T-79.1001 **Syksy 2007**
Tietojenkäsittelyteorian perusteet T
Harjoitus 7
Demonstraatiotehtävien ratkaisut

Lemma (Säännöllisten kielten 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.
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**: *Hahmolausekkeet* ovat esimerkiksi UN*X-järjestelmien tekstityökaluissa käytetty säännöllisten lausekkeiden yleistys, jossa sallitaan merkkijonoarvoisten muuttujien käyttö lausekkeissa. Sovitettaessa merkkijonoa annettuun lausekkeeseen vaaditaan, että tietynnimisen muuttujan arvoksi tulee eri kohdissa sama osamerkkijono. Siten esimerkiksi aXb^*Xa ja $aX(a \cup b)^*YX(a \cup b)^*Ya$ ovat aakkoston $\{a, b\}$ hahmolausekkeita, joista ensimmäinen kuvaa kielen $\{awb^nwa \mid w \in \{a, b\}^*,\ n \geq 0\}$. Osoita, että hahmolausekkeet ovat säännöllisten lausekkeiden aito yleistys, so. niillä voidaan kuvata myös joitakin ei-säännöllisiä kieliä.

   Vastaus: Osoittaaksemme, että tehtävän hahmolausekkeet on säännöllisten lausekkeiden aito yleistys, tulee meidän löytää hahmolauseke, jonka määrittämä kieli ei ole säännöllinen.

   Tarkastellaan hahmolauseketta XX vastaavaa kieltä $L = \{zz \mid z = \{a, b\}^*\}$. Oletetaan, että L on säännöllinen. Valitaan $x = a^pba^pb \in L$, missä n on pumppauslemmassa esiintyvä kielestä L riippuva kokonaisluku. Nyt $|x| = 2n + 2 > n$. Pumppauslemman mukaan voidaan kirjoittaa $x = uvw$, missä $|uv| \leq n$ ja $|v| \geq 1$. Siis $u = a^{n-|v|-k}$, $v = a^{|v|}$ ja $w = a^kba^nb$, missä $0 \leq k < n$. Nyt pumppauslemman mukaan kaikille $i \geq 0$ tulisi päteä $uv^iw \in L$. Kuitenkin $uv^0w = uw = a^{n-|v|}ba^nb \notin L$, sillä se ei ole muotoa zz, koska vaadittiin $|v| \geq 1$. Päädyttiin siis ristiriitaan oletuksen kanssa. L ei näin ollen voi olla säännöllinen.

   Löydettiin siis ei-säännöllinen kieli, joka voidaan kuvata hahmolausekkeella. Näin ollen hahmolausekkeet ovat säännöllisten lausekkeiden aito yleistys. □

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$\}$ vo 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:

$$S \to aSbS \mid bSaS \mid \varepsilon$$

6. **Tehtävä**: Laadi algoritmi, joka testaa onko annetun yhteydettömän kieliopin $G = (V, \Sigma, P, S)$ tuottama kieli epätyhjä, so. voidaanko kieliopin lähtösymbolista S johtaa yhtään päätejonoa $x \in \Sigma^*$.

   Vastaus: Allaoleva proseduuri ?GeneratesNonemptyLanguage(G) ottaa syötteenä yhteydettömän kieliopin G, ja palauttaa arvon `true`, jos G:n generoima kieli ei ole tyhjä.

```
?GeneratesNonemptyLanguage(G = (V, Σ, P, S): context-free grammar)
  T ← Σ
  repeat |V − Σ| times
    for each A → X_1 ··· X_k ∈ P
      if A ∉ T ∧ X_1 ··· X_k ∈ T^k
        T ← T ∪ {A}
  if S ∈ T
    return true
  else
    return false
```

   Algoritmin idea on lähteä terminaalisymbolien joukosta Σ ja testata, onko näistä mahdollista perääntyä S:ään käyttäen joukon P produktioita käänteisesti. Perääntymistä simuloidaan iteroimalla $|V - \Sigma|$ kertaa saavutettavien symbolien joukkoa T.

   Perusteluksi sille, että $|V - \Sigma|$ askelta riittää, tarkastellaan sanaa $z \in L(G)$, jolla on kielen sanoista kaikkein pienin jäsennyspuu. Jos z:lla on muotoa

$$S \to^* uAy \to^* uvAxy \to^* uvwxy$$

   oleva johto, missä $u, v, w, x, y \in \Sigma^*$, niin myös sana $z' = uwy$ voidaan johtaa kieliopin säännöillä[1]. Tällöin kuitenkin z':n jäsennyspuu on pienempi kuin z:n jäsennyspuu, mikä on ristiriidassa sen oletuksen kanssa, että z:n puu on pienin. Tästä seuraa se, että missään z:n minimaalisen jäsennyspuun haaroissa ei voi esiintyä sama välike kahteen kertaan,

[1] Vertaa yhteydettömien kielten pumppauslemmaan.

joten algoritmissa riittää käydä sääntöjoukko läpi yhtä monta kertaa kuin kieliopissa on välikkeitä.

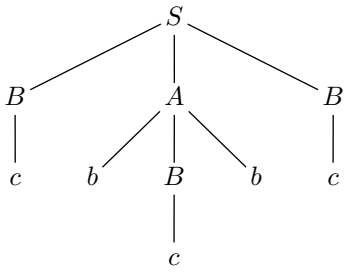
Tarkastellaan esimerkiksi kielioppia:

$$\begin{aligned} S &\to BAB \mid ABA \\ A &\to aAS \mid bBa \\ B &\to bBS \mid c \end{aligned}$$

Algoritmin laskenta etenee joukon T osalta seuraavasti:

$$\begin{aligned} T_0 &= \{a, b, c\} \\ T_1 &= \{a, b, c, B\} && (B \to c) \\ T_2 &= \{a, b, c, A, B\} && (A \to bBa) \\ T_3 &= \{a, b, c, A, B, C, S\} && (S \to BAB, S \to ABA) \end{aligned}$$

Koska $|V - \Sigma| = 3$, algoritmin suoritus päättyy ja $T = T_3$. Huomataan, että $S \in T$, joten kieli ei ole tyhjä. Pienin kieleen kuuluvan sanan jäsennyspuu on:



Liite: Chomskyn normaalimuoto

Muutetaan esimerkin vuoksi seuraava kielioppi Chomskyn normaalimuotoon:

$$\begin{aligned} P = \{S &\to aAS \mid bBS \mid \varepsilon \\ A &\to aAA \mid b, \\ B &\to bBB \mid a\} \end{aligned}$$

Kielioppi on Chomskyn normaalimuodossa, mikäli seuraavat ehdot toteutuvat:

(a) Ainoastaan alkuvälike S voi olla tyhjentyvä.

(b) Alkuvälike S ei esiinny säännöissä oikealla puolella.

(c) Mahdollisesti esiintyvää sääntöä $S \to \varepsilon$ lukuunottamatta kaikki säännöt ovat muotoa $A \to BC$ tai $A \to a$, missä A, B ja C ovat välikkeitä ja a terminaalisymboli.

Kielioppi muutetaan normaalimuotoon vaiheittain:

(a) **Poistetaan lähtösymboli sääntöjen oikealta puolelta**.
Koska kieliopissa on säännöt $S \to aAS$ ja $S \to bBS$, lisätään uusi lähtösymboli S' ja sääntö $S' \to S$. Saadaan tulokseksi sääntöjoukko:

$$\begin{aligned} S' &\to S, \\ S &\to aAS \mid bBS \mid \varepsilon \\ A &\to aAA \mid b, \\ B &\to bBB \mid a \end{aligned}$$

(b) **Poistetaan ε-produktiot**.

Koska Chomskyn normaalimuodossa ainoastaan lähtösymboli S' saa olla tyhjentyvä, täytyy muut ε-säännöt poistaa kieliopista. Lasketaan aluksi tyhjentyvien välikkeiden joukko NULL:

$$\begin{aligned} \text{NULL}_0 &= \{S\} && (S \to \varepsilon) \\ \text{NULL}_1 &= \{S, S'\} && (S' \to S) \\ \text{NULL}_2 &= \{S, S'\} = \text{NULL} \end{aligned}$$

Tämän jälkeen korvataan säännöt $A \to X_1 \cdots X_n$ joukolla sääntöjä

$$A \to \alpha_1 \cdots \alpha_2, \quad \text{missä } \alpha_i = \begin{cases} X_i, X_i \notin \text{NULL} \\ X_i \text{ tai } \varepsilon, X_i \in \text{NULL} \end{cases}$$

Lopuksi poistetaan kaikki säännöt muotoa $A \to \varepsilon$ (lukuunottamatta sääntöä $S' \to \varepsilon$). Saadaan tulokseksi sääntöjoukko[2]:

$$\begin{aligned} S' &\to S \mid \varepsilon \\ S &\to aAS \mid aA \mid bBS \mid bB \\ A &\to aAA \mid b, \\ B &\to bBB \mid a \end{aligned}$$

(c) **Poistetaan yksikköproduktiot**.

Seuraavaksi poistetaan kieliopista kaikki muotoa $A \to B$ olevat säännöt, missä sekä A että B ovat välikkeitä.

Lasketaan ensin joukot $F(A)$ kaikilla $A \in V - \Sigma$:

$$\begin{aligned} F(A) = F(B) = F(S) &= \emptyset \\ F(S') &= \{S\} \end{aligned}$$

Välike B kuuluu joukkoon $F(A)$ täsmälleen silloin, kun A:sta voidaan johtaa B käyttäen pelkkiä yksikköproduktioita.

Sääntö $A \to B$ korvataan sääntöjoukolla $\{A \to w \mid \exists C \in F(B) \cup \{B\} : C \to w \in P\}$. Tulokseksi saadaan sääntöjoukko:

$$\begin{aligned} S' &\to aAS \mid aA \mid bBS \mid bB \mid \varepsilon \\ S &\to aAS \mid aA \mid bBS \mid bB \\ A &\to aAA \mid b, \\ B &\to bBB \mid a \end{aligned}$$

(d) **Poistetaan liian pitkät produktiot**.

Viimeisessä vaiheessa lisätään kielioppiin uusi välike C_σ sekä sääntö $C_\sigma \to \sigma$ kaikille $\sigma \in \Sigma$ sekä jaetaan kaikki säännöt $A \to w$ ($|w| > 2$) ketjuksi sääntöjä, jotka kaikki johtavat tismalleen kaksi symbolia.

[2]Tarkkaan ottaen tässä vaiheessa pitäisi lisätä vielä uusi aloitusvälike S'' ja säännöt $S'' \to \varepsilon | S'$, mutta tässä tapauksessa ei synny ongelmia, vaikka käytetään S':a lähtösymbolina.

Annetun kieliopin Chomskyn normaalimuodoksi saadaankin seuraava sääntöjoukko:

$$
\begin{aligned}
S' &\to C_a S_1' \mid C_a A \mid C_b S_2' \mid C_b B \mid \varepsilon \\
S_1' &\to AS \\
S_2' &\to BS \\
S &\to C_a S_1 \mid C_a A \mid C_b S_2 \mid C_b B \\
S_1 &\to AS \\
S_2 &\to BS \\
A &\to C_a A_1 \mid b \\
A_1 &\to AA \\
B &\to C_a B_1 \mid a \\
B_1 &\to BB \\
C_a &\to a \\
C_b &\to b
\end{aligned}
$$

Liite: CYK-algoritmi

CYK-algoritmilla voidaan tutkia kuuluuko sana $x = x_1 \cdots x_n$ kieliopin G määrittelemään kieleen. Algoritmin kuluessa lasketaan välikejoukot N_{ik}. Joukko N_{ik} käsittää kaikki ne välikkeet, joista voidaan johtaa osajono $x_i \cdots x_{i+k}$, eli sanan x kohdasta i alkava k:n merkin mittainen osajono. Joukkojen laskemisessa voidaan käyttää apuna dynaamista ohjelmointia seuraavaan tapaan:

$$
\begin{aligned}
N_{i,1} &= \{A \mid (A \to x_i) \in P\} \\
N_{i,k} &= \bigcup_{j=1}^{k-1} \{A \in V - \Sigma \mid G\text{:ssä on sääntö } A \to BC, \\
&\qquad\qquad\qquad\qquad \text{missä } B \in N_{ij} \text{ ja } C \in N_{i+j,k-j}\}
\end{aligned}
$$

Tarkastellaan kielioppia G:

$$
\begin{aligned}
S &\to C_a D \mid C_a A \mid C_a E \mid BC_b \mid a \mid b \\
A &\to C_a D \mid C_a A \mid a \\
B &\to C_a E \mid BC_b \mid b \\
D &\to AC_b \\
E &\to BC_b \\
C_a &\to a \\
C_b &\to b
\end{aligned}
$$

Tarkistetaan, kuuluvatko sanat $w_1 = aabbb$ ja $w_2 = aabb$ kieleen $L(G)$. Koska w_2 on w_1:n prefiksi, täytyy taulukko tehdä ainoastaan sanalle w_1.

Lasketaan ensin joukot N_{i1}, $i \leq 5$:

		$i \to$				
		$1:a$	$2:a$	$3:b$	$4:b$	$5:b$
$k \downarrow$	1	$\underline{a}abbb$	$a\underline{a}bbb$	$aa\underline{b}bb$	$aab\underline{b}b$	$aabb\underline{b}$
		$\{S, A, C_a\}$	$\{S, A, C_a\}$	$\{S, B, C_b\}$	$\{S, B, C_b\}$	$\{S, B, C_b\}$

Kussakin taulukon ruudussa on alleviivattuna sitä vastaava sanan osajono.

Lasketaan seuraavaksi N_{12}, eli niiden välikkeiden joukko, jolla voidaan johtaa sanan alussa oleva aa. Koska normaalimuotoisessa kieliopissa ei ole yhtään sääntöä, jolla voisi suoraan johtaa useamman kuin yhden terminaalisymbolin, voidaan aa saada ainoastaan siten, että johdetaan jostain välikkeestä kaksi a:n tuottavaa välikettä. Käytännössä tämä tapahtuu

siten, että etsitään kaikki sellaiset välikkeet X, joille on olemassa sääntö $X \to YZ$, missä $Y \in N_{11}$ ja $X \in N_{21}$.

$$\begin{array}{l} N_{11} = \{S, A, C_a\} \\ N_{21} = \{S, A, C_a\} \end{array} \quad \Rightarrow \quad N_{12} = \{S, A\} \ .$$

Tässä käytettiin sääntöjä $S \to C_a A$ ja $A \to C_a A$. Ruudulle N_{22} saadaan vastaavasti:

$$\begin{array}{l} N_{21} = \{S, A, C_a\} \\ N_{31} = \{S, B, C_b\} \end{array} \quad \Rightarrow \quad N_{22} = \{D\} \ .$$

Ainoa ehdot täyttävä sääntö on $D \to AC_b$. Kokonaisuudessaan taulukon toiseksi riviksi muodostuu:

		$i \to$				
		$1:a$	$2:a$	$3:b$	$4:b$	$5:b$
	1	$\underline{a}abbb$ $\{S, A, C_a\}$	$a\underline{a}bbb$ $\{S, A, C_a\}$	$aa\underline{b}bb$ $\{S, B, C_b\}$	$aab\underline{b}b$ $\{S, B, C_b\}$	$aabb\underline{b}$ $\{S, B, C_b\}$
$k \downarrow$	2	$\underline{aa}abbb$ $\{S, A\}$	$a\underline{ab}bbb$ $\{D\}$	$aa\underline{bb}b$ $\{S, B, E\}$	$aab\underline{bb}$ $\{S, B, E\}$	

Ruudun N_{13} kohdalla huomataan, että sanan kolme ensimmäistä merkkiä (aab) voidaan johtaa kahdella eri tavalla:

(a) johdetaan a välikkeellä $X \in N_{11}$ ja ab välikkeellä $Y \in N_{22}$; tai

(b) johdetaan aa välikkeellä $X \in N_{12}$ ja b välikkeellä $Y \in N_{31}$.

Vastaavat joukot ovat:

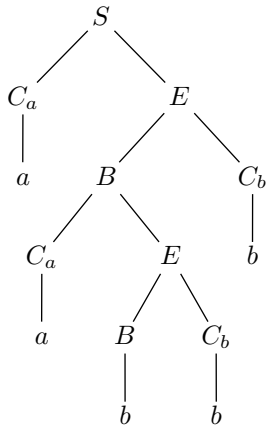
$$\begin{array}{lll} j = 1 & \Rightarrow & \begin{array}{l} N_{11} = \{S, A, C_a\} \\ N_{22} = \{D\} \end{array} \end{array} \qquad \begin{array}{lll} j = 2 & \Rightarrow & \begin{array}{l} N_{12} = \{S, A\} \\ N_{31} = \{S, B, C_b\} \end{array} \end{array}$$

Tapausta $j = 1$ vastaava välikejoukko on $\{S, A\}$ ($S \to C_a D$, $A \to C_a D$) ja tapausta $j = 2$ vastaava on $\{D\}$ ($D \to AC_b$), joten $N_{13} = \{S, A, D\}$. Samaan tapaan jatkamalla saadaan lopulta koko taulukoksi:

		$i \to$				
		$1:a$	$2:a$	$3:b$	$4:b$	$5:b$
	1	$\underline{a}abbb$ $\{S, A, \underline{C_a}\}$	$a\underline{a}bbb$ $\{S, A, \underline{C_a}\}$	$aa\underline{b}bb$ $\{S, \underline{B}, C_b\}$	$aab\underline{b}b$ $\{S, B, \underline{C_b}\}$	$aabb\underline{b}$ $\{S, B, \underline{C_b}\}$
	2	$\underline{aa}abbb$ $\{S, A\}$	$a\underline{ab}bbb$ $\{D\}$	$aa\underline{bb}b$ $\{S, B, \underline{E}\}$	$aab\underline{bb}$ $\{S, B, E\}$	
$k \downarrow$	3	$\underline{aab}bb$ $\{A, S, D\}$	$a\underline{abb}bb$ $\{S, \underline{B}\}$	$aa\underline{bbb}$ $\{S, B, E\}$		
	4	$\underline{aabb}b$ $\emptyset$	$a\underline{abbb}$ $\{S, B, \underline{E}\}$			
	5	$\underline{aabbb}$ $\{\underline{S}, B\}$				

Viimeisessä vaiheessa täytyi käydä läpi neljä eri kombinaatiota: (N_{11}, N_{24}), (N_{12}, N_{33}), (N_{13}, N_{42}) sekä (N_{14}, N_{51}).

Koska $S \in N_{15}$, niin $aabbb \in L(G)$. Toisaalta, koska $S \notin N_{1,4}$, niin $aabb \notin L(G)$. Taulukosta voidaan konstruoida suoraan sanan w_1 jäsennyspuu käymällä se läpi takaperin: S saadaan ruutuun N_{15} ruuduista N_{11} ja N_{24} käyttäen välikkeitä C_a ja E, joten ensimmäisenä käytetään sääntöä $S \to C_a E$. Ylläolevassa taulukossa on sanan w_1 johdossa käytetyt välikkeet alleviivattu. Kokonaisuudessaan jäsennyspuu on seuraavanlainen:


S
C_a
E
a
B
C_b
C_a
E
b
a
B
C_b
b
b