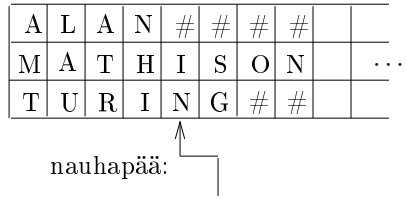
## 4.2 Turingin koneiden laajennuksia

### 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:




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

missä $a_1,\ldots,a_k$ ovat urilta $1,\ldots,k$ luetut merkit, $b_1,\ldots,b_k$ 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

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

missä muut komponentit ovat kuten standardimallissa, paitsi siirtymäfunktio:

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

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

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

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

$$\hat{\delta}(q,\begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}) = (q',\begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix},\Delta),$$

$$\text{kun} \quad \delta(q,(a_1,\ldots,a_k)) = (q',(b_1,\ldots,b_k),\Delta).$$

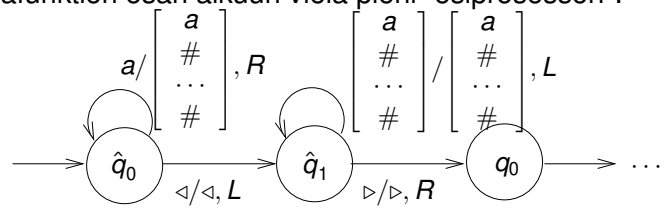




Koneen $\widehat{M}$ laskennan aluksi täytyy syötejono "nostaa" ykkösuralle, so. korvata nauhalla merkkijono $a_1 a_2 \ldots a_n$ merkkijonolla

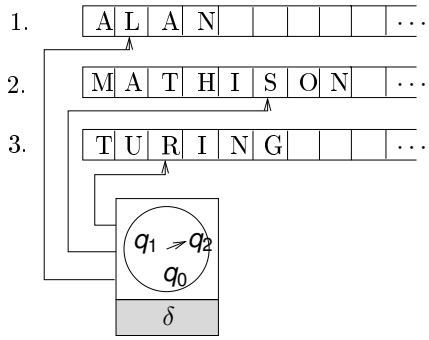
$$\begin{bmatrix} a_1 \\ \# \\ \vdots \\ \# \end{bmatrix} \begin{bmatrix} a_2 \\ \# \\ \vdots \\ \# \end{bmatrix} \cdots \begin{bmatrix} a_n \\ \# \\ \vdots \\ \# \end{bmatrix}.$$

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

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

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

Formaalisti *$k$-nauhainen Turingin kone* on seitsikko

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

missä muut komponentit ovat kuten standardimallissa, paitsi siirtymäfunktio:

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

Seuraajatilannerelaatio ym. peruskäsitteet määritellään pienin muutoksin entiseen tapaan.

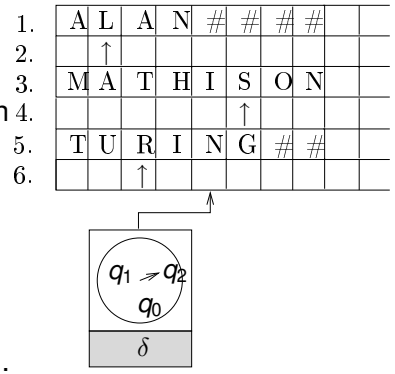




**Lause 4.2.** Jos formaali kieli $L$ voidaan tunnistaa $k$-nauhaisella Turingin koneella, se voidaan tunnistaa myös standardimallisella Turingin koneella.
*Todistus (idea).* Olkoon

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

$k$-nauhainen Turingin kone, joka tunnistaa kielen $L$. Konetta $M$ voidaan simuloida $2k$-uraisella koneella $\widehat{M}$ siten, että koneen $\widehat{M}$ parittomat urat $1, 3, 5, \ldots, 2k-1$ vastaavat $M$:n nauhoja $1, 2, \ldots, k$, 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

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

missä muut komponentit ovat kuten deterministisessä standardimallissa, paitsi siirtymäfunktio:

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

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 $(q_i, b_i, \Delta_i)$ 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:

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

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

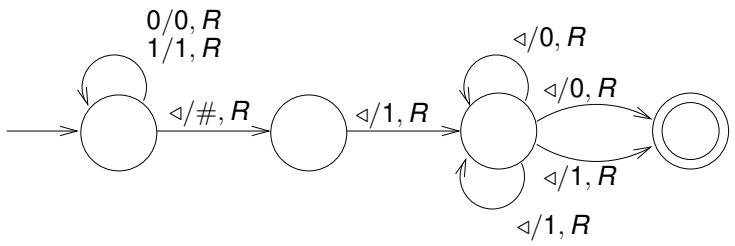
$$L(\text{CHECK_MULT}) = \{n\#p\#q \mid n, p, q \text{ binäärilukuja}, n = pq\}.$$

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





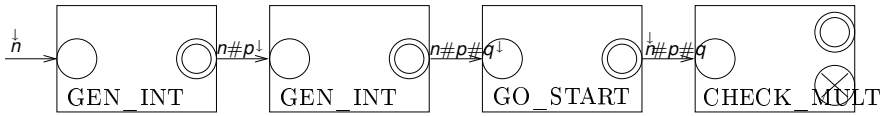
Olkoon edelleen GEN_INT seuraava mielivaltaisen ykköstä suuremman binääriluvun nauhan loppuun tuottava epädeterministinen Turingin kone:



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:

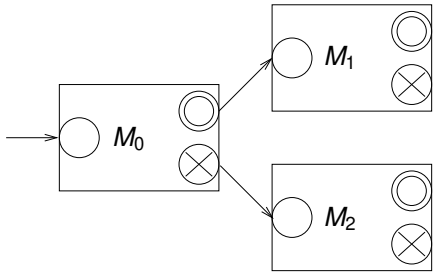







Yhdistetty kone hyväksyy syötteenä annetun binääriluvun $n$, jos ja vain jos on olemassa binääriluvut $p, q \geq 2$, joilla $n = pq$ — siis jos ja vain jos $n$ on yhdistetty luku.

*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

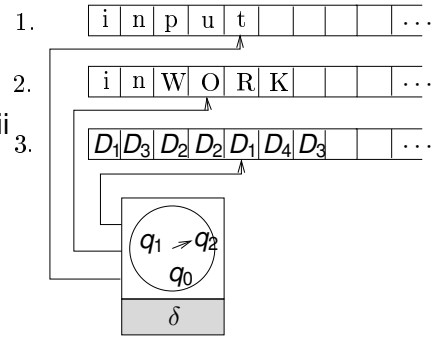
$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

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.

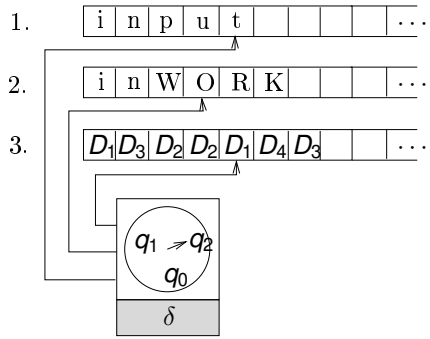




Yksityiskohtaisemmin:
Nauhalla 1 $\widehat{M}$ säilyttää kopiota syötejonosta ja nauhalla 2 se simuloi koneen $M$ työnauhaa. Kunkin simuloitavan laskennan aluksi $\widehat{M}$ kopioi syötteen nauhalta 1 nauhalle 2 ja pyyhkii pois nauhalle 2 edellisen laskennan jäljiltä mahdollisesti jääneet merkit.
Nauhalla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



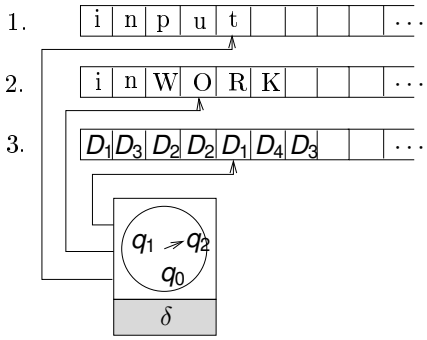
Tarkemmin sanoen, olkoon $r$ suurin $M$:n siirtymäfunktion arvojoukon koko. Tällöin $\widehat{M}$:lla on erityiset nauhamerkit $D_1, \ldots, D_r$, joista koostuvia jonoja se generoi nauhalle 3 kanonisessa järjestyksessä: $\varepsilon$, $D_1$, $D_2$, $\ldots$, $D_r$, $D_1D_1$, $D_1D_2$, $\ldots$, $D_1D_r$,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





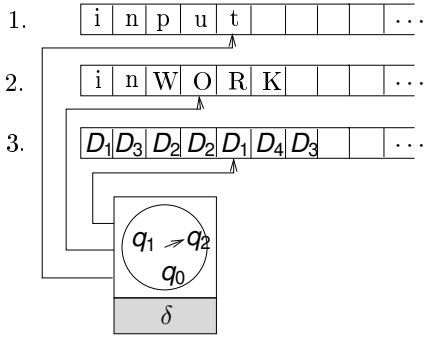



Esimerkiksi jos kolmosnauhalla on jono $D_1D_3D_2$, niin ensimmäisessä siirtymässä valitaan vaihtoehto 1, toisessa vaihtoehto 3, kolmannessa vaihtoehto 2. Ellei tämä laskenta johtanut $M$:n hyväksyvään lopputilaan, generoidaan seuraava koodijono $D_1D_3D_3$ ja aloitetaan alusta.
Jos koodijono on epäkelpo, 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. □