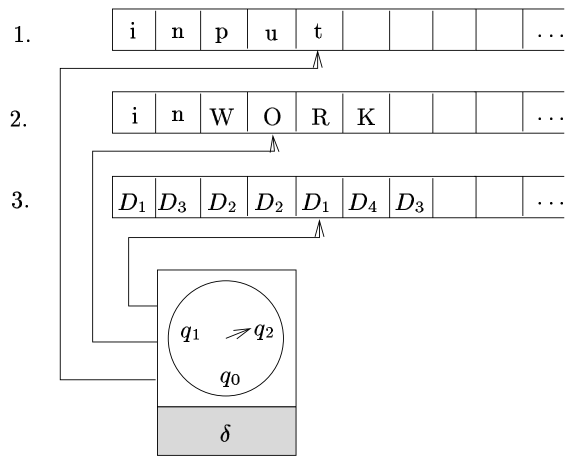
- Selvästi tämä systemaattinen koneen M laskentojen läpikäynti johtaa koneen $\widehat{M}$ hyväksymään syötejonon, jos ja vain jos koneella M on syötteen hyväksyvä laskenta. Jos hyväksyvää laskentaa ei ole, kone $\widehat{M}$ ei pysähdy. $\square$



Kuva 5.12: Epädeterministisen Turingin koneen simulointi deterministisellä

5.4 Rajoittamattomat ja kontekstiset kieliopit

Rajoittamattomat kieliopit (*unrestricted grammars*) eli yleiset muunnossysteemit (*string rewriting systems*)

- yleistetään kontekstittomia kielioppeja sallimalla produktiossa yhden välikkeen sijaan minkä tahansa välikkeistä ja päätteistä koostuvan merkkijonon korvaaminen toisella (mahd. tyhjällä merkkijonolla)
- esim. $aB \to BC|d$ ja $a \to B|\epsilon$ ovat nyt sallittuja sääntöjä, mutta $\epsilon \to A$ ei ole.
- säännöt siis muotoa $\omega \to \omega'$, missä $\omega, \omega' \in V^*$ (V koko aakkosto) ja $\omega \neq \epsilon$
- Tärkeä tulos: **Kieli voidaan tuottaa rajoittamattomalla kieliopilla jos ja vain jos se voidaan tunnistaa Turingin koneella.**

Määritelmä: Rajoittamaton kielioppi on nelikko

$$G = (V, \Sigma, P, S),$$

missä

- V on kieliopin aakkosto;
- $\Sigma \subseteq V$ on kieliopin *päätemerkkien* joukko; $N = V - \Sigma$ on *välikemerkkien* t. *-symbolien* joukko;
- $P \subseteq V^+ \times V^*$ on kieliopin *sääntöjen* t. *produktioiden* joukko ($V^+ = V^* - \{\epsilon\}$);
- $S \in N$ on kieliopin *lähtösymboli.*

Produktiota $(\omega, \omega') \in P$ merkitään tavallisesti $\omega \to \omega'$.

- Merkkijono $\gamma \in V^*$ *tuottaa* t. *johtaa suoraan* merkkijonon $\gamma' \in V^*$ kieliopissa G, merkitään
$$\gamma \underset{G}{\Rightarrow} \gamma'$$
jos voidaan kirjoittaa $\gamma = \alpha\omega\beta$, $\gamma' = \alpha\omega'\beta$ ($\alpha, \beta, \omega' \in V^*$, $\omega \in V^+$), ja kieliopissa on produktio $\omega \to \omega'$.
- Jos kielioppi G on yhteydestä selvä, merkitään yksinkertaisesti $\gamma \Rightarrow \gamma'$.
- Merkkijono $\gamma \in V^*$ *tuottaa* t. *johtaa* merkkijonon $\gamma' \in V^*$ kieliopissa G, merkitään
$$\gamma \underset{G}{\Rightarrow}^* \gamma'$$
jos on olemassa jono V:n merkkijonoja $\gamma_0, \gamma_1, \ldots, \gamma_n$ ($n \geq 0$), siten että
$$\gamma = \gamma_0 \underset{G}{\Rightarrow} \gamma_1 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} \gamma_n = \gamma'.$$
- Jälleen, jos G on yhteydestä selvä, merkitään yksinkertaisesti $\gamma \Rightarrow^* \gamma'$.
- Merkkijono $\gamma \in V^*$ on kieliopin G *lausejohdos*, jos on $S \underset{G}{\Rightarrow}^* \gamma$.
- Pelkästään päätemerkeistä koostuva G:n lausejohdos $x \in \Sigma^*$ on G:n *lause.*
- Kieliopin G *tuottama* t. *kuvaama kieli* $L(G)$ koostuu G:n lauseista, s.o.:
$$L(G) = \{x \in \Sigma^* \mid S \underset{G}{\Rightarrow}^* x\}.$$

Esimerkki: rajoittamaton kielioppi ei-kontekstittomalle kielelle $\{a^k b^k c^k \mid k \geq 0\}$.

$$
\begin{aligned}
S &\rightarrow LT \mid \epsilon \\
T &\rightarrow ABCT \mid ABC \\
BA &\rightarrow AB \\
CB &\rightarrow BC \\
CA &\rightarrow AC \\
LA &\rightarrow a \\
aA &\rightarrow aa \\
aB &\rightarrow ab \\
bB &\rightarrow bb \\
bC &\rightarrow bc \\
cC &\rightarrow cc.
\end{aligned}
$$

Esimerkiksi lauseen $aabbcc$ johto:

$$
\begin{array}{lllllll}
\underline{S} & \Rightarrow & L\underline{T} \Rightarrow LABC\underline{T} & \Rightarrow & LAB\underline{CA}BC & \Rightarrow & LA\underline{BA}CBC \\
 & \Rightarrow & LAAB\underline{CB}C & \Rightarrow & \underline{LA}ABBCC & \Rightarrow & \underline{aA}BBCC \\
 & \Rightarrow & a\underline{aB}BCC & \Rightarrow & aa\underline{bB}CC & \Rightarrow & aab\underline{bC}C \\
 & \Rightarrow & aabb\underline{cC} & \Rightarrow & aabbcc. & &
\end{array}
$$

Lause: Jos formaali kieli L voidaan tuottaa rajoittamattomalla kieliopilla, se voidaan tunnistaa Turingin koneella.

**Todistus:*

Olkoon $G = (V, \Sigma, P, S)$ kielen L tuottava rajoittamaton kielioppi. Muodostetaan kielen L tunnistava kaksinauhainen epädeterministinen Turingin kone M_G seuraavasti:

- Nauhalla 1 kone säilyttää kopiota syötejonosta.
- Nauhalla 2 on kullakin hetkellä jokin G:n lausejohdos, jota kone pyrkii muuntamaan syötejonon muotoiseksi.
- Toimintansa aluksi M_G kirjoittaa kakkosnauhalle kieliopin lähtösymbolin S.
- Koneen M_G laskenta koostuu vaiheista. Kussakin vaiheessa kone:

(i) vie kakkosnauhan nauhapään epädeterministisesti johonkin kohtaan nauhalla;

(ii) valitsee epädeterministisesti jonkin G:n produktion, jota yrittää soveltaa valittuun nauhankohtaan (produktiot on koodattu M_G:n siirtymäfunktioon);

(iii) jos produktion vasen puoli sopii yhteen nauhalla olevien merkkien kanssa, M_G korvaa ao. merkit produktion oikean puolen merkeillä;

(iv) vaiheen lopuksi M_G vertaa ykkös- ja kakkosnauhan merkkijonoja toisiinsa: jos jonot ovat samat, kone siirtyy hyväksyvään lopputilaan ja pysähtyy, muuten aloittaa uuden vaiheen (kohta (i)). $\square$

Lause: Jos formaali kieli L voidaan tunnistaa Turingin koneella, se voidaan tuottaa rajoittamattomalla kieliopilla.

**Todistus.* Olkoon $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{yes}}, q_{\text{no}})$ kielen L tunnistava standardimallinen Turingin kone. Muodostetaan kielen L tuottava rajoittamaton kielioppi G_M seuraavasti.

Idea:

- Kieliopin G_M välikkeiksi otetaan (muiden muassa) kaikkia M:n tiloja $q \in Q$ edustavat symbolit.
- Koneen M tilanne $(q, u\underline{a}v)$ esitetään merkkijonona $[uqav]$.
- M:n siirtymäfunktion perusteella G_M:ään muodostetaan produktiot, joiden ansiosta

$$[uqav] \underset{G_M}{\Rightarrow} [u'q'a'v'] \quad \text{joss} \quad (q, uav) \underset{M}{\vdash} (q', u'a'v').$$

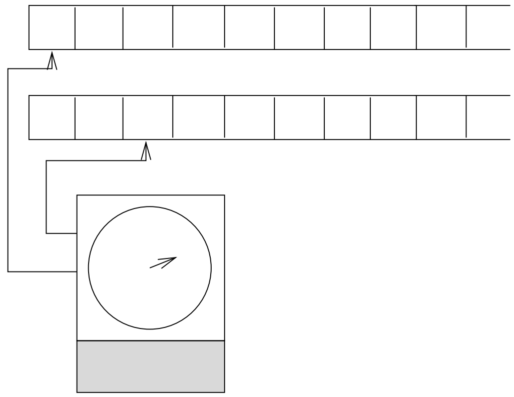
- Siten M hyväksyy syötteen x, jos ja vain jos

$$[q_0 x] \underset{G_M}{\Rightarrow}^* [u q_{\text{yes}} v]$$

joillakin $u, v \in \Sigma^*$.

Kaikkiaan kielioppiin G_M tulee kolme ryhmää produktioita:

1. Produktiot, joilla lähtösymbolista S voidaan tuottaa mikä tahansa merkkijono muotoa $x[q_0 x]$, missä $x \in \Sigma^*$ ja [, q_0,] ja ovat G_M:n välikkeitä.
2. Produktiot, joilla merkkijonosta $[q_0 x]$ voidaan tuottaa merkkijono $[u q_{\text{yes}} v]$, jos ja vain jos M hyväksyy x:n.



Kuva 5.13: Rajoittamattoman kieliopin tuottaman kielen tunnistaminen Turingin koneella.

3. Produktiot, joilla muotoa $[uq_{\text{yes}}v]$ oleva merkkijono muutetaan tyhjäksi merkkijonoksi.

Kieleen $L(M)$ kuuluvan merkkijonon x tuottaminen tapahtuu tällöin seuraavasti:

$$S \overset{(1)}{\Rightarrow^*} x[q_0x] \overset{(2)}{\Rightarrow^*} x[uq_{\text{yes}}v] \overset{(3)}{\Rightarrow^*} x.$$

Täsmällisesti määritellään $G = (V, \Sigma, P, S)$, missä

$$V = \Gamma \cup Q \cup \{S, T, [,], E_L, E_R\} \cup \{A_a \mid a \in \Sigma\},$$

ja produktiot P muodostuvat seuraavista kolmesta ryhmästä:

1. Alkutilanteen tuottaminen:

$$\begin{array}{lcll} S & \rightarrow & T[q_0] & \\ T & \rightarrow & \epsilon & \\ T & \rightarrow & aTA_a & (a \in \Sigma) \\ A_a[q_0 & \rightarrow & [q_0A_a & (a \in \Sigma) \\ A_ab & \rightarrow & bA_a & (a, b \in \Sigma) \\ A_a] & \rightarrow & a] & (a \in \Sigma) \end{array}$$

2. M:n siirtymien simulointi $(a, b \in \Gamma,\ c \in \Gamma \cup \{[\})$:

Siirtymät:	*Produktiot:*
$\delta(q,a) = (q', b, R)$	$qa \rightarrow bq'$
$\delta(q,a) = (q', b, L)$	$cqa \rightarrow q'cb$
$\delta(q,>) = (q', >, R)$	$q[\rightarrow [q'$
$\delta(q,<) = (q', b, R)$	$q] \rightarrow bq']$
$\delta(q,<) = (q', b, L)$	$cq] \rightarrow q'cb]$
$\delta(q,<) = (q', <, L)$	$cq] \rightarrow q'c]$

3. Lopputilanteen siivous:

$$
\begin{array}{lcll}
q_{\text{yes}} & \rightarrow & E_L E_R & \\
q_{\text{yes}}[& \rightarrow & E_R & \\
aE_L & \rightarrow & E_L & (a \in \Gamma) \\
[E_L & \rightarrow & \epsilon & \\
E_R a & \rightarrow & E_R & (a \in \Gamma) \\
E_R] & \rightarrow & \epsilon &
\end{array}
$$

□

Esim. Tarkastellaan seuraavaa kasvikielioppia, johon on lisätty lähtösymboli S ja sille säänöt, jotta on saatu tavallinen rajoittamaton kielioppi. Kieliopin aakkosto on $V = \{S, SIEMEN, SIRKKALEHDET, LEHDET, VARSI, NUPPU, PUNKUKKA, SINKUKKA, PAIVA, YO\}$. (Huom! Koska tämä on kasvikielioppi, emme erottele pääte- ja välikesymboleja – ”jäsennys” jatkuu ikuisesti, ellei koko populaatio satu kuolemaan.)

$S \rightarrow$ SIEMEN PAIVA | SIEMEN YO
SIEMEN YO$\rightarrow$ SIRKKALEHDET YO
SRKKALEHDET PAIVA$\rightarrow$ LEHDET PAIVA| VARSI PAIVA
LEHDET PAIVA$\rightarrow$ VARSI PAIVA
VARSI PAIVA$\rightarrow$ NUPPU PAIVA| LEHDET PAIVA
NUPPU YO$\rightarrow$ PUNKUKKA YO| SINKUKKA YO
PUNKUKKA $\rightarrow$ SIEMEN | SIEMEN SIEMEN |ϵ
SINKUKKA $\rightarrow$ SIEMEN | SIEMEN SIEMEN |ϵ
PAIVA $\rightarrow$ YO
YO $\rightarrow$ PAIVA

Nyt lähtösymbolista S voidaan johtaa esim. seuraavat lausejohdokset:

$S \Rightarrow$ SIEMEN PAIVA$\Rightarrow$ SIEMEN YO $\Rightarrow$ SIRKKALEHDET YO $\Rightarrow$ SIRKKALEHDET PAIVA $\Rightarrow$ LEHDET PAIVA $\Rightarrow$ VARSI PAIVA $\Rightarrow$ NUPPU PAIVA $\Rightarrow$ NUPPU YO $\Rightarrow$ PUNKUKKA YO $\Rightarrow$ PUNKUKKA PAIVA $\Rightarrow$ PUNKUKKA YO $\Rightarrow$ SIEMEN SIEMEN YO $\Rightarrow$ ...

Kontekstiset kieliopit (*context-sensitive grammars*)

- Rajoittamaton kielioppi on *kontekstinen*, jos sen kaikki produktiot ovat muotoa $\omega \to \omega'$, missä $|\omega'| \geq |\omega|$, tai mahdollisesti $S \to \epsilon$, missä S on lähtösymboli.
- ts.lavennettaessa merkkijono voi vain kasvaa, ei koskaan pienentyä, paitsi lähtösymboli, joka saa tuottaa ϵ:in, jos ϵ kuuluu kieleen
- Lisäksi vaaditaan, että jos kieliopissa on produktio $S \to \epsilon$, niin lähtösymboli S ei esiinny minkään produktion oikealla puolella.
- Kontekstilliset kielet ovat siis rajoittamattomien kielten osaluokka!
- esim. säännöt
  tietojenkäsittelijä $\to$ *tietojenkäpistelijä*, *tieto* $\to$ *taito*
  $SUBJ$ *on* $PREDIKAT \to SUBJ$ *oli* $PREDIKAT | SUBJ$ *on ollut* $PREDIKAT |$
  $SUBJ$ *oli ollut* $PREDIKAT$

  ovat kontekstillisia
- Formaali kieli L on *kontekstinen*, jos se voidaan tuottaa jollakin kontekstisella kieliopilla.
- Normaalimuotolause: produktiot saadaan muotoon $S \to \epsilon$ ja $\alpha A \beta \to \alpha\omega\beta$, missä A on välike ja $\omega \neq \epsilon$. (Säännön $A \to \omega$ sovellus “kontekstissa” $\alpha\,\_\,\beta$.)
- esim. $PAIVA\ SIEMEN \to PAIVA\ SIRKKALEHTI$, $PAIVA \to YO$: nyt sääntöä $SIEMEN \to SIRKKALEHTI$ saa soveltaa vain kontekstissa $PAIVA\_\epsilon$ (vain α muodostaa siis kontekstin ja β puuttuu)

Lause: Formaali kieli L on kontekstinen, jos ja vain jos se voidaan tunnistaa epädeterministisellä Turingin koneella, joka ei tarvitse enempää työtilaa kuin syötejonon pituuden verran — siis koneella, jolla ei ole muotoa $\delta(q, <) = (q', b, \Delta)$ olevia siirtymiä, missä $b \neq$ ‘$<$’.

Todistusidea: kontekstillisen kieliopin mukaisessa jäsennyksessä lähtösymboli voi vain kasvaa joka jäsennysaskeleella, joten jos lähdetään merkkijonosta kohti lähtösymbolia, voi symbolijono vain pienentyä joka jäsennysaskeleella. □

- Lauseen koneita sanotaan *lineaarisesti rajoitetuiksi automaateiksi.*
- Lineaarisesti rajoitetut automaatit tunnistavat siis täsmälleen kontekstilliset kielet.
- Avoin ongelma ("LBA = DLBA"): onko lauseessa vaadittu epädeterminismi välttämätöntä vai riittäisikö deterministinen kone?

Esimerkki Tiedämme, että kielen $L = \{a^k b^k c^k \mid k \geq 0\}$ voi tunnistaa syötteen vaatimassa tilassa (muutetaan merkkejä A:ksi, B:ksi ja C:ksi yksi kerrallaan), joten se on kontekstillinen. Voidaan antaa seuraava kielen kuvaava kontekstillinen kielioppi:

$S' \rightarrow S|\epsilon$
$S \rightarrow aSBc|abc$
$cB \rightarrow Bc$
$bB \rightarrow bb$

Esim. merkkijonon $aaabbbccc$ johto:
$\underline{S} \Rightarrow a\underline{S}Bc \Rightarrow aa\underline{S}BcBc \Rightarrow aaabc\underline{cB}cBc \Rightarrow aaa\underline{bB}ccBc \Rightarrow aaabbc\underline{cB}c \Rightarrow aaabb\underline{cB}cc \Rightarrow aaab\underline{bB}ccc \Rightarrow aaabbbccc$