5.6 Endliche Automaten

Das Drehkreuz kennt genau zwei Zustände: *gesperrt* und *frei*. Halten Sie im Zustand gesperrt Ihre Karte an den Leser, wechselt es nach frei; gehen Sie hindurch, ist es wieder gesperrt. Drücken Sie gegen ein gesperrtes Drehkreuz, bleibt es gesperrt. Mehr weiß es nicht — weder, wie viele Leute heute schon durch sind, noch, wer Sie sind. Ein Türschloss mit Tastenfeld, das aufgeht, sobald die letzten drei Tasten 7, 3, 3 waren, ist übrigens genauso gebaut wie der Automat für „endet auf `abb`“ weiter unten.

> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Zustandsmaschine**.

Eine Grammatik *erzeugt* Wörter. Für den Compiler brauchen wir das Gegenstück: eine Maschine, die ein vorgelegtes Wort *prüft*. Das leistet der endliche Automat.

Definition: Endlicher Automat

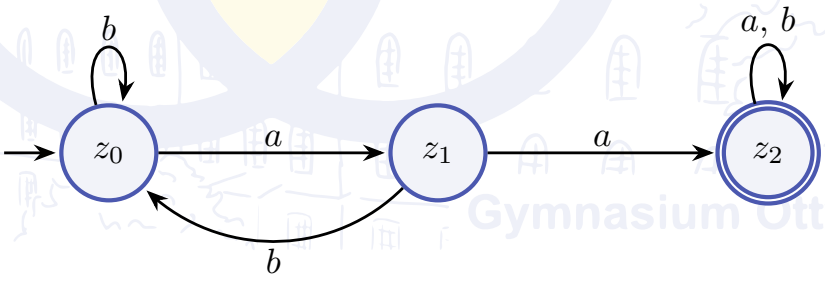
Ein **endlicher akzeptierender Automat** (Akzeptor) ist ein **5-Tupel** $M = (Z, A, f, z_0, F)$ mit

- der endlichen **Zustandsmenge** Z,
- dem **Eingabealphabet** A,
- der **Übergangsfunktion** $f : Z \times A \to Z$,
- dem **Anfangszustand** $z_0 \in Z$,
- der **Finalmenge** $F \subseteq Z$ der **Endzustände**.

Der Automat akzeptiert ein Wort, wenn er nach dem Lesen des letzten Zeichens in einem Endzustand steht.

Gezeichnet wird er als **Zustandsgraph**: Zustände sind Kreise, Übergänge beschriftete Pfeile, der Startzustand hat einen Pfeil von außen, Endzustände einen doppelten Rand.

Beispiel: Wörter über $\{a, b\}$, die `aa` enthalten.



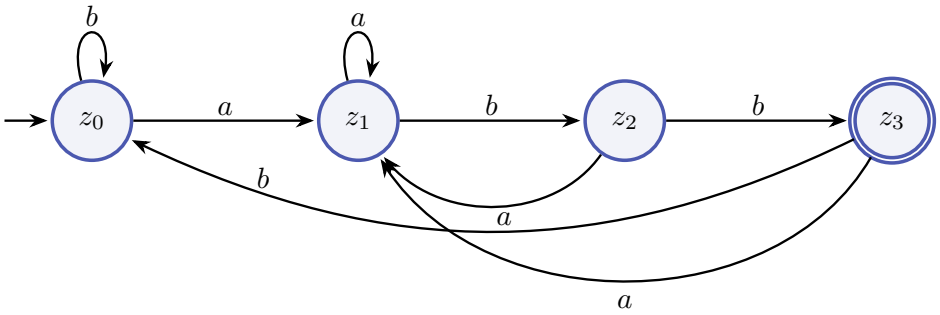
Zustandsgraph: akzeptiert alle Wörter über $\{a, b\}$, die `aa` enthalten.

In Worten: z_0 heißt „das zuletzt gelesene Zeichen war kein a“, z_1 heißt „das zuletzt gelesene Zeichen war ein a, aber noch kein aa”, und z_2 heißt „aa ist aufgetreten” — dieser Zustand wird nie wieder verlassen. Ein b in z_1 wirft zurück nach z_0. Beachten Sie: In z_0 kann der Automat schon beliebig viele a gelesen haben, nur eben nicht als letztes Zeichen. Der Zustand merkt sich nicht die Anzahl, sondern allein, wie weit das gesuchte aa gerade gediehen ist.

Dieselbe Information als **Übergangstabelle** — die Form, die man zum Programmieren braucht:

Zustand	bei a	bei b	Endzustand
z_0	z_1	z_0	nein
z_1	z_2	z_0	nein
z_2	z_2	z_2	**ja**

Beispiel: Wörter über $\{a, b\}$, die auf abb enden.



Jeder Zustand merkt sich, wie viel vom gesuchten abb schon gelesen ist.

Zustand	bei a	bei b	Bedeutung	Endzustand
z_0	z_1	z_0	nichts Passendes gesehen	nein
z_1	z_1	z_2	zuletzt a	nein
z_2	z_1	z_3	zuletzt ab	nein
z_3	z_1	z_0	zuletzt abb	**ja**

Der Zustand merkt sich also gerade so viel von der Vergangenheit, wie für die Entscheidung nötig ist — und keinen Buchstaben mehr. Das ist die zentrale Idee: **Der Zustand ist das Gedächtnis des Automaten, und dieses Gedächtnis ist endlich.**

> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Automatensimulator**.

Dasselbe Muster an einer Tür: Das Tastenschloss merkt sich nur, wie viel vom Code zuletzt getippt wurde. Alle Ziffern außer 7 und 3 stehen am Pfeil als „sonst“.

> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Zustandsmaschine**.

Arbeitsauftrag 6

21. Zeichnen Sie den Zustandsgraphen zur zweiten Tabelle und prüfen Sie die Wörter abb, aabb, abba, babb.
22. Konstruieren Sie einen DEA für alle geraden Dualzahlen und einen für alle ungeraden.
23. Konstruieren Sie einen DEA für Bezeichner nach der Regel aus 5.5.
24. Konstruieren Sie einen DEA über $\{a, b\}$, der genau die Wörter mit einer geraden Anzahl von a akzeptiert.
25. Geben Sie zu Ihrem Automaten aus Aufgabe 24 die Übergangstabelle an.
26. Welche Sprache akzeptiert der Automat mit $Z = \{z_0, z_1\}$, $F = \{z_1\}$ und den Übergängen $f(z_0, 0) = z_0$, $f(z_0, 1) = z_1$, $f(z_1, 0) = z_1$, $f(z_1, 1) = z_0$? (Tipp: Zählen Sie.)

5.6.1 Deterministisch oder nichtdeterministisch

Sie suchen in einem fremden Einkaufszentrum einen bestimmten Laden. Allein müssten Sie an jeder Gabelung raten. Mit genug Freunden geht es ohne Raten: An jeder Gabelung geht jemand in jede Richtung, und wer ankommt, ruft die anderen. Ein nichtdeterministischer Automat rechnet genau so — er verfolgt alle Wege gleichzeitig und ist zufrieden, wenn irgendeiner ankommt.

- Bei einem **deterministischen** Automaten (DEA) ist f eindeutig: Zu jedem Paar aus Zustand und Zeichen gibt es genau einen Folgezustand.
- Bei einem **nichtdeterministischen** Automaten (NEA) dürfen es mehrere sein. Der Automat müsste gewissermaßen raten, welchen Weg er nimmt, und akzeptiert ein Wort, wenn *irgendein* Weg in einen Endzustand führt.

Beim NEA ist f also keine Funktion in den Zuständen, sondern eine in die **Teilmengen** von Z: $f : Z \times A \to \mathcal{P}(Z)$. Der Automat ist gewissermaßen in mehreren Zuständen gleichzeitig.

NEAs sind oft schneller hingeschrieben — bauen lässt sich aber nur ein DEA, denn eine Maschine kann nicht raten. Der Trost ist ein Satz:

Merke

Zu **jedem** NEA gibt es einen DEA, der dieselbe Sprache akzeptiert. Nichtdeterminismus bringt keine zusätzliche Mächtigkeit, nur kürzere Diagramme.

5.6.2 Die Teilmengenkonstruktion

Der Beweis des Satzes ist zugleich ein Bauplan. Die Idee: Wenn der NEA in mehreren Zuständen gleichzeitig sein kann, macht man **jede mögliche Zustandsmenge zu einem einzigen DEA-Zustand**.

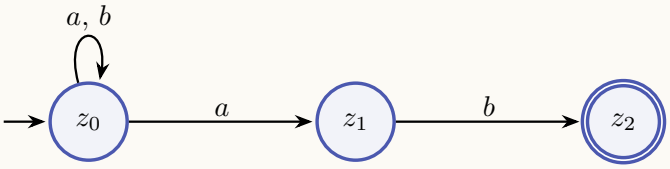
Definition: Teilmengenkonstruktion

Gegeben ein NEA mit Zustandsmenge Z. Der zugehörige DEA entsteht so:

1. **Startzustand** des DEA ist die Menge $\{z_0\}$.
2. Für jede bereits erzeugte Menge M und jedes Zeichen c bildet man die Menge aller Zustände, die der NEA von *irgendeinem* Zustand aus M mit c erreichen kann. Das ist der Folgezustand.
3. Neu entstandene Mengen werden genauso behandelt, bis keine neue mehr auftritt.
4. **Endzustände** des DEA sind genau die Mengen, die mindestens einen Endzustand des NEA enthalten.

Beispiel: Wörter über {a, b}, die auf ab enden

Der NEA rät, wann das letzte `ab` beginnt:



Der NEA. In z_0 hat er bei a zwei Möglichkeiten — das ist der Nichtdeterminismus.

	a	b
z_0	$\{z_0, z_1\}$	$\{z_0\}$
z_1	$\emptyset$	$\{z_2\}$
z_2	$\emptyset$	$\emptyset$

Endzustand ist z_2. In z_0 ist der Automat nichtdeterministisch: Bei a kann er dort bleiben oder nach z_1 wechseln.

Im Simulator sehen Sie die Zustands**menge** wandern — genau das, was die Teilmengenkonstruktion gleich zu einzelnen Zuständen zusammenfasst:

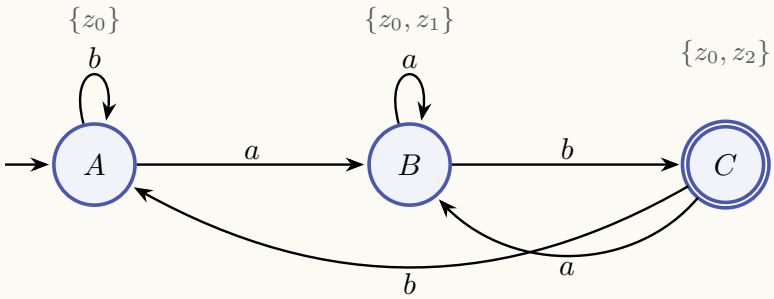
> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Automatensimulator**.

Durchführung. Start ist $A = \{z_0\}$.

Menge	mit a	mit b
$A = \{z_0\}$	$\{z_0, z_1\} = B$	$\{z_0\} = A$
$B = \{z_0, z_1\}$	$\{z_0, z_1\} = B$	$\{z_0, z_2\} = C$
$C = \{z_0, z_2\}$	$\{z_0, z_1\} = B$	$\{z_0\} = A$

Keine neue Menge mehr — fertig. C enthält z_2 und ist damit der einzige Endzustand.

Der fertige DEA:



Der konstruierte DEA. Über jedem Zustand steht die Menge von NEA-Zuständen, aus der er entstanden ist.

Übergangstabelle des DEA:

	a	b
$\to A$	B	A
B	B	C
C^*	B	A

Probe: bab läuft $A \to A \to B \to C$ — akzeptiert. aba läuft $A \to B \to C \to B$ — abgelehnt, richtig, denn aba endet nicht auf ab.

Merke

Hat der NEA n Zustände, hat der DEA höchstens 2^n — so viele Teilmengen gibt es. Meist sind es weit weniger, weil die meisten Mengen nie erreicht werden. Im Beispiel waren es 3 statt der möglichen 8.

Arbeitsauftrag 7

27. Konstruieren Sie mit der Teilmengenkonstruktion einen DEA zu dem NEA über $\{0,1\}$ mit $f(z_0,0)=\{z_0,z_1\}$, $f(z_0,1)=\{z_0\}$, $f(z_1,1)=\{z_2\}$, $F=\{z_2\}$. Welche Sprache akzeptiert er?
28. Geben Sie zum DEA aus dem Beispiel das 5-Tupel vollständig an.
29. Zeichnen Sie einen NEA, der alle Wörter über $\{a,b\}$ akzeptiert, die `aa` **oder** `bb` enthalten, und wandeln Sie ihn um. Wie viele Zustände hat der DEA?

Gymnasium Ottweiler