Løsningsforslag til Obligatorisk oppgave 2

INF1800 – Logikk og beregnbarhet – Høsten 2008

Alfred Bratterud

Oppgave 1

Vi har alfabetet $A = \{a, b\}$ og språkene
$L_1 = \{s \mid s \neq \Lambda\}$
$L_2 = \{s \mid s \text{ inneholder minst tre forekomster av } a\}$
$L_3 = \{s \mid |s| \text{ er et partall}\}$

OBS: *Husk at vi kan finne utallige regulære uttrykk for det samme språket. Nedenfor oppgir vi ett eller to alternativer; dersom du har skrevet noe annet kan du se etter om gruppelærer har gitt noen kommentar på din oblig. Det kan godt være riktig selv om det avviker fra løsningene gitt her.*

(a) Finn et regulært uttrykk R slik at $L(R) = L_1$:
Her trenger vi å lage et uttrykk som beskriver alle strenger over alfabetet, foruten Λ.

$$(a+b)(a+b)^* \text{ og}$$
$$(a+b)^*(a+b)$$

er begge riktige svar. Først har vi $(a+b)^*$ som gir alle strengene i språket;

$$\begin{aligned} L((a+b)^*) &= L(a+b)^* \\ &= (L(a) \cup L(b))^* \\ &= (\{a\} \cup \{b\})^* \\ &= \{a, b\}^* \\ &= \{\Lambda, a, b, aa, bb, ab, ba, aaa, bbb, abb, baa, \ldots\} \end{aligned}$$

Legger vi til $(a+b)$ foran eller bak er vi kvitt Λ; det spiller ingen rolle hvilken vi velger fordi fra algebraen for regulære uttrykk har vi

$$\begin{aligned} RR^* &= R^*R \ \text{ så} \\ (a+b)(a+b)^* &= (a+b)^*(a+b) \end{aligned}$$

(b) Finn et regulært uttrykk R slik at $L(R) = L_2$:
Her er det viktig å merke seg at vi ikke bare er ute etter de strengene som inneholder delstrengen aaa (mange har gjort den feilen), men alle strenger som inneholder minst tre a'er. Vi kan gjøre slik:

$$(a+b)^*a(a+b)^*a(a+b)^*a(a+b)^*$$

men det kan også skrives litt enklere, med tanke på at vi skal lage en DFA senere, der vi må ha determinisme for a:

$$b^*ab^*ab^*a(a+b)^*$$

Her legger vi altså bare et vilkårlig antall b'er mellom de tre første a'ene; siden vi må ha tre forekomster av a må det finnes tre *første* forekomster av a. Til slutt legger vi på hva som helst.

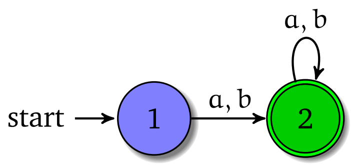
(c) Finn et regulært uttrykk R slik at $L(R) = L_3$:
Vi må bygge opp strengen av par, nærmere bestemt alle mulige par over alfabetet;

$$(aa + bb + ab + ba)^*$$

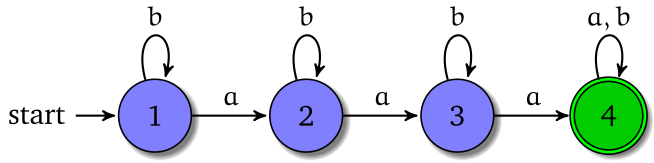
som vi, med tanke på DFA'en vi skal lage, kan forenkle til

$$((a + b)(a + b))^*$$

(d) Gi en deterministisk endelig tilstandsautomat som gjenkjenner L_1:

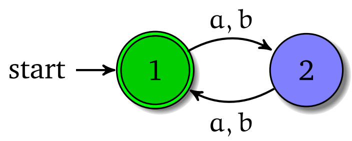


(e) Gi en deterministisk endelig tilstandsautomat som gjenkjenner L_2:



Her vil de tre a-kantene representere de første tre forekomstene av a i input-strengen.

(f) Gi en deterministisk endelig tilstandsautomat som gjenkjenner L_3:



Oppgave 2

Forenkle uttrykket $(a^*b^*)^*a + (b^*a^*)^*a$ ved å bruke algebraiske likheter.
Legg merke til at det eneste som skiller uttrykkene er rekkefølgen på $a*$ og $b*$ inne i parentesene.

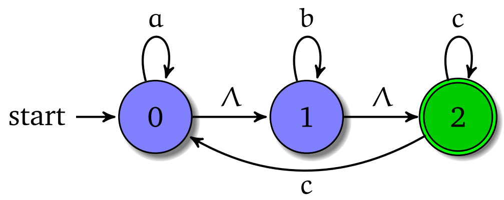
$$\begin{aligned}
(a^*b^*)^*a + (b^*a^*)^*a &= (a^*b^*)^*a + (b + a)^*a && \text{ved } (R^*S^*)^* = (R + S)^* \\
&= (a^*b^*)^*a + (a + b)^*a && \text{ved } R + S = S + R \\
&= (a^*b^*)^*a + (a^*b^*)^*a && \text{ved } (R^*S^*)^* = (R + S)^* \\
&= (a^*b^*)^*a && \text{ved } R + R = R
\end{aligned}$$

Eller gjerne også

$$= (a + b)^*a \qquad \text{ved } (R^*S^*)^* = (R + S)^*$$

Oppgave 3

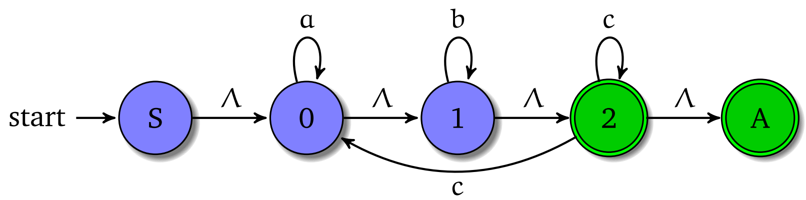
Vi har følgende NFA gitt ved transisjonstabell:



Vi kunne også tatt med $\emptyset$ som egen tilstand, men det er ikke nødvendig siden vi ikke krever determinisme av denne automaten.

(a) Vi bruker algoritmen for å gå fra NFA til regulært uttrykk. Det er ikke et krav i oppgaven å bruke algoritmen, men hvis man lærer seg den vil man være sikker på å finne riktig løsning:

1. Lag ny start- og stopptilstand:



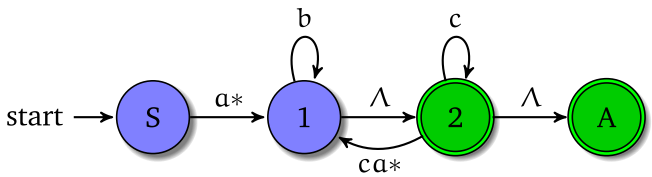
2. Vi skal hele tiden sjekke om det går to kanter fra en tilstand til en annen. I så fall skal vi slå sammen kantene, og dersom vi hadde R over den ene og S over den andre setter vi $R + S$ over den nye kanten. Vi begynner nå å fjerne tilstander, men holder øye med dette hele tiden.
3. Fjern tilstand 0, og lag nye kanter $new(i, j)$ fra hver tilstand i som har en vei til en tilstand j kun via 0. Vi har bruker formelen for nye kanter fra boka, der 0 er satt inn for k, $new(i, j) = \mathrm{old}(i, j) + \mathrm{old}(i, 0)\mathrm{old}(0, 0)^*\mathrm{old}(0, j)$. Vi har her to veier som går via $0: \; S \to 0 \to 1$ og $2 \to 0 \to 1$. Vi får da to nye kanter:

$$
\begin{aligned}
new(S,1) &= \mathrm{old}(S,1) + \mathrm{old}(S,0)\mathrm{old}(0,0)^*\mathrm{old}(0,1) && \\
&= \emptyset + \mathrm{old}(S,0)\mathrm{old}(0,0)^*\mathrm{old}(0,1) && \mathrm{old}(S,1) = \emptyset \\
&= \emptyset + \Lambda\mathrm{old}(0,0)^*\mathrm{old}(0,1) && \mathrm{old}(S,0) = \Lambda \\
&= \emptyset + \Lambda a^*\mathrm{old}(0,1) && \mathrm{old}(0,0) = a \\
&= \emptyset + \Lambda a^*\Lambda && \mathrm{old}(0,1) = \Lambda \\
&= a^* && \emptyset + \Lambda a = a
\end{aligned}
$$

Altså a^* for $S \to 1$ og for $2 \to 1$ får vi:

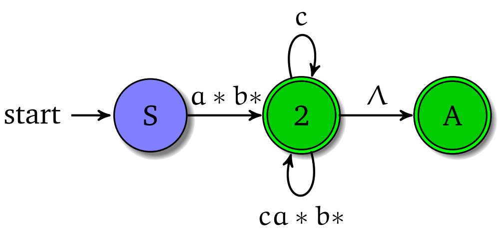
$$
\begin{aligned}
\mathit{new}(2,1) &= \mathtt{old}(2,1) + \mathtt{old}(2,0)\mathtt{old}(0,0)^*\mathtt{old}(0,1) && \\
&= \emptyset + \mathtt{old}(2,0)\mathtt{old}(0,0)^*\mathtt{old}(0,1) && \mathtt{old}(2,1) = \emptyset \\
&= \emptyset + c\,\mathtt{old}(0,0)^*\mathtt{old}(0,1) && \mathtt{old}(2,0) = c \\
&= \emptyset + ca^*\mathtt{old}(0,1) && \mathtt{old}(0,0) = a \\
&= \emptyset + ca^*\Lambda && \mathtt{old}(0,1) = \Lambda \\
&= ca^* && \emptyset + c = c
\end{aligned}
$$

Dette gir følgende bilde

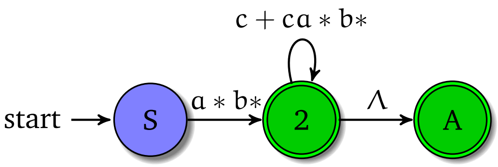


OBS: *Dette er ikke lenger en NFA, fordi vi har kanter med regulære uttrykk over, i stedet for bokstaver i alfabetet.*

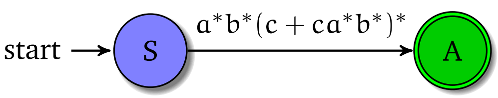
4. Vi fjerner tilstand 1 på samme måte og får:



Her har vi fått en ny løkke i tilstand 2 fordi det gikk en vei fra 2 til 2 via 1 som ble fjernet. Dermed har vi to kanter fra 2 til 2 som nå må slås sammen (her kan vi forenkle uttrykket litt, men det gjør vi til slutt):



5. Til slutt fjerner vi tilstand 2:



Og sitter igjen med det regulære uttrykket

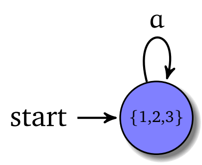
$a^*b^*(c + ca^*b^*)^*$	Som kan forkortes til
$a^*b^*(ca^*b^*)^*$	fordi $c + ca * b* = c(\Lambda + a * b*) = ca * b*$

(b) Vi bruker algoritmen for å gå fra NFA til DFA. Igjen er det ikke noe krav i oppgaven om å bruke en bestemt algoritme, men når vi gjør det vet vi at vi kommer i mål. Husk definisjonen av λ-lukning: $\lambda(S) = \{x \mid x \text{ kan nåes fra } S \text{ kun via } \Lambda\} \cup \{S\}$. Dette blir altså en mengde, siden vi må ha med alle tilstander vi kan nå med Λ fra S, gjerne via andre tilstander, ikke bare direkte.

1. Ny starttilstand er λ-lukningen av den gamle starttilstanden, altså $\lambda(0) = 0, 1, 2$
2. Da skal vi finne alle transisjonene gitt ved T_D (D for deterministisk) fra den nye starttilstanden. La $T_N(s, x)$ være transisjonsfunksjonen slik den står for NFA'en. Vi har da at

$$\begin{aligned}
T_D(0, 1, 2, a) &= \lambda(T_N(0, a)) \cup \lambda(T_N(1, a)) \cup \lambda(T_N(2, a)) \\
&= \lambda(0) \cup \lambda(\emptyset) \cup \lambda(\emptyset) \\
&= \lambda(0) \\
&= \{0, 1, 2\}
\end{aligned}$$

Så langt har vi altså dette:

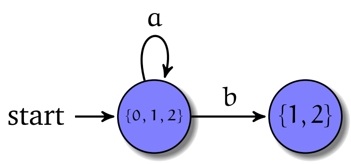


Husk at det som står inne i tilstanden bare er et navn. Her indikerer det at denne tilstanden blir en kombinasjon av tilstandene $1, 2$ og 3 fra NFA'en.

3. Vi går videre og sjekker hva b i starttilstanden gir:

$$\begin{aligned}
T_D(\{0, 1, 2\}, b) &= \lambda(T_N(0, b)) \cup \lambda(T_N(1, b)) \cup \lambda(T_N(2, b)) \\
&= \lambda(\emptyset) \cup \lambda(1) \cup \lambda(\emptyset) \\
&= \lambda(1) \text{ som vi regner ut til} \\
&= \{1, 2\} \text{ ved å se på NFA'en}
\end{aligned}$$

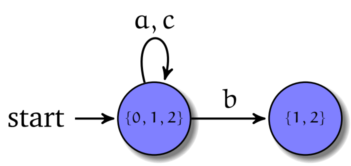
Vi kobler til den nye tilstanden og får



4. For c i starttilstanden får vi:

$$\begin{aligned}
T_D(\{0, 1, 2\}, c) &= \lambda(T_N(0, c)) \cup \lambda(T_N(1, c)) \cup \lambda(T_N(2, c)) \\
&= \lambda(\emptyset) \cup \emptyset \cup \lambda(\{0, 2\}) \\
&= \lambda(\{0, 2\}) \text{ som blir} \\
&= \lambda(0) \cup \lambda(2) \\
&= \{0, 1, 2\} \cup \emptyset \\
&= \{0, 1, 2\}
\end{aligned}$$

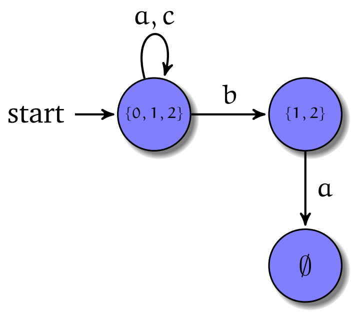
Vi får altså en ny løkke fra $\{0, 1, 2\} \rightarrow \{0, 1, 2\}$, som vi slår sammen med den løkken vi har for a.



5. Vi er ferdige med starttilstanden, så vi går til $\{1,2\}$ og finner transisjonene derfra.

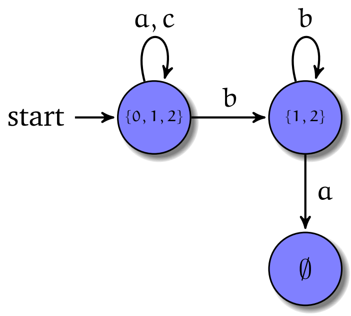
$$\begin{aligned}
T_D(\{1,2\}, a) &= \lambda(T_N(1,a)) \cup \lambda(T_N(2,a)) \\
&= \lambda(\emptyset) \cup \lambda(\emptyset) \\
&= \emptyset
\end{aligned}$$

Siden vi skal ha determinisme må vi lage en tilstand $\emptyset$ som blir en slags "Søppelhåndtering":



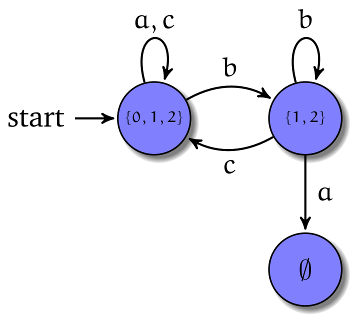
6. Videre ser vi hva b gir i $\{1,2\}$:

$$\begin{aligned}
T_D(\{1,2\}, b) &= \lambda(T_N(1,b)) \cup \lambda(T_N(2,b)) \\
&= \lambda(1) \cup \lambda(\emptyset) \\
&= \{1,2\}
\end{aligned}$$



7. Så regner vi ut hva c gir i $\{1,2\}$:

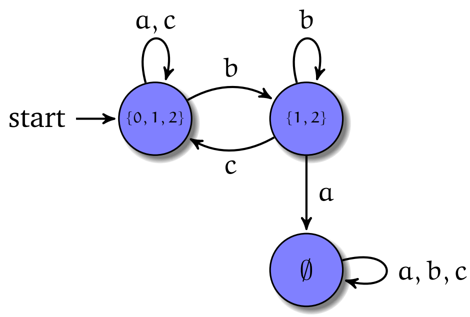
$$\begin{aligned}
T_D(\{1,2\}, b) &= \lambda(T_N(1,c)) \cup \lambda(T_N(2,c)) \\
&= \lambda(\emptyset) \cup \lambda(\{0,2\}) \\
&= \lambda(0) \cup \lambda(2) \\
&= \{0,1,2\} \cup \{2\} \\
&= \{0,1,2\}
\end{aligned}$$



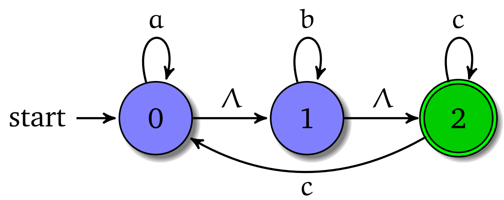
8. Til slutt finner vi hva henholdsvis a, b og c gir i $\emptyset$. Vi har ingen kanter fra $\emptyset$ i den opprinnelige automaten, så

$$\begin{aligned} T_D(\emptyset, \delta) &= \lambda(T_N(\emptyset, \delta) \\ &= \lambda(\emptyset) \\ &= \emptyset \end{aligned}$$

For $\delta = a, b$ og c. Dette gir tre nye løkker, som vi slår sammen til en:



Til sist angir vi hvilke tilstander som er aksepterende. Det er alle nye tilstander som inneholder en av de aksepterende tilstander fra den NFA'en vi begynte med. I vårt tilfelle blir dette alle bortsett fra $\emptyset$. Vi har altså gått fra denne NFA'en:



Til denne DFA'en via algoritmen for NFA→DFA: