja kirjoittaa kaikki nauhat yhdessä laskenta-askelessa. Laskennan aluksi syöte sijoitetaan ykkösnauhan vasempaan laitaan ja kaikki nauhapäät nauhojensa alkuun.

Tällaisen koneen siirtymäfunktion arvot ovat muotoa

$$\delta(q, a_1, \dots, a_k) = (q', (b_1, \Delta_1), \dots, (b_k, \Delta_k)),$$

missä $a_1, \dots, a_k$ ovat nauhoilta $1, \dots, k$ luetut merkit, $b_1, \dots, b_k$ niiden tilalle kirjoitettavat merkit, ja $\Delta_1, \dots, \Delta_k \in \{L, R\}$ nauhapäiden siirtosuunnat.

Formaalisti *k-nauhainen Turingin kone* on seitsikko

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

missä muut komponentit ovat kuten standardimallissa, paitsi siirtymäfunktio:

$$\delta : (Q - \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}) \times (\Gamma \cup \{\triangleright, \triangleleft\})^k \to Q \times ((\Gamma \cup \{\triangleright, \triangleleft\}) \times \{L, R\})^k.$$

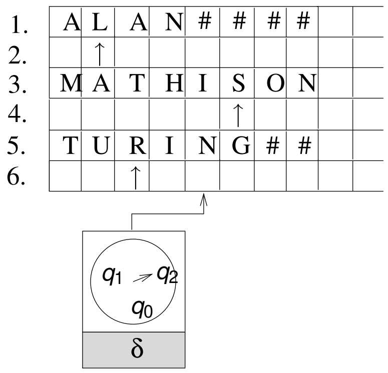
Seuraajatilannerelaatio ym. peruskäsitteet määritellään pienin muutoksin entiseen tapaan.

Lause 4.2. Jos formaali kieli L voidaan tunnistaa k-nauhaisella Turingin koneella, se voidaan tunnistaa myös standardimallisella Turingin koneella.

Todistus (idea). Olkoon

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

k-nauhainen Turingin kone, joka tunnistaa kielen L. Konetta M voidaan simuloida $2k$-uraisella koneella $\widehat{M}$ siten, että koneen $\widehat{M}$ parittomat urat $1, 3, 5, \dots, 2k-1$ vastaavat M:n nauhoja $1, 2, \dots, k$, ja kutakin paritonta uraa seuraavalla parillisella uralla on merkillä $\uparrow$ merkitty vastaavan nauhan nauhapään sijainti.



Simuloinnin aluksi syötemerkkijono sijoitetaan normaalisti koneen $\widehat{M}$ ykkösuralle. Ensimmäisessä siirtymässään $\widehat{M}$ merkitsee nauhapääosoittimet $\uparrow$ parillisten urien ensimmäisiin merkkipaikkoihin.

Tämän jälkeen $\widehat{M}$ toimii "pyyhkimällä" nauhaa edestakaisin sen alku- ja loppumerkin välillä. Vasemmalta oikealle pyyhkäisyllä $\widehat{M}$ kerää tiedot kunkin osoittimen kohdalla olevasta M:n nauhamerkistä. Kun kaikki merkit ovat selvillä, $\widehat{M}$ simuloi yhden M:n siirtymän, ja takaisin oikealta vasemmalle suuntautuvalla pyyhkäisyllä kirjoittaa $\uparrow$-osoittimien kohdalle asianmukaiset uudet merkit ja siirtää osoittimia. $\square$

3. Epädeterministiset koneet

Formaalisti *epädeterministinen Turingin kone* on seitsikko

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

missä muut komponentit ovat kuten deterministisessä standardimallissa, paitsi siirtymäfunktio:

$$\delta : (Q - \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}) \times (\Gamma \cup \{\triangleright, \triangleleft\}) \to \mathcal{P}(Q \times (\Gamma \cup \{\triangleright, \triangleleft\}) \times \{L, R\}).$$

Siirtymäfunktion arvon

$$\delta(q, a) = \{(q_1, b_1, \Delta_1), \ldots, (q_k, b_k, \Delta_k)\}$$

tulkinta on, että ollessaan tilassa q ja lukiessaan merkin a kone voi toimia jonkin kolmikon (q_i, b_i, Δ_i) mukaisesti.

Epädeterministisen koneen tilanteet, tilannejohdot jne. määritellään formaalisti samoin kuin deterministisenkin koneen tapauksessa, paitsi että ehdon $\delta(q, a) = (q', b, \Delta)$ sijaan kirjoitetaan $(q', b, \Delta) \in \delta(q, a)$.

Tämän muutoksen takia seuraajatilannerelaatio $\underset{M}{\vdash}$ ei ole enää yksiarvoinen: koneen tilanteella (q, w) voi nyt olla useita vaihtoehtoisia seuraajia, so. tilanteita (q', w'), joilla $(q, w) \underset{M}{\vdash} (q', w')$.

Koneen M tunnistama kieli määritellään:

$$L(M) = \{x \in \Sigma^* \mid (q_0, \underline{x}) \underset{M}{\vdash}^* (q_{\mathrm{acc}}, w) \text{ jollakin } w \in \Gamma^*\}.$$

Epädeterministisen koneen M tapauksessa siis merkkijono x kuuluu M:n tunnistamaan kieleen, jos *jokin* M:n kelvollinen tilannejono johtaa alkutilanteesta syötteellä x hyväksyvään lopputilanteeseen.

Esimerkki. Yhdistettyjen lukujen "tunnistaminen" epädeterministisillä Turingin koneilla.

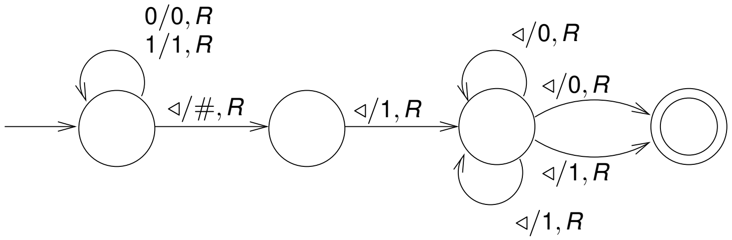
Ei-negatiivinen kokonaisluku n on *yhdistetty*, jos sillä on kokonaislukutekijät $p, q \geq 2$, joilla $pq = n$. Luku, joka ei ole yhdistetty, on *alkuluku*.

Oletetaan, että on jo suunniteltu deterministinen kone CHECK_MULT, joka tunnistaa kielen

$$L(\text{CHECK\_MULT}) = \{n\#p\#q \mid n, p, q \text{ binäärilukuja}, n = pq\}.$$

Olkoon lisäksi GO_START deterministinen Turingin kone, joka siirtää nauhapään osoittamaan nauhan ensimmäistä merkkiä.

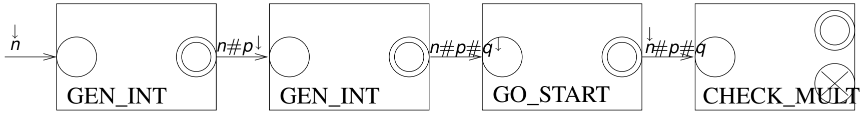
Olkoon edelleen GEN_INT seuraava mielivaltaisen ykköstä suuremman binääriluvun nauhan loppuun tuottava epädeterministinen Turingin kone:



Epädeterministinen Turingin kone TEST_COMPOSITE, joka tunnistaa kielen

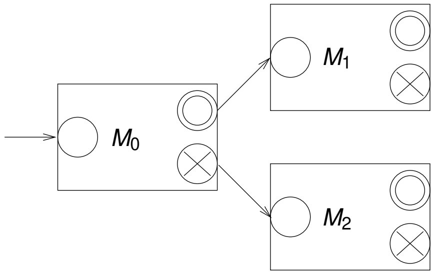
$L(\text{TEST\_COMPOSITE}) = \{n \mid n \text{ on binäärimuotoinen yhdistetty luku}\}$

voidaan muodostaa näistä komponenteista yhdistämällä:



Yhdistetty kone hyväksyy syötteenä annetun binääriluvun n, jos ja vain jos on olemassa binääriluvut $p, q \geq 2$, joilla $n = pq$ — siis jos ja vain jos n on yhdistetty luku.

Huom. Yleinen kaaviomerkintä Turingin koneiden yhdistämiselle:



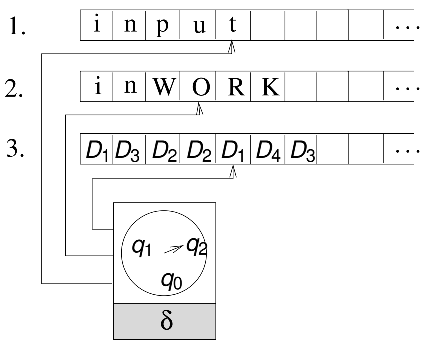
Lause 4.3 Jos formaali kieli L voidaan tunnistaa epädeterministisellä Turingin koneella, se voidaan tunnistaa myös standardimallisella deterministisellä Turingin koneella.

Todistus (idea). Olkoon

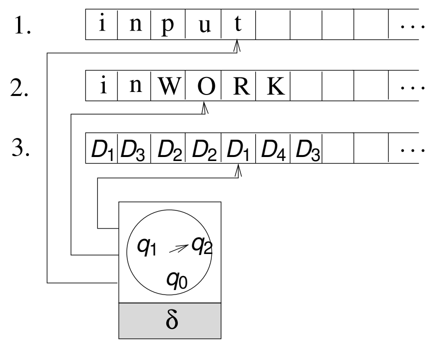
$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

epädeterministinen Turingin kone, joka tunnistaa kielen L. Konetta M voidaan simuloida kolmenauhaisella deterministisellä koneella $\widehat{M}$, joka käy systemaattisesti läpi M:n mahdollisia laskentoja (tilannejonoja), kunnes löytää hyväksyvän — jos sellainen on olemassa. Kone $\widehat{M}$ voidaan edelleen muuntaa standardimalliseksi edellisten lauseiden konstruktioilla.

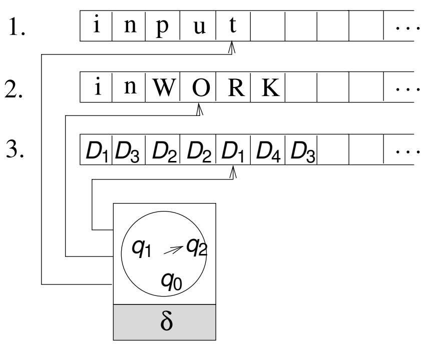
Yksityiskohtaisemmin:
Nauhalla 1 $\widehat{M}$ säilyttää kopiota syötejonosta ja nauhalla 2 se simuloi koneen M työnauhaa.
Kunkin simuloitavan laskennan aluksi $\widehat{M}$ kopioi syötteen nauhalta 1 nauhalle 2 ja pyyhkii pois nauhalle 2 edellisen laskennan jäljiltä mahdollisesti jääneet merkit.
Nauhalla 3 $\widehat{M}$ pitää kirjaa vuorossa olevan laskennan “järjestysnumerosta”.

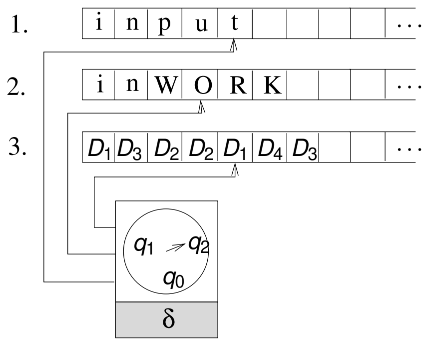


Tarkemmin sanoen, olkoon r suurin M:n siirtymäfunktion arvojoukon koko. Tällöin $\widehat{M}$:lla on erityiset nauhamerkit $D_1, \dots, D_r$, joista koostuvia jonoja se generoi nauhalle 3 kanonisessa järjestyksessä: ε, D_1, D_2, $\dots$, D_r, D_1D_1, D_1D_2, $\dots$, D_1D_r, D_2D_1, $\dots$.
Kutakin generoitua jonoa kohden $\widehat{M}$ simuloi yhden M:n osittaisen laskennan, jossa epädeterministiset valinnat tehdään kolmosnauhan koodijonon ilmaisemalla tavalla.



Esimerkiksi jos kolmosnauhalla on jono $D_1D_3D_2$, niin ensimmäisessä siirtymässä valitaan vaihtoehto 1, toisessa vaihtoehto 3, kolmannessa vaihtoehto 2. Ellei tämä laskenta johtanut M:n hyväksyvään lopputilaan, generoidaan seuraava koodijono $D_1D_3D_3$ ja aloitetaan alusta.
Jos koodijono on epäkelpo, so. jos siinä jossakin kohden on tilanteeseen liian suuri koodi, simuloitu laskenta keskeytetään ja generoidaan seuraava jono.





Selvästi tämä systemaattinen koneen M laskentojen läpikäynti johtaa koneen $\widehat{M}$ hyväksymään syötejonon, jos ja vain jos koneella M on syötteen hyväksyvä laskenta. Jos hyväksyvää laskentaa ei ole, kone $\widehat{M}$ ei pysähdy. $\square$