NTNU
Institutt for matematiske fag

TMA4140
Diskret matematikk

Plenumsregning 13

Høst 2023

1 La $\mathcal{S}$ være språket av bitstrenger b slik at 3 deler antallet 1-ere i b.

a) Vis at $\mathcal{S}$ er regulært ved å finne en endelig tilstandsmaskin som gjenkjenner nøyaktig $\mathcal{S}$ blant alle endelige bitstrenger.

b) Finn et regulært uttrykk som vil bli tolket som det regulære språket $\mathcal{S}$.

2 I denne oppgaven skal vi skrive en endelig tilstandsmaskin som skal simulere en enkel digital kodelås. Kodelåsen skal 'låses' ved å taste inn mellom 1 og 4 nullere, og åpnes igjen ved å taste inn samme antallet 1-ere og så avslutte med 0. For eksempel representerer strengen `0001110` at låsen først ble låst, og så åpnet på korrekt vist. Dermed skal denne strengen aksepteres. Strengen `000110` skal ikke aksepteres, fordi låsen i dette tilfellet ikke låses opp riktig.

Teori:

Regulære språk og tilsvarende endelige tilstandsmaskiner er nyttige verktøy både i teoretisk datavitenskap og ingeniørvitenskap. Likevel er det viktig å vite at disse maskinene langt ifra kan gjøre 'alt'. En endelig tilstandsmaskin har et veldig begrenset minnne, siden den bare kan være i et gitt antall tilstander. Hvis maskinen skal kunne lagre for eksempel k bits med informasjon må den ha minst 2^k tilstander.

3 (Utfordring) La $\mathcal{S}$ være språket som består av bitstrenger som består av en rekke med 0-ere og så en like lang rekke 1-ere (altså $\mathcal{S} = \{0^n 1^n \in \mathcal{B} \mid n \in \mathbb{N}\}$).

Vis at det ikke eksisterer en endelig tilstandsmaskin som aksepterer nøyaktig bitstrengene i $\mathcal{S}$.

Simulering av endelige tilstandsmaskiner

Vi kan teste endelige tilstandsmaskiner og regulære utrykk med for eksempel denne online-simulatoren: `https://ivanzuzak.info/noam/webapps/fsm_simulator/`

En tilstandsmaskin som løser oppgave 1 kan skrives på følgende måte (skriv inn i fanen 'input automaton')

```
#states
s0
s1
s2
#initial
s0
#accepting
s0
#alphabet
0
1
#transitions
s0:0>s0
s1:0>s1
s2:0>s2
s0:1>s1
s1:1>s2
s2:1>s0
```