Ukeoppgaver i INF3110/4110

Uke 42 (15.-17.10.2003)

Oppgave 1

(Dette er en fortsettelse av oppgave 4 fra forrige uke, hvor de to første punktene ble gitt.)

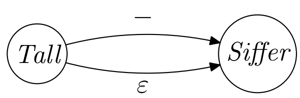
$$
\begin{array}{lll}
\langle \textit{FloatingPointLiteral} \rangle & \rightarrow & \langle \textit{Digits} \rangle \; \mathbf{.} \; \langle \textit{Digits} \rangle^{?} \; \langle \textit{ExponentPart} \rangle^{?} \; | \\
 & & \mathbf{.} \; \langle \textit{Digits} \rangle \; \langle \textit{ExponentPart} \rangle^{?} \; | \\
 & & \langle \textit{Digits} \rangle \; \langle \textit{ExponentPart} \rangle \\
\langle \textit{Digits} \rangle & \rightarrow & \langle \textit{Digit} \rangle^{+} \\
\langle \textit{Digit} \rangle & \rightarrow & \mathbf{0} \; | \; \mathbf{1} \\
\langle \textit{ExponentPart} \rangle & \rightarrow & \mathbf{E} \; [\![\mathbf{+} \; | \; \mathbf{-}]\!]^{?} \; \langle \textit{Digits} \rangle
\end{array}
$$

1. Skriv grammatikken i form av et jernbanediagram.
2. Dette er grammatikken til et regulært språk (selv om det ikke umiddelbart ser slik ut). Skriv om grammatikken slik at den oppfyller kravene til en klassisk BNF-definisjon av et regulært språk.
3. Lag en ikke-deterministisk automat for grammatikken, basert på jernbanediagrammet i punkt 1.
4. Bruk standard-algoritmen for å gjøre automaten i forrige punkt om til en deterministisk automat.
5. Lag en ikke-deterministisk automat basert på den endrede BNF-grammatikken fra punkt 2.

   Eksempel:
   En definisjon

   $\langle \textit{Tall} \rangle \rightarrow - \; \langle \textit{Siffer} \rangle \; | \; \langle \textit{Siffer} \rangle$

   vil typisk bli oversatt til:

6. Gjør automaten du kom frem til i forrige punkt om til en deterministisk automat.

   Konstruer også den tilsvarende tilstandstabellen.
7. Sammenlign de to deterministiske automatene fra punkt 4 og 6. (De bør helst beskrive det samme regulære språket!)

Oppgave 2

Gitt følgende regulære uttrykk

$$\mathbf{R} : [\![\mathbf{0}|\mathbf{1}]\!][\![\mathbf{0}|\mathbf{1}]\!]^*\mathbf{0}[\![\mathbf{0}|\mathbf{1}]\!]$$

Vi kaller i det følgende også språket beskrevet av det regulære uttrykket for **R**.

1. Beskriv med ord hvordan setningene i språket **R** ser ut.
2. Lag en ID-automat som aksepterer setninger i språket **R**.
3. Ta utgangspunkt i ID-automaten fra forrige punkt og lag en tilsvarende D-automat.
4. Lag en BNF-grammatikk som beskriver det samme språket som **R**.

Oppgave 3

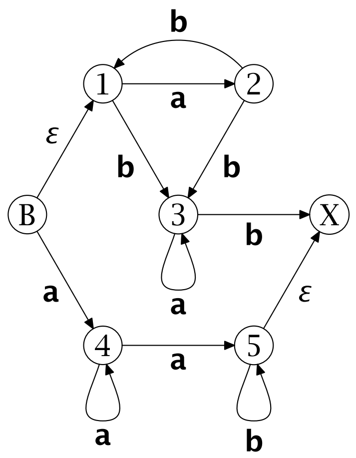
Se på følgende grammatikk over alfabetet $\{\mathsf{a}, \mathsf{b}\}$:

$$\langle S \rangle \to \langle A \rangle\ \langle B \rangle$$
$$\langle A \rangle \to \mathbf{3}\ \langle A \rangle\ \mathbf{3} \mid \varepsilon$$
$$\langle B \rangle \to \mathbf{5}\ \langle B \rangle\ \mathbf{5} \mid \varepsilon$$

1. Gi en avledning av setningen **3333555555**
2. Finn en kort og intuitiv beskrivelse av språket som denne grammatikken definerer.

Oppgave 4

Gitt følgende endelige, ikke-deterministiske automat:



B angir starttilstanden, X angir sluttilstanden/aksepterende tilstand og 1,. . . , 5 er navn på de andre tilstandene.

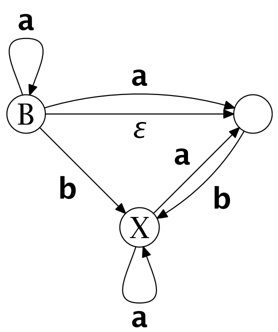
Finn ut om denne automaten godtar følgende strenger:

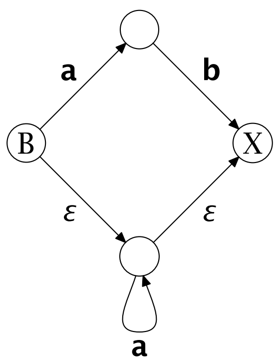
1. **abb**
2. **abbbb**
3. **aabb**
4. **abab**
5. **ababb**
6. **ababab**

Hvis automaten gjør det, så gi den tilsvarende sekvensen av tilstander.

Oppgave 5

Gitt følgende endelige, *ikke-deterministiske* automater:





Konstruer (ved metoden gitt i kompendiet og på forelesning) endelige, deterministiske automater fra disse.

Oppgave 6 (kan puffes til neste uke)

Gitt følgende BNF-eksempel for $\langle Program \rangle$:

$$
\begin{array}{ll}
\langle \mathit{Program} \rangle & \rightarrow \textbf{begin}\ \langle \mathit{Deklsekv} \rangle\ \langle \mathit{Setnsekv} \rangle\ \textbf{end} \\
\langle \mathit{Deklsekv} \rangle & \rightarrow \langle \mathit{Deklsekv} \rangle\ \langle \mathit{Dekl} \rangle\ \textbf{;}\ |\ \varepsilon \\
\langle \mathit{Setnsekv} \rangle & \rightarrow \langle \mathit{SetnsekvX} \rangle\ |\ \varepsilon \\
\langle \mathit{SetnsekvX} \rangle & \rightarrow \langle \mathit{Setn} \rangle\ |\ \langle \mathit{SetnsekvX} \rangle\ \textbf{;}\ \langle \mathit{Setn} \rangle \\
\langle \mathit{Dekl} \rangle & \rightarrow \textbf{dekl...} \\
\langle \mathit{Setn} \rangle & \rightarrow \textbf{put...}\ |\ \textbf{if...}\ |\ \textbf{while...}
\end{array}
$$

Vi oppgir ikke her nøyaktig hvordan deklarasjonene og setningene ser ut, bare at de starter med et eget nøkkelord, og at de har en indre struktur som ikke skaper problemer.

1. Skriv noen setninger i språket og tegn syntakstreet for hver av disse.
2. Hvilke av metasymbolene kan produsere den tomme setning? (Dette kalles *meta-til-tom mengden*.)
3. For hvert metasymbol $\langle X \rangle$ skal du finne mengden av de grunnsymboler eller terminaler som kan stå *først* i en avledning som starter med $\langle X \rangle$. (Dette kalles *startmengden* til metasymbolet $\langle X \rangle$.)