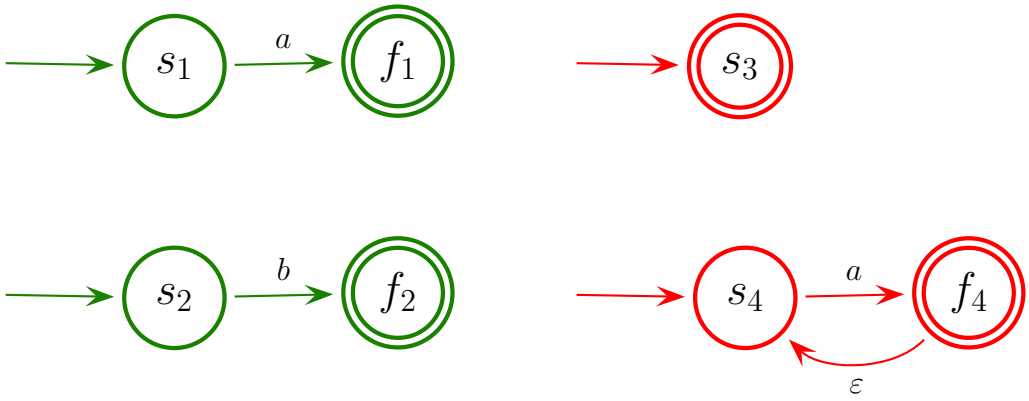
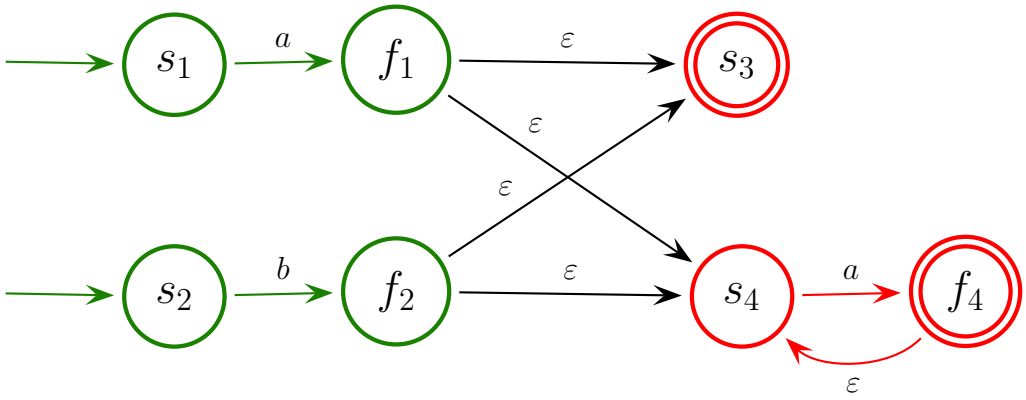
Lösungen zu Übungsblatt 3

Aufgabe 1

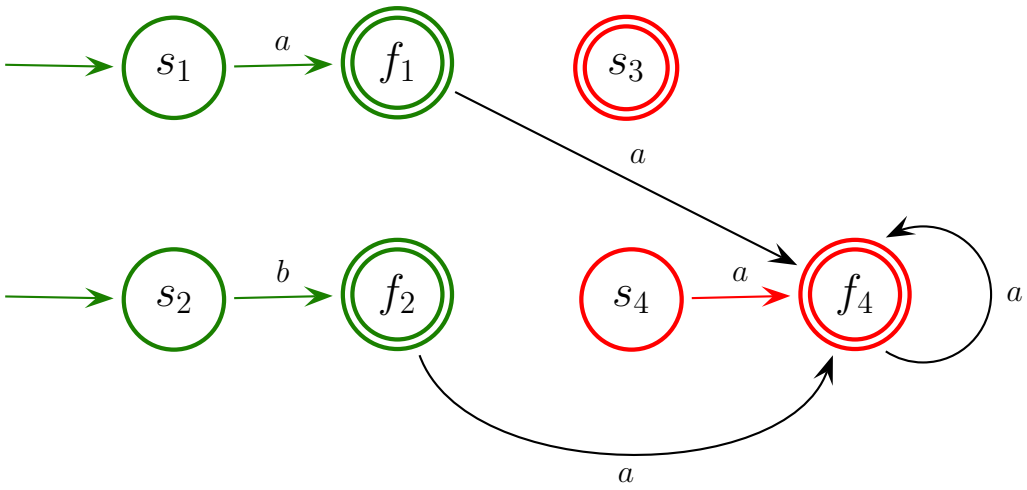
a. Es gilt $L((a|b)a^\star) = (\{a\} \cup \{b\}) \circ \{a\}^* = (\{a\} \cup \{b\}) \circ (\{\varepsilon\} \cup \{a\}^+)$. Mit dem Verfahren aus dem Beweis zu Satz 2.20 erhalten wir zunächst die folgenden beiden ε-NDEAs für die Sprachen $\{a\} \cup \{b\}$ und $\{\varepsilon\} \cup \{a\}^+$



und dann durch "Hintereinanderschalten" der beiden ε-NDEAs einen ε-NDEA für die Konkatenation der beiden Sprachen

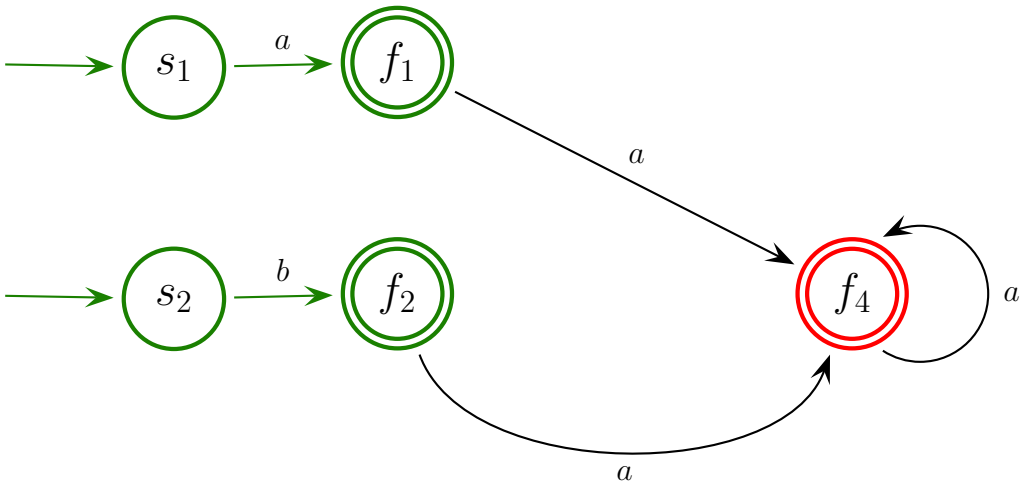


b. Aus dem ε-NDEA in **a.** erhalten wir mit dem Verfahren aus dem Beweis zu Satz 2.18 den NDEA



Man beachte, dass f_1 und f_2 jetzt (wieder) Endzustände sind, weil beide einen ε-übergang zum Endzustand s_3 besaßen.

Schließlich ergibt sich durch Entfernung der unerreichbaren Zustände der NDEA



Aufgabe 2

Gegeben seien zwei DEAs $A_1 = (\Sigma, Q_1, s_1, F_1, \delta_1)$ und $A_2 = (\Sigma, Q_2, s_2, F_2, \delta_2)$.

a. Gesucht ist ein DEA A mit $L(A) = L(A_1) \cup L(A_2)$.

Die Idee besteht darin, die Automaten A_1 und A_2 *parallel* auf dem Eingabewort w laufen zu lassen und das Wort genau dann zu akzeptieren, wenn (mindestens) einer der beiden Automaten in einen Endzustand gelangt. Eine solche Parallelausführung simuliert der folgende DEA A.

$A = (\Sigma, Q, s, F, \delta)$ mit

- $Q = Q_1 \times Q_2$
- $s = (s_1, s_2)$
- $F = F_1 \times Q_2 \ \cup \ Q_1 \times F_2$
- $\delta : Q \times \Sigma \to Q$
  $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$

Aus der Definition von δ folgt durch Induktion über $|w|$, dass

$$\delta^*(s, w) \quad = \quad (\delta_1^*(s_1, w), \delta_2^*(s_2, w)) \tag{1}$$

für alle $w \in \Sigma^*$. Also gilt tatsächlich

$$\begin{aligned}
w \in L(A) \quad &\Leftrightarrow \quad \delta^*(s, w) \in F \\
&\qquad \text{per Definition des Akzeptierens} \\
&\Leftrightarrow \quad (\delta_1^*(s_1, w), \delta_2^*(s_2, w)) \in F \\
&\qquad \text{wegen (1)} \\
&\Leftrightarrow \quad \delta_1^*(s_1, w) \in F_1 \text{ oder } \delta_2^*(s_2, w) \in F_2 \\
&\qquad \text{per Definition von } F \\
&\Leftrightarrow \quad w \in L(A_1) \text{ oder } w \in L(A_2) \\
&\qquad \text{per Definition des Akzeptierens} \\
&\Leftrightarrow \quad w \in L(A_1) \cup L(A_2)
\end{aligned}$$

b. Gesucht ist ein DEA A mit $L(A) = L(A_1) \cap L(A_2)$.

Die Idee ist die gleiche wie in **a.** mit einem einzigen Unterschied: Das Eingabewort w wird akzeptiert, wenn *beide* Automaten in einen Endzustand gelangen. Dementsprechend definiert man $A = (\Sigma, Q, s, F, \delta)$ mit Q, s und δ wie in **a.**, aber mit Endzustandsmenge $F = F_1 \times F_2$.

Damit erhält man

$$\begin{aligned} w \in L(A) &\Leftrightarrow \delta^*(s,w) \in F \\ &\Leftrightarrow (\delta_1^*(s_1,w), \delta_2^*(s_2,w)) \in F \\ &\Leftrightarrow \delta_1^*(s_1,w) \in F_1 \text{ und } \delta_2^*(s_2,w) \in F_2 \\ &\Leftrightarrow w \in L(A_1) \text{ und } w \in L(A_2) \\ &\Leftrightarrow w \in L(A_1) \cap L(A_2) \end{aligned}$$

Vergleich mit den in der Vorlesung angegebenen Algorithmen:

Durch die Verfahren aus der Vorlesung entstehen zunächst nichtdeterministische Automaten, die man mit der Potenzmengen-Konstruktion wieder in deterministische umwandeln muss. Da die Potenzmengen-Konstruktion im schlechtesten Fall exponentiellen Zeit- und Platzbedarf hat, sind die hier vorgestellten Verfahren wesentlich effizienter.

Aufgabe 3

Sei $A = (\Sigma, Q, s, F, \delta)$.

'(a) $\Rightarrow$ (c)':

Sei $L(A)$ unendlich. Da es in Σ^* nur endlich viele Wörter der Länge $< |Q|$ gibt, enthält $L(A)$ sogar unendlich viele Wörter der Länge $\geq |Q|$.

'(c) $\Rightarrow$ (b)':

Sei $w \in L(A)$ mit $|w| \geq |Q|$. Der Lauf $(s,w) \vdash_A^* (f,\varepsilon)$ besteht aus $|w|$ Übergangsschritten und damit aus $|w| + 1 > |Q|$ Konfigurationen. Da es nur $|Q|$ verschiedene Zustände gibt, muss also (mindestens) ein Zustand in diesem Lauf mehrmals auftreten.

'(b) $\Rightarrow$ (a)':

Sei $w \in L(A)$ ein Wort, in dessen Lauf der Zustand p mehrmals auftritt, d.h. es existieren $x, v, y \in \Sigma^*$ mit $w = xvy$, $v \neq \varepsilon$ und

$$(s,w) = (s,xvy) \ \vdash_A^* \ (p,vy) \ \vdash_A^+ \ (p,y) \ \vdash_A^* \ (f,\varepsilon)$$

Da man beim Lesen des Teilwortes v eine Schleife von p nach p durchläuft, kann man durch i-malige Wiederholung dieser Schleife auch mit jeder Potenz v^i ($i \in \mathbb{N}$) von p nach p gelangen. Also gilt auch

$$(s,xv^iy) \ \vdash_A^* \ (p,v^iy) \ \vdash_A^+ \ (p,y) \ \vdash_A^* \ (f,\varepsilon)$$

d.h. $uv^iy \in L(A)$ für alle $i \in \mathbb{N}$, und wegen $v \neq \varepsilon$ sind dies unendlich viele Wörter.

Aufgabe 4

Sei $A = (\Sigma, Q, s, F, \delta)$.

a. ε liegt genau dann in $L(A)$, wenn s ein Endzustand ist.

b. Nach Aufgabe 3 gilt

$$L(A) \text{ ist unendlich} \quad \Leftrightarrow \quad \exists\, w \in L(A).\ |w| \geq |Q| \tag{2}$$

Da es *unendlich viele* Wörter der Länge $\geq |Q|$ gibt, liefert uns (2) noch keinen Entscheidungs-Algorithmus. Wir benötigen dazu noch das Ergebnis aus Aufgabe 5 b von Übungsblatt 4: Wenn $L(A)$ ein Wort der Länge $\geq |Q|$ enthält, dann enthält $L(A)$ sogar ein Wort, dessen Länge zwischen $|Q|$ und $2 \cdot |Q|$ liegt. Also gilt sogar

$$L(A) \text{ ist unendlich} \quad \Leftrightarrow \quad \exists\, w \in L(A).\ |Q| \leq |w| < 2 \cdot |Q| \tag{3}$$

Mit (3) erhalten wir einen Entscheidungsalgorithmus: Um zu testen ob $L(A)$ unendlich ist, braucht man nur zu überprüfen, ob eines der *endlich vielen* Wörter $w \in \Sigma^*$ mit $|Q| \leq |w| < 2 \cdot |Q|$ von A akzeptiert wird.

c. Gesucht ist ein Algorithmus, der überprüft ob $L(A) \neq \emptyset$. Laut Vorlesung existiert ein Algorithmus, der überprüft ob $L(A) = \emptyset$. Daraus erhält man den gewünschten Algorithmus, indem man die Antworten "ja" und "nein" vertauscht. (In der Terminologie der Berechenbarkeitstheorie: Das Komplement einer entscheidbaren Menge ist ebenfalls entscheidbar.)

d. $L(A)$ enthält genau dann ein Wort mit drei aufeinander folgenden Nullen, wenn es Zustände $p, q \in Q$ gibt mit

- p ist erreichbar
- $\delta^*(p, 000) = q$
- q ist lebendig

Da man die Menge der erreichbaren Zustände und (ganz analog) die Menge der lebendigen Zustände von A berechnen kann, hat man den gewünschten Algorithmus: Für jeden erreichbaren Zustand p bestimmt man $q = \delta^*(p, 000)$ und überprüft ob q lebendig ist.

Einen alternativen Algorithmus erhält man mit Hilfe von Aufgabe 2: $L(A)$ enthält genau dann ein Wort mit drei aufeinander folgenden Nullen, wenn der Durchschnitt von $L(A)$ mit der regulären Sprache

$L_0 = \{w \in \Sigma^* \mid w \text{ enthält } 000\}$ nicht leer ist. Sei also A_0 ein DEA, der L_0 erkennt. Dann arbeitet der gewünschte Algorithmus wie folgt: Mit dem Verfahren aus Aufgabe 2 konstruiert er einen DEA A' mit $L(A') = L(A) \cap L(A_0)$ und überprüft mit dem Algorithmus aus **c.** ob $L(A') \neq \emptyset$ ist. (In der Terminologie der Berechenbarkeitstheorie: Das Entscheidungsproblem **d.** wird auf das Entscheidungsproblem **c.** reduziert.)

e. Nach Aufgabe 3 gilt: Wenn $L(A)$ endlich ist, dann enthält $L(A)$ nur Wörter der Länge $< |Q|$. Also gibt es im Falle $|L(A)| \geq 2$ nur folgende Alternativen:

- entweder $L(A)$ ist endlich und enthält mindestens zwei Wörter der Länge $< |Q|$
- oder $L(A)$ ist unendlich

Damit hat man den gewünschten Entscheidungs-Algorithmus: Zuerst überprüft man, ob unter den endlich vielen Wörtern der Länge $< |Q|$ bereits zwei existieren, die von A akzeptiert werden. Wenn dies nicht der Fall ist, so kann $|L(A)| \geq 2$ nur noch gelten, wenn $L(A)$ unendlich ist, und das kann man mit **b.** überprüfen.

Aufgabe 5

Alle Fragestellungen aus Aufgabe 4 beziehen sich nur auf die Sprache $L(A)$ und nicht auf den Automaten A selbst. Um eine solche Fragestellung für einen ε-NDEA A zu beantworten, konstruiert man mit den Verfahren aus der Vorlesung einen zu A äquivalenten DEA A' und beantwortet die Frage für A'. Wegen $L(A) = L(A')$ ist das dann auch die richtige Antwort für A. (In der Terminologie der Berechenbarkeitstheorie: Das Entscheidungsproblem für ε-NDEAs lässt sich jeweils auf das entsprechende Entscheidungsproblem für DEAs reduzieren.)