Øving 11 - TDAT2005 Algoritmer og datastrukturer

IDI – NTNU Høsten 2019

Oppgave 1

Vi vet at unionen av to regulære språk er et regulært språk. Ved Kleenes teorem, så vet vi at regulære språk er nøyaktig de som aksepteres av endelige automater. Kan du «snekre» et eksempel på en automat som gjenkjenner unionen i del b) nedenfor?

a) La $\Sigma = \{0,1\}$. La automater som aksepterer språkene

- $L_1 = \{s \in \Sigma^* \mid \text{ antall enere er delelig med } 2\}$
- $L_2 = \{s \in \Sigma^* \mid \text{ antall enere er delelig med } 3\}$

b) Basert på automatene i a), lag en automat som gjenkjenner $L_1 \cup L_2$.

(Hint: Kartesisk produkt!)

Oppgave 2

a) Lag en grammatikk over terminalalfabetet $\{0,1\}$, som beskriver språket bestående av alle ord som palindromer, dvs som er like om de leses forlengs eller baklengs.

b) Bevis at dette språket ikke er regulært.
(Hint: Her kan dere bruke samme idé som i beviset for at språket $\{a^n b^n \mid n > 0\}$ ikke er regulært, eksempel 12.2.8 i boka, side 804, og som vi snakket om på forelesning.)

Oppgave 3

Lag en klasse Automat i et programmeringsspråk. Tilstandene nummeres med heltall, med 0 som starttilstanden. Inputalfabetet er en liste med tegn (char).

a) Klassen skal ha en konstruktør med argumenter som gir opplysninger om

- Inputalfabetet
- Aksepterende tilstander
- Neste-tilstand-tabellen, som en todimensjonal tabell med like mange linjer som tilstander, og like mange kolonner som antall tegn i inputalfabetet.

b) Klassen skal ha en metode sjekkInput(input) som tar en liste/array, og som returnerer logisk «True» dersom automaten aksepterer input, «False» ellers.

c) Test koden på automatene i oppgave 8 i øving 10:

- Test automaten i øving 10, oppgave 8 a) med inputstrenger ϵ (tomme streng), $010, 111, 010110$ og 001000.
- Test automaten i oppgave 8 b) med inputstrenger $abbb$, $aaab$ og $babab$.

Oppgave 4

a) Beskriv en måte å generere strengen $babbab$ på i grammatikken gitt under:

$$V = \{a, b, S, A\}$$
$$\Sigma = \{a, b\}$$
$$R = \{S \to AA, A \to AAA, A \to a, A \to bA, A \to Ab\}$$

b) (Fra eksamen h12 i LC119 Diskret matematikk) La L være språket bestående av alle bitstrenger med et odde antall enere. Forklar hvorfor den kontekstfrie grammatikken under ikke genererer nøyaktig dette språket:

Alfabetet er $V = \{S, N, 0{,}1\}$ med terminalsymboler $\Sigma = \{0{,}1\}$, startsymbolet er S og reglene er som følger:

$$\begin{aligned}
&(1)\quad S \to N1NS \\
&(2)\quad N \to 0N \\
&(3)\quad N \to \varepsilon \\
&(4)\quad S \to \varepsilon
\end{aligned}$$

Oppgave 5

Lag en grammatikk som generer alle lovlige boolske uttrykk som bruker operatorene ! (ikke), || (eller) og && (og) i et programmeringsspråk. For enkelhets skyld antar vi at p, q og r er eneste lovlige variabelnavn. Grammatikken skal kun generere lovlige uttrykk. Eksempler på ulovlige uttrykk er pq og $!\,p\&\&||r$. Terminalsymbolene er

$$\Sigma = \{p, q, r, (,), !, ||, \&\&\}$$

Utover dette velger du selv alfabet og regler. Bestem deg for hvordan du vil bruke paranteser (det er enklest med mange), så lenge uttrykkene gir matematisk mening.

(For at uttrykkene skal ha entydig tolkning, kan vi bruke paranteser «over alt», eller bruke presedensregler. Vanligst (?) er at ! (negasjon) har høyest prioritet, deretter følger && og så ||. (Og det regnes fra venstre mot høyre for operatorer med lik presedens, men det trengs ikke her, siden vi har gitt dem forskjellige presedens og de er assosiative.))