

Noen utfyllende kommentarer til INF 2820 V2012, Obligatorisk innleveringsoppgave 1, deloppgave A

Er det i oppgave 1 lov til å skrive flere bokstaver, som "ha" på en kant?

Spørsmålet deler seg i to:

- Hva er symbolmengden (kalles også alfabetet) A ?
- Kan vi skrive mer enn et symbol på en kant?

Et formelt språk består av en mengde symboler A , gjerne kalt alfabetet, og en mengde strenger $L \subseteq A^*$ (Forelesning 1, time 2, slide 4). A kan velges forskjellig, men i denne oppgaven skal det bestå av bokstaver og evt. andre tegn, altså $A1 = \{h, a, e, \dots\}$. Det er også det eneste som JFLAP tillater, og denne oppgaven skal løses i JFLAP.

Men noen vil ha oppdaget at selv om JFLAP bare tillater symboler som består av enkle karakterer, så tillater den at vi skriver flere symboler på en kant. Vil det være tillatt i denne oppgaven (spm. b over)? Nei. Hvorfor ikke? Fordi vi her skal lage en DFA. Hvis vi skriver flere symboler på en kant, vil vi få en NFA – men ikke en DFA. JFLAP bruker den alternative definisjonen av NFA som vi så på i forelesning 2 som tillater mer enn et symbol på en kant. Men at dette ikke er en DFA kan vi også se hvis vi skriver for eksempel "ha" på en kant og prøver JFLAPs "convert to DFA".

Hva når vi går videre inn i dfa.py? Da skal kantene i JFLAP-representasjonen representeres kant for kant. Symbolene skal stadig være enkeltbokstaver og nettverket lages etter modell av brekenettverketsom står øverst i kommentarfeltet i dfa.py.

Hva med oppgave 2 og 3 – kan vi der også bare skrive enkeltbokstaver på hver kant?

Nei. I oppgave 2 og 3 skal vi la symbolmengden være mengder av ord, altså $A2 = \{\text{tre, halv, fem, ti, på, ...}\}$ og $A3 = \{\text{Kari, ga, barnet, dyret, overrakte, ...}\}$. Dette er altså byggestenene. Vi er interessert i å se hvordan vi kan bruke NFA-er til å beskrive fraser og setninger bygget opp fra disse stenene.

Men også her vil vi bare ha ett symbol på hver kant, men nå vil altså "tre" eller for den saks skyld "ha" bli betraktet som ett symbol (se på eksemplene Forelesning 1, time 2, slide 5). Nettverket kan nå skrives etter modell av template.nfa.

Hvordan representeres tapen i Python-programmene – streng eller liste?

Noen vil ha sett at i kommentarene i kodene bruker jeg strenger, for eksempel `recognize2(mydfa, "baaaa", 1)`. Det virker fint når kantene er merket med enkelttegn. Det virker både i dfa.py og nfa.py. Men den samme koden virker også hvis vi representerer tapen med en liste `recognize2(mydfa, ["b", "a", "a", "a", "a"], 1)`. Det er litt overraskende at begge deler virker med samme resultat. Det går fordi Python behandler strenger og lister litt forskjellig. De to uttrykkene `"baaaa"[0]` og `["b", "a", "a", "a", "a"][0]` vil gi samme resultat "b".

I oppgave 2 og 3 må vi bruke lister av strenger for tape, som `iklokke.recog(["kvart", "over", "ni"])`. For å gjøre dette enkelt kan vi starte med en streng og bruke `split()` som `"kvart over ni".split()`. Flere tips på gruppa på torsdag.

Lykke til!