

3.3 KIELIOPPIEN JÄSENNYSONGELMA

Ratkaistava tehtävä:

“Annettu yhteydetön kielioppi G ja merkkijono x . Onko $x \in L(G)$?”

Ratkaisumenetelmä = *jäsennysalgoritmi*.

Useita vaihtoehtoisia menetelmiä, erityisesti kun G on jotain rajoitettua (käytännössä esiintyvää) muotoa.



Johdot ja jäsenyspuut

Olkoon $\gamma \in V^*$ kieliopin $G = (V, \Sigma, P, S)$ lausejohdos.

Lähtösymbolista S merkkijonoon γ johtavaa suorien johtojen jonoa

$$S = \gamma_0 \Rightarrow \gamma_1 \Rightarrow \dots \Rightarrow \gamma_n = \gamma$$

sanotaan γ :n *johdoksi* G :ssä.

Johdon *pituus* on siihen kuuluvien suorien johtojen määrä (edellä n).

Esimerkki: lauseen $a + a$ johtoja kieliopissa G_{expr} :

$$\begin{aligned} \text{(i)} \quad E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \\ &\Rightarrow a + T \Rightarrow a + F \Rightarrow a + a \\ \text{(ii)} \quad E &\Rightarrow E + T \Rightarrow E + F \Rightarrow T + F \\ &\Rightarrow F + F \Rightarrow F + a \Rightarrow a + a \\ \text{(iii)} \quad E &\Rightarrow E + T \Rightarrow E + F \Rightarrow E + a \\ &\Rightarrow T + a \Rightarrow F + a \Rightarrow a + a. \end{aligned}$$



Johto $\gamma \Rightarrow^* \gamma'$ on *vasen johto*, merkitään

$$\gamma \Rightarrow_{\text{lm}}^* \gamma',$$

jos kussakin johtoaskeleessa on produktiota sovellettu merkkijonon vasemmanpuoleisimpaan välikkeeseen (edellä johto (i)).

Vastaavasti määritellään *oikea johto* (edellä (iii)), jota merkitään

$$\gamma \Rightarrow_{\text{rm}}^* \gamma'$$

Suuria vasempia ja oikeita johtoaskelia merkitään $\gamma \Rightarrow_{\text{lm}} \gamma'$ ja

$$\gamma \Rightarrow_{\text{rm}} \gamma'.$$



Olkoon $G = (V, \Sigma, P, S)$ yhteydetön kielioppi.

Kieliopin G mukainen *jäsennyspuu* on järjestetty puu, jolla on seuraavat ominaisuudet:

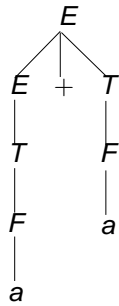
(i) puun solmut on nimetty joukon $V \cup \{\varepsilon\}$ alkioilla siten, että sisäsolmujen nimet ovat välikkeitä (so. joukosta $N = V - \Sigma$) ja juurisolmun nimenä on lähtösymboli S ;

(ii) jos A on puun jonkin sisäsolmun nimi, ja X_1, \dots, X_k ovat sen jälkeläisten nimet järjestyksessä, niin $A \rightarrow X_1 \dots X_k$ on G :n produktio.

Jäsennyspuun τ *tuotos* on merkkijono, joka saadaan liittämällä yhteen sen lehtisolmujen nimet esijärjestyksessä (“vasemmalta oikealle”).



Esimerkki. Lauseen $a + a$ jäsenyspuu kieliopissa G_{expr} :



Lauseen johto:

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \\ &\Rightarrow a + T \Rightarrow a + F \Rightarrow a + a \end{aligned}$$



Johtoa

$$S = \gamma_0 \Rightarrow \gamma_1 \Rightarrow \dots \Rightarrow \gamma_n = \gamma$$

vastaavan jäsenyspuun muodostaminen:

- (i) puun juuren nimeksi tulee S ; jos $n = 0$, niin puussa ei ole muita solmuja; muuten
 - (ii) jos ensimmäisessä johtoaskelella on sovellettu produktiota $S \rightarrow X_1 X_2 \dots X_k$, niin juurelle tulee k jälkeläissolmua, joiden nimet vasemmalta oikealle ovat X_1, X_2, \dots, X_k ;
 - (iii) jos seuraavassa askelella on sovellettu produktiota $X_i \rightarrow Y_1 Y_2 \dots Y_l$, niin juuren i :nnelle jälkeläissolmulle tulee l jälkeläistä, joiden nimet vasemmalta oikealle ovat Y_1, Y_2, \dots, Y_l ; ja niin edelleen.
- Konstruktioista huomataan, että jos τ on jotakin johtoa $S \Rightarrow^* \gamma$ vastaava jäsenyspuu, niin τ :n tuotos on γ .



Olkoon τ kieliopin G mukainen jäsenyspuu, jonka tuotos on päätamerkkijono x .

Tällöin τ :sta saadaan vasen johto x :lle käymällä puun solmut läpi esijärjestyksessä ("ylhäältä alas, vasemmalta oikealle") ja laventamalla vastaan tulevat välitteet järjestyksessä puun osoittamalla tavalla.

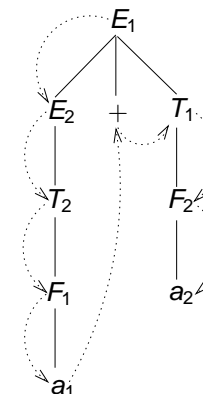
Oikea johto saadaan käymällä puu läpi käänteisessä esijärjestyksessä ("ylhäältä alas, oikealta vasemmalle").

Muodostamalla annetusta vasemmasta johdosta $S \xRightarrow{*} x$ ensin jäsenyspuu edellä esitetyllä tavalla, ja sitten jäsenyspuusta vasen johto, saadaan takaisin alkuperäinen johto; vastaava tulos pätee myös oikeille johdoille.



Esimerkki. Lauseen $a + a$ vasemman johdon muodostaminen jäsenyspuusta.

Jäsenyspuu:



Solmut esijärjestyksessä:

$$E_1 E_2 T_2 F_1 a_1 + T_1 F_2 a_2$$

Vasen johto:

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \\ &\xRightarrow[1m]{1m} a + T \xRightarrow[1m]{1m} a + F \xRightarrow[1m]{1m} a + a \end{aligned}$$



Lause 3.3 Olkoon $G = (V, \Sigma, P, S)$ yhteydetön kielioppi.

Tällöin:

- (i) jokaisella G :n lausejohdoksella γ on G :n mukainen jäsennysspuu τ , jonka tuotos on γ ;
- (ii) jokaista G :n mukaista jäsennysspuuta τ , jonka tuotos on päätemerkkijono x , vastaavat yksikäsitteiset vasen ja oikea johto $S \xRightarrow{\text{lm}}^* x$ ja $S \xRightarrow{\text{rm}}^* x$.

Seuraus 3.4 Jokaisella G :n lauseella on vasen ja oikea johto.

Siis: yhteydetön kieliopin tuottamien lauseiden jäsennysspuut, vasemmat ja oikeat johdot vastaavat yksikäsitteisesti toisiaan.

Jäsennyssongelman ratkaisuun katsotaan usein kuuluvan pelkän päätösongelman “Onko $x \in L(G)$?” ratkaisemisen lisäksi jonkin näistä jäsennyssesityksistä tuottaminen.



Moniselitteisyys on tietojenkäsittelysovelluksissa yleensä ei-toivottu ominaisuus, koska se merkitsee että annetulla lauseella on kaksi vaihtoehtoista “tulkintaa.”

Yhteydetön kieli, jonka tuottavat kieliopit ovat kaikki moniselitteisiä, on *luonnostaan moniselitteinen*.

Esimerkiksi kielioppi G'_{expr} on moniselitteinen, kieliopit G_{expr} ja G_{match} yksiselitteisiä. Kieli $L_{\text{expr}} = L(G'_{\text{expr}})$ ei ole luonnostaan moniselitteinen, koska sillä on myös yksiselitteinen kielioppi G_{expr} . Luonnostaan moniselitteinen on esimerkiksi kieli

$$\{a^i b^j c^k \mid i = j \text{ tai } j = k\}.$$

(Todistus sivuutetaan.)



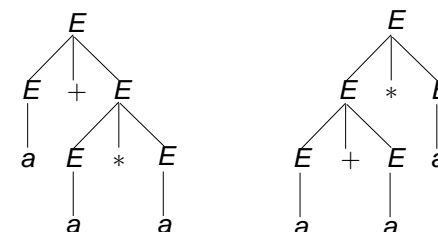
Kieliopin moniselitteisyys

Lauseella voi olla kieliopissa useita jäsennyksiä.

Esimerkki. Tarkastellaan yksinkertaisten aritmeettisten lausekkeiden kielioppia:

$$G'_{\text{expr}} = \{E \rightarrow E + E, E \rightarrow E * E, E \rightarrow a, E \rightarrow (E)\}.$$

Lauseella $a + a * a$ on tässä kieliopissa kaksi jäsennyttä:



Yhteydetön kielioppi G on *moniselitteinen*, jos jollakin G :n lauseella x on kaksi erilaista G :n mukaista jäsennysspuuta. Muuten kielioppi on *yksiselitteinen*.



3.4 Osittava jäsentäminen

Yksi (yleisessä muodossa tehoton!) tapa etsiä vasenta johtoa (jäsennysspuuta) annetun kieliopin G mukaiselle lauseelle x on aloittaa G :n lähtösymbolista ja generoida systemaattisesti kaikki mahdolliset vasemmat johdot (jäsennysspuut), samalla sovittaen muodostetun lausejohdoksen päätemerkkejä (puun lehtiä) x :n merkkeihin. Ei-yhteensopivuuden ilmetessä peruutetaan viimeksi tehty produktiovalinta ja kokeillaan järjestyksessä seuraavaa vaihtoehtoa.

Tällaista lauseenjäsennysspuuta sanotaan *osittavaksi*, koska siinä tarkasteltu lause yritetään johtaa kieliopin lähtösymbolista osittamalla se valittujen produktioiden mukaisiin rakenneseisiin ja yrittämällä näin, tarvittaessa toistuvasti edelleen osittamalla, sovittaa kieliopin tuottamaa rakennetta yhteen lauseen rakenteen kanssa.



Esim. Tarkastellaan kielioppia G :

$$\begin{aligned} E &\rightarrow T + E \mid T - E \mid T \\ T &\rightarrow a \mid (E). \end{aligned}$$

Lauseen $a - a$ osittava jäsenitys G :n suhteen:

$$\begin{aligned} E &\Rightarrow T + E \Rightarrow a + T && \text{[ristiriita; peruutetaan]} \\ &\Rightarrow (E) + T && \text{[ristiriita; peruutetaan]} \\ &\Rightarrow T - E \Rightarrow a - E \Rightarrow a - T + E \Rightarrow a - a + E \\ &&& \text{[ristiriita; peruutetaan]} \\ &\Rightarrow T - E \Rightarrow a - E \Rightarrow a - T + E \Rightarrow a - (E) + E \\ &&& \text{[ristiriita; peruutetaan]} \\ &\Rightarrow a - T - E \Rightarrow a - a - E \\ &&& \text{[ristiriita; peruutetaan]} \\ &\Rightarrow a - T - E \Rightarrow a - (E) - E \\ &&& \text{[ristiriita; peruutetaan]} \\ &\Rightarrow a - T \Rightarrow a - a && \text{[OK!]} \end{aligned}$$



Em. osittava jäsenystekniikka saadaan huomattavasti tehokkaammaksi, jos kieliopilla on sellainen ominaisuus, että jäsennyksen joka vaiheessa määrää tavoitteena olevan lauseen seuraava merkki yksikäsitteisesti sen, mikä lavennettavana olevaan välikkeeseen liittyvä produktio on valittava. Kielioppia, jolla on tämä ominaisuus, sanotaan *LL(1)-tyyppiseksi*.

Muokataan G :stä välikkeen E produktiot "tekijöimällä" ekvivalentti kielioppi G' :

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +E \mid -E \mid \varepsilon \\ T &\rightarrow a \mid (E). \end{aligned}$$

Esimerkiksi lauseen $a - a$ jäsentäminen G' :n suhteen (kulloisenkin produktiovalinnan määräävä syötemerkki on tässä merkitty vastaavan johtonuolen päälle):

$$E \Rightarrow TE' \xrightarrow[\text{lm}]{a} aE' \xrightarrow[\text{lm}]{-} a - E \Rightarrow a - TE' \xrightarrow[\text{lm}]{a} a - aE' \xrightarrow[\text{lm}]{\varepsilon} a - a.$$



LL(1)-tyyppiselle kieliopille on helppo kirjoittaa jäsennysohjelma suoraan rekursiivisina proseduureina. Esimerkiksi kieliopin G' pohjalta voidaan muodostaa seuraava C-kielinen funktiokokoelma, joka syötejonon jäsennyksen yhteydessä tulostaa sen tuottavan vasemman johdon produktiot järjestyksessä.

```
#include <stdio.h>
```

```
int next;
void E(void); void Eprime(void); void T(void);
```

```
void E(void)
{
    printf("E -> TE'\n");
    T(); Eprime();
}
```



```
void Eprime(void)
{
    if (next == '+') {
        printf("E' -> +E\n");
        next = getchar();
        E();
    }
    else if (next == '-') {
        printf("E' -> -E\n");
        next = getchar();
        E();
    }
    else
        printf("E' -> \n");
}
```



```

void T(void)
{
    if (next == 'a') {
        printf("T -> a\n");
        next = getchar();
    }
    else if (next == '(') {
        printf("T -> (E)\n");
        next = getchar();
        E();
        if (next != ')')
            ERROR(") expected.");
        next = getchar();
    }
    else ERROR("T cannot start with this.");
}

```



```

void ERROR(char *msg)
{
    printf("%s\n",msg); exit(1);
}

int main(void)
{
    next = getchar();
    E();
    exit(0);
}

```



Esimerkiksi syötejonoa $a-(a+a)$ käsitellessään ohjelma tulostaa seuraavat rivit:

```

E → TE'
T → a
E' → -E
E → TE'
T → (E)
E → TE'
T → a
E' → +E
E → TE'
T → a
E' →
E' →

```

Tulostus vastaa vasenta johtoa:

$$\begin{aligned}
 E &\Rightarrow TE' \Rightarrow aE' \Rightarrow a - E \Rightarrow a - TE' \\
 &\Rightarrow a - (E)E' \Rightarrow a - (TE')E' \\
 &\Rightarrow a - (aE')E' \Rightarrow a - (a + E)E' \\
 &\Rightarrow a - (a + TE')E' \Rightarrow a - (a + aE')E' \\
 &\Rightarrow a - (a + a)E' \Rightarrow a - (a + a).
 \end{aligned}$$
