Leksiline analüüs

Leksiline analüüs

- *Leksiline analüüs* kontrollib programmi sõnade (literaalsümbolite) vastavust leksilistele reeglitele ning teisendab programmi sümbolite (tokens) jadaks:
  - eemaldab tühisümbolid ja kommentaarid;
  - identifitseerib võtmesõnad, identifikaatorid ja konstandid;
  - konstrueerib sümbolite tabeli;
  - leiab sümbolite rea/veeru numbrid;
  - teavitab vajadusel leksiliste vigadest.
- Leksilist analüüsi kutsutakse *skaneerimiseks* (scanning) ning vastavat analüsaatorit nimetatakse *skanneriks* (scanner).

Regulaaravaldised

- *Regulaaravaldised* üle (lõpliku) tähestiku Σ

$$E ::= \emptyset \mid \varepsilon \mid a \mid (E\ E) \mid (E \mid E) \mid (E)^\star$$

kus $a \in \Sigma$.

- Regulaaravaldis E defineerib *keele* $L(E) \subseteq \Sigma^\star$

$$\begin{array}{lcllcl} L(\emptyset) & = & \emptyset & L(E_1\ E_2) & = & \{uv \mid u \in L(E_1),\ v \in L(E_2)\} \\ L(\varepsilon) & = & \{\varepsilon\} & L(E_1 \mid E_2) & = & L(E_1) \cup L(E_2) \\ L(a) & = & \{a\} & L(E^\star) & = & \{w^i \mid w \in L(E),\ i \geq 0\} \end{array}$$

kus $w^0 = \varepsilon$ ja $w^{n+1} = w\,w^n$.

Regulaaravaldised

- Näiteid:

Regulaaravaldis	Defineeritav keel
$a \mid b$	$\{a, b\}$
$abba$	$\{abba\}$
$ab^\star a$	$\{aa, aba, abba, abbba, \ldots\}$
$(ab)^\star$	$\{\varepsilon, ab, abab, ababab, \ldots\}$

- Regulaaravaldistes esinevate sulgude vähendamiseks on operaatoritele määratud prioriteedid:
  - sulundioperaator $(\cdot)^\star$ seob kõige tugevamalt;
  - valikuoperaator $(\cdot \mid \cdot)$ seob kõige nõrgemalt.

Regulaaravaldised

- *Regulaarne kirjeldus* tähestikus Σ on reeglite hulk

$$\begin{array}{lcl} d_1 & \to & E_1 \\ d_2 & \to & E_2 \\ & \cdots & \\ d_n & \to & E_n \end{array}$$

  kus d_i on (unikaalne) nimi ja E_i on regulaaravaldis tähestikus $\Sigma \cup \{d_1, \ldots, d_{i-1}\}$.

- Lühendavaid tähistusi regulaaravaldiste esitamiseks:
  - *mittetühi sulund*: $E^+ = E\,E^\star$;
  - *optsioon*: $E? = \varepsilon \mid E$;
  - *märgiklassid*: näit. $[a,b,c] = a \mid b \mid c$ või $[a-z] = a \mid \ldots \mid z$.

Regulaaravaldised

Näiteid regulaarsetest kirjeldustest:

Identifikaatorid:

$$\begin{aligned} \text{Letter} &\rightarrow [a-z,\ A-Z] \\ \text{Digit} &\rightarrow [0-9] \\ \text{Identifier} &\rightarrow \text{Letter}\,(\text{Letter} \mid \text{Digit})^\star \end{aligned}$$

Arvkonstandid:

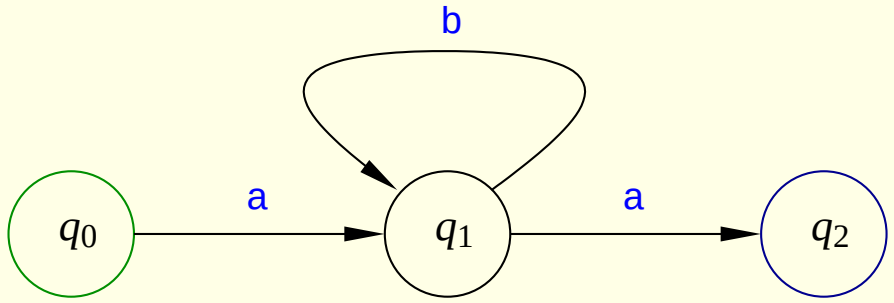
$$\begin{aligned} \text{Sign} &\rightarrow (+ \mid -)? \\ \text{Integer} &\rightarrow 0 \mid \text{Sign}\,[1-9]\,\text{Digit}^\star \\ \text{Decimal} &\rightarrow \text{Integer}\,.\,\text{Digit}^+ \\ \text{Real} &\rightarrow (\text{Integer} \mid \text{Decimal})\,E\,\text{Integer} \end{aligned}$$

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 $\delta : Q \times \Sigma \to Q$.
- Vastasel korral on lõplik automaat *mittedetermineeritud* (NFA).

Lõplikud automaadid

- Lõplike automaate esitatakse tihti *üleminekudiagrammidena*:



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

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

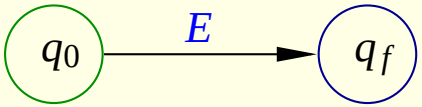
kus $\delta^\star \subseteq Q \times \Sigma^\star \times Q$ on üleminekurelatsiooni δ refleksiivne transitiivne sulund.

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

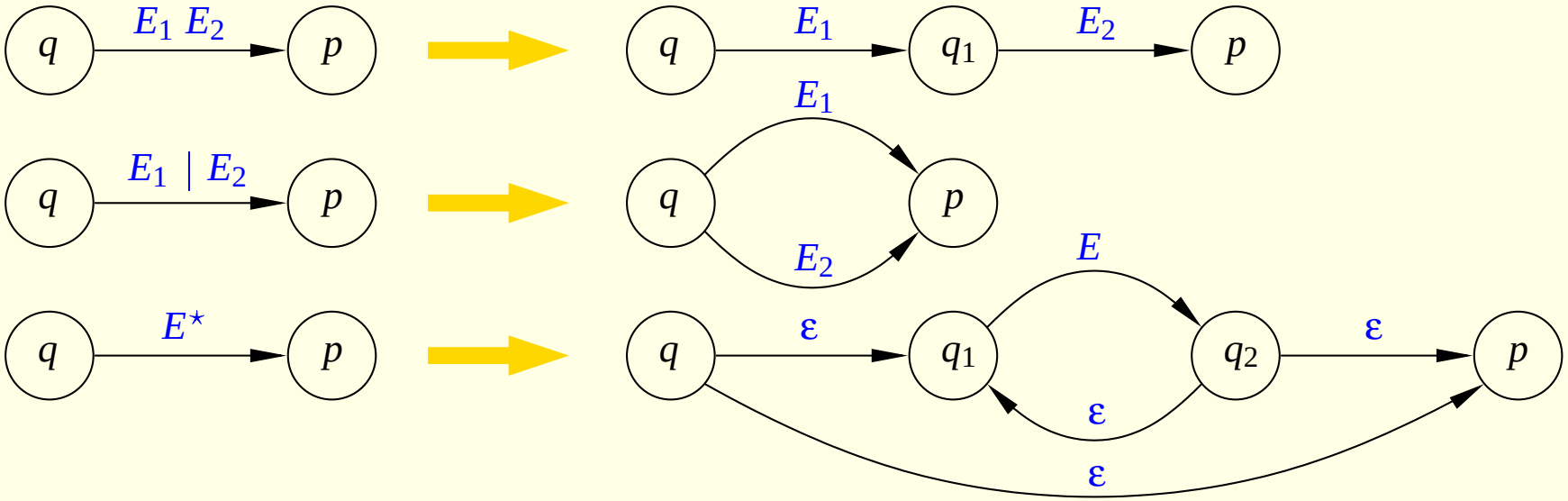
Regulaaravaldise teisendamine automaadiks

Thompsoni konstruktsioon regulaaravaldise teisendamiseks (mittedetermineeritud) lõplikuks automaadiks:

– regulaaravaldisele E seame vastavusse ”automaadi”:

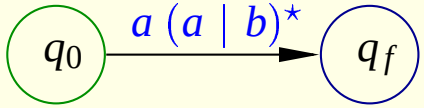


– teisendame ”automaati” järgmiste reeglite abil, kuni kõik üleminekud on kas ε või üksikud tähed:



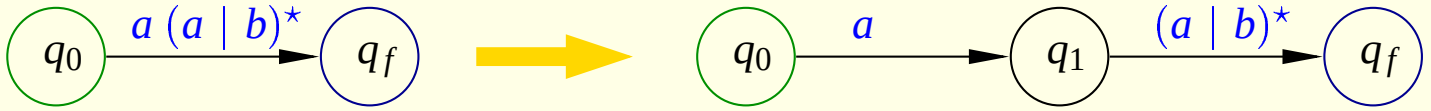
Regulaaravaldise teisendamine automaadiks

Näide:



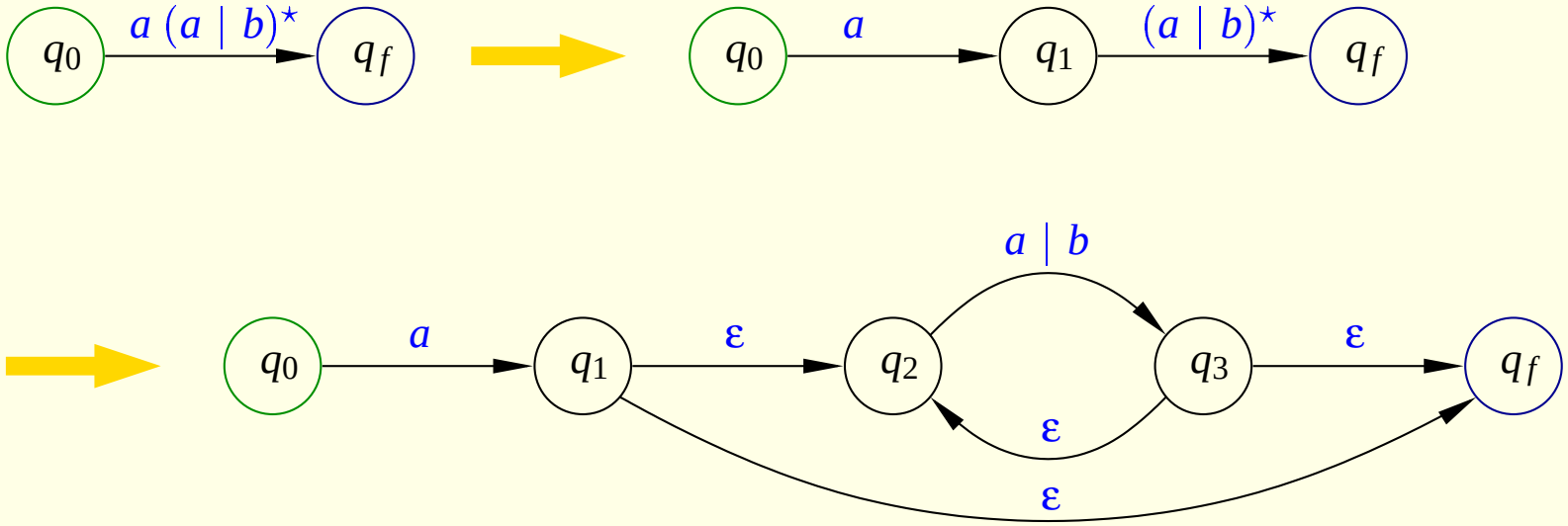
Regulaaravaldise teisendamine automaadiks

Näide:



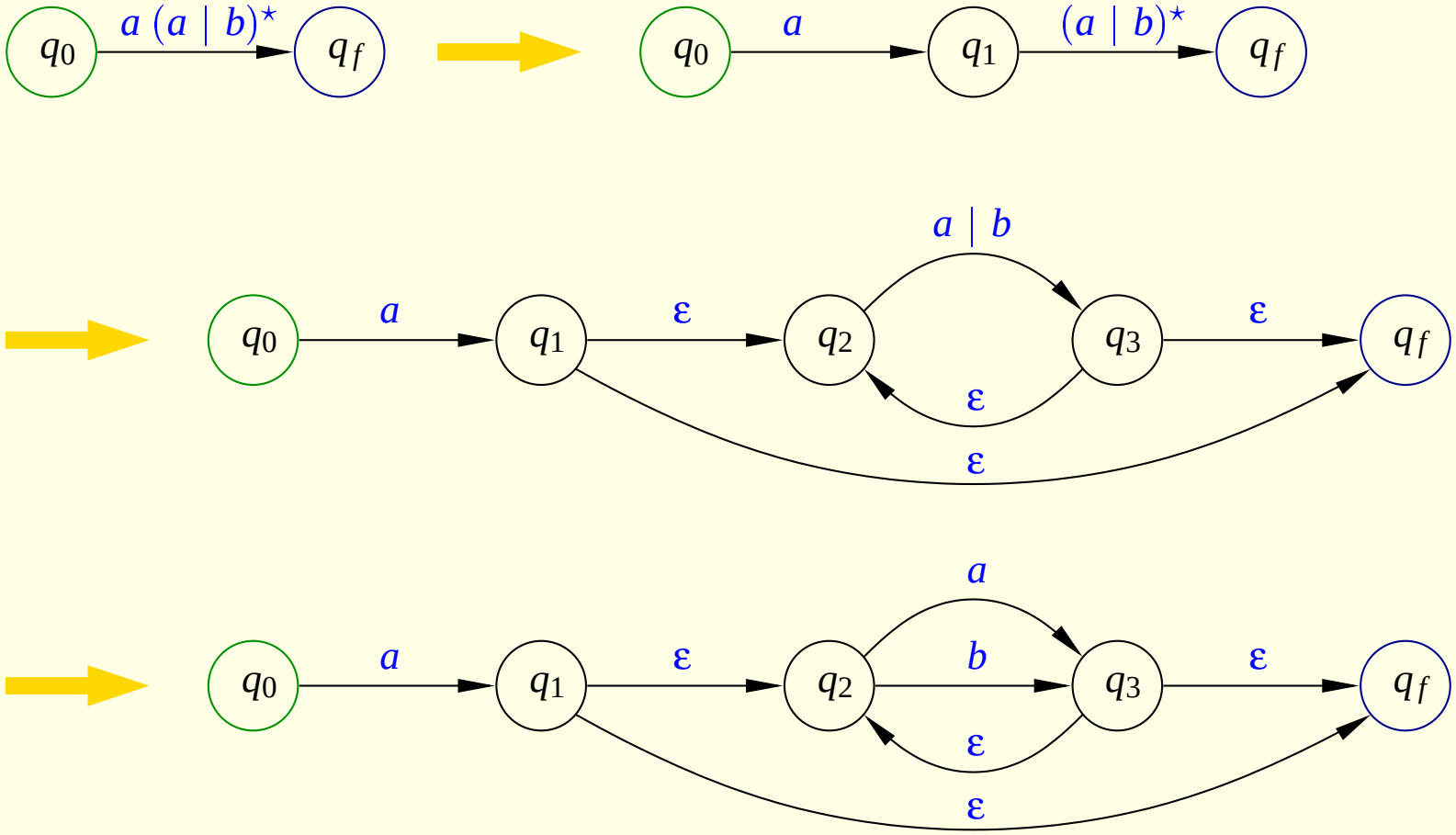
Regulaaravaldise teisendamine automaadiks

Näide:



Regulaaravaldise teisendamine automaadiks

Näide:



Determineeritud lõpliku automaadi koostamine

- Mittedeterminееritud lõpliku automaadiga $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ ekvivalentse determineeritud lõpliku automaadi $A' = \langle Q', \Sigma, \delta', q_0', F' \rangle$ konstrueerimine *osahulkade moodustamise* abil.
- Abifunktsioonid:
  - tühikäigusulundi funktsioon ε-*closure* : $2^Q \to 2^Q$

$$\varepsilon\text{-}closure(S) = \{\, p \mid q \in S,\ (q, \varepsilon, p) \in \delta^\star \,\}$$

  - ühe sammu funktsioon *move* : $2^Q \times \Sigma \to 2^Q$

$$move(S, a) = \{\, p \mid q \in S,\ (q, a, p) \in \delta \,\}$$

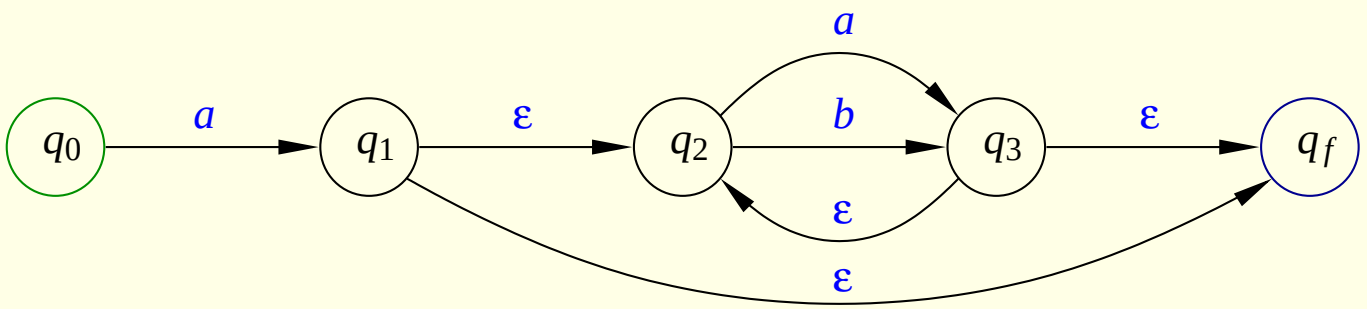
Determineeritud lõpliku automaadi koostamine

Algoritm:

$Q' := \emptyset$; $F' := \emptyset$; $\delta' := \emptyset$;
$q'_0 := \varepsilon\text{-}closure(\{q_0\})$; $U := \{q'_0\}$;
while $\exists S \in U$ **do**
    $U := U/S$; $Q' := Q' \cup \{S\}$;
    foreach $a \in \Sigma$ **do**
        $T := \varepsilon\text{-}closure(move(S,a))$;
        if $T \notin U \cup Q'$ **then** $U := U \cup \{T\}$;
        $\delta' := \delta' \cup \{(S,a) \mapsto T\}$;
    end
end
$F' := \{S \in Q' \mid S \cap F \neq \emptyset\}$;

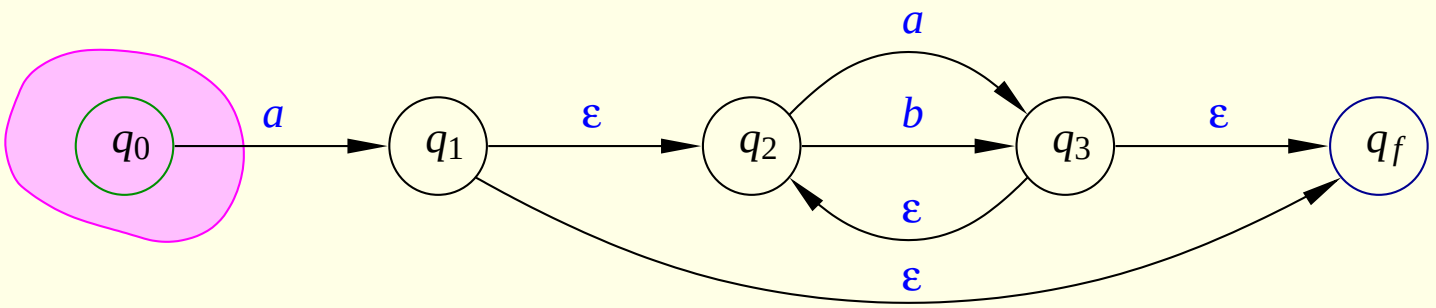
Determineeritud lõpliku automaadi koostamine

Näide:



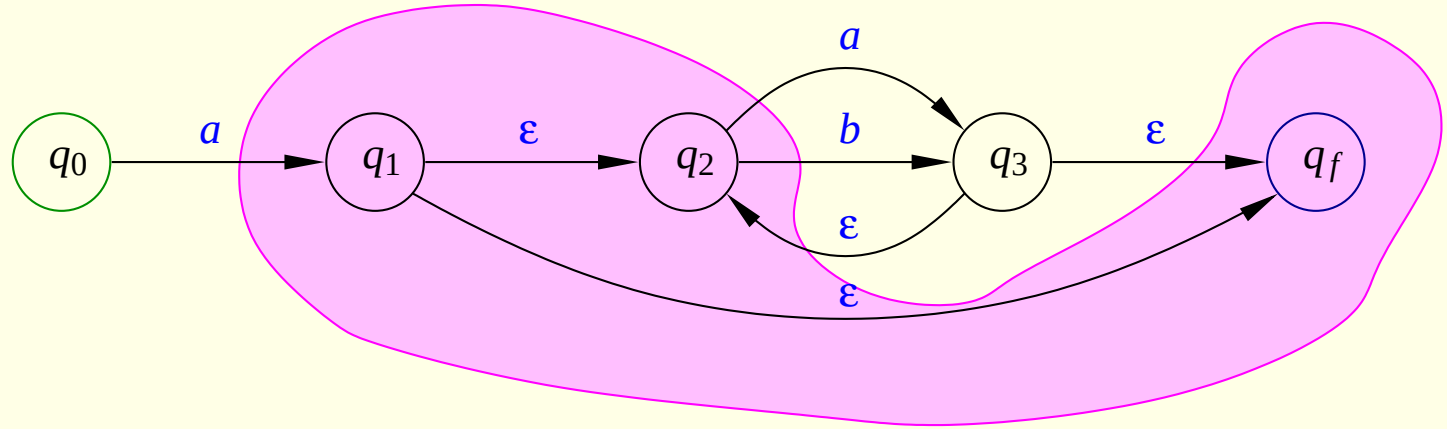
Determineeritud lõpliku automaadi koostamine

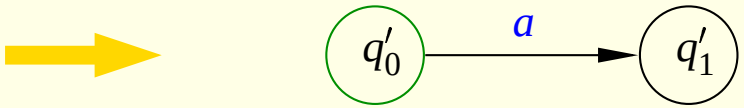
Näide:



Determineeritud lõpliku automaadi koostamine

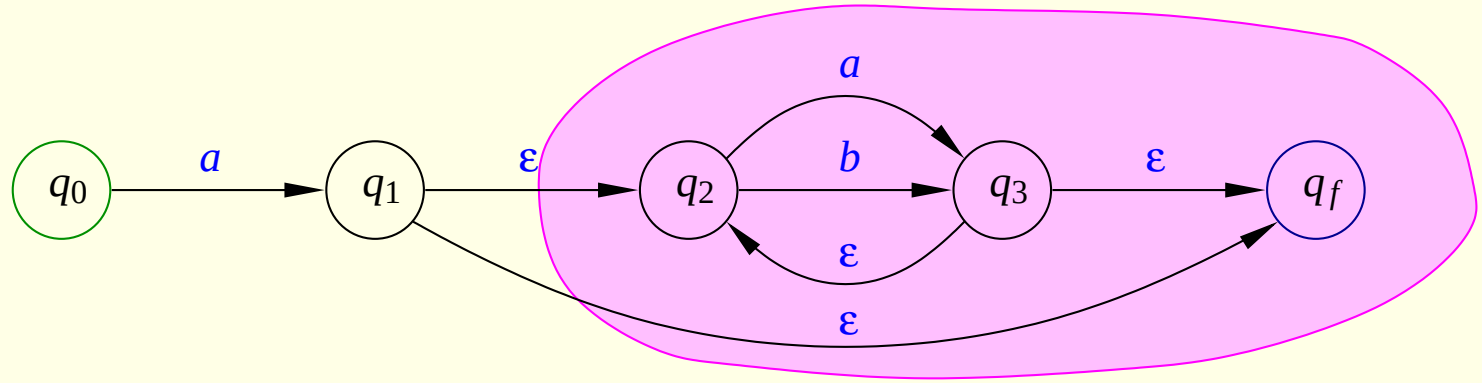
Näide:

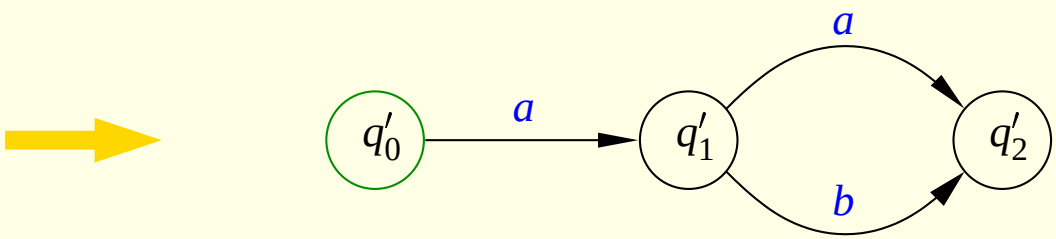




Determineeritud lõpliku automaadi koostamine

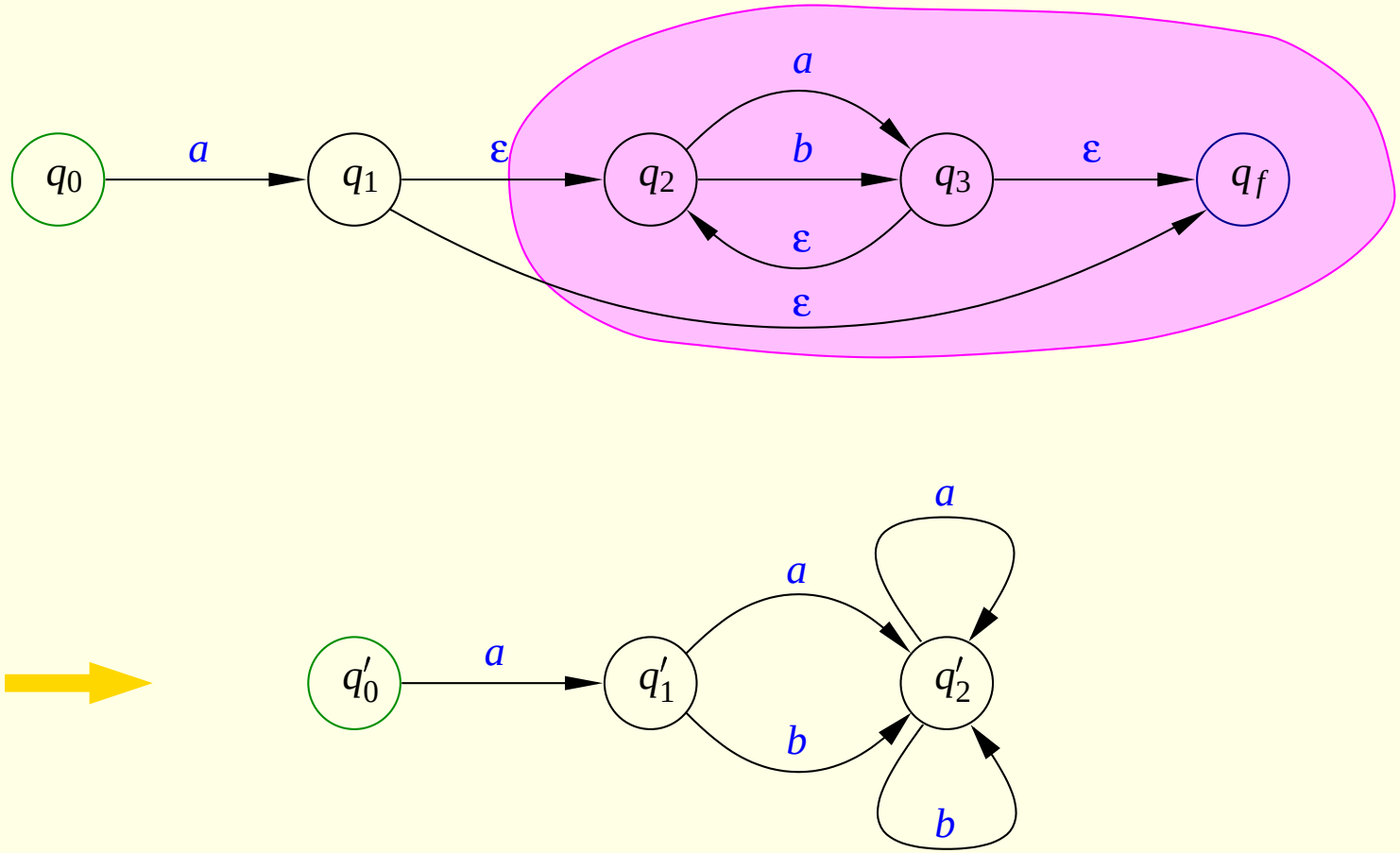
Näide:





Determineeritud lõpliku automaadi koostamine

Näide:



Determineeritud lõpliku automaadi koostamine

Näide: