Tik-79.148 **Kevät 2000**
Tietojenkäsittelyteorian perusteet
Laskuharjoitus 6
Ratkaisut

1. (3.1.2)

   Yhteydettömässä kieliopissa $G = (V, \Sigma, R, S)$, V on kieleen kuuluvien merkkien joukko, $\Sigma \subset V$ on kielen terminaalisymbolien joukko. Kaikki kieleen kuuluvat sanat koostuvat pelkästään terminaalisymboleista, joukkoon $V - \Sigma$ kuuluvia symboleita (nonterminaaleja) käytetään pelkästään kielen sanojen johtamisessa. $R \in (V-\Sigma)\times V^*$ on joukko muunnossääntöjä. $S \in V - \Sigma$ on kielen aloitussymboli.

   Tehtävän kielioppi on

   $$\begin{aligned} V &= \{a, b, S, A\} \\ \Sigma &= \{a, b\} \\ R &= \{S \to aAa, S \to bAb, S \to e, A \to SS\} \end{aligned}$$

   Sana *baabbb* voidaan johtaa tällä seuraavasti:

   $$\begin{aligned} S &\to bAb \to bSSb \to baAaSb \to baSSaSb \\ &\to baSaSb \to baaSb \to baabAbb \\ &\to baabSSbb \to baabSbb \to baabbb \end{aligned}$$

   Kielen sanoissa esiintyvät a ja b aina pareittain, siten että jokaisen parin sisällä voi olla muita pare-ja.

2. (3.1.3 a, c)

   a) Kieltä $L = \{wcw^R \mid w \in \{a, b\}^*\}$ vastaa yhteydetön kielioppi:

   $$\begin{aligned} G =& (V, \Sigma, R, S) \\ \Sigma =& \{a, b, c\} \\ V =& \{a, b, c, S\} \\ R =& \{S \to aSa \\ & \quad S \to bSb \\ & \quad S \to c\} \end{aligned}$$

   Esimerkiksi kieleen kuuluvan sanan *abbcbba* johto menee seuraavasti:

   $$S \to aSa \to abSba \to abbSbba \to abbcbba$$

c) Kieltä $L = \{w \in \{a,b\}^* \mid w = w^R\}$ vastaa yhteydetön kielioppi

$$\begin{aligned} G =& (V, \Sigma, R, S) \\ \Sigma =& \{a, b\} \\ V =& \{a, b, S\} \\ R =& \{S \to aSa \\ & S \to bSb \\ & S \to e \\ & S \to a \\ & S \to b\} \end{aligned}$$

Esimerkiksi *ababa* johdetaan:

$$S \to aSa \to abSba \to ababa,$$

ja *abbbba*:

$$S \to aSa \to abSba \to abbSbba \to abbbba$$

3. (3.1.4)

Aakkoston $\Sigma = \{a, b, (,), \cup, ^*, \emptyset\}$ sallitut säännölliset lausekkeet voi generoida kieliopilla $G = (V, \Sigma, R, S)$,

$$\begin{aligned} V =& \Sigma \cup \{S, T, F\} \\ R =& \{S \to T \cup S, S \to T \\ & T \to F, T \to F^*, \\ & T \to FT, T \to F^*T \\ & F \to \emptyset, F \to a, \\ & F \to b, F \to (S)\} \end{aligned}$$

Esimerkiksi säännöllinen lauseke $(a \cup bb)^*b$ johdetaan $S \to T \to F^*T \to (S)^*T \to (T \cup S)^*T \to (F \cup S)^*T \to (a \cup S)^*T \to (a \cup T)^*T \to (a \cup FT)^*T \to (a \cup bT)^*T \to (a \cup bF)^*T \to (a \cup bb)^*T \to (a \cup bb)^*F \to (a \cup bb)^*b$

Formaalit kielet muodostavat hierarkian, jossa jokainen porras on aina aidosti edellistä vahvempi ilmaisuvoimaltaan. Niinpä esimerkiksi säännölliset kielet voidaan kuvata säännöllisillä lausekkeilla, ja kielet, joihin kuuluvat kaikki tietyllä aakkostolla muodostettavat säännölliset lausekkeet, voidaan kuvata yhteydettömillä kieliopeilla. Ei kuitenkaan yleisesti päde, että tietyn kieliluokan tuottavat kieliopit voitaisiin aina kuvata seuraavan luokan kieliopeilla.

4. (3.1.9 a,d / 3.1.8 a,d)

a) Kieltä $L = \{a^m b^n \mid m \geq n\}$ vastaava yhteydetön kielioppi on:

$$\begin{aligned} \Sigma =& \{a, b\} \\ V =& \{a, b, S, A, B\} \\ R =& \{S \to ASB, S \to e \\ & A \to Aa, A \to a, \\ & B \to b, B \to e\} \end{aligned}$$

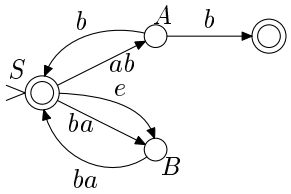
d) Kieltä $L = \{uawb \mid u, w \in \{a, b\}^*, |u| = |w|\}$ vastaava yhteydetön kielioppi on

$$\begin{aligned}\Sigma =& \{a, b\}\\ V =& \{a, b, S, T\}\\ R =& \{S \to Tb, T \to aTa,\\ & T \to aTb, T \to bTb\\ & T \to bTa, T \to a\}\end{aligned}$$

5. (3.1.10a / 3.2.1)

Kielioppi on säännöllinen, mikäli $R \subseteq (V - \Sigma) \times \Sigma^*((V - \Sigma) \cup \{e\})$, eli säännöissä saa olla oikealla puolella korkeintaan yksi nonterminaali, ja sen täytyy olla viimeisenä.

Säännöllistä kielioppia vastaavan epädeterministisen tilakoneen voi muodostaa ottamalla koneesen tila jokaista kieliopin nonterminaalia varten. Aina kun kieliopissa on sääntö $A \to uB, A, B \in (V - \Sigma), u \in \Sigma^*$, asetetaan A:ta vastaavasta tilasta kaari u tilaan B. Lisäksi otetaan yksi erityinen lopputila, johon viedään kaikki kaaret, jotka ovat muotoa $R \to u, u \in \Sigma^*$. Lopputiloiksi kelpaavat kaikki ne tilat, joihin sana voi päättyä.



Tilakoneessa A:sta lähtevä kaari erityiseen lopputilaan on tässä tapauksessa turha, sillä myös lopputilaan S päästään b-kaarella.

Kielioppi generoi sanan *abba* seuraavasti: $S \to abA \to abbS \to abbB \to abbaS \to abba$. Tilakone puolestaan käy läpi tilat S, A, S, B, S, mikä vastaa tarkalleen sanan johdossa läpikäytyjä nonterminaaleja.

6. Melkein kaikki nykyisin käytössä olevat ohjelmointikielet pohjautuvat yhteydettömiin kielioppeihin, joilla kuvataan kielen rakenne. Vain kaikkein yksinkertaisimmat kielet ovat puhtaasti yhteydettömiä, sillä esimerkiksi muuttujien tyyppitarkistuksen tekeminen pelkillä yhteydettömillä säännöillä on mahdotonta. Käytännössä asia hoidetaan siten, että ohjelma jäsennetään kieliopin mukaan puurakenteeksi välittämättä mitään muuttujien tyypeistä. Kun ohjelma on saatu jäsennetyksi, tarkistetaan saadusta rakenteesta onko muuttujia käytetty oikein.

Alla on määritelty yksinkertainen ohjelmointikieli, joka toteuttaa luonnollisilla luvuilla tehtävät peruslaskutoimitukset. Niiden lisäksi kielessä on tuki muuttujille, silmukkarakenne ja ehtolause. Kieliopissa on käytetty seuraavanlaisia merkintöjä:

- Kielen nonterminaalit on kirjoitettu *kursiivilla*.
- Kielen terminaalisymbolit on kirjoitettu **lihavoituna**.

- Välimerkit ja operaattorit ovat myös terminaalisymboleita.
- Kaikki saman nonterminaalin säännöt on kerätty yhteen, ja ne eroitetaan toisistaan —-merkillä.

$$
\begin{aligned}
\mathit{program} &\rightarrow \mathit{statement_list} \\
\\
\mathit{statement_list} &\rightarrow \mathit{statement_list}\ \mathit{statement} \\
&\mid \mathit{statement} \\
\\
\mathit{statement} &\rightarrow \textbf{while}\ \mathit{expression}\ \{\ \mathit{statement_list}\ \} \\
&\mid \textbf{if}\ \mathit{expression}\ \{\ \mathit{statement_list}\ \} \\
&\mid \textbf{if}\ \mathit{expression}\ \{\ \mathit{statement_list}\ \}\ \textbf{else}\ \{\ \mathit{statement_list}\ \} \\
&\mid \textbf{variable} = \mathit{expression}\ ; \\
&\mid \mathit{expression}\ ; \\
\\
\mathit{expression} &\rightarrow -\ \mathit{expression} \\
&\mid \mathit{expression} + \mathit{expression} \\
&\mid \mathit{expression} - \mathit{expression} \\
&\mid \mathit{expression} * \mathit{expression} \\
&\mid \mathit{expression} \,/\, \mathit{expression} \\
&\mid \mathit{expression} == \mathit{expression} \\
&\mid \mathit{expression} < \mathit{expression} \\
&\mid (\ \mathit{expression}\ ) \\
&\mid \textbf{number} \\
&\mid \textbf{variable}
\end{aligned}
$$

Kurssin kotisivulla on C++-kielinen lähdekoodi yksinkertaiseen tulkkiin, joka toteuttaa ylläolevan kielen.

Kielellä voi toteuttaa esimerkiksi luvun kertoman laskemisen seuraavasti (#-merkki aloittaa kommentin):

```
# computes the factorial of the variable 'n'
n = 8;

a = 1;              # initial value
m = 1;              # loop index
while (m < n+1) {
 a = a * m;
 m = m+1;
}
a;                  # return the factorial
```