Lektion 12

In dieser Lektion werden zwei wichtige Eigenschaften deterministischer endlicher Automaten gezeigt: Zu jedem DEA kann ein äquivalenten Automaten mit minimaler Zustandszahl bestimmt werden und es kann durch ein algorithmisches Verfahren geprüft werden, ob zwei Automaten äquivalent sind, d.h. die gleiche Sprache akzeptieren.

7.5 Minimalautomaten

Wie in den vorangehenden Beispielen zu sehen war, kann es für die gleiche Sprache unterschiedlich komplex aufgebaute Automaten geben. In diesem Abschnitt geht es um die Frage, ob man Automaten systematisch zu äquivalenten, möglichst einfachen Automaten vereinfachen kann. "Möglichst einfach" heißt dabei, mit einer möglichst kleinen Zahl an Zuständen.

Definition 7.14 - Minimalautomat

Ein DEA A heißt **Minimalautomat** für eine Sprache L, falls der Automat die Sprache L akzeptiert, d.h. $L = L(A)$, und es keinen anderen Automaten A' gibt, der

- äquivalent zu A ist (d.h. auch die gleiche Sprache L akzeptiert) und
- weniger Zustände als A hat.

Es kann also auch unterschiedliche Minimalautomaten für eine Sprache geben, die dann aber alle die gleich Zahl an Zuständen haben. Z.B. können einfach die Zustände anders benannt sein.

Im Folgenden lernen Sie ein Verfahren kennen, wie aus einem gegebenen DEA ein Minimalautomat berechnet werden kann. Dieses Verfahren hat auch praktische Anwendungen, z.B. im Compilerbau in sog. Scanner-Generatoren, die für die lexikalischen Analyse den Code für einen DEA generieren. Durch eine Minimierung des DEA wird der Scanner dann zwar nicht schneller, aber die Codegröße wird reduziert.

Ein wichtiger Schritt für die Vereinfachung wird sein, Zustände im Automat zu erkennen, die das gleiche bewirken (werden dann als *äquivalente Zustände* bezeichnet). Hat man zueinander äquivalente Zustände, können sie zu einem Zustand zusammengelegt werden.

7.5.1 Äquivalenz von Zuständen

Definition 7.15 - Äquivalente und unterscheidbare Zustände

- Zwei **Zustände** z_1 und z_2 eines DEA A=(Z, Σ, δ, z_0, E) heißen **äquivalent**, notiert

  $$\boldsymbol{z_1 \equiv z_2},$$

  wenn für alle Wörter $w \in \Sigma^*$ gilt:

  $z_1 \xrightarrow{w} z'$ für irgend einen Endzustand $z' \in E$

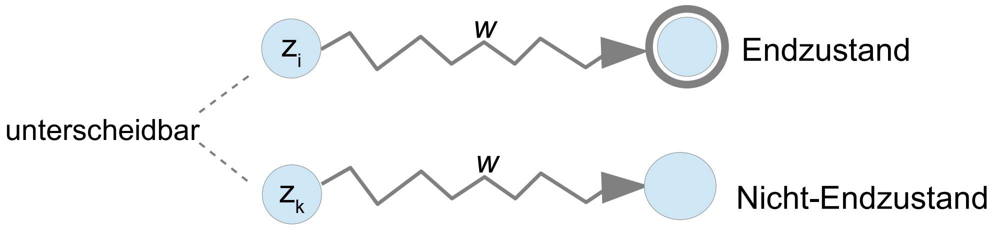
  genau dann, wenn

  $z_2 \xrightarrow{w} z''$ für irgend einen Endzustand $z'' \in E$

- Zwei **Zustände**, die *nicht äquivalent* sind, heißen **unterscheidbar.**

Erläuterung

- Zwei Zustände z_1 und z_2 sind **äquivalent**, wenn von beiden Zuständen aus mit genau den gleichen Wörtern Endzustände erreichbar sind (es müssen aber nicht die identischen Endzustände sein!). Hat man zwei äquivalente Zustände dann ist das zukünftige Verhalten bei der Analyse eines Worts für beide Zustände gleich: Es ist egal, ob man von dem einen oder anderen Zustand ausgeht, man wird genau mit den gleichen (Rest-)Wörtern Endzustände erreichen.
- Zwei Zustände z_1 und z_2 sind somit **unterscheidbar**, wenn es mindestens ein Wort *w* gibt, so dass von z_1 aus mit *w* ein Endzustand erreicht werden kann, aber von z_2 aus nicht (bzw. umgekehrt).

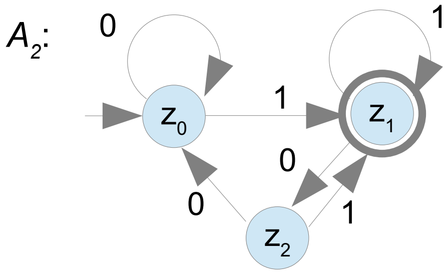


Wir sagen in diesem Fall, dass die Zustände z_i und z_k *durch das Wort w unterscheidbar* sind.

Sind zwei Zustände unterscheidbar, macht es also einen Unterschied, ob man bei der Analyse eines Worts in den einen oder in den anderen Zustand gelangt.

Beispiel 7.16 - Äquivalente und unterscheidbare Zustände

Welche Zustände sind bei folgendem DEA A_2 zueinander äquivalent bzw. voneinander unterscheidbar?



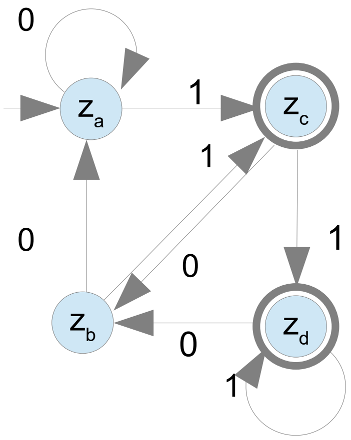
- ***z_0* und *z_1*** sind **unterscheidbar**: Betrachte Wort ε. Man bleibt mit dem Wort in z_0 bzw. in z_1. z_1 ist Endzustand, z_0 aber nicht. Also sind z_0 und z_1 durch das Wort ε unterscheidbar, d.h. nicht äquivalent.
- ***z_1* und *z_2*** sind **unterscheidbar**: Gleiche Argumentation gilt hier auch: z_1 ist Endzustand, z_2 aber nicht, also sind z_1 und z_2 durch das Wort ε unterscheidbar, also nicht äquivalent.
- ***z_0* und *z_2*** sind **äquivalent**: Wir betrachten alle möglichen Wörter $w \in \{0,1\}^*$. Es gibt folgende drei Fälle: Es kann das leere Wort ε sein, das Wort kann mit 0 anfangen oder es kann mit 1 anfangen.
  - Fall $w=\varepsilon$: Beide Zustände sind keine Endzustände, somit durch ε nicht unterscheidbar.
  - Fall $w=0w'$: Nach Transition mit 0 befindet sich Automat A_2 in Zustand z_0, egal ob man bei z_0 oder z_2 gestartet ist. Danach geht es für w' somit gleich weiter, somit sind beide Zustände durch keine Wort $0w'$ unterscheidbar.
  - Fall $w=1w'$: nach Transition mit 1 befindet sich Automat A_2 im Zustand z_1, egal ob man bei z_0 oder z_2 gestartet ist. Danach geht es für w' folglich gleich weiter. Somit sind beide Zustände durch keine Wort $1w'$ unterscheidbar.

  Die Zustände z_0 und z_2 sind durch kein Wort $w \in \{0,1\}^*$ unterscheidbar, also sind sie äquivalent, d.h. $z_0 \equiv z_2$.

Aufgabe 7.17 - Äquivalente/unterscheidbare Zustände

Überlegen Sie sich in analoger Weise, welche Zustände bei folgendem DEA A_3 zueinander äquivalent sind bzw. welche voneinander unterscheidbar sind.

A_3:



Bei den vorangehenden Beispielen/Aufgaben haben wir durch eine ad hoc-Vorgehensweise, d.h. durch genaues Hinschauen und intelligentes Überlegen, für das spezifische Beispiel gezeigt, dass Zustände äquivalent bzw. unterscheidbar sind. Gibt es auch ein systematisches Verfahren, das bei jedem beliebigen Automaten angewendet werden kann und das berechnet, welche Zustände äquivalent sind? Die Antwort ist "ja".

Zu zeigen, dass zwei Zustände äquivalent sind, ist allerdings schwieriger als zu zeigen, dass zwei Zustände unterscheidbar sind. Für die Unterscheidbarkeit muss nur ein Wort als "Zeuge" dafür gefunden werden, dass die Zustände sich unterschiedlich verhalten, während für die Äquivalenz gezeigt werden muss, dass sich beide Zustände für alle unendlich viele mögliche Wörter gleichartig verhalten. Das Verfahren zur Bestimmung äquivalenter Zustände nimmt deshalb den "Umweg", dass zunächst die unterscheidbaren Zustände bestimmt werden. Die Zustände, für die sich ergibt, dass sie nicht unterscheidbar sind, bleiben am Ende als äquivalente Zustände übrig.

Beobachtung 7.18 - Unterscheidbare Zustände

Woran kann man erkennen, dass zwei Zustände unterscheidbar sind?

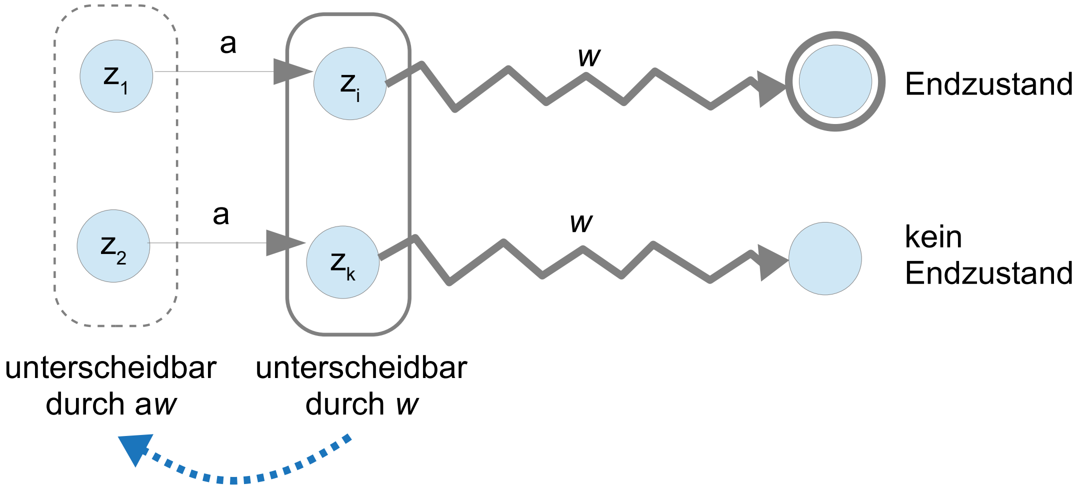
(1) Ist z_i **Endzustand** und z_k **kein Endzustand**, dann sind beide Zustände durch das leere Wort ε unterscheidbar.

(2) **"Propagierungsregel":** Ist schon bekannt, dass zwei Zustände z_i und z_k unterscheidbar sind (durch irgend ein Wort w) und gibt es ein Zeichen a, so dass für zwei weitere Zustände z_1 und z_2 gilt

$$z_1 \xrightarrow{a} z_i$$

$$z_2 \xrightarrow{a} z_k \quad ,$$

dann sind auch z_1 und z_2 unterscheidbar (durch das Wort aw).
Dies ist in folgender Abbildung veranschaulicht:



7.5.2 Berechnung äquivalenter Zustände

Die Bestimmung unterscheidbarer Zustände basiert auf diesen zwei Beobachtungen: Man beginnt gemäß Punkt (1) mit den Paaren Endzustand/Nicht-Endzustand als unterscheidbare Zustände. Dann wendet man iterativ immer wieder die Propagierungsregel nach (2) an, um abzuleiten, dass weitere Zustandspaare auch unterscheidbar sind.

Algorithmus 7.19 - Berechnung äquivalenter Zustände

Ausgangspunkt: Wir bilden eine Tabelle in Dreiecksform für alle Paare $\{z_i, z_k\}$ von Zuständen. Jeder Zustand ist zu sich selbst äquivalent, bei allen anderen Paaren ist noch unklar, ob sie unterscheidbar oder äquivalent sind.

z_0	$\equiv$				
z_1		$\equiv$			
z_2			$\equiv$		
z_3				$\equiv$	
z_4					$\equiv$
	z_0	z_1	z_2	z_3	z_4

In diese Tabelle wird eingetragen, ob schon bekannt ist, ob zwei Zustände unterscheidbar oder äquivalent sind.

Eintrag bei $\{z_i, z_k\}$:

$\equiv$	(für i = k): jeder Zustand ist zu sich selbst äquivalent
X	z_i und z_k sind unterscheidbar
leer	es ist noch unbekannt, ob z_i und z_k unterscheidbar sind

Vorgehensweise:

1. Phase (Endzustand/Nicht-Endzustand): Markiere alle Paare $\{z_j, z_k\}$, bei denen entweder z_j oder z_k Endzustand ist, aber nicht beide, mit X als unterscheidbar.

2. Phase (Propagierungsregel anwenden):

Wiederhole für jedes noch nicht markierte Paare $\{z_j, z_k\}$:

Wiederhole für alle Zeichen $a \in \Sigma$:

Falls $z_j \xrightarrow{a} z_j'$,

$z_k \xrightarrow{a} z_k'$

und $\{z_j', z_k'\}$ in Tabelle schon mit X markiert

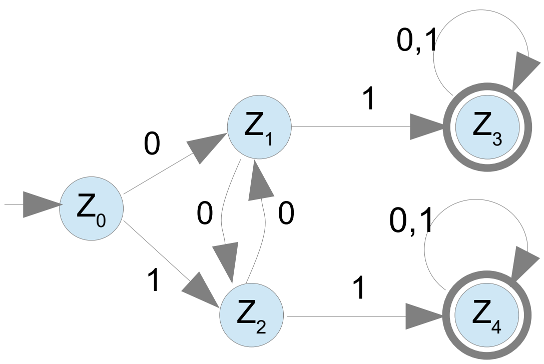
dann markiere auch $\{z_j, z_k\}$ mit X als unterscheidbar.

so lange, bis sich in einem kompletten Durchlauf durch alle leeren Einträge keine neue Markierung mehr ergibt.

Ergebnis: Für alle am Ende nicht markierten Paare $\{z_i, z_j\}$ gilt, dass sie nicht unterscheidbar und somit äquivalent sind, d.h. $z_i \equiv z_j$.

Beispiel 7.20 - Bestimmung äquivalenter Zustände

Folgender DEA ist gegeben:



Berechnung der äquivalenten Zustände nach obigem Verfahren:

1. Phase: Alle Paare **Endzustand/Nicht-Endzustand** mit X als unterscheidbar markieren:

z_0	$\equiv$				
z_1		$\equiv$			
z_2			$\equiv$		
z_3	X	X	X	$\equiv$	
z_4	X	X	X		$\equiv$
	z_0	z_1	z_2	z_3	z_4

2. Phase:

1. Durchlauf: Noch unmarkiert sind die Zustandspaare $\{z_0,z_1\}$, $\{z_0,z_2\}$, $\{z_1,z_2\}$, $\{z_3,z_4\}$. Paar jeweils gemäß Propagierungsregel prüfen.

$\{z_0,z_1\}$:

Betrachte Zeichen 0:

$z_0 \xrightarrow{0} z_1$

$z_1 \xrightarrow{0} z_2$,

$\{z_1,z_2\}$ ist nicht markiert,

$\{z_0,z_1\}$ bleibt unmarkiert

Betrachte Zeichen 1:

$z_0 \xrightarrow{1} z_2$

$z_1 \xrightarrow{1} z_4$,

$\{z_2,z_4\}$ schon markiert,

$\{z_0,z_1\}$ wird auch mit X als unter scheidbar markiert

z_0	≡				
z_1	X	≡			
z_2			≡		
z_3	X	X	X	≡	
z_4	X	X	X		≡
	z_0	z_1	z_2	z_3	z_4

$\{z_0,z_2\}$:

Es wir nun ein Ausschnitt aus der Zustandsübergangstabelle als etwas kompaktere Darstellung verwendet:

	0	1
z_0	z_1	z_2
z_2	z_1	z_4

z_2 und z_4 sind schon unterscheidbar, somit werden auch z_0 und z_2 als unterscheidbar markiert.

z_0	≡				
z_1	X	≡			
z_2	X		≡		
z_3	X	X	X	≡	
z_4	X	X	X		≡
	z_0	z_1	z_2	z_3	z_4

$\{z_1,z_2\}$:

	0	1
z_1	z_2	z_3
z_2	z_1	z_4

Weder $\{z_1,z_2\}$ noch $\{z_3,z_4\}$ sind bisher markiert. Kein neuer Eintrag.

$\{z_3,z_4\}$:

	0	1
z_3	z_3	z_3
z_4	z_3	z_3

Jeweils gleicher Zielzustand z_3, somit keine unterscheidbaren Zielzustände, keine neuer Eintrag.

1. Durchlauf durch alle freien Einträge ist abgeschlossen. Es wurde mindestens eine neue Markierung eingetragen, somit ist ein weiterer Durchlauf nötig:

2. Durchlauf: Noch unmarkiert sind die Zustandskombinationen $\{z_1,z_2\}$, $\{z_3,z_4\}$.

$\{z_1,z_2\}$:

	0	1
z_1	z_2	z_3
z_2	z_1	z_4

Weder $\{z_1,z_2\}$ noch $\{z_3,z_4\}$ sind bisher markiert. Kein neuer Eintrag.

$\{z_3,z_4\}$:

	0	1
z_3	z_3	z_3
z_4	z_3	z_3

Jeweils gleicher Zielzustand z3, somit Zielzustände nicht unterscheidbar, keine neuer Eintrag.

Im 2. Durchlauf haben sich keine neuen Markierungen ergeben. Somit endet das Verfahren:

Endergebnis: Die Paare $\{z_1, z_2\}$ und $\{z_3, z_4\}$ sind noch nicht mit X markiert, d.h. nicht unterscheidbar und somit *äquivalent*:

z_0	$\equiv$				
z_1	X	$\equiv$			
z_2	X	$\equiv$	$\equiv$		
z_3	X	X	X	$\equiv$	
z_4	X	X	X	$\equiv$	$\equiv$
	z_0	z_1	z_2	z_3	z_4

also gilt $z_1 \equiv z_2$, $z_3 \equiv z_4$.

Anmerkungen

- Bei diesem relativ einfachen Beispiel ist bereits nach dem 2. Durchgang die Berechnung der unterscheidbaren Zustände fertig. Bei größeren, komplexeren Automaten können auch mehr als zwei Durchgänge nötig sein. Das Verfahren wird aber immer nach einer endliche Anzahl von Durchgängen enden, da in jedem Durchgang (bis auf den letzten) mindestens ein Eintrag gefüllt wird.
- Es ist klar, dass zwei Zustände unterscheidbar sind, wenn in der Tabelle eine Markierung X eingetragen wurde, da sich aus den Prüfungen immer auch eine

Zeichenkette als Zeuge bestimmen lässt. Es kann umgekehrt auch bewiesen werden, dass im Ablauf des Verfahrens garantiert irgendwann ein X eingetragen wird, wenn zwei Zustände unterscheidbar sind. Somit folgt dann, dass am Ende die freien Einträge genau die nicht unterscheidbaren, somit äquivalenten Zustände angeben.

- Das Ergebnis ist immer eine Äquivalenzrelation. Es können auch mehr als zwei Zustände zueinander äquivalent sein. Hat man drei Zustände z_a, z_b und z_c, für die $z_a \equiv z_b$ und $z_b \equiv z_c$ gilt, dann ergibt sich automatisch auch $z_a \equiv z_c$ (Transitivität).
- Das Berechnungsverfahren ist sehr aufwendig, wenn es von Hand durchgeführt wird. Es läuft aber alles rein "mechanisch" ab, ist auf jeden DEA anwendbar und könnte z.B. auch durch ein Programm implementiert werden.

7.5.3 Minimierungsverfahren für DEAs

Das zuvor vorgestellte Verfahren zur Berechnung äquivalenter Zustände eines DEAs wird nun für die Minimierung von DEAs verwendet. Bei der Minimierung werden folgende zwei Eigenschaften verwendet:

- Zustände, die vom Startzustand aus nicht erreicht werden können, sind überflüssig und können entfernt werden.
- Wenn mehrere Zustände äquivalent sind, können sie zu einem Zustand vereinigt werden.

Wir verwenden dazu den Begriff der Äquivalenzklassen, den Sie sich in Erinnerung rufen sollten.

Erinnerung: Äquivalenzrelationen und Äquivalenzklassen

- Ist ein *Äquivalenzrelation R* für eine Menge *M* gegeben (= Relation, die reflexiv, transitiv und symmetrisch ist), so ergibt sich eine Aufteilung ("Partitionierung") der gesamten Menge in einzelne sog. *Äquivalenzklassen*. Eine Äquivalenzklasse ist eine maximal große Teilmenge, so dass alle Elemente der Teilmenge zueinander äquivalent sind.
- Ist $x \in M$, dann bezeichnet $[x]_R$ die *Äquivalenzklasse* von *x*, d.h. die Menge aller Elemente, die zu *x* äquivalent sind (*x* ist auf jeden Fall auch dabei, da jeder Wert zu sich selbst äquivalent ist).

Beispiel 7.21 - Äquivalenzklassen

Wir betrachten die Menge aller Personen, die in Ulm, Neu-Ulm, Langenau oder Erbach wohnen. Die Relation "wohnt am gleichen Ort" ist ein Äquivalenzrelation.

- Durch diese Äquivalenzrelation ergeben sich dann vier Äquivalenzklassen:
  (1) Alle Personen, die in Ulm wohnen,

(2) Alle Personen, die in Neu-Ulm wohnen,
(3) alle Personen, die in Langenau wohnen,
(4) alle Personen, die in Erbach wohnen.

- Wohnt eine Person p_1 in Ulm, dann ist $[p_1]_R$ die Äquivalenzklasse von p_1, d.h. die Menge aller Personen, die in Ulm wohnen.

Algorithmus 7.22 - Minimierung von DEAs

Gegeben ist ein DEA $A = (Z, \Sigma, \delta, z_0, E)$.

(1) Entferne alle Zustände, die nicht vom Startzustand z_0 aus erreichbar sind.

(2) Berechne die Äquivalenzrelation $\equiv$ zwischen den Zuständen (mit *Algorithmus 7.19 - Berechnung äquivalenter Zustände,* s.o.)

(3) Bilde den Minimalautomaten durch Zusammenlegen von Zuständen, die zueinander äquivalent sind, d.h.

- Neue Zustände sind die Äquivalenzklassen bezüglich $\equiv$ (d.h. Mengen der Zustände, die jeweils zueinander äquivalent sind).
- Neuer Startzustand: Äquivalenzklasse, die den bisherigen Startzustand z_0 enthält.
- Neue Endzustände: Alle Äquivalenzklassen, die nur aus Endzuständen bestehen.
- Neue Zustandsübergänge werden aus dem ursprünglichen Automaten *A* folgendermaßen übernommen: Gibt es im Automaten *A* einen Zustandsübergang

  $$z_i \xrightarrow{a} z_k$$

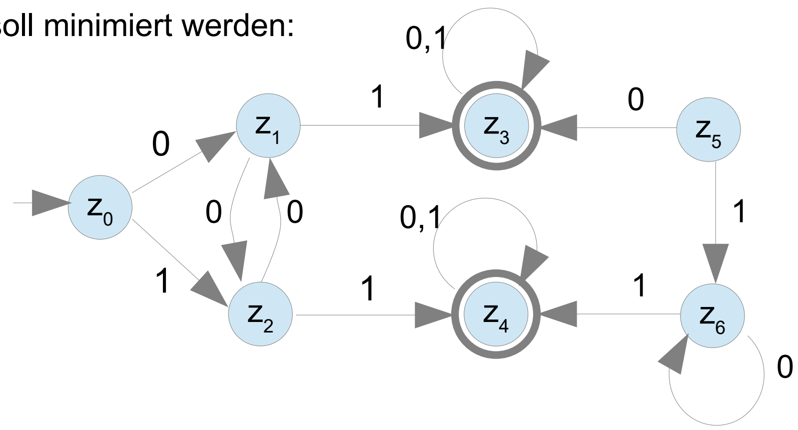
  dann gibt es im Minimalautomaten einen Zustandsübergang

  $$[z_i]_\equiv \xrightarrow{a} [z_k]_\equiv$$

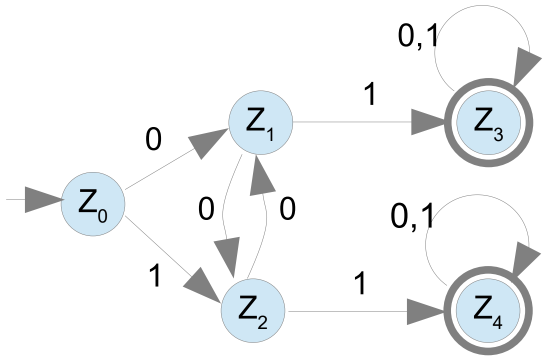
  d.h. von der Menge äquivalenter Zustände, in der z_i enthalten ist, zu der Menge, in der z_k enthalten ist.

Beispiel 7.23 - Minimierung eines DEA

Folgender DEA soll minimiert werden:



(1) Nicht erreichbare Zustände (mit zugehörigen Zustandsübergängen) entfernen. z_5 und z_6 sind nicht vom Startzustand z_0 aus erreichbar und werden entfernt. Übrig bleibt folgender DEA:



(2) Äquivalenzrelation ≡ auf Zuständen berechnen. Ergebnis, siehe Beispiel 7.20:

$z_1 \equiv z_2$

$z_3 \equiv z_4$

Somit ergeben sich drei Äquivalenzklassen:

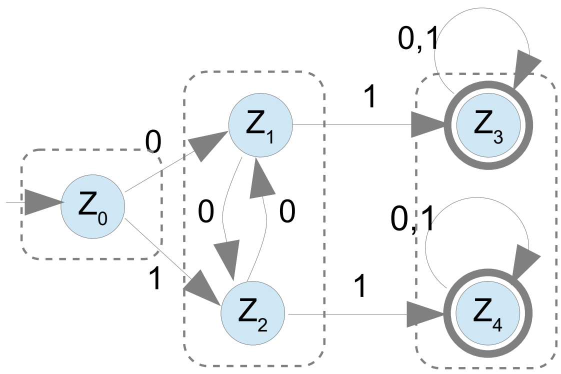
$\{z_0\}$ z_0 ist nur zu sich selbst äquivalent

$\{z_1, z_2\}$

$\{z_3, z_4\}$

(3) Äquivalente Zustände zusammenlegen (d.h. Äquivalenzklassen als Zustände nehmen)

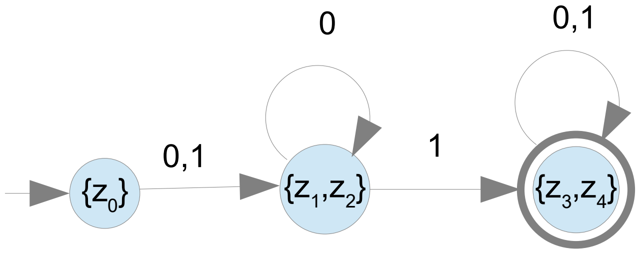
- Startzustand: $\{z_0\}$
- Endzustände: nur ein Endzustand $\{z_3, z_4\}$



(4) Zustandsübergänge übernehmen:

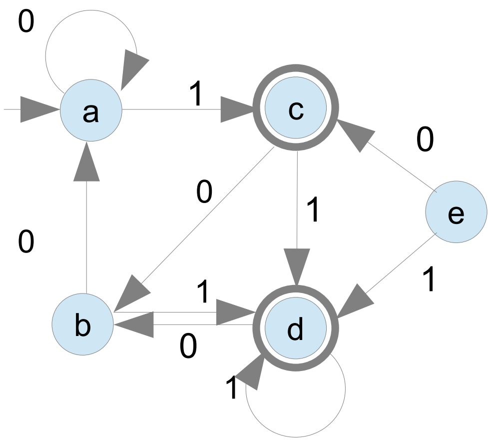
- Im Ausgangsautomat $z_0 \xrightarrow{0} z_1$, somit im Minimalautomat $\{z_0\} \xrightarrow{0} \{z_1, z_2\}$
- Im Ausgangsautomat $z_0 \xrightarrow{1} z_2$, somit im Minimalautomat $\{z_0\} \xrightarrow{1} \{z_1, z_2\}$
- usw.

Ergebnis: Insgesamt ergibt sich somit folgender Minimalautomat:



Aufgabe 7.24 - Minimierung von DEA

Minimieren Sie mit dem vorgestellten Verfahren folgenden DEA:



7.6 Äquivalenz von Automaten

In *Aufgabe 7.10 - Sprache von DEAs* war bereits zu sehen, dass es für eine Sprache unterschiedliche Automaten geben kann.

Definition 7.25 - Äquivalenz von Automaten

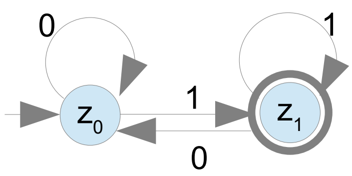
Zwei deterministische endliche **Automaten** A_1 und A_2 heißen **äquivalent**, wenn beide die gleiche Sprache akzeptieren, d.h. wenn

$$L(A_1) = L(A_2),$$

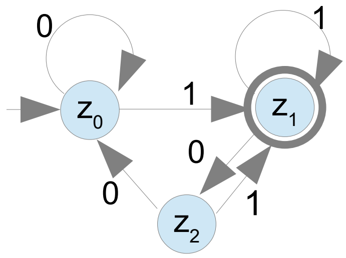
Beispiel 7.26 - Äquivalenz von Automaten

- Welche Automaten sind äquivalent?

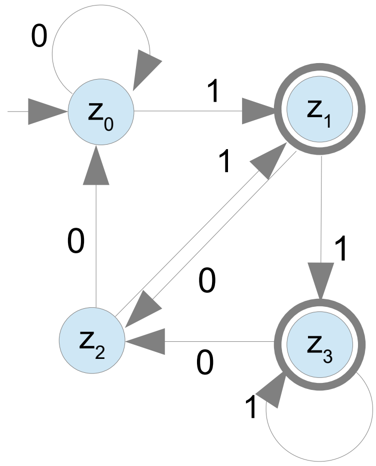
A_1:



A_2:



A_3:



- Die Automaten A_1 und A_2 sind schon aus Aufgabe 7.10 bekannt. Beide akzeptieren die gleiche Sprache, d.h.

  $L(A_1) = L(A_2) = \{ w1 \mid w \in \Sigma^* \}$,

  Also sind A_1 und A_2 äquivalent.

- Auch bei A_3 gilt, dass alle Wörter akzeptiert werden, die mit 1 enden. Mit 1, und nur mit 1, kommt man in einen Endzustand. Mit 0 kommt man immer in einen Nicht-Endzustand. Ebenso ist man mit dem leeren Wort ε, d.h. im Startzustand, in einem nicht akzeptierenden Zustand.

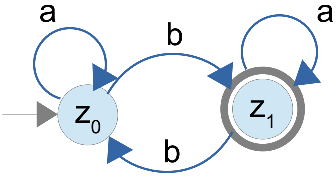
  $L(A_3) = \{ w1 \mid w \in \Sigma^* \} = L(A_1) = L(A_2)$

  D.h. alle drei Automaten A_1, A_2 und A_3 sind folglich zueinander äquivalent.

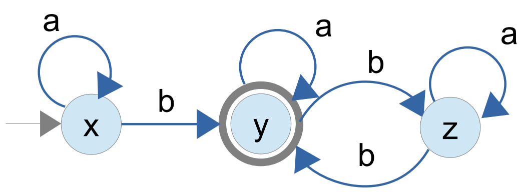
Aufgabe 7.27 - Äquivalenz von Automaten

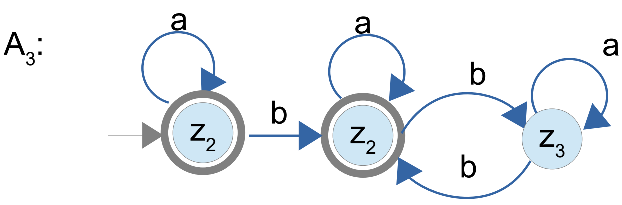
Welche der folgenden DEAs zum Alphabet $\Sigma = \{a, b\}$ sind äquivalent? Überlegen Sie sich dazu, welche Sprachen jeweils von den Automaten akzeptiert werden.

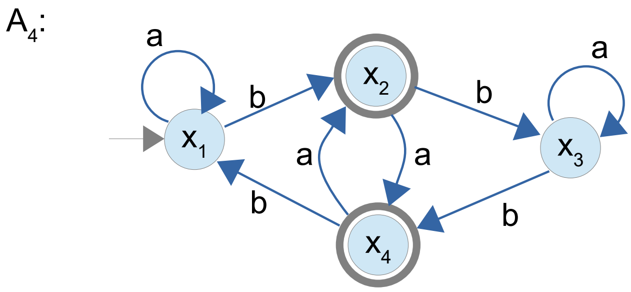
A_1:



A_2:





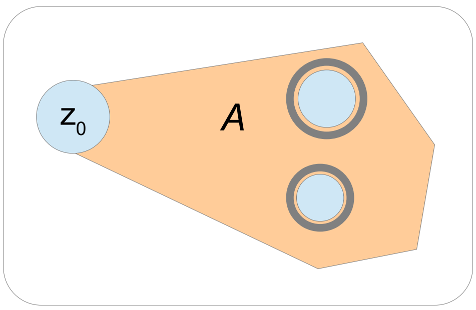


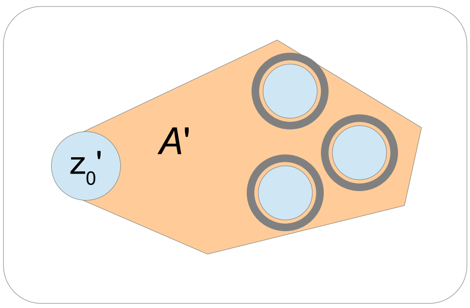
7.6.1 Nachweis der Äquivalenz von Automaten

Neben der Minimierung von Automaten, die im vorigen Abschnitt betrachtet wurde, kann noch eine weitere grundlegende Fragestellung in Zusammenhang mit DEAs auf die Äquivalenzberechnung von Zuständen zurückgeführt werden: Gibt es ein Verfahren, um zu prüfen, ob zwei Automaten äquivalent sind, d.h. die gleiche Menge von Wörtern akzeptieren?

Definition 7.28 - Äquivalenzproblem für DEAs

Äquivalenzproblem für DEAs: Gegeben sind zwei DEAs $A = (Z, \Sigma, \delta, z_0, E)$ und $A' = (Z', \Sigma', \delta', z_0', E')$:



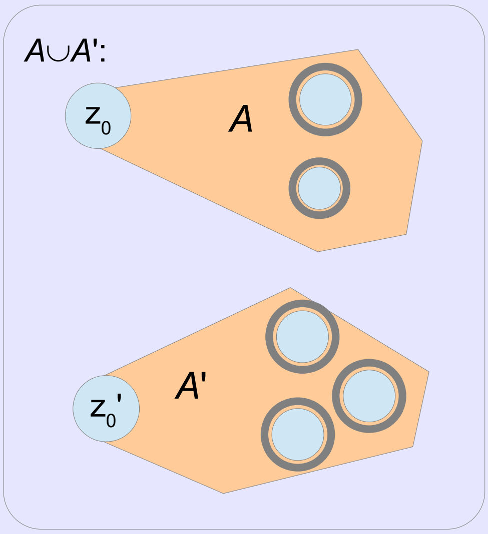


Lässt sich algorithmisch prüfen, ob beide Automaten *A* und *A'* äquivalent sind, d.h. $L(A) = L(A')$?

Algorithmus 7.29 - Äquivalenznachweis für DEAs

Zwei DEAs A und A' seien gegeben.

(3) Bilde einen gemeinsamen Automaten $A \cup A'$ durch disjunkte Vereinigung der Zustände von A und A' (d.h. gleich benannte Zustände ggf. dabei umbenennen).



(4) Berechne die äquivalenten Zustände von $A \cup A'$ mit Algorithmus 7.19.

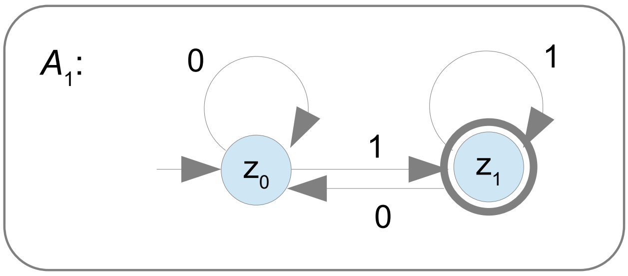
(5) Ergibt sich, dass die *Startzustände* z_0 von A und und z_0' von A' zueinander *äquivalent* sind, dann sind auch die *Automaten* A und A' *äquivalent*, da von beiden Zuständen aus mit genau den gleichen Wörtern Endzustände erreicht werden können.

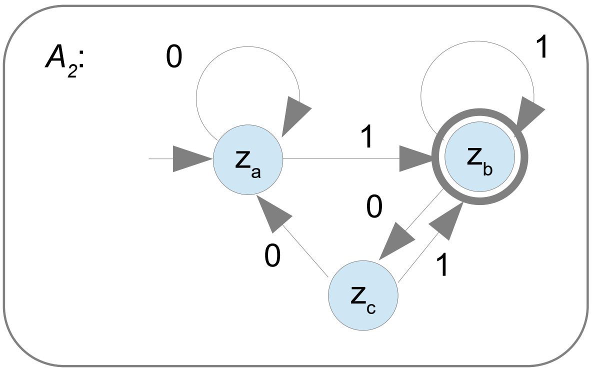
Anmerkung

- Durch die Vereinigung $A \cup A'$ der beiden Automaten ist noch kein DEA vollständig definiert, da nicht angegeben ist, welcher Zustand der Startzustand sein soll. Für die Berechnung der äquivalenten Zustände spielt das aber keine Rolle.

Beispiel 7.30 - Äquivalenz von Automaten

Sind die Automaten A_1 und A_2 äquivalent?

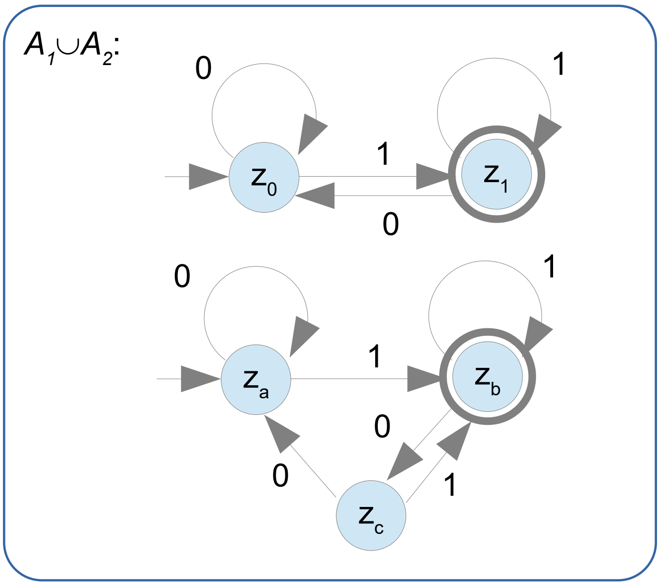




THU Technische Hochschule Ulm

Prüfung auf Äquivalenz:

- Vereinigten Automat betrachten:



- Äquivalente Zustände von $A_1 \cup A_2$ berechnen:

  1. Phase: Alle Kombinationen aus Endzustand und Nicht-Endzustand als unterscheidbar markieren.

z_0	≡				
z_1	X	≡			
z_a		X	≡		
z_b	X		X	≡	
z_c		X		X	≡
	z_0	z_1	z_a	z_b	z_c

2. Phase: Propagierungsregel für alle noch freien Einträge prüfen:

1. Durchgang: $\{z_0,z_a\}$, $\{z_0,z_c\}$, $\{z_1,z_b\}$, $\{z_a, z_c\}$.

   ⇒ Es gibt keine neuen Markierungen.

Somit ist die Berechnung der äquivalenten Zustände abgeschlossen mit folgendem Ergebnis:

z_0	≡				
z_1	X	≡			
z_a	≡	X	≡		
z_b	X	≡	X	≡	
z_c	≡	X	≡	X	≡
	z_0	z_1	z_a	z_b	z_c

- Die Startzustände z_0 und z_a sind nicht unterscheidbar, somit *äquivalent*. Das heißt, dass von beiden Zuständen mit genau den gleichen Wörtern Endzustände erreicht werden können. Somit akzeptieren A_1 und A_2 die gleichen Wörter, d.h. beide *Automaten* sind äquivalent.

Anmerkung

- Dass durch ein algorithmisches Verfahren entschieden werden kann, ob zwei beliebige Automaten die gleiche Sprache akzeptieren, ist ein tiefgehendes, wichtiges Ergebnis aus dem Bereich der formalen Sprachen.
- Bei einigen anderen Beschreibungsmethoden für Sprachen ist dies nicht entsprechend der Fall. Beispielsweise ist es algorithmisch nicht prüfbar (nicht berechenbar), ob zwei beliebige kontextfreie Grammatiken die gleiche Sprache definieren.

Zusammenfassung zu Lektion 12

Diese Fragen sollten Sie nun beantworten können

- Wann sind zwei Zustände eines DEA äquivalent bzw. unterscheidbar?
- Wie kann berechnet werden, ob zwei *Zustände* eines DEA äquivalent sind?
- Wie kann ein DEA minimiert werden?
- Was bedeutet, dass zwei Automaten äquivalent sind?
- Wie kann geprüft werden, ob zwei DEAs äquivalent sind?