T-79.148 **Syksy 2004**
Tietojenkäsittelyteorian perusteet
Harjoitus 6
Demonstraatiotehtävien ratkaisut

Lemma (Pumppauslemma). Olkoon A säännöllinen kieli. Tällöin on olemassa $n \geq 1$ siten, että kaikki A:n merkkijonot x , joiden pituus $|x| \geq n$ ovat ilmaistavissa muodossa $x = uvw$, missä $|uv| \leq n$, $|v| \geq 1$ ja merkkijonot muotoa uv^iw kuuluvat kieleen A kaikilla $i \geq 0$.
Kompaktimmin, painottaen pumppauslemman asettamia vaatimuksia, voimme kirjoittaa seuraavasti:

$\forall$ säännöllisille kielille A

$\exists n \geq 1$ s.e.
$\forall x \in A : |x| \geq n$
$\exists$ osiinjako $x = uvw$, missä $|uv| \leq n$ ja $|v| \geq 1$
$\forall i \geq 0$ $uv^iw \in A$.

Pumppauslemmaa voidaan käyttää hyväksi, kun halutaan osoittaa kieli L *ei-säännölliseksi*. Tehdään ensin vastaoletus, eli oletetaan L säännölliseksi kieleksi. Tavoitteena on päästä ristiriitaan tämän oletuksen kanssa seuraten pumppauslemman asettamia vaatimuksia säännöllisille kielille.
Pumppauslemmaa käytettäessä täytyy aina muistaa, että sillä voi osoittaa vain kielen epäsäännöllisyyden, ja sitä *ei* voi käyttää toiseen suuntaan. Esimerkiksi kieli

$$I = \{c^i a^n b^n \mid i > 0 \wedge n \geq 0\} \cup L(a^*b^*)$$

ei ole säännöllinen, mutta kaikki siihen kuuluvat sanat (tyhjää sanaa lukuunottamatta) voidaan osittaa pumppauslemman ehtojen mukaisesti. Näin ollen kieltä I ei voida suoraan todistaa epäsäännölliseksi, vaan todistuksessa täytyy käyttää apuna säännöllisten kielten sulkeumaominaisuuksia 5. tehtävän vastauksessa esitettävään tapaan.
Jos halutaan osoittaa kieli säännölliseksi, voidaan muodostaa sen hyväksyvä äärellinen automaatti, sillä pätee: Kieli L on säännöllinen $\Leftrightarrow$ on olemassa äärellinen automaatti M, joka hyväksyy kielen L (merkitään $L(M) = L$).

4. **Tehtävä**:Modernissa WWW-sivujen kuvaamiseen käytetyssä XML-kielessä on sivujen suunnittelijan mahdollista laatia omia ns. dokumenttityyppimäärityksiä (engl. *Document Type Definition*, lyh. DTD), jotka ovat oleellisesti sivulla esitettävän tekstin tai muun datan rakennetta kuvaavia yhteydettömiä kielioppeja. Tutustu tämän XML/DTD-kuvauskielen notaatioon (esim. WWW-sivulta `http://www.rpbourret.com/xml/xmldtd.htm`), ja laadi seuraavaa XML/DTD-kuvausta vastaava yhteydetön kielioppi:

```
<!DOCTYPE Book [
   <!ELEMENT Book (Title, Chapter+)>
   <!ATTLIST Book Author CDATA #REQUIRED>
   <!ELEMENT Title (#PCDATA)>
   <!ELEMENT Chapter (#PCDATA)>
   <!ATTLIST Chapter id ID #REQUIRED>
]>
```

Vastaus: Yllä oleva DTD-kuvaus määrittelee rakenteen kirjalle. Määrittelyssä esiintyy kahdenlaisia asioita: *osia* (element) ja *attribuutteja* (attlist). Perusajatuksena on, että kirja itsessään koostuu osista, ja attribuutit puolestaan liittävät kirjan osiin ylimääräistä tietoa.

Yleisesti ottaen attribuutteja ei voida esittää puhtailla kieliopeilla, vaan niitä varten tarvitaan *attribuuttikieliopit* (ks. opetusmoniste s. 60–68). Näin ollen DTD-kuvauksesta mallinnetaan ensin vain osat, eli käytännössä ainoastaan rivit, jotka alkavat merkinnällä `!ELEMENT`.

Näistä riveistä ensimmäinen:

```
<!ELEMENT Book (Title, Chapter+)>
```

kertoo, että kirja (Book) sisältää otsikon (Title) ja listan lukuja (Chapter). Lukuja täytyy olla vähintään yksi. Seuraava rivi:

```
<!ELEMENT Title (#PCDATA)>
```

puolestaan määrittelee otsikon merkkijoukoksi (#PCDATA). Merkkijoukkoon voi kuulua periaattessa mitä tahansa tietokoneen merkkijärjestelmään kuuluvia symboleita.

Lopuksi, rivi:

```
<!ELEMENT Chapter (#PCDATA)>
```

kertoo luvun olevan taas joukko merkkejä.

Kirjan rakenteen kuvaa siis kielioppi:[1]

$$\begin{aligned} \mathit{Book} &\rightarrow \mathit{Title}\ \mathit{Chapters} \\ \mathit{Title} &\rightarrow \mathbf{data} \\ \mathit{Chapters} &\rightarrow \mathit{Chapter}\ \mathit{Chapters} \mid \mathit{Chapter} \\ \mathit{Chapter} &\rightarrow \mathbf{data} \end{aligned}$$

XML-kielessä erotetaan dokumentin osat toisistaan käyttäen apuna `<A>` ja `</A>` koodeja. Kun nämä koodit lisätään yllä olevaan kielioppiin, saadaan tulokseksi seuraavanlainen yhteydetön kielioppi:

$$\begin{aligned} \mathit{Book} &\rightarrow \langle\mathbf{Book}\rangle\ \mathit{Title}\ \mathit{Chapters}\ \langle/\mathbf{Book}\rangle \\ \mathit{Title} &\rightarrow \langle\mathbf{Title}\rangle\ \mathbf{data}\ \langle/\mathbf{Title}\rangle \\ \mathit{Chapters} &\rightarrow \mathit{Chapter}\ \mathit{Chapters} \mid \mathit{Chapter} \\ \mathit{Chapter} &\rightarrow \langle\mathbf{Chapter}\rangle\ \mathbf{data}\ \langle/\mathbf{Chapter}\rangle \end{aligned}$$

Vaikka attribuutteja ei voidakaan täysin esittää yhteydettömällä kieliopilla, niiden syntaksi voidaan kuitenkin kuvata. XML-kielessä osan attribuutit kirjoitetaan osan aloittavan koodin sisään. Lisäämällä nämä ylläolevaan kielioppiin saadaan tulokseksi:

$$\begin{aligned} \mathit{Book} &\rightarrow \langle\ \mathbf{Book}\ \mathit{BookAttributes}\ \rangle\ \mathit{Title}\ \mathit{Chapters}\ \langle/\mathbf{Book}\rangle \\ \mathit{Title} &\rightarrow \langle\mathbf{Title}\rangle\ \mathbf{data}\ \langle/\mathbf{Title}\rangle \\ \mathit{Chapters} &\rightarrow \mathit{Chapter}\ \mathit{Chapters} \mid \mathit{Chapter} \\ \mathit{Chapter} &\rightarrow \langle\ \mathbf{Chapter}\ \mathit{ChapterAttributes}\ \rangle\ \mathbf{data}\ \langle/\mathbf{Chapter}\rangle \\ \mathit{BookAttributes} &\rightarrow \mathbf{author} = \mathbf{data} \\ \mathit{ChapterAttributes} &\rightarrow \mathbf{id} = \mathbf{data} \end{aligned}$$

Tässä on huomattava, että yhteydettömällä kieliopilla ei voida määritellä ehtoa, jonka mukaan kaikilla luvuilla on eri tunnus. Tällaisten ehtojen toteutuminen täytyy tarkistaa jollain erillisellä ohjelmakoodilla.

[1] *Kursiivilla* kirjoitetut sanat ovat välikkeitä, **vahvennetut** päätemerkkejä.

5. **Tehtävä**: Osoita, että kieli $\{w \in \{a,b\}^* \mid w$:ssä on yhtä monta a:ta ja b:tä$\}$ ei ole säännöllinen, ja laadi yhteydetön kielioppi sen kuvaamiseen.

   Vastaus: Kielen $L = \{w \in \{a,b\}^* \mid w$:ssä on yhtä monta a:ta ja b:tä$\}$ voisi todistaa ei-säännölliseksi suoraan pumppauslemmalla. Tässä esitetään kuitenkin hieman monimutkaisempi ratkaisu esimerkkinä siitä, miten "hankalia" kieliä voidaan käsitellä.

   Määritellään kieli $L' = L \cap L(a^*b^*)$. Oletetaan, että L on säännöllinen. Koska $L(a^*b^*)$ on säännöllinen ja säännöllisten kielten joukko on suljettu leikkauksen suhteen, täytyy myös L':n olla säännöllinen. (Toisinpäin ehto ei päde: L' voi olla säännöllinen vaikka L ei olisi, sillä esim. $A \cap \emptyset = \emptyset$ kaikille kielille A).

   Huomataan, että $L' = \{a^k b^k \mid k \geq 0\}$. Tarkastellaan sanaa $w = a^n b^n$, missä n on pumppauslemmassa esiintyvä parametri. Yritetään osittaa w lemman ehtojen mukaisesti. Koska $|xy| \leq n$, osituksen täytyy olla muotoa;

$$\begin{aligned} x &= a^{n-i-k} \\ y &= a^i \\ z &= a^k b^n, \end{aligned}$$

   missä $0 < i \leq n$ ja $i + k \leq n$. Nyt $xz = a^{n-i}b^n$, joten $xz \notin L'$. Näin ollen sanaa w ei voida pumpata, eikä L' ole säännöllinen, joten myöskään L ei ole säännöllinen.

   Alla kielen L kuvaava yhteydetön kielioppi G:

   $G = (V, \Sigma, P, S)$, missä
   $V = \{S, T, a, b\}$,
   $\Sigma = \{a, b\}$,
   $P = \{\ S \to SS \mid aT \mid Ta \mid \varepsilon,$
   $\quad T \to ST \mid TS \mid b\}$

   Tässä kieliopissa välikkeellä S johdetaan merkkijonot, joissa on molempia aakkosia yhtä monta, ja T:llä kaikki joissa on b-merkkejä yksi enemmän kuin a-merkkejä.

   Esimerkki. Annetaan johto merkkijonolle $aababb \in L$.

$$\begin{aligned} S &\Rightarrow aT \\ &\Rightarrow aST \\ &\Rightarrow aaTT \\ &\Rightarrow aabT \\ &\Rightarrow aabST \\ &\Rightarrow aabaTT \\ &\Rightarrow aababT \\ &\Rightarrow aababb \end{aligned}$$

6. **Tehtävä**: Laadi yhteydetön kielioppi, joka tuottaa kaikki seuraavan esimerkin tapaiset, yksinkertaisista sisäkkäisistä `for`-silmukoista, `begin`- ja `end`-sulkeilla kootuista lauseista ja alkeisoperaatioista `a` rakentuvat "ohjelmat":

```
a;
for 3 times do
begin
  for 5 times do a;
  a; a
end
```

   Silmukkalaskureiden voit olettaa olevan kokonaislukuja väliltä $0, \ldots, 9$.

   Vastaus: Ohjelmointikielten kieliopit määritellään useimmiten siten, että aakkostoksi otetaan kielessä esiintyvät syntaktiset elementit (lekseemit). Tässä tapauksessa niitä ovat numerot, `a` sekä varatut sanat. Ohjelman jäsentäminen jaetaan kahteen osaan:

(a) Muutetaan ohjelman teksti jonoksi lekseemeitä tilakoneiden avulla.

(b) Muodostetaan lekseemijonon jäsennyspuu.

Tehtävän kieliopin voi määritellä monellakin eri tapaa, tässä on yksi mahdollinen tulkinta:

$$
\begin{aligned}
G =& (V, \Sigma, P, C) \\
V =& \{C, S, N, \textbf{begin}, \textbf{do}, \textbf{end}, \textbf{for}, \textbf{times}, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ;, a\} \\
\Sigma =& \{\textbf{begin}, \textbf{do}, \textbf{end}, \textbf{for}, \textbf{times}, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ;, a\}
\end{aligned}
$$

Tässä välikkeen S tulkintana on ”lause” (*statement*), C:n ”yhdistetty lause” (*compound statement*) ja N:n ”numero”. Kieliopin säännöt määritellään seuraavasti:

$$
\begin{aligned}
P = \{&C \to S \mid S; C \\
&S \to a \mid \textbf{begin}\ C\ \textbf{end} \mid \textbf{for}\ N\ \textbf{times do}\ S \\
&N \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\}
\end{aligned}
$$

Esimerkki. Tehtävänannossa esiintyneen ohjelman jäsennyspuu: