

Definition 2.9 (Erweiterte Übergangsfunktion eines NFA). Sei $\mathcal{M} = (Q, \Sigma, \delta, Q_0, F)$ ein NFA. In Analogie zu Definition 2.4 (Seite 21) erweitern wir die Übergangsfunktion eines NFA zu einer (ebenfalls mit δ bezeichneten) Abbildung

$$\delta : 2^Q \times \Sigma^* \rightarrow 2^Q.$$

Intuitiv ist $\delta(P, x)$ die Menge aller Zustände, die man mit dem Wort x von einem Zustand $p \in P$ erreichen kann. Die formale Definition von $\delta(P, x)$ erfolgt durch Induktion nach der Länge von x . Sei $P \subseteq Q$ und $a \in \Sigma, x \in \Sigma^+$. Dann ist

$$\delta(P, \varepsilon) \stackrel{\text{def}}{=} P \text{ und } \delta(P, ax) \stackrel{\text{def}}{=} \bigcup_{p \in P} \delta(\delta(p, a), x).$$

Ist $q \in Q$ und $w \in \Sigma^*$, so ist $\delta(q, w) \stackrel{\text{def}}{=} \delta(\{q\}, w)$. ■

Definition 2.10 (Akzeptierte Sprache eines NFA). Sei $\mathcal{M} = (Q, \Sigma, \delta, Q_0, F)$ ein NFA. Die von \mathcal{M} akzeptierte Sprache $\mathcal{L}(\mathcal{M})$ ist wie folgt definiert:

$$\mathcal{L}(\mathcal{M}) \stackrel{\text{def}}{=} \{w \in \Sigma^* : \delta(Q_0, w) \cap F \neq \emptyset\}.$$

■

Beispiel 2.11 (NFA). Wir betrachten den NFA \mathcal{M} mit dem Alphabet $\Sigma = \{0, 1\}$ in Abbildung 10. Dieser hat zwei Startzustände q_0 und q_1 zwischen denen eine nichtdeterministische Wahl stattfindet. Da $\delta(q_0, 0) = \{q_0, q_1\}$ zweielementig ist, verhält sich \mathcal{M} im Startzustand q_0 für das Eingabezeichen 0 nichtdeterministisch und entscheidet sich willkürlich für einen der beiden Folgezustände q_0 oder q_1 .

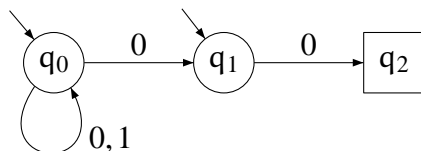


Figure 10: NFA \mathcal{M}

Da $\delta(q_0, 1) = \{q_0\}$ einelementig ist, ist das Verhalten von \mathcal{M} in Zustand q_0 bei Lesen einer 1 deterministisch. Weiter ist $\delta(q_1, 1) = \emptyset$. Liest \mathcal{M} also in Zustand q_1 das Eingabezeichen 1, so gibt es keinen entsprechenden Übergang und \mathcal{M} verwirft. Analoges gilt für den Endzustand q_2 , in dem weder eine 0 noch eine 1 gelesen werden kann, da $\delta(q_2, 0) = \delta(q_2, 1) = \emptyset$.

Für die erweiterte Übergangsfunktion gilt:

$$\begin{aligned} \delta(q_0, 0100) &= \delta(\delta(q_0, 0), 100) &= \delta(\{q_0, q_1\}, 100) \\ &= \delta(q_0, 100) \cup \delta(q_1, 100) &= \delta(\delta(q_0, 1), 00) \\ &= \delta(q_0, 00) &= \delta(\delta(q_0, 0), 0) \\ &= \delta(\{q_0, q_1\}, 0) &= \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1, q_2\} \end{aligned}$$

und somit $\delta(Q_0, 0100) = \delta(\{q_0, q_1\}, 0100) = \{q_0, q_1, q_2\}$. Wegen $\delta(Q_0, 0100) \cap F = \{q_2\} \neq \emptyset$ gilt $0100 \in \mathcal{L}(\mathcal{M})$. ■

In Analogie zur Definition der Läufe in einem DFA (Seite 20) können wir $\mathcal{L}(\mathcal{M})$ über die Läufe charakterisieren. Sei $w = a_1 a_2 \dots a_n \in \Sigma^*$. Ein Lauf für w in \mathcal{M} ist eine Zustandsfolge $q_0 q_1 \dots q_m$ mit

$$q_0 \in Q_0 \text{ und } q_{i+1} \in \delta(q_i, a_{i+1}), i = 1, \dots, m-1.$$

Weiter fordern wir, dass entweder $m = n$ oder $m < n$ und $\delta(q_m, a_{m+1}) = \emptyset$. Ist $m = n$ und $q_n \in F$, dann nennen wir $q_0 q_1 \dots q_n$ einen akzeptierenden Lauf für w in \mathcal{M} . Ist $m < n$ und $\delta(q_m, a_{m+1}) = \emptyset$ oder $m = n$ und $q_n \notin F$, dann wird $q_0 q_1 \dots q_m$ verwerfender Lauf für w in \mathcal{M} genannt. Die von \mathcal{M} akzeptierte Sprache ist also genau die Menge aller Wörter $w \in \Sigma^*$, für die es wenigstens einen akzeptierenden Lauf gibt:

$$\begin{aligned} \mathcal{L}(\mathcal{M}) &= \{ w \in \Sigma^* : \text{es gibt einen akzeptierenden Lauf für } w \text{ in } \mathcal{M} \} \\ &= \{ w \in \Sigma^* : \delta(Q_0, w) \cap F \neq \emptyset \} \end{aligned}$$

Im Gegensatz zu DFA kann ein Wort w viele Läufe in einem NFA haben. Für die Akzeptanz wird lediglich gefordert, dass einer der Läufe für w akzeptierend ist.

Beispiel 2.12 (Läufe und akzeptierte Sprache eines NFA). Wir betrachten nochmals den NFA aus Abbildung 10 auf Seite 28. Das Wort $w = 0100$ hat mehrere Läufe, nämlich:

$q_0 q_0 q_0 q_0 q_0$	ist nicht akzeptierend
$q_0 q_0 q_0 q_0 q_1$	ist nicht akzeptierend
$q_0 q_0 q_0 q_1 q_2$	ist akzeptierend
$q_0 q_1$	ist nicht akzeptierend
$q_1 q_2$	ist nicht akzeptierend

Da nur die Existenz eines akzeptierenden Laufs entscheidend ist, gilt $w = 0100 \in \mathcal{L}(\mathcal{M})$, obwohl es für w auch verwerfende Läufe gibt. Tatsächlich akzeptiert \mathcal{M} genau diejenigen Wörter, die entweder 0 sind oder mit 00 enden:

$$\mathcal{L}(\mathcal{M}) = \{0\} \cup \{x00 : x \in \{0,1\}^*\}.$$

Für den Nachweis der Inklusion “ \supseteq ” genügt es für jedes der Wörter 0 oder $x00$ mit $x \in \{0,1\}^*$ einen akzeptierenden Lauf in \mathcal{M} anzugeben. Für das Wort 0 ist es der Lauf $q_1 q_2$. Für die Wörter $x00$ mit $|x| = k$ ist

$$\underbrace{q_0 q_0 \dots q_0}_{k+1\text{-mal}} q_1 q_2$$

ein akzeptierender Lauf, in denen \mathcal{M} bei Scannen des Präfix x in Zustand q_0 verharrt und mit den letzten beiden Nullen von q_0 über q_1 in den Endzustand q_2 geht.

Die Inklusion “ \subseteq ” folgt aus der Beobachtung, dass (1) nur das Wort 0 einen in q_1 beginnenden akzeptierenden Lauf besitzt und (2) alle Wörter, die einen in q_0 beginnenden akzeptierenden Lauf haben, mit 00 enden. Aussage (1) ist offensichtlich. Aussage (2) ergibt sich aus der Tatsache, dass alle Läufe, die in q_0 beginnen und in q_2 enden, die Gestalt $q_0 q_0 \dots q_0 q_1 q_2$ haben. Da Übergänge von q_0 nach q_1 und von q_1 nach q_2 nur durch Lesen des Zeichens 0 möglich sind, muss das betreffende Eingabewort die Form $x00$ haben. ■

Beispiel 2.13 (NFA für Mustererkennung). Wir betrachten das Problem der Mustererkennung, bei dem ein Muster $M = a_1 \dots a_m$ und ein Text $T = b_1 \dots b_t$ gegeben sind und gefragt ist, ob M in T vorkommt. Die Idee für einen nichtdeterministischen Mustererkennungsalgorithmus besteht darin, nichtdeterministisch eine Textposition i zu erraten, und dann zeichenweise zu prüfen, ob das Muster im Text ab Position i beginnt.

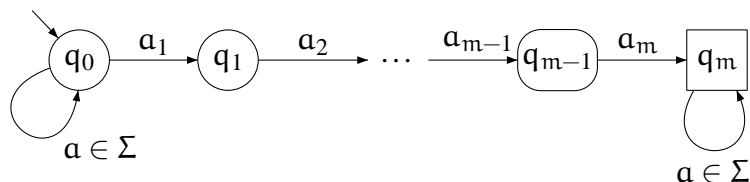


Figure 11: NFA für das Mustererkennungsproblem

Für festes Muster kann dieses sehr simple Verfahren durch den in Abbildung 11 skizzierten NFA \mathcal{M} realisiert werden. Wird ein Text $T = b_1 b_2 \dots b_t$ über diesem NFA “gescannt” und stimmt das aktuelle Eingabezeichen b_i mit dem ersten Zeichen a_1 des Musters überein, dann rät der NFA nichtdeterministisch, ob das Muster ab Position i beginnt. Wenn ja, wechselt der NFA von dem Anfangszustand q_0 in den Zustand q_1 und prüft nun deterministisch, ob die restlichen Zeichen $a_2 \dots a_m$ des Musters mit den folgenden Textzeichen $b_{i+1} b_{i+2} \dots b_{i+m-1}$ übereinstimmen. Die von \mathcal{M} akzeptierte Sprache ist also die Menge aller Texte $T \in \Sigma^*$, so dass M ein Teilwort von T ist. ■

Äquivalenz von DFA und NFA

Jeder DFA kann als NFA aufgefasst werden. Dies ist wie folgt einsichtig. Ist $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$ ein DFA, dann definieren wir einen NFA wie folgt. Sei $\mathcal{M}' = (Q, \Sigma, \delta', \{q_0\}, F)$, wobei

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\} & : \text{ falls } \delta(q, a) \neq \perp \\ \emptyset & : \text{ sonst.} \end{cases}$$

Offenbar ist jeder Lauf von \mathcal{M}' für w zugleich ein Lauf von \mathcal{M} in w und umgekehrt. Somit gilt $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}')$. Wir zeigen nun, dass es auch umgekehrt zu jedem NFA einen DFA gibt, welcher dieselbe Sprache akzeptiert.

Definition 2.14 (Äquivalenz von NFA). Seien \mathcal{M}_1 und \mathcal{M}_2 zwei NFA mit demselben Alphabet Σ . \mathcal{M}_1 und \mathcal{M}_2 heißen *äquivalent*, falls $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$.³ ■

Satz 2.15 (NFA \rightsquigarrow DFA, Potenzmengenkonstruktion). Zu jedem NFA gibt es einen äquivalenten DFA.

Proof. Sei $\mathcal{M} = (Q, \Sigma, \delta, Q_0, F)$ ein NFA. Wir wenden die sog. *Potenzmengenkonstruktion* an und definieren einen DFA, dessen Zustände Mengen von Zuständen in \mathcal{M} sind.

$$\mathcal{M}_{\text{det}} \stackrel{\text{def}}{=} \left(2^Q, \Sigma, \delta_{\text{det}}, Q_0, F_{\text{det}} \right),$$

³Wir fassen DFA als NFA auf, so dass der Äquivalenzbegriff zugleich für DFA definiert ist.

wobei

$$F_{\text{det}} \stackrel{\text{def}}{=} \{P \subseteq Q : P \cap F \neq \emptyset\}$$

$$\delta_{\text{det}}(P, \alpha) \stackrel{\text{def}}{=} \bigcup_{p \in P} \delta(p, \alpha).$$

Insbesondere ist $\delta_{\text{det}}(\emptyset, \alpha) = \emptyset$ für alle $\alpha \in \Sigma$. Man beachte, dass die Übergangsfunktion δ_{det} von \mathcal{M}_{det} total ist. Wir zeigen nun, dass $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}_{\text{det}})$.

Die einfachste Argumentation benutzt die erweiterten Übergangsfunktionen in \mathcal{M} und \mathcal{M}_{det} . Tatsächlich stimmt die deterministische Übergangsfunktion $\delta_{\text{det}} : 2^Q \times \Sigma^* \rightarrow 2^Q$ von \mathcal{M}_{det} mit der in Definition 2.9 angegebenen Erweiterung der nichtdeterministischen Übergangsfunktion δ von \mathcal{M} überein. Der Nachweis dieser Aussage wird erbracht, indem man durch Induktion nach der Wortlänge von w zeigt, dass $\delta_{\text{det}}(P, w) = \delta(P, w)$. Es gilt $\delta_{\text{det}}(P, \varepsilon) = \delta(P, \varepsilon)$ und $\delta_{\text{det}}(P, \alpha) = \delta(P, \alpha)$ (Definition von δ_{det}). Weiter ist für $\alpha \in \Sigma$ und $w \in \Sigma^+$

$$\begin{aligned} w \in \mathcal{L}(\mathcal{M}_{\text{det}}) &= \delta_{\text{det}}(\delta_{\text{det}}(P, \alpha), w) \\ &= \delta_{\text{det}}(\delta(P, \alpha), w) \\ &= \delta(\delta(P, \alpha), w) \\ &= \delta(P, \alpha w) \end{aligned}$$

Nun folgt:

$$\begin{aligned} w \in \mathcal{L}(\mathcal{M}_{\text{det}}) &\text{ gdw } \delta_{\text{det}}(Q_0, w) \in F_{\text{det}} \\ &\text{ gdw } \delta(Q_0, w) \in F_{\text{det}} \\ &\text{ gdw } \delta(Q_0, w) \cap F \neq \emptyset \\ &\text{ gdw } w \in \mathcal{L}(\mathcal{M}) \end{aligned}$$

Wir geben nun einen expliziten Nachweis der Gleichung $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}_{\text{det}})$ an, der mit akzeptierenden Läufen in \mathcal{M} bzw. \mathcal{M}_{det} argumentiert und daher zugleich verdeutlicht, inwiefern \mathcal{M}_{det} die möglichen Berechnungen von \mathcal{M} simuliert.

“ \subseteq ”: Sei $w = a_1 a_2 \dots a_n \in \mathcal{L}(\mathcal{M})$ und $q_0 q_1 \dots q_n$ ein akzeptierender Lauf von \mathcal{M} für w . Dann ist $q_0 \in Q_0$, $q_{i+1} \in \delta(q_i, a_{i+1})$ für $i = 0, 1, \dots, n-1$, und $q_n \in F$. Seien $P_0 = Q_0$ und $P_i = \delta(P_{i-1}, a_i)$. Damit ist $P_i = \delta_{\text{det}}(P_{i-1}, a_i)$. Durch Induktion nach i kann man zeigen, dass

$$q_i \in P_i \text{ für } i = 0, 1, \dots, n.$$

Somit ist $P_0 P_1 \dots P_n$ der zu w gehörende Lauf in \mathcal{M}_{det} . Insbesondere ist $q_n \in P_n \cap F$ und daher $P_n \cap F \neq \emptyset$. Hieraus folgt $P_n \in F_{\text{det}}$. Also ist $P_0 P_1 \dots P_n$ ein akzeptierender Lauf. Somit ist $w \in \mathcal{L}(\mathcal{M}_{\text{det}})$.

“ \supseteq ”: Sei $w \in \mathcal{L}(\mathcal{M}_{\text{det}})$ und $w = a_1 \dots a_n$. Weiter sei $P_0 P_1 \dots P_n$ der zu w gehörende Lauf in \mathcal{M}_{det} . Dann ist $P_0 = Q_0$, $P_{i+1} = \delta_{\text{det}}(P_i, a_{i+1})$, $i = 0, 1, \dots, n-1$, und P_n ein Endzustand in \mathcal{M}_{det} , also $P_n \cap F \neq \emptyset$. Wir konstruieren nun “rückwärts” einen akzeptierenden Lauf $q_0 q_1 \dots q_n$ für w in \mathcal{M} :

Wir wählen einen beliebigen Zustand $q_n \in P_n \cap F$.

Wähle $q_{n-1} \in P_{n-1}$ mit $q_n \in \delta(q_{n-1}, a_n)$.

Wähle $q_{n-2} \in P_{n-2}$ mit $q_{n-1} \in \delta(q_{n-2}, a_{n-1})$.

⋮

Wähle $q_0 \in P_0$ mit $q_1 \in \delta(q_0, a_1)$.

Tatsächlich gibt es solche Zustände q_i , da $P_n \cap F \neq \emptyset$ und für $i = n-1, \dots, 0$:

$$P_{i+1} = \delta_{\text{det}}(P_i, a_{i+1}) = \bigcup_{q \in P_i} \delta(q, a_{i+1}).$$

Daher gibt es zu jedem Zustand $q_{i+1} \in P_{i+1}$ einen Zustand $q_i \in P_i$ mit $q_{i+1} \in \delta(q_i, a_{i+1})$. Wegen $P_0 = Q_0$ ist q_0 ein Anfangszustand und $q_0 q_1 \dots q_n$ ein Lauf von \mathcal{M} für w . Wegen $q_n \in F$ ist der Lauf $q_0 q_1 \dots q_n$ akzeptierend. Also ist $w \in \mathcal{L}(\mathcal{M})$. \square

In Abbildung 12 betrachten wir ein einfaches Beispiel für die im Beweis von Satz 2.15 angegebene Potenzmengenkonstruktion. Links ist der NFA \mathcal{M} skizziert, rechts dessen Potenzmengenkonstruktion. Beispielsweise kann der akzeptierende Lauf $q_0 q_1 q_0 q_1 q_0$ für das Wort 1010 in \mathcal{M} durch den akzeptierenden Lauf $\{q_0\}\{q_0, q_1\}\{q_0\}\{q_0, q_1\}\{q_0\}$ im DFA \mathcal{M}_{det} (Potenzmengenkonstruktion) simuliert werden. Zustand $\{q_1\}$ in der Potenzmengenkonstruktion ist unerreichbar. Er kann daher weggelassen werden, ohne die akzeptierte Sprache zu ändern. Auch auf den Zustand \emptyset kann verzichtet werden, da von ihm kein Endzustand erreichbar ist.

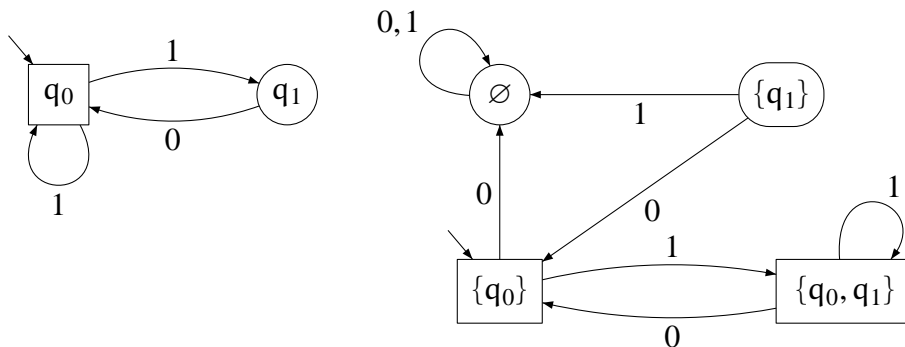


Figure 12: Beispiel zur Potenzmengenkonstruktion

Beispiel 2.16 (Das Teilsummenproblem). Das Teilsummenproblem bezeichnet folgende kombinatorische Fragestellung:

- Gegeben ist eine ganze Zahl $K \geq 2$ und eine nichtleere Folge $a_1 a_2 \dots a_n \in \{1, \dots, K\}^*$.
- Gefragt ist, ob es eine Teilmenge I von $\{1, \dots, n\}$ gibt, so dass $\sum_{i \in I} a_i = K$.

Ein einfacher nichtdeterministische Algorithmus, der das Teilsummenproblem löst, rät nichtdeterministisch eine Teilmenge I von $\{1, \dots, n\}$ und prüft dann, ob die Teilsumme der geratenen Indexmenge gleich K ist. Für festes K können wir diese Idee durch einen NFA realisieren. Intuitiv bearbeitet der Automat die Eingabewerte a_1, a_2, \dots, a_n der Reihe nach und entscheidet

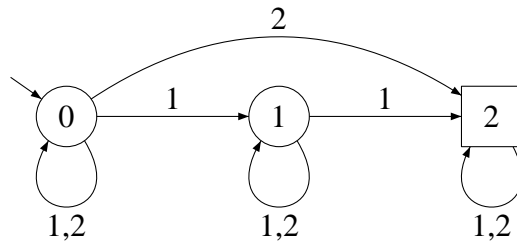


Figure 13: NFA $\mathcal{M}^{(K)}$ für das Teilsummenproblem ($K = 2$)

nichtdeterministisch, die i -te Zahl a_i an der Teilsumme zu beteiligen oder nicht. Wir betrachten folgenden NFA $\mathcal{M}^{(K)}$, dessen Zustände die Zahlen $q \in \{0, 1, \dots, K\}$ sind, welche jeweils für den Wert der bereits erzielten Teilsummen stehen. Der Automat verfügt über einen eindeutigen Anfangs- und Endzustand. Der Anfangszustand ist 0; dies entspricht der initialen Teilsumme 0. Der Endzustand ist K (der gewünschte Wert der Teilsumme). Wir definieren den NFA $\mathcal{M}^{(K)}$ wie folgt:

$$\mathcal{M}^{(K)} \stackrel{\text{def}}{=} (\{0, 1, \dots, K\}, \{1, 2, \dots, K\}, \delta, \{0\}, \{K\}),$$

wobei für jeden Zustand $q \in \{0, 1, \dots, K\}$ und jedes Zeichen $a \in \{1, \dots, K\}$:

$$\delta(q, a) \stackrel{\text{def}}{=} \{q, q+a\} \cap \{0, 1, \dots, K\} = \begin{cases} \{q\} & : \text{ falls } q+a > K \\ \{q, q+a\} & : \text{ falls } q+a \leq K \end{cases}$$

Für $K = 2$ hat der NFA $\mathcal{M}^{(K)}$ die in Abbildung 13 angegebene Gestalt. Man überlegt sich leicht, dass die von $\mathcal{L}(\mathcal{M}^{(K)})$ akzeptierte Sprache genau die Menge aller Zahlenfolgen $a_1 a_2 \dots a_n \in \{1, 2, \dots, K\}^*$ ist, für die es eine Teilfolge $a_{i_1} a_{i_2} \dots a_{i_\ell}$ gibt, deren Summe K ergibt.

Die Potenzmengenkonstruktion angewandt auf $\mathcal{M}^{(K)}$ liefert einen DFA $\mathcal{M}_{\text{det}}^{(K)}$, dessen Zustände Teilmengen von $\{0, 1, \dots, K\}$ sind:

$$\mathcal{M}_{\text{det}}^{(K)} = (2^{\{0, 1, \dots, K\}}, \{1, \dots, K\}, \delta_{\text{det}}, \{0\}, F_{\text{det}}),$$

wobei

$$\begin{aligned} F_{\text{det}} &= \{P \subseteq \{0, 1, \dots, K\} : K \in P\} \\ \delta_{\text{det}}(P, a) &= \{q \in \{0, 1, \dots, K\} : q \in P \text{ oder } q-a \in P\} \\ &= P \cup \{p+a : p \in P, p+a \leq K\} \end{aligned}$$

für $a \in \{1, \dots, K\}$ und $P \subseteq \{0, 1, \dots, K\}$. Man überzeugt sich nun leicht davon, dass für jedes Eingabewort $w = a_1 a_2 \dots a_n$ über dem Alphabet $\{1, 2, \dots, K\}$ gilt:

$$\begin{aligned} w \text{ hat genau dann einen akzeptierenden Lauf in } \mathcal{M}_{\text{det}}^{(K)}, \\ \text{wenn } \sum_{i \in I} a_i = K \text{ für eine Teilmenge } I \text{ von } \{1, \dots, n\}. \end{aligned}$$

Der DFA $\mathcal{M}_{\text{det}}^{(K)}$ kann also nun als Vorlage für einen deterministischen Algorithmus für das Teilsummenproblem (z.B. basierend auf Backtracking oder dynamischem Programmieren) dienen.

■