

# Syntaks og semantik

## Lektion 3

15 februar 2007

Endelige automater

Regulære sprog

Lukningsegenskaber

## Forord

- 1 Endelige automater
- 2 Regulære sprog
- 3 Lukningsegenskaber ved regulære sprog

- tilstande + transitioner
- $(Q, \Sigma, \delta, q_0, F)$ : tilstande, (input)alfabetet, transitionsfunktionen, starttilstand, accepttilstande
- $\delta : Q \times \Sigma \rightarrow Q$
- **deterministisk**: givet en tilstand og et inputsymbol, kender vi næste tilstand
- **accepterer** et ord  $w \in \Sigma^*$  hvis der findes  $w_1, w_2, \dots, w_n \in \Sigma$  og  $r_0, r_1, \dots, r_n \in Q$  således at  $w = w_1 w_2 \dots w_n$  og
  - 1  $r_0 = q_0$ ,
  - 2  $r_{i+1} = \delta(r_i, w_{i+1})$  for alle  $i = 0, 1, \dots, n-1$ , og
  - 3  $r_n \in F$ .
- **genkender** sproget  $\llbracket M \rrbracket = \{w \mid M \text{ accepterer } w\}$

3 / 30

Endelige automater

Regulære sprog

Lukningsegenskaber

- **Definition**: Et sprog siges at være **regulært** hvis der findes en endelig automat der genkender det.
- **Vigtig**, hidtil ubevist **Sætning**: Et sprog er regulært hvis og kun hvis det kan beskrives ved et **regulært udtryk**.
- **Ligeså vigtig**, også hidtil ubevist **Sætning**: Der findes sprog der **ikke** er regulære.

Lad  $\Sigma$  være et alfabet og  $A_1, A_2 \subseteq \Sigma^*$ . Hvis  $A_1$  og  $A_2$  er regulære sprog, da er også følgende sprog regulære:

- $A_1 \cup A_2$  ✓
- $A_1 \cap A_2$  ✓
- $\overline{A_1} = \Sigma^* \setminus A_1$

- $A_1 \circ A_2$

- $A_1^*$

5 / 30

Lad  $\Sigma$  være et alfabet og  $A_1, A_2 \subseteq \Sigma^*$ . Hvis  $A_1$  og  $A_2$  er regulære sprog, da er også følgende sprog regulære:

- $A_1 \cup A_2$  ✓
- $A_1 \cap A_2$  ✓
- $\overline{A_1} = \Sigma^* \setminus A_1$  ✓

Lad  $M = (Q, \Sigma, \delta, q_0, F)$  være en endelig automat med  $\llbracket M \rrbracket = A_1$ . Lad  $F' = Q \setminus F$  og  $M' = (Q, \Sigma, \delta, q_0, F')$ , da er  $\llbracket M' \rrbracket = \overline{A_1}$ .

- $A_1 \circ A_2$  ?
- Problem: Flertydigheder i sammensætninger. F.eks. ved  $A_1 = \{a, ab\}, A_2 = \{ba\}$
- $A_1^*$  ?

6 / 30

## Non-determinisme

- 4 Motivation
- 5 Non-deterministiske endelige automater
- 6 At genkende sprog
- 7 Nondeterminisme er ligegyldig (?)
- 8 Lukningsegenskaber ved regulære sprog
- 9 Regulære udtryk genererer regulære sprog

7 / 30

**Ønske:** Givet endelige automater  $M_1$  og  $M_2$ , konstruér en "sammensat" automat  $M$  således at  $\llbracket M \rrbracket = \llbracket M_1 \rrbracket \circ \llbracket M_2 \rrbracket$ .



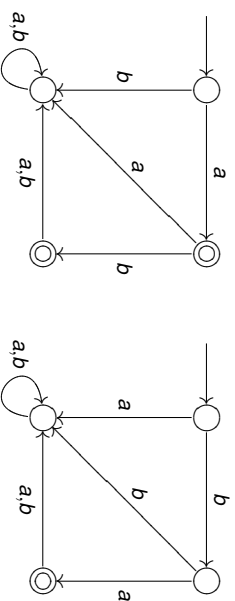
8 / 30

** nske:** Givet endelige automater  $M_1$  og  $M_2$ , konstru r en "sammensat" automat  $M$  s ledes at  $\llbracket M \rrbracket = \llbracket M_1 \rrbracket \circ \llbracket M_2 \rrbracket$ .



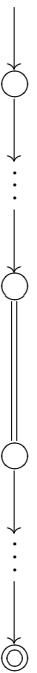
**Problem:** Hvis  $M_1$  har transitioner mellem accepttilstande, bliver transitionsfunktionen uspecificeret.

Eksempel, med  $\llbracket M_1 \rrbracket = \{a, ab\}$ ,  $\llbracket M_2 \rrbracket = \{ba\}$ :



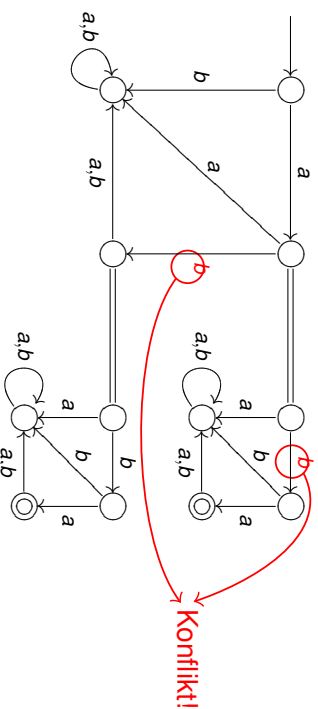
9 / 30

** nske:** Givet endelige automater  $M_1$  og  $M_2$ , konstru r en "sammensat" automat  $M$  s ledes at  $\llbracket M \rrbracket = \llbracket M_1 \rrbracket \circ \llbracket M_2 \rrbracket$ .



**Problem:** Hvis  $M_1$  har transitioner mellem accepttilstande, bliver transitionsfunktionen uspecificeret.

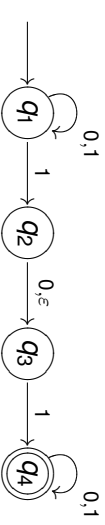
Eksempel, med  $\llbracket M_1 \rrbracket = \{a, ab\}$ ,  $\llbracket M_2 \rrbracket = \{ba\}$ :



10 / 30

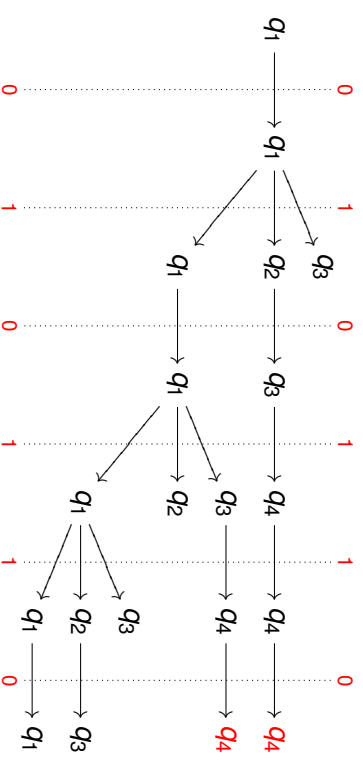
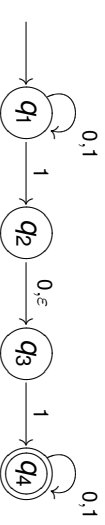
**Id :** Tillad hvad vi ikke kan undg !

- tillad at der er **flere end  n transition** med samme label fra en tilstand
- tillad at der er **ingen transitioner** med et bestemt label fra en tilstand
- tillad transitioner der **ikke l ser input-symboler**



- ved flere end  n mulige transitioner: g  til alle mulige tilstande *samtidigt*
- hvis ingen mulige transitioner: d 
- ved  $\epsilon$ -transitioner: bliv i tilstanden, men g  ogs  hen til den anden
- accept r hvis en accept-tilstand kan n s

11 / 30



12 / 30

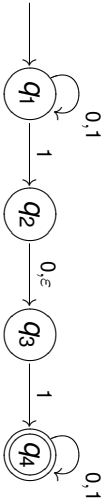
**Definition 1.37:** En **nondeterministisk endelig automat** er en 5-tupel  $(Q, \Sigma, \delta, q_0, F)$ , hvor delene er

- 1  $Q$  : en endelig mængde af tilstande
- 2  $\Sigma$  : input-alfabetet
- 3  $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$  : transitions-funktionen
- 4  $q_0 \in Q$  : starttilstanden
- 5  $F \subseteq Q$  : mængden af accepttilstande

Transitions-funktionen:

- **deterministisk** automat (fra sidste lektion):  $\delta : Q \times \Sigma \rightarrow Q$   
*input*: tilstand + tegn  
*output*: ny tilstand
- **nondeterministisk** automat:  $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$   
*input*: tilstand + tegn eller  $\epsilon$   
*output*: en **mængde** af nye tilstande
- $\mathcal{P}(Q)$ : **potensmængden** af  $Q$ ; mængden af alle delmængder af  $Q$ :  $\mathcal{P}(Q) = \{S \mid S \subseteq Q\}$

**Eksempel 1.38:**



$Q = \{q_1, q_2, q_3, q_4\} \quad \Sigma = \{0, 1\}$   
 $q_0 = q_1 \quad F = \{q_4\}$

$\delta$	0	1	$\epsilon$
$q_1$	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
$q_2$	$\{q_3\}$	$\emptyset$	$\{q_3\}$
$q_3$	$\emptyset$	$\{q_4\}$	$\emptyset$
$q_4$	$\{q_4\}$	$\{q_4\}$	$\emptyset$

**Terminologi:** Fra nu af:

- **deterministisk** endelig automat (**DFA**): dem fra sidste lektion med  $\delta : Q \times \Sigma \rightarrow Q$
- **nondeterministisk** endelig automat (**NFA**): dem med  $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$
- $\Sigma \cup \{\epsilon\}$  skrives også  $\Sigma_\epsilon$
- Husk:  $\mathcal{P}(Q) = \text{potensmængden ("power set") af } Q$ :  $\mathcal{P}(Q) = \{S \mid S \subseteq Q\}$
- enhver DFA er også en NFA
- og enhver NFA kan laves om til en DFA! (bevis kommer lige om lidt)

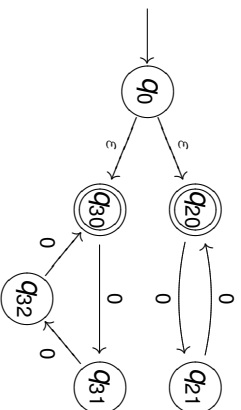
**Definition:** Lad  $M = (Q, \Sigma, \delta, q_0, F)$  være en endelig automat og  $w \in \Sigma^*$ . Da siges  $M$  at **acceptere**  $w$  hvis der findes  $m \in \mathbb{N}$  og  $y_1, y_2, \dots, y_m \in \Sigma_\epsilon$  (!) og  $r_0, r_1, \dots, r_m \in Q$  således at  $w = y_1 y_2 \dots y_m$  og

- 1  $r_0 = q_0$ ,
- 2  $r_{i+1} \in \delta(r_i, y_{i+1})$  for alle  $i = 0, 1, \dots, m - 1$ , og
- 3  $r_m \in F$ .

Sproget som **genkendes** af  $M$  er

$$\llbracket M \rrbracket = L(M) = \{w \mid M \text{ accepterer } w\}$$

### Eksempel 1.33:



- $w = 00 = \epsilon 00$ :  
 $q_0 \rightarrow \{q_{20}, q_{30}\} \rightarrow \{q_{21}, q_{31}\} \rightarrow \{q_{20}, q_{32}\} \Rightarrow$  **Jep.**
- $w = 000 = \epsilon 000$ :  
 $q_0 \rightarrow \{q_{20}, q_{30}\} \rightarrow \{q_{21}, q_{31}\} \rightarrow \{q_{20}, q_{32}\} \rightarrow \{q_{21}, q_{30}\} \Rightarrow$  **Jep.**
- $w = 0000 = \epsilon 0000$ :  
 $q_0 \rightarrow q_{20} \rightarrow q_{21} \rightarrow q_{20} \rightarrow q_{21} \rightarrow q_{20} \Rightarrow$  **Jep.**  
 (Nok med * t* accepterende run.)
- $w = 00000 = \epsilon 00000$ :  
 $q_0 \rightarrow \{q_{20}, q_{30}\} \rightarrow \{q_{21}, q_{31}\} \rightarrow \{q_{20}, q_{32}\} \rightarrow \{q_{21}, q_{30}\} \rightarrow \{q_{20}, q_{31}\} \rightarrow \{q_{21}, q_{32}\} \Rightarrow$  **Nej.**  
 (Alle runs er ikke-accepterende.)

17 / 30

**Vigtig s tning 1.39:** Til enhver NFA  $N$  findes der en DFA  $M$  med  $\llbracket M \rrbracket = \llbracket N \rrbracket$ .

Eller: Til enhver NFA findes der en DFA der genkender samme sprog.

Eller: Til enhver NFA findes der en * kvivalent* DFA.

(Hvis vi siger at to maskiner er  kvivalente hvis de genkender samme sprog.)

18 / 30

**Vigtig s tning 1.39:** Til enhver NFA  $N$  findes der en DFA  $M$  med  $\llbracket M \rrbracket = \llbracket N \rrbracket$ .

**Beviside:** N r vi ser efter om vores NFA  $N$  accepterer et ord, skal vi holde styr p  **m ngder af tilstande**.

Dvs. vi skal konstruere en DFA  $M$  der holder styr p  m ngder af tilstande i  $N$ .  $\Rightarrow$  *Tilstandene* i  $M$  afspejler *m ngder af tilstande* i  $N$ .

19 / 30

**Vigtig s tning 1.39:** Til enhver NFA  $N$  findes der en DFA  $M$  med  $\llbracket M \rrbracket = \llbracket N \rrbracket$ .

**Bevis:** Skriv  $N = (Q, \Sigma, \delta, q_0, F)$ . Vi konstruerer en DFA

- $Q' = \mathcal{P}(Q)$  tilstande i  $M$  er m ngder af tilstande i  $N$
- $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$  vi accepterer hvis  n af  $N$ s tilstande er accepterende
- $q'_0 = \{q_0\}$   $M$  starter i  $N$ s starttilstand

Transitionsfunktionen: f rste fors g:  $\delta'(R, a) = \{\delta(r, a) \mid r \in R\}$

Virker ikke helt: mangler at tage  $\epsilon$ -transitioner:  $\delta'(R, a)$  skal v re den m ngde af tilstande vi kan n  fra tilstande i  $R$  ved at l se et  $a$ , *plus* alle de tilstande vi s  kan n  via  $\epsilon$ -transitioner!

**Hovsa!** der er ogs  problemer med  $q'_0$ :  $q'_0$  skal best  af alle de tilstande i  $N$  der kan n s fra  $q_0$  via  $\epsilon$ -transitioner.

20 / 30

**Vigtig s tning 1.39:** Til enhver NFA  $N$  findes der en DFA  $M$  med  $\llbracket M \rrbracket = \llbracket N \rrbracket$ .

**Bevis:** Skriv  $N = (Q, \Sigma, \delta, q_0, F)$ . Vi konstruerer en DFA  $M = (Q', \Sigma, \delta', q'_0, F')$ :

- $Q' = \mathcal{P}(Q)$
- $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$

For enhver delm nge  $R \subseteq Q$  lad

$$E(R) = \{q \in Q \mid q \text{ kan n s fra } R \text{ ved 0 eller flere } \varepsilon\text{-transitioner}\}$$

–  **$\varepsilon$ -aflukningen** af  $R$ .

- $q'_0 = E(\{q_0\})$
- $\delta'(R, a) = \{q \in Q \mid q \in E(\{\delta(r, a)\}) \text{ for et } r \in R\}$   
 $= \bigcup_{r \in R} E(\{\delta(r, a)\})$

For at vise at  $\llbracket M \rrbracket = \llbracket N \rrbracket$ , skal vi vise at

- ethvert  $w \in \llbracket M \rrbracket$  accepteres af  $N$ , og at
- ethvert  $w \in \llbracket N \rrbracket$  accepteres af  $N$ .

21 / 30

**S tning 1.45:** (havde vi allerede, men nu med nyt bevis!)

Hvis  $A_1$  og  $A_2$  er regul re sprog over et alfabet  $\Sigma$ , da er ogs   $A_1 \cup A_2$  et regul rt sprog over  $\Sigma$ .

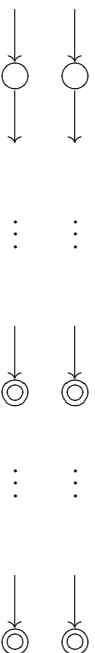
**Bevis:** Lad  $N_1$  og  $N_2$  v re NFAs med  $\llbracket N_1 \rrbracket = A_1$  og  $\llbracket N_2 \rrbracket = A_2$ .

Skriv  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ ,  $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ , og konstru r en ny NFA  $N = (Q, \Sigma, \delta, q_0, F)$  ved

- $Q = Q_1 \cup Q_2 \cup \{q_0\}$ , (en ekstra ny starttilstand)
- $F = F_1 \cup F_2$  og

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \\ \delta_2(q, a) & \text{hvis } q \in Q_2 \\ \{q_1, q_2\} & \text{hvis } q = q_0 \text{ og } a = \varepsilon \\ \emptyset & \text{hvis } q = q_0 \text{ og } a \neq \varepsilon \end{cases}$$

Da er  $\llbracket N \rrbracket = A_1 \cup A_2$ .



22 / 30

**S tning 1.45:** (havde vi allerede, men nu med nyt bevis!)

Hvis  $A_1$  og  $A_2$  er regul re sprog over et alfabet  $\Sigma$ , da er ogs   $A_1 \cup A_2$  et regul rt sprog over  $\Sigma$ .

**Bevis:** Lad  $N_1$  og  $N_2$  v re NFAs med  $\llbracket N_1 \rrbracket = A_1$  og  $\llbracket N_2 \rrbracket = A_2$ .

Skriv  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ ,  $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ , og konstru r en ny NFA  $N = (Q, \Sigma, \delta, q_0, F)$  ved

- $Q = Q_1 \cup Q_2 \cup \{q_0\}$ , (en ekstra ny starttilstand)
- $F = F_1 \cup F_2$  og

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \\ \delta_2(q, a) & \text{hvis } q \in Q_2 \\ \{q_1, q_2\} & \text{hvis } q = q_0 \text{ og } a = \varepsilon \\ \emptyset & \text{hvis } q = q_0 \text{ og } a \neq \varepsilon \end{cases}$$

Da er  $\llbracket N \rrbracket = A_1 \cup A_2$ .



**Intuitivt!**

23 / 30

**S tning 1.47:** Hvis  $A_1$  og  $A_2$  er regul re sprog over et alfabet  $\Sigma$ , da er ogs   $A_1 \circ A_2$  et regul rt sprog over  $\Sigma$ .

**Bevis:** Lad  $N_1$  og  $N_2$  v re NFAs med  $\llbracket N_1 \rrbracket = A_1$  og  $\llbracket N_2 \rrbracket = A_2$ .

Skriv  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ ,  $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ , og konstru r en ny NFA  $N = (Q, \Sigma, \delta, q_1, F_2)$  ved

- $Q = Q_1 \cup Q_2$
- (starttilstanden er  $q_1$ , accepttilstandene er  $F_2$ )

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \text{ og } q \notin F_1 \\ \delta_1(q, a) & \text{hvis } q \in F_1 \text{ og } a \neq \varepsilon \\ \delta_1(q, a) \cup \{q_2\} & \text{hvis } q \in F_1 \text{ og } a = \varepsilon \\ \delta_2(q, a) & \text{hvis } q \in Q_2 \end{cases}$$

Da er  $\llbracket N \rrbracket = A_1 \circ A_2$ .



24 / 30

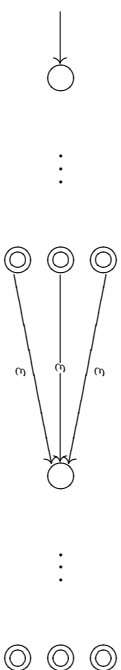
**Sætning 1.47:** Hvis  $A_1$  og  $A_2$  er regulære sprog over et alfabet  $\Sigma$ , da er også  $A_1 \circ A_2$  et regulært sprog over  $\Sigma$ .

**Bevis:** Lad  $N_1$  og  $N_2$  være NFAs med  $\llbracket N_1 \rrbracket = A_1$  og  $\llbracket N_2 \rrbracket = A_2$ . Skriv  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ ,  $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ , og konstruér en ny NFA  $N = (Q, \Sigma, \delta, q_1, F_2)$  ved

- $Q = Q_1 \cup Q_2$
- (starttilstanden er  $q_1$ , acceptitilstandene er  $F_2$ )

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \text{ og } q \notin F_1 \\ \delta_1(q, a) & \text{hvis } q \in F_1 \text{ og } a \neq \epsilon \\ \delta_1(q, a) \cup \{q_2\} & \text{hvis } q \in F_1 \text{ og } a = \epsilon \\ \delta_2(q, a) & \text{hvis } q \in Q_2 \end{cases}$$

Da er  $\llbracket N \rrbracket = A_1 \circ A_2$ .

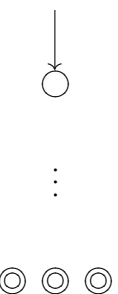


26 / 30

**Sætning 1.49:** Hvis  $A$  er et regulært sprog over et alfabet  $\Sigma$ , da er også  $A^*$  et regulært sprog over  $\Sigma$ .

**Bevis:** Lad  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$  være en NFA med  $\llbracket N_1 \rrbracket = A$ . Konstruér en ny NFA  $N = (Q, \Sigma, \delta, q_0, F)$  ved

- $Q = Q_1 \cup \{q_0\}$ ,
  - $F = F_1 \cup \{q_0\}$  og
- $$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \text{ og } q \notin F_1 \\ \delta_1(q, a) & \text{hvis } q \in F_1 \text{ og } a \neq \epsilon \\ \delta_1(q, a) \cup \{q_1\} & \text{hvis } q \in F_1 \text{ og } a = \epsilon \\ \emptyset & \text{hvis } q = q_0 \text{ og } a = \epsilon \\ & \text{hvis } q = q_0 \text{ og } a \neq \epsilon \end{cases}$$



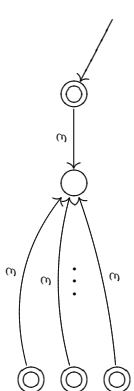
26 / 30

**Sætning 1.49:** Hvis  $A$  er et regulært sprog over et alfabet  $\Sigma$ , da er også  $A^*$  et regulært sprog over  $\Sigma$ .

**Bevis:** Lad  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$  være en NFA med  $\llbracket N_1 \rrbracket = A$ . Konstruér en ny NFA  $N = (Q, \Sigma, \delta, q_0, F)$  ved

- $Q = Q_1 \cup \{q_0\}$ ,
- $F = F_1 \cup \{q_0\}$  og

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{hvis } q \in Q_1 \text{ og } q \notin F_1 \\ \delta_1(q, a) & \text{hvis } q \in F_1 \text{ og } a \neq \epsilon \\ \delta_1(q, a) \cup \{q_1\} & \text{hvis } q \in F_1 \text{ og } a = \epsilon \\ \emptyset & \text{hvis } q = q_0 \text{ og } a = \epsilon \\ & \text{hvis } q = q_0 \text{ og } a \neq \epsilon \end{cases}$$



27 / 30

**Sætning 1.54:** Et sprog er regulært hvis og kun hvis det kan beskrives ved et regulært udtryk.

*Eller:* Givet et alfabet  $\Sigma$  og  $L \subseteq \Sigma^*$ , da er  $L$  et regulært sprog hvis og kun hvis der findes et regulært udtryk  $R$  over  $\Sigma$  således at  $L = \llbracket R \rrbracket$ .

(Bevis til første halvdel nu, til anden halvdel næste gang.)

28 / 30

**Lemma 1.55:** Givet et alfabet  $\Sigma$  og et sprog  $L \subseteq \Sigma^*$ . Hvis der findes et regul ert udtryk  $R$  over  $\Sigma$  med  $\llbracket R \rrbracket = L$ , da er  $L$  regul ert.

**Bevis** (ved **strukturel induktion**):

- 1 Hvis  $L = \llbracket a \rrbracket$  for et  $a \in \Sigma$ : Lad  $M = \text{---} \circ \xrightarrow{a} \circ \text{---}$ , da er  $\llbracket M \rrbracket = \{a\} = L$ .
- 2 Hvis  $L = \llbracket \varepsilon \rrbracket$ : Lad  $M = \text{---} \circ \xrightarrow{\varepsilon} \circ \text{---}$ , da er  $\llbracket M \rrbracket = \{\varepsilon\} = L$ .
- 3 Hvis  $L = \llbracket \emptyset \rrbracket$ : Lad  $M = \text{---} \circ \text{---}$ , da er  $\llbracket M \rrbracket = \emptyset = L$ .
- 4 Hvis  $L = \llbracket R_1 \cup R_2 \rrbracket$ : Ved **induktionsantagelsen** har vi NFAs  $M_1$  og  $M_2$  s ledes at  $\llbracket M_1 \rrbracket = \llbracket R_1 \rrbracket$  og  $\llbracket M_2 \rrbracket = \llbracket R_2 \rrbracket$ . Derfor er  $\llbracket R_1 \rrbracket$  og  $\llbracket R_2 \rrbracket$  regul ere sprog, med s tning 1.45 alts  ogs   $\llbracket R_1 \rrbracket \cup \llbracket R_2 \rrbracket = \llbracket R_1 \cup R_2 \rrbracket = L$ .
- 5 Hvis  $L = \llbracket R_1 \cup R_2 \rrbracket$  eller  $L = \llbracket R_1^* \rrbracket$ : Analogt til tilf lde 4, bortset fra at s tning 1.47 hhv. 1.49 skal benyttes.

**Eksempel 1.56:** Konvert r  $(ab \cup a)^*$  til en NFA.

