# Chomskyn hierarkia

Formaalit kielet sijoitetaan perinteisesti vaikeutensa perusteella eri tasoille Chomskyn hierarkiaan:

taso 0: tunnistettavat kielet, voidaan tunnistaa Turingin koneella

taso 2: yhteydettömät kielet, voidaan tunnistaa pinoautomaatilla ja tuottaa yhteydettömällä kieliopilla

taso 3: säännölliset kielet, voidaan tunnistaa äärellisellä automaatilla ja tuottaa oikealle lineaarisella kieliopilla (ks. harj. 7) tai säännöllisellä lausekkeella.

Tarkastelemme tässä lyhyesti kurssin "formaalien kielten maailmankuvaan" ajanpuutteen vuoksi jääneitä aukkoja:

- Millaiset kieliopit vastaavat Turingin koneita (taso 0)?
- Mitä on tasolla 1?

Tämä ei kuulu kurssin koealueeseen, mutta saattaa helpottaa yleiskuvan saamista kurssin sisällöstä.

Turingin konetta vastaava kielioppiformalismi on rajoittamattomat kieliopit.

Rajoittamaton kielioppi on nelikko $G = (V, \Sigma, R, S)$, missä

1. $V$ on äärellinen muuttujien joukko
2. $\Sigma$ on päätesymbolien joukko, jolla $\Sigma \cap V = \emptyset$
3. $R$ on äärellinen joukko sääntöjä muotoa $u \to v$, missä $u \in (\Sigma \cup V)^+$ ja $v \in (\Sigma \cup V)^*$ ja
4. $S \in V$ on lähtösymboli.

Tämä laajentaa yhteydettömiä kielioppeja siten, että säännön $u \to v$ vasen puoli $u$ saa olla mikä tahansa epätyhjä merkkijono, ei pelkästään yksi muuttujasymboli.

Kuten yhteydettömien kielioppien tapauksessa, merkitsemme $w \Rightarrow w'$, ja sanomme että $w$ johtaa suoraan merkkijonon $w'$, jos voidaan kirjoittaa $w = xuy$ ja $w' = xvy$, missä $(u \to v) \in R$. Jos $w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_{n-1} \Rightarrow w_n$, sanomme että $w_0$ johtaa merkkijonon $w_n$, ja merkitsemme $w_0 \stackrel{*}{\Rightarrow} w_n$. Kieliopin tuottama kieli on $L(G) = \left\{ w \in \Sigma^* \mid S \stackrel{*}{\Rightarrow} w \right\}$.

**Esimerkki** Kieli

$$A = \{ \mathrm{a}^i\mathrm{b}^i\mathrm{c}^i \mid i \in \mathbb{N} \}$$

tunnetusti ei ole yhteydetön. Se voidaan kuitenkin tuottaa rajoittamattomalla kieliopilla

$$\begin{array}{rcl} S & \to & \mathrm{a}A\mathrm{bc} \mid \mathrm{abc} \mid \varepsilon \\ A & \to & \mathrm{a}A\mathrm{b}C \mid \mathrm{ab}C \\ C\mathrm{b} & \to & \mathrm{b}C \\ C\mathrm{c} & \to & \mathrm{cc}, \end{array}$$

missä on käytetty samoja merkintäkonventioita kuin yhteydettömille kieliopeille.

Esim. merkkijonolle aaabbbccc saadaan johto

$$\begin{aligned} S \Rightarrow \mathrm{a}A\mathrm{bc} \Rightarrow \mathrm{aa}A\mathrm{b}C\mathrm{bc} \Rightarrow \mathrm{aaab}C\mathrm{b}C\mathrm{bc} \Rightarrow \mathrm{aaab}C\mathrm{bb}C\mathrm{c} \\ \Rightarrow \mathrm{aaabb}C\mathrm{b}C\mathrm{c} \Rightarrow \mathrm{aaabbb}CC\mathrm{c} \Rightarrow \mathrm{aaabbb}C\mathrm{cc} \Rightarrow \mathrm{aaabbbccc}. \end{aligned}$$

□

Kielioppien ja Turingin koneiden yhteys on seuraava:

**Lause** Kieli on Turing-tunnistettava, jos ja vain jos jokin rajoittamaton kielioppi tuottaa sen.

**Todistushahmotelma:** Kun on annettu rajoittamaton kielioppi $G$, kieli $L(G)$ voidaan luetella seuraavasti:

1. Käy läpi kaikki yhden pituiset johdot $S \Rightarrow w_1$ ja tulosta kaikki päätemerkkijonot $w_1 \in \Sigma^*$.
2. Käy läpi kaikki kahden pituiset johdot $S \Rightarrow w_1 \Rightarrow w_2$ ja tulosta kaikki päätemerkkijonot $w_2 \in \Sigma^*$.
3. Käy läpi kaikki kolmen pituiset johdot $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow w_3$ ja tulosta kaikki päätemerkkijonot $w_3 \in \Sigma^*$.
4. ....

Muodostetaan Turingin kone $M$, joka syötteellä $w$ simuloi tätä prosessia ja hyväksyy, jos merkkijono $w$ tulostuu jossain vaiheessa. Jos merkkijonoa $w$ koskaan ei tule, $M$ jää ikuiseen silmukkaan (paitsi jos $L(G)$ on äärellinen, jolloin $M$ voi hylätä, kun uusia merkkijonoja ei enää tule). Nyt $M$ tunnistaa kielen $L(G)$.

Mielenkiintoisempi suunta on muodostaa annetulle Turingin koneelle $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ kielioppi $G = (V, \Sigma, R, S)$, jolla $L(G) = L(M)$.

Perusidea on tulkita Turingin koneen tilanteet

$$u_1 u_2 \dots u_n \, q \, v_1 v_2 \dots v_n, \qquad u_i, v_i \in \Gamma, q \in Q,$$

merkkijonoiksi. Tätä varten valitaan kieliopin muuttujiksi $V = Q \cup (\Gamma - \Sigma)$, jolloin tilanteet ovat suoraan aakkoston $V \cup \Sigma$ merkkijonoja.

Turingin koneen laskennan esittämiseksi liitetään kielioppiin sääntöjä seuraavasti:

- Jos $\delta(r, a) = (s, b, \mathsf{R})$, lisätään sääntö $ra \to bs$.
- Jos $\delta(r, a) = (s, b, \mathsf{L})$, lisätään sääntö $cra \to scb$ kaikilla $c \in \Gamma$.

Nyt Turingin koneen laskenta-askelta $uqv \vdash u'q'v'$ vastaa kieliopin suora johto $uqv \Rightarrow u'q'v'$.

Laskennan alun ja lopun vaatimat yksityiskohdat sivuutetaan (ks. esim. Sudkamp: Languages and Machines). □

Yhteydettömien kielioppien ohella toinen tärkeä erikoistapaus on yhteysherkät (context-sensitive) kieliopit, jotka määrittävät hierarkian tason 1.

Yhteysherkässä kieliopissa kaikilla säännöillä $u \to v$ pitää olla $|u| \leq |v|$. Poikkeuksena sallitaan kuitenkin sääntö $S \to \varepsilon$ edellyttäen, että $S$ ei esiinny minkään säännön oikealla puolella. Kieli $A$ on yhteysherkkä, jos $A = L(G)$ jollain yhteysherkällä $G$.

**Esimerkki** Yhteydettömän kieliopin säännöissä $u \to v$ pätee aina $|u| = 1$. Poistamalla $\varepsilon$-säännöt (kuten Chomskyn normaalimuodon yhteydessä) yhteydetön kielioppi saadaan yhteysherkkään muotoon. Siis yhteydettömät kielet ovat yhteysherkkiä.

Toisaalta edellinen esimerkki itse asiassa osoittaa kielen $\{ \mathrm{a}^i\mathrm{b}^i\mathrm{c}^i \mid i \in \mathbb{N} \}$ yhteysherkäksi. Siis kaikki yhteysherkät kielet eivät ole yhteydettömiä. □

Nimitys "yhteysherkkä" tulee siitä, että tällaisen kieliopin säännöt voidaan muuntaa muotoon $xAy \to xwy$, missä $A \in V$, $x, y \in (V \cup \Sigma)^*$ ja $w \in (V \cup \Sigma)^+$. Siis sääntöä $A \to w$ saadaan soveltaa vain "kontekstissa" $x \_ y$.

**Lause** Yhteysherkät kielet ovat ratkeavia.

**Todistushahmotelma:** Erikoistapausta $S \Rightarrow \varepsilon$ lukuunottamatta missä tahansa yhteysherkän kieliopin johdossa

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n$$

pätee

$$1 \leq |w_1| \leq |w_2| \leq \ldots \leq |w_n|.$$

Siis erityisesti jos kysytään jostakin merkkijonosta $w$, päteekö $S \stackrel{*}{\Rightarrow} w$, niin riittää tarkastella johtoja, joiden välivaiheiden $w_i$ pituudet $|w_i|$ ovat korkeintaan $|w|$. Koska tällaisia välivaiheita on äärellinen määrä, voidaan esim. käydä läpi kaikki niiden järjestykset ja katsoa, muodostuuko laillinen johto. □

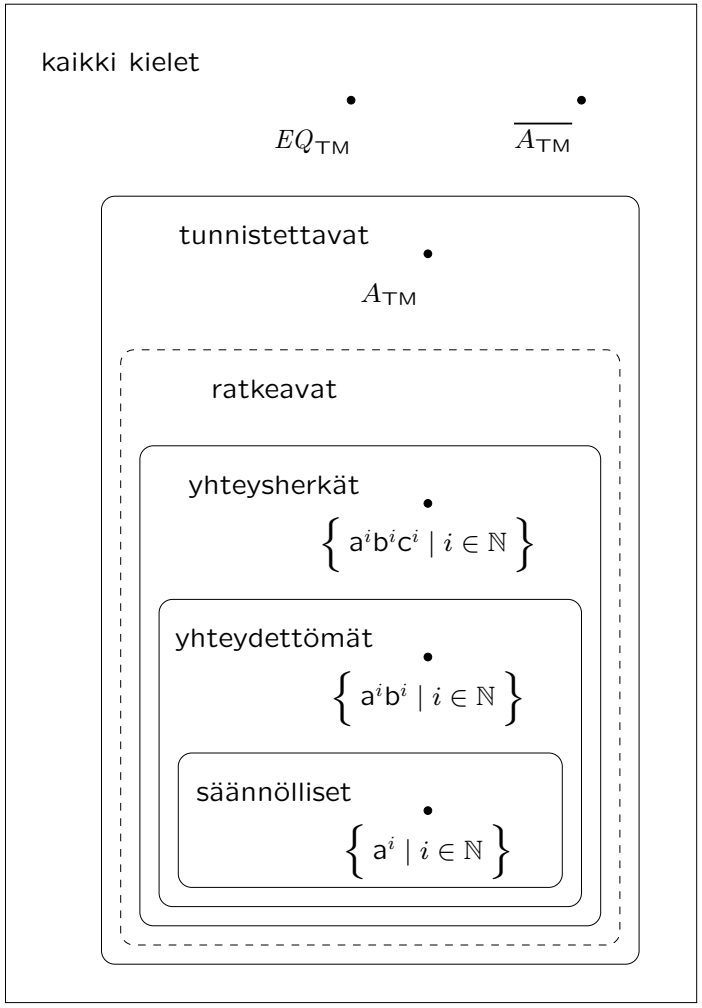
Tarkemmin voidaan osoittaa, että kieli on yhteysherkkä, jos ja vain jos se voidaan tunnistaa lineaarisesti rajoitetulla automaatilla (linear-bounded automaton, LBA). Tällainen automaatti on epädeterministinen Turingin kone, joka ei käytä nauhatilaa enempää kuin syötteen pituuden verran, ts. ei koskaan kirjoita mitään tyhjämerkin päälle.

Yleisiä, yhteysherkkiä, yhteydettömiä ja oikealle lineaarisia kielioppeja kutsutaan vastaavasti tyypin 0, 1, 2 ja 3 kieliopeiksi. Saadaan seuraava Chomskyn hierarkia:

| tyyppi | kieli | kielioppi | automaatti |
|---|---|---|---|
| 0 | tunnistettava | rajoittamaton | Turingin kone |
| 1 | yhteysherkkä | yhteysherkkä | lin. rajoitettu |
| 2 | yhteydetön | yhteydetön | pinoautom. |
| 3 | säännöllinen | oikealle lineaarinen | äärellinen autom. |

(Oikealle lineaarisista kieliopeista ks. harjoitus 7.) Ylempi taso sisältää myös kaikkien alempien tasojen kielet. Lisäksi tiedämme, että

- kaikki kielet eivät ole edes tunnistettavia,
- tasot ovat erillisiä (esim. on olemassa yhteysherkkiä ei-yhteydettömiä kieliä) ja
- yhteysherkät $\subset$ ratkeavat $\subset$ tunnistettavat.




Chomskyn hierarkia ja eräiden kielten sijainti sen suhteen