Aufgabe 1

1. L_1 ist eine NKA-Sprache und nicht regulär.

   - Wir zeigen zunächst, dass L_1 nicht regulär ist.
     Sei $N \in \mathbb{N}$ beliebig. Wähle $x = a^{3\cdot N+1}c^N = uvw \in L_1$ mit $|uv| \leq N$. Daher haben wir $v = a^k$, $k > 0$. Für $i = 0$ gilt dann offensichtlich $uv^0w = a^{3\cdot N+1-k}c^N$ mit $3 \cdot N + 1 - k \leq 3 \cdot N$. Folglich ist
     $$\#_a(uv^0w) + 2 \cdot \#_b(uv^0w) = 3 \cdot N + 1 - k + 2 \cdot 0 = 3 \cdot N + 1 - k \leq 3 \cdot N$$
     und damit $uv^0w \notin L_1$.
   - Wir zeigen nun, dass L_1 eine NKA-Sprache ist.
     Lege für jedes gelesene a ein Element auf den Keller und für jedes b zwei. Lösche für jedes c drei Elemente aus dem Keller. Kommt nach dem ersten b noch ein a, verwerfe. Kommt nach dem ersten c noch ein a oder ein b, verwerfe. Wenn das gesamte Wort gelesen wurde, prüfe, ob der Keller noch Elemente enthält. Falls ja, akzeptiere. Sonst verwerfe.
     Formal: M akzeptiert durch Endzustand, $M := \{\Sigma, \Gamma, e, Q, s, \{f\}, \Delta\}$, mit $\Sigma = \{a, b, c\}, \Gamma = \{e, E\}, Q = \{s, t, v_0, v_1, v_2, f\}$ und:
     $$\begin{aligned}\Delta = &\{(s, x, a, xE, s), (s, x, \varepsilon, x, t) \mid \forall x \in \Gamma\}\\ &\cup \{(t, x, b, xEE, t), (t, x, \varepsilon, x, v_0) \mid \forall x \in \Gamma\}\\ &\cup \{(v_0, E, c, \varepsilon, v_1), (v_1, E, \varepsilon, \varepsilon, v_2), (v_2, E, \varepsilon, \varepsilon, v_0), (v_0, E, \varepsilon, E, f)\}\end{aligned}$$

2. L_2 is eine NKA-Sprache und nicht regulär.

   - Wir zeigen zunächst, dass L_2 nicht regulär ist.
     Sei $N \in \mathbb{N}$ beliebig. Wähle $x = a^N b^{2\cdot N} = uvw \in L_2$ mit $|uv| \leq N$ und damit $v = a^k$, $k > 0$. Für $i = 2$ ergibt sich $uv^2w = a^{N+k}b^{2\cdot N}$. Da $2 \cdot (N + k) = 2 \cdot N + 2 \cdot k > 2 \cdot N$, gilt $uv^2w \notin L_2$.
   - Wir zeigen nun, dass L_2 eine NKA-Sprache ist.
     Der NKA kann sich in drei verschiedenen Phasen befinden. In Phase 1 gilt $\#_b(w) > 2 \cdot \#_a(w)$, in Phase 2 gilt $\#_b(w) = 2 \cdot \#_a(w)$ und in Phase 3 gilt $\#_b(w) < 2 \cdot \#_a(w)$.
     - Befindet sich der NKA in Phase 1 (Zustand t) und liest ein b, so lege ein B auf den Keller und bleibe in Phase 1. Wird ein a gelesen, prüfe, ob sich noch zwei B im Keller befinden. Falls ja, lösche zwei B aus dem Keller und bleibe in Phase 1. Ist der Keller nun leer, so gehe in Phase 2 über. Ist nur ein B im Keller, dann lösche dieses aus dem Keller und lege ein A auf dem Keller, gehe in diesem Fall Phase 3 über.
     - Befindet sich der NKA in Phase 2 (Zustand s) und liest ein b, so lege ein B auf den Keller und gehe in Phase 1 über. Wird ein a gelesen, so lege zwei A auf den Keller und gehe in Phase 3 über.
     - Befindet sich der NKA in Phase 3 (Zustand v) und liest ein b, so entferne ein A aus dem Keller. Ist der Keller nun leer, so gehe in Phase 2 über. Andernfalls bleibe in Phase 3. Wird ein a gelesen, so lege zwei A auf den Keller und bleibe in Phase 3.

     Der NKA akzeptiert, wenn am Ende des Wortes der Keller leer ist oder nur B auf dem Keller liegen. Formal: M akzeptiert durch Endzustand, $M := \{\Sigma, \Gamma, e, Q, s, \{f\}, \Delta\}$, mit $\Sigma = \{a, b, c\}, \Gamma = \{e, A, B\}, Q = \{s, t, t', v, f\}$ und:
     $$\begin{aligned}\Delta = &\{(s, e, a, eAA, v), (s, e, b, eB, t), (s, e, \varepsilon, e, f)\}\\ &\cup \{(v, A, b, \varepsilon, v), (v, A, a, AAA, v), (v, e, \varepsilon, e, s)\}\\ &\cup \{(t, B, b, \varepsilon, BB, t), (t, B, a, \varepsilon, t'), (t, e, \varepsilon, e, s), (t, B, \varepsilon, B, f)\}\\ &\cup \{(t', B, \varepsilon, \varepsilon, t), (t', e, \varepsilon, eA, v)\}\end{aligned}$$

3. L_3 ist regulär. Der Automat $M = (Q, \Sigma, s, F, \Delta)$ akzeptiert L_3 mit:
   $$\begin{aligned}Q &= \{q_{(i,j)} \mid 0 \leq i \leq 4 \ \wedge \ 0 \leq j \leq 2\}\\ \Sigma &= \{a, b\}\\ s &= q_{(0,0)}\\ F &= \{q_{(i,j)} \mid i \leq j\}\\ \Delta &= \{(q_{(i,j)}, a, q_{(i+1 \bmod 5, j)}) \mid q_{(i,j)} \in Q\}\\ &\quad \cup \{(q_{(i,j)}, b, q_{(i, j+1 \bmod 3)}) \mid q_{(i,j)} \in Q\}\end{aligned}$$

4. L_4 ist keine NKA-Sprache. Wir zeigen dies mit Hilfe des Pumping-Lemmas für NKA-Sprachen. Sei $N \in \mathbb{N}$ beliebig. Wir wählen eine Primzahl $p > N$ und $z = a^{p^2} b^p \in L_4$. Fallunterscheidung über $z = uvwxy$, wobei $|vwx| \leq N$ und $|vx| > 0$.

   (a) v und x bestehen nur aus as.
   Dann müssen die p bs alle in y enthalten sein. Sei $v = a^k$ und $x = a^\ell$. Für $i = 2$ gilt $uv^2wx^2y = a^{p^2+k+\ell}b^p$. Da $|vwx| \leq N < p$ gilt, gilt offensichtlich $k + \ell < p$ und somit $p^2 + k + \ell \bmod p \neq 0$. Also gilt $uv^2wx^2y \notin L_4$.

   (b) v und x bestehen zusammen aus mehr bs als as.
   Wir wählen i groß genug, sodass $\#_b(uv^iwx^iy) > \#_a(uv^iwx^iy)$ gilt (dies ist möglich, da wir durch das „Aufpumpen" von v und x immer mehr bs generieren). Dann gilt

   $$\#_a(uv^iwx^iy) \bmod \#_b(uv^iwx^iy) = \#_a(uv^iwx^iy) \neq 0$$

   Somit folgt $uv^iwx^iy \notin L_4$.

   (c) v und x bestehen zusammen aus mindestens so vielen as wie bs.
   Sei k die Anzahl der as und ℓ die Anzahl der bs in v und x gemeinsam. Dann gilt nach Annahme $0 < \ell \leq k$. Wähle $i = p^2 + 1$. Dann gilt

   $$uv^iwx^iy = a^{p^2+(i-1)\cdot k}b^{p+(i-1)\cdot\ell} = a^{p^2+p^2\cdot k}b^{p+p^2\cdot\ell} = a^{p^2\cdot(1+k)}b^{p\cdot(1+p\cdot l)}$$

   Es gilt
   $$p^2\cdot(1+k) \bmod p\cdot(1+p\cdot\ell) = 0 \;\Leftrightarrow\; p\cdot(1+k) \bmod 1+p\cdot\ell = 0$$

   Da p eine Primzahl ist, gilt $p\cdot(1+k) \bmod 1+p\cdot\ell = 0 \;\Leftrightarrow\; 1+k \bmod 1+p\cdot\ell = 0$. Da aber $k \leq N$, $p > N$ und $\ell \in \mathbb{N}$ nach Annahme, gilt $1+p\cdot\ell > 1+k$. Somit gilt $1+k \bmod 1+p\cdot\ell \neq 0$. Daraus folgt, dass $uv^iwx^iy \notin L_4$ für $i = p^2+1$.

Aufgabe 2

Der NKA $M := (\Sigma, \Gamma, L, Q, s, \emptyset, \Delta)$ mit $\Sigma = \{a, b\}$, $\Gamma = \{A, L\}$, $Q = \{s\}$ und

$$\Delta = \{(s, L, a, AL, s), (s, L, \varepsilon, \varepsilon, s), (s, A, b, \varepsilon, s)\}$$

erkennt die gegebene Sprache.
Die Idee hinter dem NKA ist, dass er zuerst nur a's einließt und die Buchstaben im Keller speichert. Irgendwann entscheidet sich der Automat nichdeterministisch, dass er glaubt in der Mitte des Wortes angekommen zu sein. Jetzt ließ dem NKA nur b's ein. Betrachtet der Automat nun beim Lesen den obersten Buchstaben des Kellers und überprüft, ob sich noch A's im Keller befinden. Ist dies der Fall wird der Buchstabe vom Keller entfernt und weitergelesen. Hält der Automat auf leerem Keller, war das Wort in L_{ab}.

Der Automat merkt sich, ob er sich in der ersten oder Zweiten Phase des Erkennens befindet, mit Hilfe des Buchstabens L, der während der ersten Phase immer oben auf dem Keller liegt. Liegt L nicht mehr auf dem Keller, so befindet sich der Automat in der zweiten Phase und kann nur b's lesen und A's aus dem Keller entfernen. Der Wechsel zwischen den Phasen findet in akzeptierenden Berechenungen mit Lesen eines ε statt .

Akzeptierende Berechnung für Wort $aaabbb$:

$(L, s, aaabbb) \vdash_M (AL, s, aabbb) \vdash_M (AAL, s, abbb) \vdash_M (AAAL, s, bbb) \vdash_M (AAA, s, bbb) \vdash_M (AA, s, bb) \vdash_M (A, s, b) \vdash_M (\varepsilon, s, \varepsilon)$

Aufgabe 3

1.

$$
\begin{aligned}
merge(abc, bd) &= \{a\} \cdot merge(bc, bd) \;\cup\; \{b\} \cdot merge(abc, d) \\
&= \{a\} \cdot \big(\{b\} \cdot merge(c, bd) \;\cup\; \{b\} \cdot merge(bc, d)\big) \;\cup\; \{b\} \cdot \big(\{a\} \cdot merge(bc, d) \;\cup\; \{d\} \cdot merge(abc, \varepsilon)\big) \\
&= \{a\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot merge(\varepsilon, bd) \;\cup\; \{b\} \cdot merge(c, d)\big) \;\cup\; \{b\} \cdot \big(\{b\} \cdot merge(c, d) \;\cup\; \{d\} \cdot merge(bc, \varepsilon)\big)\big) \\
&\quad \cup\; \{b\} \cdot \big(\{a\} \cdot \big(\{b\} \cdot merge(c, d) \;\cup\; \{d\} \cdot merge(bc, \varepsilon)\big) \;\cup\; \{d\} \cdot \{abc\}\big) \\
&= \{a\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot \{bd\} \;\cup\; \{b\} \cdot \big(\{c\} \cdot merge(\varepsilon, d) \;\cup\; \{d\} \cdot merge(c, \varepsilon)\big)\big) \;\cup\; \{b\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot merge(\varepsilon, d) \;\cup\; \{d\} \cdot merge(c, \varepsilon)\big) \;\cup\; \{d\} \cdot \{bc\}\big)\big) \\
&\quad \cup\; \{b\} \cdot \big(\{a\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot merge(\varepsilon, d) \;\cup\; \{d\} \cdot merge(c, \varepsilon)\big) \;\cup\; \{d\} \cdot merge(bc, \varepsilon)\big) \;\cup\; \{d\} \cdot \{abc\}\big) \\
&= \{a\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot \{bd\} \;\cup\; \{b\} \cdot \big(\{c\} \cdot \{d\} \;\cup\; \{d\} \cdot \{c\}\big)\big) \;\cup\; \{b\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot \{d\} \;\cup\; \{d\} \cdot \{c\}\big) \;\cup\; \{d\} \cdot \{bc\}\big)\big) \\
&\quad \cup\; \{b\} \cdot \big(\{a\} \cdot \big(\{b\} \cdot \big(\{c\} \cdot \{d\} \;\cup\; \{d\} \cdot \{c\}\big) \;\cup\; \{d\} \cdot \{bc\}\big) \;\cup\; \{d\} \cdot \{abc\}\big) \\
&= \{a\} \cdot \big(\{b\} \cdot \big(\{cbd\} \;\cup\; \{b\} \cdot \big(\{cd\} \;\cup\; \{dc\}\big)\big) \;\cup\; \{b\} \cdot \big(\{b\} \cdot \big(\{cd\} \;\cup\; \{dc\}\big) \;\cup\; \{dbc\}\big)\big) \;\cup\; \{b\} \cdot \big(\{a\} \cdot \big(\{b\} \cdot \big(\{cd\} \;\cup\; \{dc\}\big) \;\cup\; \{dbc\}\big) \;\cup\; \{dabc\}\big) \\
&= \{a\} \cdot \big(\{b\} \cdot \big(\{cbd\} \;\cup\; \{bcd\} \;\cup\; \{bdc\}\big) \;\cup\; \{b\} \cdot \big(\{bcd\} \;\cup\; \{bdc\} \;\cup\; \{dbc\}\big)\big) \;\cup\; \{b\} \cdot \big(\{a\} \cdot \big(\{bcd\} \;\cup\; \{bdc\} \;\cup\; \{dbc\}\big) \;\cup\; \{dabc\}\big) \\
&= \{a\} \cdot \big(\{bcbd\} \;\cup\; \{bbcd\} \;\cup\; \{bbdc\} \;\cup\; \{bbcd\} \;\cup\; \{bbdc\} \;\cup\; \{bdbc\}\big) \;\cup\; \{b\} \cdot \big(\{abcd\} \;\cup\; \{abdc\} \;\cup\; \{adbc\} \;\cup\; \{dabc\}\big) \\
&= \{abcbd\} \;\cup\; \{abbcd\} \;\cup\; \{abbdc\} \;\cup\; \{abbcd\} \;\cup\; \{abbdc\} \;\cup\; \{abdbc\} \;\cup\; \{babcd\} \;\cup\; \{babdc\} \;\cup\; \{badbc\} \;\cup\; \{bdabc\} \\
&= \{abcbd, abbcd, abbdc, abdbc, babcd, babdc, badbc, bdabc\}
\end{aligned}
$$

2. Ja, die regulären Sprachen sind unter $merge()$ abgeschlossen. Seien L_1 und L_2 regulär. Wir geben eine Konstruktion eines NEA an, der $merge(L_1, L_2)$ beschreibt. Mit Hilfe der Potenzmengenkonstruktion kann man daraus einen DEA konstruieren, der $merge(L_1, L_2)$ beschreibt und somit ist $merge(L_1, L_2)$ nach Definition regulär.
   Da L_1 und L_2 nach Annahme regulär sind, existieren DEAs $M_1 = (Q_1, \Sigma_1, s_1, F_1, \Delta_1)$ und $M_2 = (Q_2, \Sigma_2, s_2, F_2, \Delta_2)$, die L bzw. L' beschreiben. Wir konstruieren einen NEA $M_3 = (Q_3, \Sigma_3, s_3, F_3, \Delta_3)$, der $merge(L, L')$ beschreibt, wie folgt:

   - $Q_3 := Q_1 \times Q_2$
   - $\Sigma_3 := \Sigma_1 \cup \Sigma_2$
   - $s_3 := (s_1, s_2)$
   - $F_3 := \{(p, q) \mid p \in F_1 \ \wedge \ q \in F_2\}$
   - $\Delta_3 := \{((p,q), \sigma, (p', q)) \mid (p, \sigma, p') \in \Delta_1\} \ \cup \ \{((p,q), \sigma, (p, q')) \mid (q, \sigma, q') \in \Delta_2\}$

   Der NEA besteht aus $|Q_1|\cdot|Q_2|$ Zuständen, die alle Kombinationsmöglichkeiten der Zustände aus M_1 und M_2 repräsentieren. Für jeden Zustand (p, q) behalten wir alle Transitionen von p in M_1 sowie alle Transitionen von q in M_2 bei (entsprechend in die zugehörigen Zustände aus $Q_1 \times Q_2$, wobei der Zustand des Automaten, dessen Transition zur Zeit nicht betrachtet wird, unverändert bleibt). Hierbei tritt Nichtdeterminismus auf, wenn $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ und zwei Wörter $w \in L_1$ und $v \in L_2$ ein gemeinsames Zeichen beinhalten. Der Startzustand ist offensichtlich das Tupel der beiden Startzustände von M_1 und M_2. Akzeptierende Zustände sind all solche, die sich aus akzeptierenden Zuständen aus M_1 und M_2 zusammensetzen. So wird sichergestellt, dass sich ein Wort, das akzeptiert wird, aus zwei Wörtern aus L_1 und L_2 zusammensetzt.

3. Nein, die NKA Sprachen sind nicht unter Merge abgeschlossen.
   Beweis: Die Sprachen $L := \{a^n b^n \mid n \in \mathbb{N}\}$ und $L' := \{c^n d^n \mid n \in \mathbb{N}\}$ sind NKA Sprachen. Sei nun N das N aus dem Pumping-Lemma für NKA-Sprachen. Die Sprache $merge(L, L')$ enthält offensichtlich nur Worte w für die gilt: $\#_a(w) = \#_b(w) \wedge \#_c(w) = \#_d(w)$. Das Wort $z := a^N c^N b^N d^N \in merge(L, L')$ ist also ein Wort dieser Sprache. Nun gilt jedoch für jede Unterteilung $z = uvwxy$ mit $|vwx| \leq N$ und $|vw| > 0$, dass uvw nicht sowohl das Zeichen a, als auch das Zeichen b enthalten kann. Ebenso kann es nicht sowohl das Zeichen c, als auch das Zeichen d enthalten. Da für i=0 $uv^iwx^iy \neq uvwxy$ hat sich mindestens die Anzahl eines Zeichens im Wort geändert. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass es sich dabei um a gehandelt hat. Da b nicht in vwx enthalten ist, ist nun $\#_a(w) \neq \#_b(w)$ und $uv^iwx^iy \notin merge(L, L')$. Daher ist $merge(L, L')$ nach dem Pumping-Lemma für NKAs keine NKA-Sprache.

4. Ist L eine NKA-Sprache und L' eine reguläre Sprache, so ist $merge(L, L')$ eine NKA-Sprache. Dies folgt, da wir beim einlesen eines Wortes nichtdeterministisch entscheiden können, ob der gelesene Buchstabe aus L oder aus L' stammte. L kann mit einem NKA mit nur einem Zustand erkannt werden der bei leerem Keller akzeptiert und zur Erkennung von L' benötigt man keinen Keller. Daher können wir die Zustände eines NKAs nutzen um den Automaten für L' zu simulieren und der NKA kann trotzdem noch L erkennen, da die Zustände unwichtig sind.
   Sei also $M_{L'} := (Q, \Sigma_1, s_0, F, \Delta_1)$ und $M_L := (\Sigma_2, \Gamma, L, \{s\}, s, \emptyset, \Delta_2)$. Dann erkennt der NKA

   $$M := (\Sigma_1 \cup \Sigma_2, \Gamma, L, Q \cup \{F_{end}\}, s_0, \{F_{end}\}, \Delta)$$

   die Sprache, wobei

   $$\begin{aligned}\Delta := &\{(q, \gamma, \sigma, W, q') \,|\, (\sigma \in \Sigma_2 \wedge q = q' \wedge (s, \gamma, \sigma, W, s) \in \Delta_2) \vee (\sigma \in \Sigma_1 \wedge (q, \sigma, q') \in \Delta_1 \wedge W = \gamma)\} \\ &\cup \{(q, \varepsilon, \varepsilon, \varepsilon, F_{end} \,|\, q \in F)\}\end{aligned}$$

   Die letzte Transition und der Endzustand F_{end}, verhindern, dass der Automat akzeptiert, obwohl, der Keller noch nicht leer wahr.