Nachname:________________________ Mat-Nr:________________________

Aufgabe 1: Multiple Choice **(7 P.)**

Kreuzen Sie die richtige Antwort an. Jede richtige Antwort ist einen Punkt wert, für jede falsche Antwort wird ein Punkt abgezogen. Eine unbeantwortete Frage ist 0 Punkte wert. Die minimale Anzahl der zu erreichenden Punkte ist 0.

(a) Für eine Sprache L gelte: $\exists k \in \mathbb{N}$ so daß für jedes $w \in L$ mit $|w| > k$ gilt: es gibt eine Zerlegung $w = xyz$ mit $\mid xy \mid \leq k$ und $\mid y \mid > 0$ und $\forall i$ gilt $xy^iz \in L$. Daraus folgt, daß L regulär ist. □**Ja** □**Nein**

Antwort: Nein. Das ist das Pumpinglemma verkehrtherum, aber die Aussage gilt dann nicht.

(b) Ist G eine kontextfreie Grammatik, so gilt: $\mathcal{L}(G)$ ist nicht regulär. □**Ja** □**Nein**

Antwort: Nein. Rechts- und linklineare Grammatiken sind auch kontextfrei, d.h. i.A erzeugen nicht alle kontextfreien Grammatiken nicht-reguläre Sprachen.

(c) Ist G eine kontextfreie Grammatik, so gilt: Gibt es einen regulären Ausdruck α mit $\mathcal{L}(\alpha) = \mathcal{L}(G)$, so gibt es eine linkslineare Grammatik G' mit $\mathcal{L}(G) = \mathcal{L}(G')$. □**Ja** □**Nein**

Antwort: Jawohl, die Linkslinearen Grammatiken erkennen genau die regulären Sprachen, und durch α ist $\mathcal{L}(G)$ regulär.

(d) Ist G eine kontextfreie Grammatik, so gilt: Gibt es keinen DEA M mit $\mathcal{L}(M) = \mathcal{L}(G)$, so ist G nicht linkslinear.

□**Ja** □**Nein**

Antwort: Korrekt, denn wäre G linkslinear, wäre $\mathcal{L}(G)$ regulär, also gäbe es einen DEA M mit $\mathcal{L}(M) = \mathcal{L}(G)$.

(e) Ist L eine reguläre, ε-freie Sprache und G eine Grammatik mit $L = \mathcal{L}(G)$, dann läßt sich G im Allgemeinen nicht mit dem aus der Vorlesung bekannten Algorithmus in Greibach-Normalform bringen.

□**Ja** □**Nein**

Antwort: Stimmt. Es gibt Typ-x-Grammatiken mit $x = 0, 1$, die nicht von Typ $x = 2, 3$ sind, die aber dennoch reguläre Sprachen beschreiben. Für diese Grammatiken ist die Greibach-Normalform aber nicht definiert, und der Algorithmus funktioniert darum nicht.

(f) Jede nicht-deterministische, kontextfreie Sprache ist immer inhärent mehrdeutig.

□**Ja** □**Nein**

Antwort: Nein. Wes. Mehrdeutigkeit und Nichtdeterminismus haben nichts miteinander zu tun. Betrachte z.B. $\Sigma = \{a_1, \ldots, a_n\}$ und $L = \{ww^R \mid w \in \Sigma^*\}$. Diese Sprache ist nichtdeterministisch, aber $G = \{S \rightarrow a_iSa_i|\varepsilon, 1 \leq i \leq n\}$, ist eine eindeutige Grammatik mit $\mathcal{L}(G) = L$.

(g) Kann man für eine gegebene kontextfreie Grammatik G, mit $\mathcal{L}(G)$ deterministisch, und eine durch einen endlichen Automaten gegebene reguläre Sprache L_1 entscheiden, ob $\mathcal{L}(G) = L_1$ ist?

□**Ja** □**Nein**

Antwort: Ja. Das Leerheitsproblem für deterministisce kontextfreie Sprachen ist entscheidbar. Wenn $\mathcal{L}(G) \cap \overline{L_1} = \emptyset = \overline{L(G)} \cap L_1$, dann ist $\mathcal{L}(G) = L_1$.

Nachname:____________________ Mat-Nr:____________________

Aufgabe 2: Pumpinglemma für Kontextfreie Sprachen **(3+5 P.)**

(a) Zitieren Sie das *Pumpinglemma für Kontextfreie Sprachen* (nicht den Beweis).

(b) Zeigen Sie mit Hilfe des Pumpinglemmas für kontextfreie Sprachen, daß die Sprache $L_2 = \{a^{n^2} b^n\} \subseteq \{a, b\}^*$ nicht kontextfrei ist.

Lösung

(a) Sei L eine cfl. Dann gibt es eine natürliche Zahl k sodaß sich jedes wort z mit $|z| \geq k$ (**0.5 Pkt.**) wie folgt zerlegen lässt: $z = uvwxy$ (**0.5 Pkt.**) und

- (i) $|vx| \geq 1$ (**0.5 Pkt.**)
- (ii) $|vwx| \leq n$ (**0.5 Pkt.**)
- (iii) $uv^iwx^iy \in L$ für alle $i \geq 0$ (**1 Pkt.**).

(b) (klug gewähltes Wort: **1 Punkt**) Wähle z.B. $z = a^{k^2} b^k$. Mögliche Faktorisierungen: $v = a \cdots a, x = a \cdots a$ oder $v = a \cdots a, x = b \cdots b$ oder $v = b \cdots b, x = b \cdots b$. (Jede angegebene Faktorisierung je **einen halben Punkt**, maximal **2 Punkte**). Einzige sinnvolle Faktorisierung ist $v = a \cdots a, x = b \cdots b$. (Beweis, daß sich diese Faktorisierung nicht pumpen läßt: 2 Punkte)

$\square$

Nachname:____________________ Mat-Nr:____________________

Aufgabe 3: Ardens Lemma **(5 P.)**

Sei Σ ein Alphabet. Seien $A, B, K, L_1, L_2 \subseteq \Sigma^*$. Eine Verallgemeinerung von Ardens Lemma besagt: Wenn

$$AK \cup B \subseteq K, \text{dann ist} A^*B \subseteq K,$$

bzw. wenn

$$KA \cup B \subseteq K,\ \text{dann ist}\ BA^* \subseteq K.$$

Beweisen Sie mit diesem Satz die folgenden Aussage:

$$(L_1 \cup L_2)^* \subseteq L_1^*(L_2L_1^*)^*.$$

Hinweis: Wählen Sie geeignete A, B und K und beweisen Sie, daß die Vorraussetzung des verallgemeinerten Lemma von Arden erfüllt ist.

Lösung

(Korrektes Patternmatching ($K = L_1^*(L_2L_1^*)^*,\ A = L1 + L2,\ B = \{\varepsilon\}$) **2 Punkte** Beweis dass vorraussetzung Lemma stimmt **3 Punkte**)
Setze $K = L_1^*(L_2L_1^*)^*$, $A = L_1 \cup L_2$ und $B = \{\varepsilon\}$, beweise, daß $\{\varepsilon\} \cup (L_1 \cup L_2)K \subseteq K$.

$$\begin{aligned} L_1K &\subseteq K \\ L_2K &\subseteq K \quad \text{also} \\ (L_1 \cup L_2)K &\subseteq K \end{aligned}$$

Ardens Lemma liefert den Rest.

□

Nachname:________________________ Mat-Nr:________________________

Aufgabe 4: Getränkeautomaten **(4+10 P.)**

In den folgenden beiden Aufgaben sollen Sie das Verhalten eines Getränkeautomaten m.H. von endlichen bzw. Kellerautomaten modellieren. Das Verhalten wird in Prosa beschrieben, wobei reale Aktionen am Getränkeautomaten durch die Symbole des Alphabets abstrakt beschrieben werden sollen, z.B. Münzeneinwerfen durch *M?*[1], Ausgabe von Kaffee durch *Kaffee*![1], usw. Die Zuordnung von Aktionen zu Symbolen wird im Text in Klammern verdeutlicht.
Hinweis: Die Prosa-Spezifikation der Getränkeautomaten ist notwendigerweise unvollständig. Falls Ihr Automat mehr kann als angegeben, ist das in Ordnung, solange es der Spezifikation nicht widerspricht.

Kommentieren Sie Ihre Lösungen.

(a) Ein Getränkeautomat mit Münzeinwurf hat die folgenden Eigenschaften:

- Für eine Münze (*M?*) bekommt man Tee.
- Für zwei Münzen (*M?*) bekommt man Kaffee.
- Der Automat kann nur zwei Münzen aufnehmen, danach ist der Münzschacht geschlossen.
- Jede Münze kann nur genau einmal für ein Getränk verwendet werden, und ist nach Benutzung aus dem Getränkeautomaten verschwunden.
- Es gibt zwei Knöpfe, um Kaffee (K_K?) oder Tee (K_T?) anzufordern. Befindet sich genug Geld im Automaten, wird auf Knopfdruck das entsprechende Getränk (*Tee*! bzw. *Kaffee*!) ausgegeben.
- Sind die Münzen im Automaten aufgebraucht, geht der Automat in den Startzustand zurück.

Modellieren Sie den Getränkeautomaten als einen (nicht notwendigerweise vollständigen) deterministischen minimalen endlichen Automaten, so daß die erkannte Sprache die Wörter aller vollständigen korrekten Verkaufstransaktionen ist. Benutzen Sie das Alphabet

$$\Sigma = \{M?, K_K?, K_T?, \mathit{Tee}!, \mathit{Kaffee}!\}.$$

[1] ! und ? werden benutzt um zwischen Eingaben und Ausgaben des Getränkeautomaten zu unterscheiden. Sonst haben ! und ? keine weitere Bedeutung.

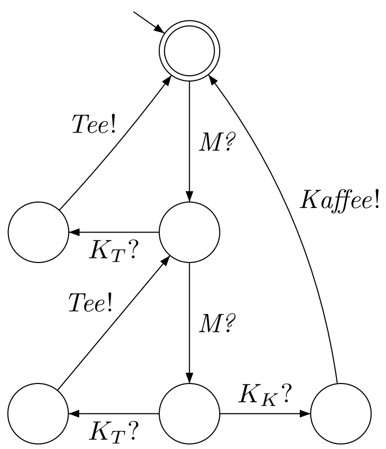
(b) Wir betrachten nun einen ähnlichen Getränkeautomaten, mit den folgenden Änderungen:

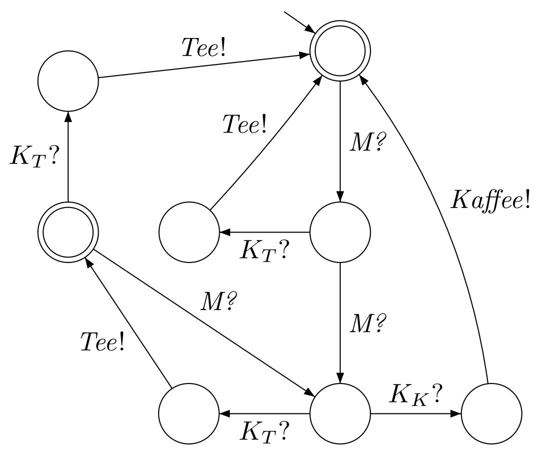
- Der Automat kann nun eine beliebige Anzahl von Münzen (*M?*) aufnehmen. Je nach Knopfdruck und Münzvorrat wird das entsprechende Getränk (*Tee*! bzw. *Kaffee*!) ausgegeben.
- Wenn der Knopf K_ε? einmal gedrückt wird, werden alle unverbrauchten Münzen, eine nach der anderen, wieder zurückgegeben (*M!*). Wärend der Ausgabe von Münzen nimmt der Automat keine weiter Münzen an.

Definieren Sie einen Kellerautomaten über das Eingabealphabet $\Sigma \cup \{K_\varepsilon?, M!\}$ und einem Stackalphabet Γ Ihrer Wahl, der das beschriebene Verhalten modelliert. Die Sprache, die erkannt werden soll, ist wieder die Menge aller vollständigen korrekten Verkaufstransaktionen inklusive Rückgabe überzähliger Münzen.

Lösung

Das problem an dieser Aufgabe ist, daß die Spezifikation nicht eindeutig ist. Somit gibt es verschiedene Akzeptanzmöglichkeiten: ein wort wird akzeptiert, wenn keine Münze mehr im Automat ist. Oder es wird akzeptiert, wenn ein Getränk ausgegeben wurde. Insofern ist alles, was ungefähr korrekt aussieht, bepunktet worden. Punktabzug z.B. wenn kein Tee angefordert werden kann, wenn sich nur eine Münze im Automat befindet.
Mögliche Lösungen:

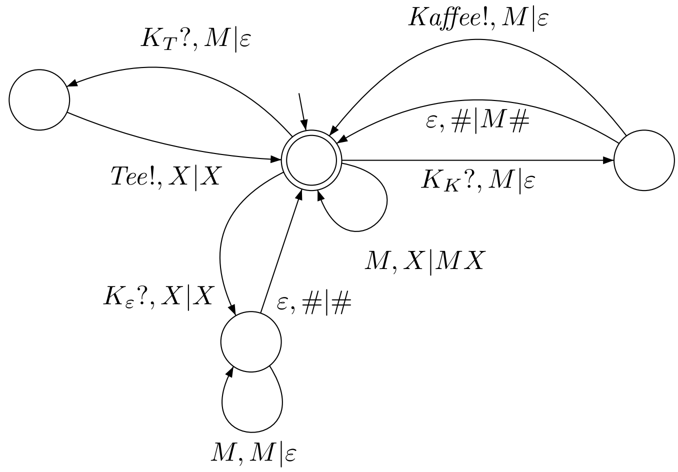




(a) oder

(b)
- korrekter kellerautomat **2 Pkt**
- kan bel Anzahl Munzen aufnehmen **1 Pkt**
- Gibt alle Muenzen nacheinander zurück auf keps **2 Pkt**
- Check für 2 Münzen bei Kaffee korrekt **2 Pkt**
- korrektes verhalten bei kaffee, wenn nur 1 Münze **2 Pkt**
- korrektes verhalten bei tee **1 Pkt**

Mögliche Lösung: Offensichtlich kann man beliebig viele Münzen einwerfen, und das wird schon als korrektes Wort betrachtet. Auch das leere wort beschreibt korrektes verhalten.



Alternativ kann man als korrekte sequenz betrachten, was mit *Kaffee*!, *Tee*!, oder *M!* endet. Dafür muss man also dann einen extra Zustand einführen, der als Endzustand fungiert.

$\square$

Aufgabe 5: Verkehrte Sprachen **(5 P.)**

Sei Σ ein Alphabet, und $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ ein NEA. Ist $L \subseteq \Sigma^*$, so wird Sprache L^R definiert durch $L^R = \{x_n x_{n-1} \cdots x_0 \mid x_0 \cdots x_{n-1} x_n \in L\}$. Sei $L = \mathcal{L}(\mathcal{A})$. Wir definieren den (nicht notwendigerweise vollständigen) NEA $\mathcal{A}^R = (Q', \Sigma, \delta', Q'_0, F')$, wobei

- $Q' = Q$
- $F' = Q_0$
- $Q'_0 = F$
- $\delta'(q, a) = \{q' \mid q \in \delta(q', a)\}$

Also alle Pfeile werden umgedreht, und Start- und Endzustandsmengen werden vertauscht.
Um zu beweisen Sie, daß $\mathcal{L}(\mathcal{A}^R) = L^R$, muss man zwei Richtungen zeigen:

(a) $w \in L \implies w^R \in \mathcal{L}(\mathcal{A}^R)$

(b) $w \in \mathcal{L}(\mathcal{A}^R) \implies w^R \in L$

Skizzieren[2] Sie einen Beweis für eine der beiden Richtungen.

Lösung

Wer die Testklausur geschrieben hat war klar im Vorteil: diese Aufgabe ist nur eine Variation von Aufgabe 6 der Testklausur.
Wir haben zwei Richtungen zu zeigen: $\mathcal{L}(\mathcal{A}^R) \subseteq L^R$ und $L^R \subseteq \mathcal{L}(\mathcal{A}^R)$.

$L^R \subseteq \mathcal{L}(\mathcal{A}^R)$: Falls $w = x_0 x_1 x_2 \ldots x_n \in L$ (also $w^R \in L^R$), $x_i \in \Sigma$ für $i = 0, \ldots n$, dann gibt es einen Pfad $q_0 q_1 q_2 \ldots q_{n+1}$ durch $\mathcal{A}$ mit $q_{i+1} \in \delta(q_i, x_i)$, und $q_0 \in Q_0$, und $q_{n+1} \in F$. Nach der Definition von $\mathcal{A}'$ gilt ausserdem: $q_i \in \delta'(q_{i+1}, x_i)$, für $i = 0, \ldots n$. Also gibt es für w^R einen Pfad $q_{n+1} q_n \cdots q_0$ in $\mathcal{A}^R$ mit $q_{n+1} \in Q'_0$ und $q_0 \in F$. Also ist $w^R \in \mathcal{L}(\mathcal{A}^R)$.

$\mathcal{L}(\mathcal{A}^R) \subseteq L^R$: Der Beweis geht genauso.

- w akzeptiert dann gibt es pfad in automat **2 Pkt**
- umgekehrter pfad in A^r mit start und endzustand vertauscht **2 Pkt**
- also $w^r \in L(Ar)$ **1 Pkt**

$\square$

[2] Skizzieren heisst hier, daß Sie eine Beweisidee angeben, aber keinesfalls den Beweis bis zum Schluss durchführen.

Aufgabe 6: Restklassen-Automaten **(4+6+11 P.)**

Sei $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. Wir interpretieren Wörter $w \in \Sigma^*$ als Binärzahlen, und berechnen Dezimalzahlen aus w wie folgt:

- $[w] = 0$ falls $w = \varepsilon$
- $[w] = 0$ falls $w = \mathbf{0}$
- $[w] = 1$ falls $w = \mathbf{1}$
- $[x \cdot w] = [x] + 2[w]$ falls $x \in \{\mathbf{0}, \mathbf{1}\}$ und $w \in \Sigma^+$.

Beachten Sie, daß in dieser Interpretation die Ziffern in umgekehrter Reihenfolge zur gewöhnliche Lesart betrachtet werden.

In den folgenden Aufgaben sollen Sie verschiedene endliche Automaten konstruieren. Wenden Sie dazu, wo nötig, die Konstruktionen Ihrer Wahl an, wie z.B. für $\cup, \cap, \bar{\cdot}$, etc. Auch die Konstruktion der vorherigen Aufgabe für L^R kann sich als nützlich erweisen.

Erläutern Sie Ihre Konstruktionsschritte.

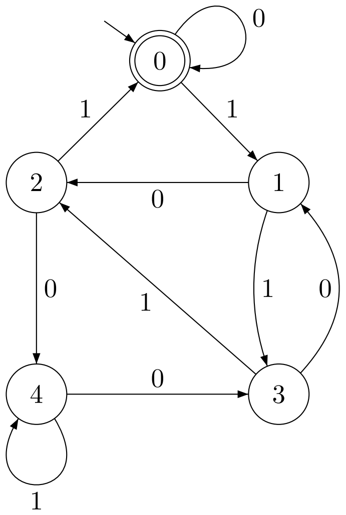
(a) Konstruieren Sie einen deterministischen, vollständigen, minimalen endlichen Automaten $\mathcal{A}_1$, der alle Wörter $w \in \Sigma^*$ erkennt, so daß $[w]$ durch 5 teilbar ist.

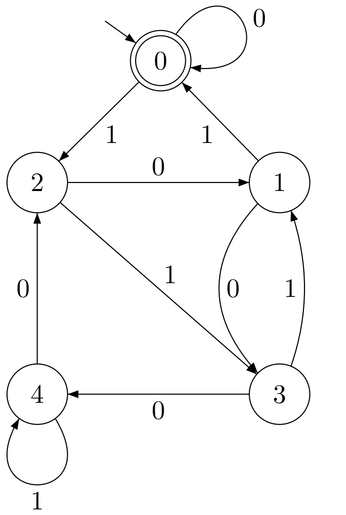
(b) Konstruieren Sie einen deterministischen, vollständigen, minimalen endlichen Automaten $\mathcal{A}_2$, der alle Wörter $w \in \Sigma^*$ erkennt, so daß $[w]$ durch 4 teilbar ist.

(c) Konstruieren Sie mit Hilfe von $\mathcal{A}_1$ und $\mathcal{A}_2$ einen deterministischen, vollständigen, minimalen endlichen Automaten $\mathcal{A}_3$, der alle Wörter $w \in \Sigma^*$ erkennt, so daß $[w]$ durch 5 oder durch 4 teilbar ist.

Lösung

(a) Links ist der Automat für die *most-significnt-bit-first*-Interpretation (also der herkömlichen Interpretation) von w. Diese ist einfacher zu konstruieren als der gewünschte Automat. Den Automaten $\mathcal{A}_1$ erhält man dadurch, daß man alle Pfeile umdreht.



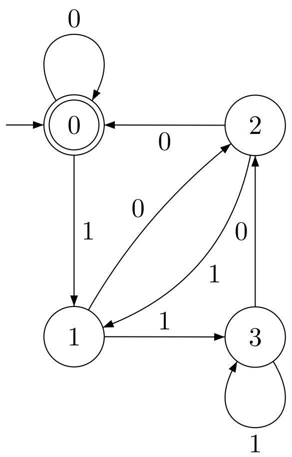


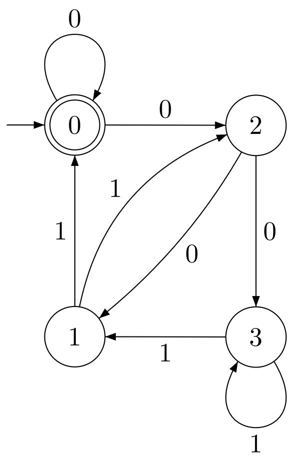
$\mathcal{A}_1^R$ $\mathcal{A}_1$

Glücklicherweise ist Automat $\mathcal{A}_1$ deterministisch und minimal, also sind wir schon fertig!

- Konstruktion von RKA gewohnt **2 Pkt**
- umdrehen **2 Pkt**

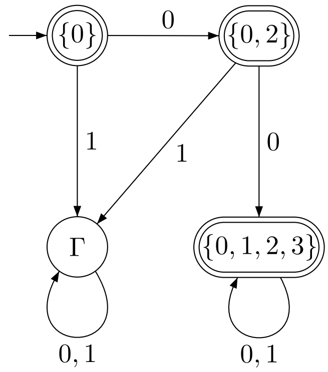
(b) Wir wenden denselben Trick an, also wir konstruieren zuerst $\mathcal{A}_2^R$.





$\mathcal{A}_2^R$ $\mathcal{A}_2'$

Die Konstruktion von $\mathcal{A}_2'$ erfolgt wieder durch Umdrehen der Pfeile. $\mathcal{A}_2'$ ist offensichtlich nicht deterministisch, und nicht vollständig. Darum müssen wir also den Automaten determinisieren, vervollständigen und minimieren (man hätte auch erst $\mathcal{A}_2^R$ minimieren können, 1 und 3 sind äquivalent):

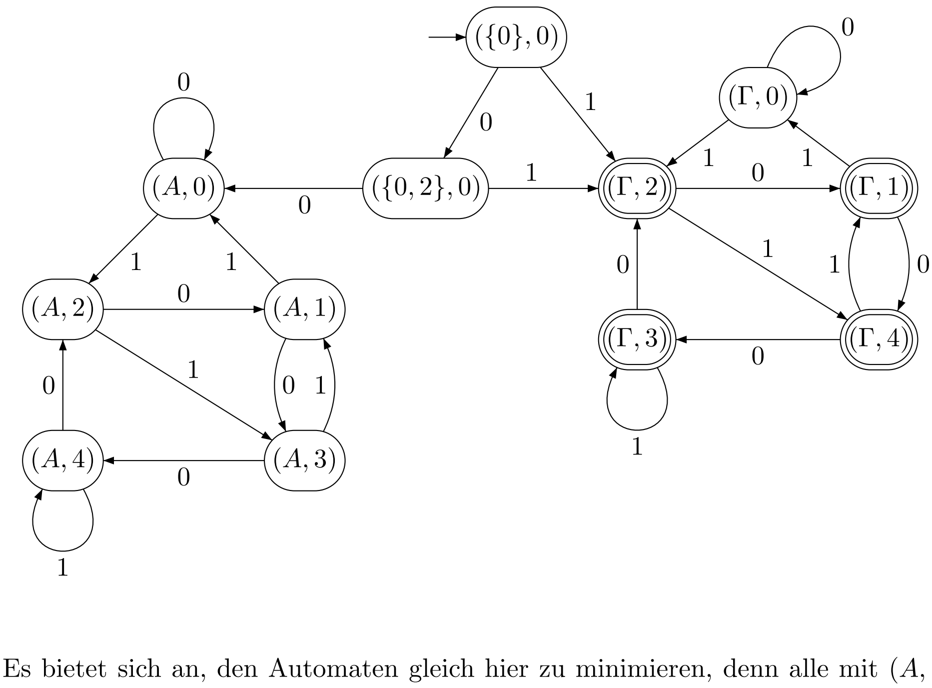


$\mathcal{A}_2$

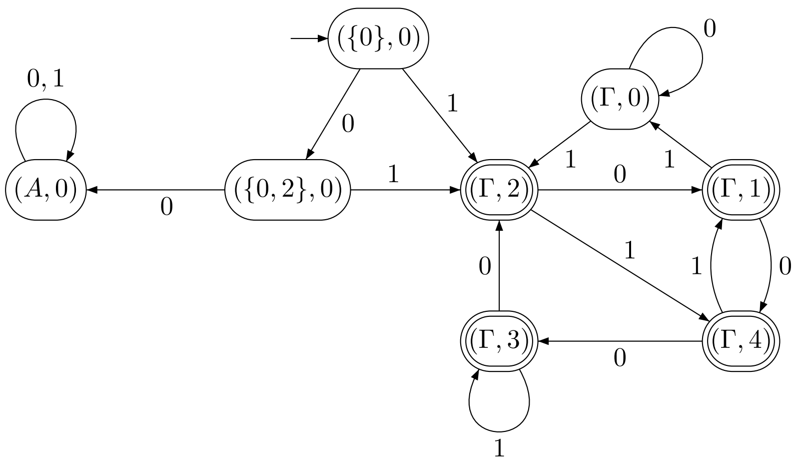
Offensichtlich ist ein Wort w nur genau dann durch 4 teilbar, wenn $[w] = 0$ oder $w = 00w'$. (Hätte man auch gleich drauf kommen können).

- Konstruktion von RKA gewohnt 2
- umdrehen 2
- determinisierung 2
- Abkuerzung (alle wörter die aus 0en bestehen oder mit 00 anfangen) volle Punktzahl

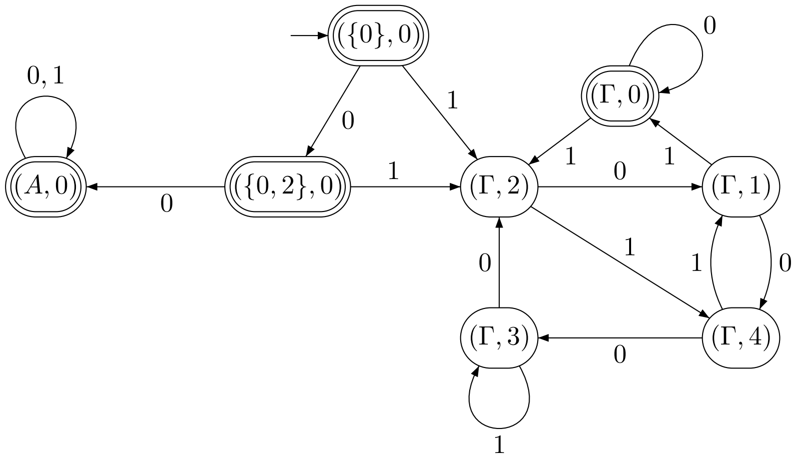
(c) Wir nutzen aus, daß $A \cup B = \overline{\overline{A} \cap \overline{B}}$ (wir können natürlich auch die herkömliche Methode verwenden, aber dann müssen wir noch determinisieren). Wir bilden also erst einen Automaten $\overline{\mathcal{A}_1} \cap \overline{\mathcal{A}_2}$ und invertieren diesen dann. Wir setzen $A = \{0, 1, 2, 3\}$.



Es bietet sich an, den Automaten gleich hier zu minimieren, denn alle mit $(A, \cdot)$ markierten Zustände sind äquivalent:



Der Automat ist nach wie vor deterministisch und vollständig. Der invertierte Automat is somit unser gesuchter Automat $\mathcal{A}_3$.



- vereinigungskonstruktion **2 Pkt**
- determinisierung **3 Pkt**
- minimierung **3 Pkt**
- korrekt **3 Pkt**

______________________________ □