Programmer som data trial exam

Albert Ross Johannessen

December 12, 2024

1

Betragt den ikke-deterministiske endelige automat ("nondeterministic finite automaton", NFA) nedenfor. Det anvendte alfabet er $\{a, b\}$. Der er i alt 5 tilstande, hvor tilstand 5 er den eneste accepttilstand.

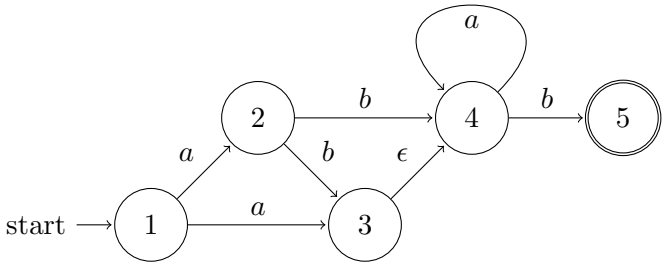

Figure 1: The labeled NFA

1.1 Angiv alle årsager til at automaten er ikke-deterministisk

- Fra knude 1 er der 2 a kanter.
- Fra knude 2 er der 2 b kanter.
- Fra knude 3 er der èn ϵ kant.

1.2 Giv tre eksempler på strenge der genkendes af automaten

- abb
- abaaaab
- aaaaaab

1.3 Giv en uformel beskrivelse af sproget (mængden af all strenge) der beskrives af automaten

- Der er to måder at starte på denne automat.
  1. Ét a og herefter ét b.
  2. Ét eller flere a.

  Herefter er slutningen ens man kan vælge et vilkårligt antal a og herefter **skal** man slutte af på ét b. Kanten fra 2 til 3 er overflødig og det samme er knude 3, hvis man sætter en kant mellem 1 og 4.

1.4 Konstruer og tegn en deterministisk endelig automat

Konstruer og tegn en deterministisk endelig automat ("deterministic finite automaton", DFA) der svarer til automaten ovenfor. Husk at angive starttilstand og accepttilstand (e). Du skal enten bruge en systematisk konstruktion svarende til den i forelæsningen eller som i Introduction to Compiler Design (ICD), eller Basics of Compiler Design (BCD), eller forklare hvorfor den resulterende automat er korrekt.
Vi starter med at konstruerer vores subset.

	a	b	NFA State
S_1	$\{2,3,4\}^{S_2}$	$\{\}$	$\{1\}$
S_2	$\{4\}^{S_3}$	$\{3,4,5\}^{S_4}$	$\{2,3,4\}$
S_3	$\{\}^{S_3}$	$\{5\}^{S_5}$	$\{4\}$
S_4	$\{\}^{S_3}$	$\{5\}^{S_5}$	$\{3,4,5\}$
S_5	$\{\}$	$\{\}$	$\{5\}$

Dette giver følgende DFA, vi skal dog verificere at den er minimal.

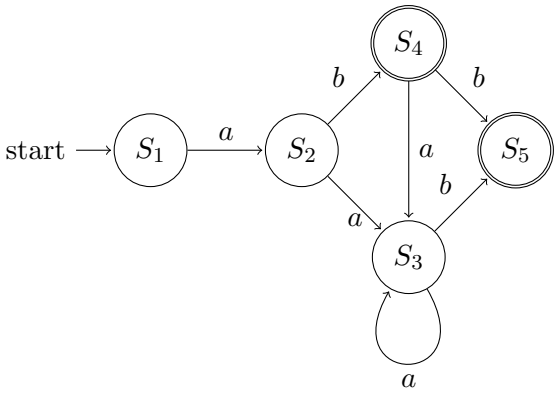

Figure 2: The labeled DFA

Vi starter med at tjekke om grupperne er konsistente.

G_1	$\{S_1, S_2, S_3\}$
G_2	$\{S_4, S_5\}$

G_1	a	b
S_1	$\{G_1\}$	-
S_2	$\{G_1\}$	$\{G_2\}$
S_3	$\{G_1\}$	$\{G_2\}$

G_1 er ikke konsistent så vi opretter 2 nye grupper.

G_2	$\{S_4, S_5\}$
G_3	$\{S_1\}$
G_4	$\{S_2, S_3\}$

G_2	a	b
S_4	$\{G_4\}$	$\{G_2\}$
S_5	-	-

G_3	$\{S_1\}$
G_4	$\{S_2, S_3\}$
G_5	$\{S_4\}$
G_6	$\{S_5\}$

G_3	a	b
S_1	$\{G_4\}$	-

G_4	a	b
S_2	$\{G_4\}$	$\{G_5\}$
S_3	$\{G_4\}$	$\{G_6\}$

G_3	$\{S_1\}$
G_5	$\{S_4\}$
G_6	$\{S_5\}$
G_7	$\{S_2\}$
G_8	$\{S_3\}$

There are only singletons left, this means that we have a minimal DFA.

1.5 Angiv et regulært udtryk

Angiv et regulær udtryk der beskriver mængden af strenge over alfabete $\{a, b\}$, som beskrives af automaten ovenfor. Check og forklar at det regulære udtryk også beskriver mængden af strenge for din DFA.

$$ab?a * b \tag{1}$$

2 Opgave 2

$$\dfrac{p_1\ \dfrac{}{\rho \vdash 21 : \texttt{int}} \qquad \dfrac{p_3\ \dfrac{\rho\,(\texttt{x}) = \forall\alpha.\alpha \to \texttt{int}}{\rho \vdash \texttt{x} : \texttt{int}} \quad p_4\ \dfrac{p_1\ \dfrac{}{\rho \vdash 2 : \texttt{int}} \quad \dfrac{}{\rho \vdash 3 : \texttt{int}}\ p_1}{\rho \vdash \texttt{2+3} : \texttt{int}} \quad \dfrac{}{\rho \vdash 40 : \texttt{int}}\ p_1}{\rho\,[f \to \forall\alpha_1 \ldots \alpha_n.t_x \to t_r] \vdash \texttt{x within [2+3, 40]} : \texttt{bool}}\ inCheck}{\rho \vdash \texttt{let x = 23 in x within [2+3, 40] end} : \texttt{bool}}\ p_6$$

3 Opgave 3

3.1

Givet følgende kode

```
let exVarEnv : varEnv =
  ([
    ("a", (Locvar 6, TypA(TypI, Some 2)));
    ("pn", (Locvar 3, TypA(TypP TypI, Some 1)));
    ("p", (Locvar 1, TypP TypI));
    ("n", (Locvar 0, TypI));
    ("g", (Glovar 0, TypI))
  ], 7)
```

Figure 3

Kan vi se at værdien 7 angiver stak dybden.

- a - har et lokalt offset på 6 med typen array af ints som er på to elementer, da der også skal være plads til en header bliver dette til 3 elementer der bliver allokeret til.
- pn - har et lokalt offset på 3 med typen array af pointers til ins på længden 1, igen på grund af header skal der allokeres til 2 elementer.
- p - har et lokalt offset på 1 med typen pointer til int og dertil skal kun allokeret ét element.
- n - har et lokalt offset på 0 med typen int, og skal kun have allokeret ét element.
- g - har en absolut adresse på 0 med typen int.

Dette vil altså sige at den næste variabel der skal allokeres til skal starte på position 7.