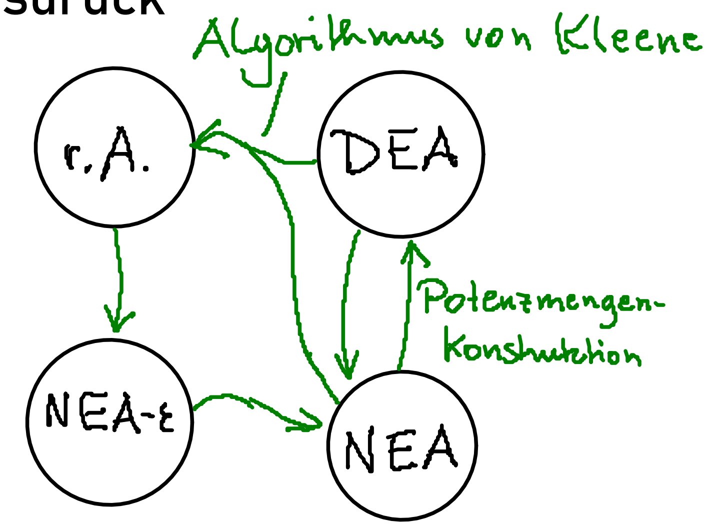
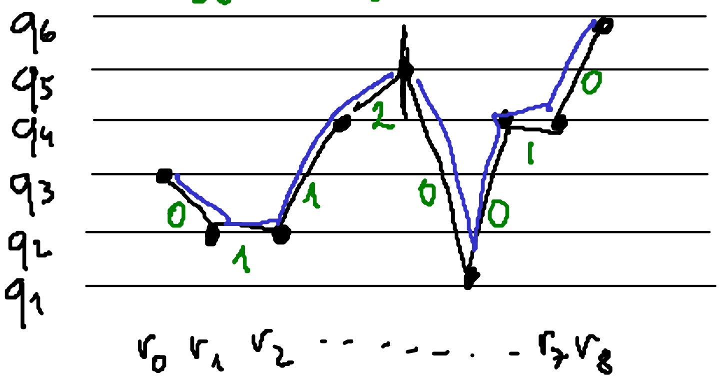
Endlicher Automat ⇝ regulärer Ausdruck



$Q = \{q_1, q_2, \ldots, q_n\}$

$\notin R_{36}^{(4)}$ $\in R_{36}^{(5)}$ ✓



für einen NEA $(Q, \Sigma, q_0, \delta, F)$

Def. Ein Berechnungsweg für ein Wort $x_1 x_2 \ldots x_m \in \Sigma^*$ ist eine abwechselnde Folge $(r_0, x_1, r_1, x_2, r_2, \ldots, x_m, r_m)$ von Zuständen $r_i \in Q$ und Eingabesymbolen $x_i \in \Sigma$, mit

- ~~$r_0 = q_0$~~
- $(r_{i-1}, x_i, r_i) \in \delta$ für $i = 1, \ldots, m$

Ein akzeptierender Berechnungsweg ist einer mit $r_m \in F$.

Wir betrachten Berechnungswege von einem beliebigen Anfangszustand $r_0 \in Q$ zu einem beliebigen Endzustand $r_m \in Q$

Beobachtung:

Wenn B ein Berechnungsweg von q nach q' für ein Wort $x \in \Sigma^*$ ist,
und B' ein Berechnungsweg von q' nach q'' für ein Wort $x' \in \Sigma^*$ ist,
dann ist $B \circ B'$ ein Berechnungsweg von q nach q'' für das Wort $xx' \in \Sigma^*$.

$$R_{ij}^{(k)} = \{ \text{ Wörter } x = x_1 x_2 \dots x_m \in \Sigma^* \mid$$

Es gibt einen Berechnungsweg $(\overset{q_i}{r_0}, x_1, \underbrace{r_1, \dots, r_{m-1}}, x_m, \overset{q_j}{r_m})$

von q_i nach q_j für x, der als Zwischenzustände

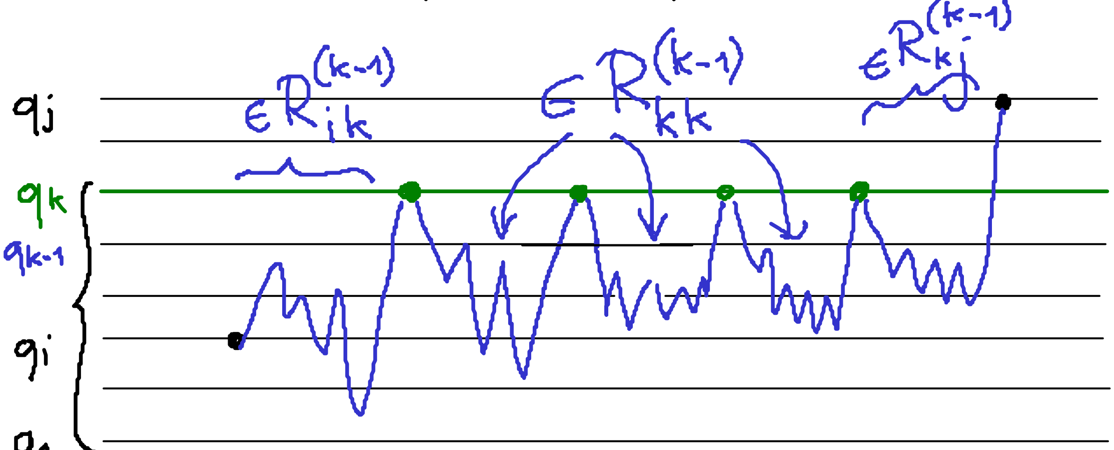
nur Zustände $\in \{q_1, q_2, \dots, q_k\}$ verwendet.

$$(i,j = 1,2,\dots,n \;;\; k = 0,1,\dots,n)$$

- Berechnungswege der Länge $m=0$ und $m=1$ haben keine Zwischenzustände. (Die Zusatzbedingung ist immer erfüllt.)

Induktiver Aufbau der Sprachen $R_{ij}^{(k)}$ für $k=0,1,2,\dots,n$

$k=0$ $$R_{ij}^{(0)} = \begin{cases} \{x \in \Sigma \mid (q_i, x, q_j) \in \delta\}, & i \neq j \\ \{x \in \Sigma \mid (q_i, x, q_j) \in \delta\} \cup \{\varepsilon\} & i = j \end{cases}$$



$k-1 \to k$

$$\boxed{R_{ij}^{(k)} = R_{ij}^{(k-1)} \cup R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}}$$

$i,j = 1,2,\dots,n$

Für $i=k$ bzw. $j=k$ sind Vereinfachungen möglich:

$$R_{kj}^{(k)} = \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)} \qquad (i=k,\ j\neq k)$$

$$R_{ik}^{(k)} = R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* \qquad (i\neq k,\ j=k)$$

$$R_{kk}^{(k)} = \left(R_{kk}^{(k-1)}\right)^* \qquad (i=j=k)$$

- Jede Sprache $R_{ij}^{(k)}$ kann durch einen regulären Ausdruck beschrieben werden.

Beweis: vollständige Induktion nach k.

$k=0$: $R_{ij}^{(0)}$ sind endliche Sprachen: $R_{ij}^{(0)} \subseteq \Sigma \cup \{\varepsilon\}$

$k-1 \to k$: $R_{ij}^{(k)}$ wird aus $R_{..}^{(k-1)}$ mit den Operationen $\cup, \bullet, {}^*$ gebildet. $\square$.

$$L(A) = \bigcup_{j\in F} R_{q_0,j}^{(n)} \quad \text{... regulärer Ausdruck für } L(A).$$

endliche Vereinigung.

$\square$

Zum Algorithmus von Kleene verwandte Algorithmen

<u>Dynamische Programmierung</u> (ein allgemeines Entwurfsprinzip für Algorithmen)

> Systematisches Lösen von <u>Teilproblemen</u>
> unter Rückgriff auf (zuvor gelöste)
> „kleinere" Teilprobleme.

- Eigentlich ist der Algorithmus von Kleene ein Algorithmus über Wege in Graphen.
  - Algorithmus von Warshall (Erreichbarkeit, transitive Hülle einer Relation)
  - Algorithmus von Floyd (kürzeste Wege)
  - Gauß-Jordan-Elimination zur Matrizeninversion

$$O(n^3)$$