3.6 Cocke-Younger-Kasami -jäsennysalgoritmi

Osittava jäsentäminen on selkeä ja tehokas jäsennysmenetelmä LL(1)-kieliopeille: n merkin mittaisen syötemerkkijonon käsittely sujuu ajassa $O(n)$. LL(1)-kieliopit ovat kuitenkin melko rajoitettu luokka; yleisen jäsennysongelman ratkaisu ei ole yhtä helppoa.

Periaatteessa ongelma voidaan ratkaista esim. soveltamalla yleistä (peruuttavaa) osittavaa jäsennystä, mutta käytännössä vaikeudeksi muodostuu erilaisten kokeiltavien johtovaihtoehtojen suuri määrä. (Tyypillisesti $O(c^n)$ kpl jollakin $c \geq 2$.)

Cocke-Younger-Kasami -algoritmi on yleiseen ns. dynaamisen ohjelmoinnin tekniikkaan (t. osaratkaisujen taulukointiin) perustuva menetelmä mielivaltaisen yhteydettömän kieliopin tuottamien merkkijonojen tunnistamiseen. Menetelmä toimii ajassa $O(n^3)$, missä n on tutkittavan merkkijonon pituus.

Algoritmia varten määritellään ensin joitakin kielioppimuunnoksia.

1. ε-produktioiden poistaminen

Olkoon $G = (V, \Sigma, P, S)$ yhteydetön kielioppi. Välike $A \in V - \Sigma$ on *tyhjentyvä*, jos $A \underset{G}{\Rightarrow}^* \varepsilon$.

Lemma 3.5. Mistä tahansa yhteydettömästä kieliopista G voidaan muodostaa ekvivalentti kielioppi G', jossa enintään lähtösymboli on tyhjentyvä.

Todistus. Olkoon $G = (V, \Sigma, P, S)$. Selvitetään ensin G:n tyhjentyvät välikkeet seuraavasti:

(i) asetetaan aluksi

NULL := $\{A \in V - \Sigma \mid A \to \varepsilon$ on G:n produktio$\}$;

(ii) toistetaan sitten seuraavaa NULL-joukon laajennusoperaatiota, kunnes joukko ei enää kasva:

$$\text{NULL} := \text{NULL} \cup \{A \in V - \Sigma \mid A \to B_1 \dots B_k \text{ on } G\text{:n prod.}, B_i \in \text{NULL kaikilla } i = 1, \dots, k\}.$$

Tämän jälkeen korvataan kukin G:n produktio $A \to X_1 \dots X_k$ kaikkien sellaisten produktioiden joukolla, jotka ovat muotoa

$$A \to \alpha_1 \dots \alpha_k, \quad \text{missä}$$

$$\alpha_i = \begin{cases} X_i, & \text{jos } X_i \notin \text{NULL}; \\ X_i \text{ tai } \varepsilon, & \text{jos } X_i \in \text{NULL}. \end{cases}$$

Lopuksi poistetaan kaikki muotoa $A \to \varepsilon$ olevat produktiot. Jos poistettavana on myös produktio $S \to \varepsilon$, otetaan muodostettavaan kielioppiin G' uusi lähtösymboli S' ja sille produktiot $S' \to S$ ja $S' \to \varepsilon$. □

Esimerkki. Poistetaan ε-produktiot kieliopista:

$$\begin{aligned} S &\to A \mid B \\ A &\to aBa \mid \varepsilon \qquad \Rightarrow \qquad (\text{NULL} = \{A, B, S\}) \\ B &\to bAb \mid \varepsilon \end{aligned}$$

$$\begin{aligned} S &\to A \mid B \mid \varepsilon \\ A &\to aBa \mid aa \mid \varepsilon \qquad \Rightarrow \\ B &\to bAb \mid bb \mid \varepsilon \end{aligned}$$

$$\begin{aligned} S' &\to S \mid \varepsilon \\ S &\to A \mid B \\ A &\to aBa \mid aa \\ B &\to bAb \mid bb. \end{aligned}$$

2. Yksikköproduktioiden poistaminen

Produktio muotoa $A \to B$, missä A ja B ovat välikkeitä, on *yksikköproduktio*.

Lemma 3.6. Mistä tahansa yhteydettömästä kieliopista G voidaan muodostaa ekvivalentti kielioppi G', jossa ei ole yksikköproduktioita.

Todistus. Olkoon $G = (V, \Sigma, P, S)$. Selvitetään ensin G:n kunkin välikkeen "yksikköseuraajat" seuraavasti:

(i) asetetaan aluksi kullekin $A \in V - \Sigma$:

$F(A) := \{B \in V - \Sigma \mid A \to B$ on G:n produktio$\}$;

(ii) toistetaan sitten seuraavia F-joukkojen laajennusoperaatiota, kunnes joukot eivät enää kasva:

$F(A) := F(A) \cup \bigcup \{F(B) \mid A \to B$ on G:n produktio$\}$.

Tämän jälkeen poistetaan G:stä kaikki yksikköproduktiot ja lisätään niiden sijaan kaikki mahdolliset produktiot muotoa $A \to \omega$, missä $B \to \omega$ on G:n ei-yksikköproduktio jollakin $B \in F(A)$. □

Esimerkki. Poistetaan yksikköproduktiot aiemmin muodostetusta kieliopista:

$$\begin{aligned} S' &\to S \mid \varepsilon \\ S &\to A \mid B \\ A &\to aBa \mid aa \\ B &\to bAb \mid bb. \end{aligned}$$

Välikkeiden yksikköseuraajat ovat: $F(S') = \{S, A, B\}$, $F(S) = \{A, B\}$, $F(A) = F(B) = \emptyset$. Korvaamalla yksikköproduktiot edellä esitetyllä tavalla saadaan kielioppi:

$$\begin{aligned} S' &\to aBa \mid aa \mid bAb \mid bb \mid \varepsilon \\ S &\to aBa \mid aa \mid bAb \mid bb \\ A &\to aBa \mid aa \\ B &\to bAb \mid bb. \end{aligned}$$

(Huomataan, että välike S on nyt itse asiassa "turha", so. se ei voi esiintyä minkään kieliopin lauseen johdossa. Myös turhat välikkeet voidaan haluttaessa poistaa kieliopista samantapaisella algoritmilla (HT).)

Chomskyn normaalimuoto

Yhteydetön kielioppi $G = (V, \Sigma, P, S)$ on *Chomskyn normaalimuodossa*, jos sen välikkeistä enintään S on tyhjentyvä, ja mahdollista produktiota $S \to \varepsilon$ lukuunottamatta muut produktiot ovat muotoa

$$A \to BC \qquad \text{tai} \qquad A \to a,$$

missä A, B ja C ovat välikkeitä ja a on päätemerkki.

Lisäksi vaaditaan yksinkertaisuuden vuoksi, että lähtösymboli S ei esiinny minkään produktion oikealla puolella.

Lause 3.7. Mistä tahansa yhteydettömästä kieliopista G voidaan muodostaa ekvivalentti Chomskyn normaalimuotoinen kielioppi G'.

Todistus. Olkoon $G = (V, \Sigma, P, S)$. Poistetaan ensin G:stä ε-produktiot ja yksikköproduktiot lemmojen 3.5 ja 3.6 konstruktioilla. Tämän jälkeen kaikki G:n produktiot ovat muotoa $A \to a$ tai $A \to X_1 \dots X_k$, $k \geq 2$ (tai $S \to \varepsilon$).

Lisätään ensin kielioppiin kutakin päätemerkkiä a varten uusi välike C_a ja sille produktio $C_a \to a$. Korvataan sitten kussakin muotoa $A \to X_1 \dots X_k$, $k \geq 2$, olevassa produktiossa ensin kaikki päätemerkit em. uusilla välikkeillä, ja sitten koko produktio produktiojoukolla

$$\begin{aligned} A &\to X_1 A_1 \\ A_1 &\to X_2 A_2 \\ &\vdots \\ A_{k-2} &\to X_{k-1} X_k, \end{aligned}$$

missä $A_1, \dots, A_{k-2}$ ovat jälleen uusia välikkeitä. $\square$

Esimerkki. Kielioppi:

$$\begin{aligned} S &\to aBCd \mid bbb \\ B &\to b \\ C &\to c \end{aligned}$$

Em. konstruktiolla saatu Chomskyn normaalimuoto:

$$\begin{aligned} S &\to C_a S_1^1 \\ S_1^1 &\to B S_2^1 \\ S_2^1 &\to C C_d \\ S &\to C_b S_1^2 \\ S_1^2 &\to C_b C_b \\ B &\to b \\ C &\to c \\ C_a &\to a \\ C_b &\to b \\ C_c &\to c \\ C_d &\to d. \end{aligned}$$

CYK-algoritmi

Olkoon $G = (V, \Sigma, P, S)$ yhteydetön kielioppi. Lauseen 3.7 nojalla voidaan olettaa, että G on Chomskyn normaalimuodossa. Kysymys, kuuluuko annettu merkkijono x kieleen $L(G)$ voidaan tällöin ratkaista seuraavasti:

Jos $x = \varepsilon$, niin $x \in L(G)$ joss $S \to \varepsilon$ on G:n produktio.

Muussa tapauksessa merkitään $x = a_1 \dots a_n$ ja tarkastellaan x:n eri osajonojen tuottamista.

Merkitään N_{ik}:lla niiden välikkeiden A joukkoa, joista voidaan tuottaa x:n positiosta i alkava, k merkin mittainen osajono:

$$\begin{aligned} N_{ik} &= \{A \in V - \Sigma \mid A \underset{G}{\Rightarrow}^* a_i \dots a_{i+k-1}\}, \\ &\quad 1 \leq i \leq i + k - 1 \leq n. \end{aligned}$$

Joukot N_{ik} voidaan laskea taulukoimalla lyhyistä osajonoista pitempiin seuraavassa esitettävällä tavalla. Selvästi on $x \in L(G)$ joss $S \in N_{1n}$.

Joukkojen N_{ik} laskeminen:

(i) asetetaan aluksi kaikilla $i = 1, \ldots, n$:

$N_{i1} := \{A \in V - \Sigma \mid A \to a_i$ on G:n produktio$\}$;

(ii) lasketaan sitten kaikilla $k = 2, \ldots, n$ ja kullakin k kaikilla $i = 1, \ldots, n-k+1$:

$$N_{ik} := \bigcup_{j=1}^{k-1} \{A \in V - \Sigma \mid A \to BC \text{ on } G\text{:n produktio, missä } B \in N_{ij} \text{ ja } C \in N_{i+j,k-j}\}. \quad \square$$

Esimerkki. Chomskyn normaalimuotoinen kielioppi G:

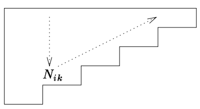
$$\begin{array}{lcl} S & \to & AB \mid BC \\ A & \to & BA \mid a \\ B & \to & CC \mid b \\ C & \to & AB \mid a \end{array}$$

CYK-algoritmin laskenta tällä kieliopilla ja syötejonolla $x = baaba$:

		$i \to$				
	N_{ik}	$1:b$	$2:a$	$3:a$	$4:b$	$5:a$
	1	B	A,C	A,C	B	A,C
	2	S,A	B	S,C	S,A	–
$k\downarrow$	3	$\emptyset$	B	B	–	
	4	$\emptyset$	S,A,C	–		
	5	S,A,C	–			

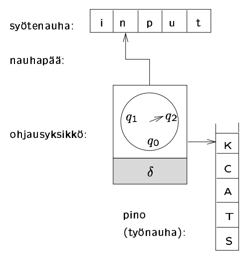
Koska lähtösymboli S kuuluu joukkoon N_{15}, päätellään että x kuuluu kieleen $L(G)$.

Yleisesti ottaen CYK-algoritmissa jotakin joukkoa N_{ik} määritettäessä edetään samanaikaisesti sarakkeessa N_{ij} joukkoa N_{ik} "kohti" ja diagonaalia $N_{i+j,k-j}$ pitkin siitä "poispäin":



3.7 Pinoautomaatit

Yhteydettömille kielille saadaan automaattikarakterisointi ns. *pinoautomaattien* avulla:



Intuitiivisesti pinoautomaatti on äärellinen automaatti, johon on lisätty yksi potentiaalisesti ääretön *työnauha*. Työnauhan käyttö ei kuitenkaan ole rajoittamatonta, vaan tieto on sillä organisoitu *pinoksi*: automaatti voi lukea ja kirjoittaa vain nauhan toiseen päähän, ja päästäkseen lukemaan aiemmin kirjoittamiaan merkkejä sen täytyy pyyhkiä viimeisin merkki pois.

Määritelmä 3.2 *Pinoautomaatti* on kuusikko

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F),$$

missä

- Q on *tilojen* äärellinen joukko;
- Σ on *syöteaakkosto*;
- Γ on *pinoaakkosto*;
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to \mathcal{P}(Q \times (\Gamma \cup \{\varepsilon\}))$ on (joukkoarvoinen) *siirtymäfunktio*;
- $q_0 \in Q$ on *alkutila*;
- $F \subseteq Q$ on *(hyväksyvien) lopputilojen* joukko.

Siirtymäfunktion arvon

$$\delta(q, \sigma, \gamma) = \{(q_1, \gamma_1), \ldots, (q_k, \gamma_k)\}$$

tulkinta on, että ollessaan tilassa q ja lukiessaan syötemerkin σ ja pinomerkin γ automaatti voi siirtyä johonkin tiloista $q_1, \ldots, q_k$ ja korvata vastaavasti pinon päällimmäisen merkin jollakin merkeistä $\gamma_1, \ldots, \gamma_k$. Pinoautomaatit ovat siis yleisessä tapauksessa *epädeterministisiä*.

Jos $\sigma = \varepsilon$, automaatti tekee siirtymän syötemerkkiä lukematta. Jos $\gamma = \varepsilon$, automaatti ei lue pinomerkkiä ja uusi kirjoitettu merkki tulee pinon päälle vanhaa päällimmäistä merkkiä poistamatta ("push"-operaatio). Jos pinosta luettu merkki on $\gamma \neq \varepsilon$ ja kirjoitettavana on $\gamma_i = \varepsilon$, pinosta poistetaan sen päällimmäinen merkki ("pop"-operaatio).

Automaatin *tilanne* on kolmikko $(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$; erityisesti automaatin *alkutilanne syötteellä* x on kolmikko (q_0, x, ε).

Intuitio: tilanteessa (q, w, α) automaatti on tilassa q, syötemerkkijonon käsittelemätön osa on w ja pinossa on ylhäältä alas lukien merkkijono α.

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

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

jos voidaan kirjoittaa $w = \sigma w'$, $\alpha = \gamma\beta$, $\alpha' = \gamma'\beta$ ($|\sigma|, |\gamma|, |\gamma'| \leq 1$), siten että

$$(q', \gamma') \in \delta(q, \sigma, \gamma).$$

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

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

jos on olemassa tilannejono (q_0, w_0, α_0), (q_1, w_1, α_1), $\ldots$, (q_n, w_n, α_n), $n \geq 0$, siten että

$$\begin{aligned} (q, w, \alpha) &= (q_0, w_0, \alpha_0) \underset{M}{\vdash} (q_1, w_1, \alpha_1) \underset{M}{\vdash} \cdots \\ &\underset{M}{\vdash} (q_n, w_n, \alpha_n) = (q', w', \alpha'). \end{aligned}$$

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

$$(q_0, x, \varepsilon) \underset{M}{\vdash}^* (q_f, \varepsilon, \alpha) \quad \text{joillakin } q_f \in F \text{ ja } \alpha \in \Gamma^*,$$

siis jos se syötteen loppuessa on jossakin hyväksyvässä lopputilassa; muuten M *hylkää* x:n.

Automaatin M *tunnistama kieli* on:

$$\begin{aligned} L(M) &= \{x \in \Sigma^* \mid (q_0, x, \varepsilon) \underset{M}{\vdash}^* (q_f, \varepsilon, \alpha) \\ &\qquad \text{joillakin } q_f \in F \text{ ja } \alpha \in \Gamma^*\}. \end{aligned}$$

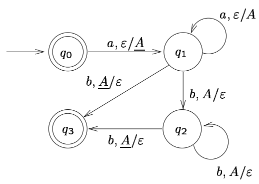
Esimerkki. Pinoautom. kielelle $\{a^k b^k \mid k \geq 0\}$:

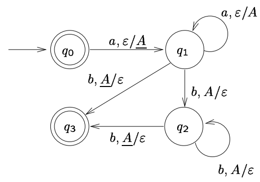
$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{A, \underline{A}\}, \delta, q_0, \{q_0, q_3\})$,

missä

$$\begin{aligned}
\delta(q_0, a, \varepsilon) &= \{(q_1, \underline{A})\}, \\
\delta(q_1, a, \varepsilon) &= \{(q_1, A)\}, \\
\delta(q_1, b, A) &= \{(q_2, \varepsilon)\}, \\
\delta(q_1, b, \underline{A}) &= \{(q_3, \varepsilon)\}, \\
\delta(q_2, b, A) &= \{(q_2, \varepsilon)\}, \\
\delta(q_2, b, \underline{A}) &= \{(q_3, \varepsilon)\}, \\
\delta(q, \sigma, \gamma) &= \emptyset \qquad \text{muilla } (q, \sigma, \gamma).
\end{aligned}$$

Kaavioesitys:





Automaatin toiminta syötteellä $aabb$:

$$\begin{aligned}
(q_0, aabb, \varepsilon) &\vdash (q_1, abb, \underline{A}) \vdash (q_1, bb, A\underline{A}) \\
&\vdash (q_2, b, \underline{A}) \quad \vdash (q_3, \varepsilon, \varepsilon).
\end{aligned}$$

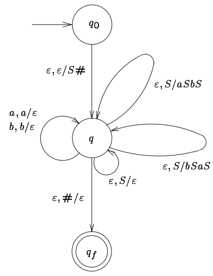
Koska $q_3 \in F = \{q_0, q_3\}$, on siis $aabb \in L(M)$.

Pinoautomaatit ja yhteydettömät kielet

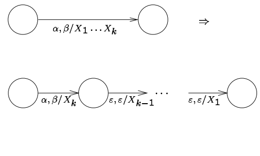
Lause 3.8 Kieli on yhteydetön, jos ja vain jos se voidaan tunnistaa jollakin (epädeterministisellä) pinoautomaatilla. □

Em. lauseen todistus sivuutetaan tässä, mutta periaatteena esim. annettua kielioppia G vastaavan pinoautomaatin M_G toiminnassa on, että M_G:n pinon käyttäytyminen syötteellä x noudattelee G:n mukaisen vasemman lausejohdon $S \underset{\text{lm}}{\Rightarrow}^* x$ etenemistä: jos pinon päällimmäisenä on välikemerkki, sovelletaan jotain G:n produktiota ja lisätään pinon pinnalle vastaavat merkit; jos pinon päällimmäisenä on päätemerkki, se sovitetaan yhteen seuraavan syötemerkin kanssa.

Esimerkki. Kielioppia $\{S \to aSbS \mid bSaS \mid \varepsilon\}$ vastaava pinoautomaatti:



Automaatin kuvauksessa on käytetty seuraavaa luonnollista lyhennemerkintää:



Esimerkiksi syötteellä $abab$ on em. automaatilla seuraava hyväksyvä laskenta:

$$
\begin{array}{llll}
(q_0, abab, \varepsilon) & \vdash (q, abab, S\#) & \vdash^* & (q, abab, aSbS\#) \\
 & \vdash (q, bab, SbS\#) & \vdash^* & (q, bab, bSaSbS\# \\
 & \vdash (q, ab, SaSbS\#) & \vdash & (q, ab, aSbS\#) \\
 & \vdash (q, b, SbS\#) & \vdash & (q, b, bS\#) \\
 & \vdash (q, \varepsilon, S\#) & \vdash & (q, \varepsilon, \#) \\
 & \vdash (q_f, \varepsilon, \varepsilon). & &
\end{array}
$$

Tämä vastaa annetun kieliopin mukaista lauseen $abab$ vasenta johtoa:

$$
\begin{array}{l}
\underline{S} \Rightarrow a\underline{S}bS \Rightarrow ab\underline{S}aSbS \Rightarrow aba\underline{S}bS \\
\quad \Rightarrow abab\underline{S} \Rightarrow abab.
\end{array}
$$

Pinoautomaatti M on *deterministinen*, jos jokaisella tilanteella (q, w, α) on enintään yksi mahdollinen seuraaja (q', w', α'), jolla

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

Toisin kuin äärellisten automaattien tapauksessa, *epädeterministiset pinoautomaatit ovat aidosti vahvempia kuin deterministiset.* Esimerkiksi kieli $\{ww^R \mid w \in \{a, b\}^*\}$ voidaan tunnistaa epädeterministisellä, mutta ei deterministisellä pinoautomaatilla. (Tod. siv.)

Yhteydetön kieli on *deterministinen*, jos se voidaan tunnistaa jollakin deterministisellä pinoautomaatilla. Deterministiset kielet voidaan jäsentää ajassa $O(n)$; yleiset yhteydettömät kielet vaativat tunnetuilla menetelmillä lähes ajan $O(n^3)$.