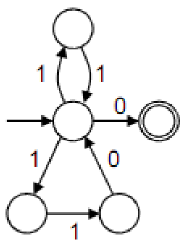
Nondeterministiske automater

- **Nondeterministiske automater** M = $(Q, \Sigma, q_0, A, \delta)$
  - Q er en endelig mængde af tilstande
  - Σ er et alfabet
  - $q_0 \in Q$ er en starttilstand
  - $A \subseteq Q$ er accepttilstande
  - $\delta: Q \times \Sigma \to 2^Q$ er en transitionsfunktion

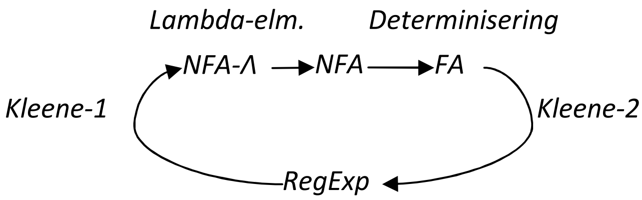
$$\delta^*(q, x) = \begin{cases} \{q\} & \text{hvis } x = \Lambda \\ \bigcup_{r \in \delta^*(q, y)} \delta(r, a) & \text{hvis } x = ya \text{ hvor } y \in \Sigma^* \text{ og } a \in \Sigma \end{cases}$$

$x \in \Sigma^*$ *accepteres* af M hvis og kun hvis $\delta^*(q_0, x) \cap A \neq \varnothing$

  - Evt. eksempel



  - Gætter vej til accept-tilstand.
- **Determinisering**: NFA → FA



  - Nondeterminismen skal fjernes.

Determinisering, theorem 4.1, induktionsbevis

$NFA: M = (Q, \Sigma, q_0, A, \delta)$

$FA: M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$

Enhver tilstand i FA'en er en mængde af tilstande i NFA'en: $Q_1 = 2^Q$

$q_1 = \{q_0\}$

$A_1 = \{q \in Q_1 \mid q \cap A \neq \varnothing\}$

$q \in Q_1 \mid a \in \Sigma : \delta_1(q, a) = \bigcup_{r \in q} \delta(r, a)$

Da Q_1 er en mængde, af mængder af tilstande, vil q bestå af en mængde af tilstande.

Da $\delta_1^*(q_1, x) = \delta^*(q_0, x)$ accepterer M_1 det samme sprog som *M*. Bevis for korrekthed:

Basis:

$x = \Lambda$

$$\delta_1^*(q_1, \Lambda) \overset{def\delta^*, FA}{=} q_1 \overset{NFA \to FA}{=} \{q_0\} \overset{def\delta^* NFA}{=} \delta^*(q_0, \Lambda)$$

I.H:

For en streng x gælder det at $\delta_1^*(q_1, x) = \delta^*(q_0, x)$

Induktionsskridt:

Bevis at $\delta_1^*(q_1, xa) = \delta^*(q_0, xa)$

$$\delta_1^*(q_1, xa) \overset{def\delta^*, FA}{=} \delta_1(\delta_1^*(q_1, x), a) \overset{I.H}{=} \delta_1(\delta^*(q_0, x), a) \overset{def.\delta_1}{=} \bigcup_{r \in \delta^*(q_0, x)} \delta(r, a) \overset{def.\delta^*, NFA}{=} \delta^*(q_0, xa)$$

- **RegExp → FA (kleene)**
  - Λ – eliminering (hvis der er tid): NFA- Λ → NFA
  - Så kan Kleene-1 bruges til at finde en FA der svarer til et RegExp