

Leksiline analüüs

Lõplikud automaadid

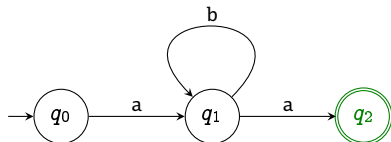
Regulaaravaldise teisendamine automaadiks

Lõplikud automaadid

- **Lõplik automaat** on viisik $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, kus
 - Q on lõplik **olekute** hulk;
 - Σ on lõplik **tähestik**;
 - $\delta \subseteq Q \times (\Sigma \cup \varepsilon) \times Q$ on **üleminekurelatsioon**;
 - $q_0 \in Q$ on **algolek**;
 - $F \subseteq Q$ on **lõppolekute** hulk.
- Lõplik automaat on **determineeritud (DFA)**, kui üleminekurelatsioon on funktsioon $\delta : Q \times \Sigma \rightarrow Q$.
- Vastasel korral on lõplik automaat **mittedetermineeritud (NFA)**.

Lõplikud automaadid

- Lõplikke automaate esitatakse tihti üleminekudiagrammidena:



- Lõplik automaat $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ aktsepteerib keele

$$L(A) = \{w \in \Sigma^* \mid (q_0, w, q_f) \in \delta^*, q_f \in F\}$$

kus $\delta^* \subseteq Q \times \Sigma^* \times Q$ on üleminekurelatsiooni δ refleksiivne transitiivne sulund.

- Teoreem:** Lõplike automaatide poolt aktsepteeritavate keelte klass langeb kokku regulaarsete keeltega.

Determineeritud lõpliku automaadi simuleerimine

Rekursiivse laskumise meetod:

- igale olekule vastab funktsioon, mis loeb ühe sümboli ja ...
- kutsub rekursiivselt välja sellele sümbolile vastava ülemineku sihtoleku funktsiooni.

Determineeritud lõpliku automaadi simuleerimine

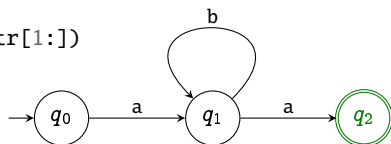
Rekursiivse laskumise meetod:

- igale olekule vastab funktsioon, mis loeb ühe sümboli ja ...
- kutsub rekursiivselt välja sellele sümbolile vastava ülemineku sihtoleku funktsiooni.

```
def q0(inpStr):  
    if inpStr == "":        error()  
    elif inpStr == "a":    return q1(inpStr[1:])  
    else:                  error()
```

```
def q1(inpStr):  
    if inpStr == "":        error()  
    elif inpStr == "a":    return q2(inpStr[1:])  
    elif inpStr == "b":    return q1(inpStr[1:])  
    else:                  error()
```

```
def q2(inpStr):  
    if inpStr == "":        return True  
    else:                  error()
```



Determineeritud lõpliku automaadi simuleerimine

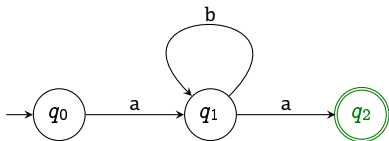
Tabeljuhitav DFA simulaator:

```
Char := next character;  
State :=  $q_0$ ;  
while Char  $\neq$  EOF do  
    State :=  $\delta$ (State, Char);  
    Char := next character;  
end  
if (State  $\in$   $F$ )  
    then return(success);  
    else return(failure);
```

Determineeritud lõpliku automaadi simuleerimine

Tabeljuhitav DFA simulaator:

```
Char := next character;  
State := q0;  
while Char ≠ EOF do  
    State := δ(State, Char);  
    Char := next character;  
end  
if (State ∈ F)  
    then return(success);  
    else return(failure);
```



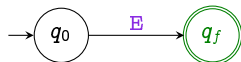
δ	a	b	x
q ₀	q ₁	Error	Error
q ₁	q ₂	q ₁	Error
q ₂	Error	Error	Error

Regulaaravaldiste teisendamine automaadiks

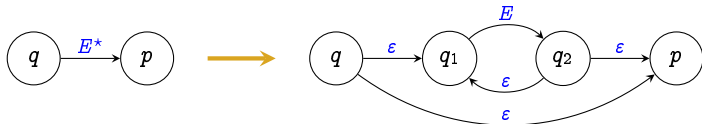
Regulaaravaldise teisendamine automaadiks

Thompsoni konstruktsioon regulaaravaldise teisendamiseks (mitedetermineeritud) lõplikuks automaadiks:

- regulaaravaldisele E seame vastavusse "automaadi":

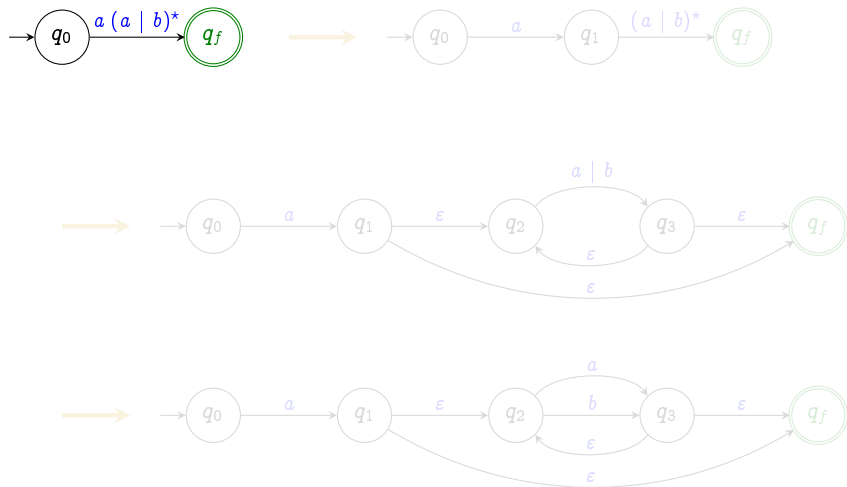


- teisendame "automaati" järgmiste reeglite abil, kuni kõik üleminekud on kas ϵ või üksikud tähed:



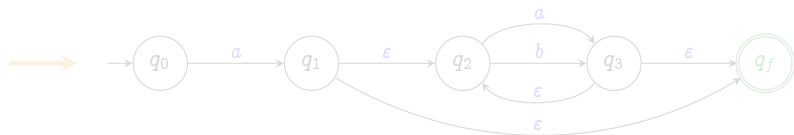
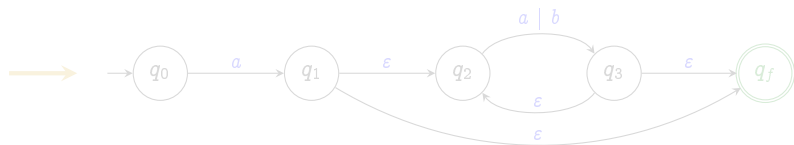
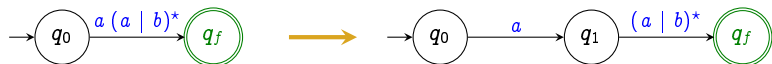
Regulaaravaldise teisendamine automaadiks

Näide:



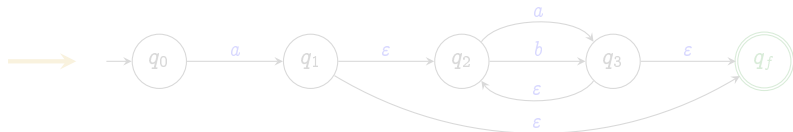
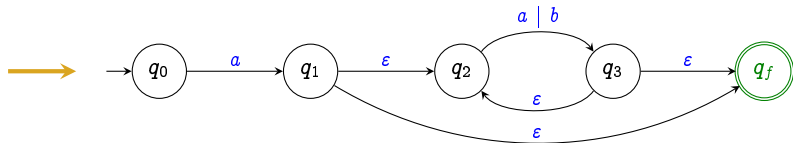
Regulaaravaldise teisendamine automaadiks

Näide:



Regulaaravaldise teisendamine automaadiks

Näide:



Regulaaravaldise teisendamine automaadiks

Näide:

