

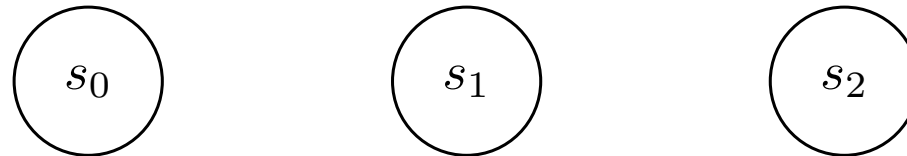
Endliche Automaten

- Einfaches Modellierungswerkzeug (z.B. UML-Statecharts)
- Verarbeiten Wörter/Ereignisfolgen
- Erkennen Sprachen
- Erlauben schnelle Spracherkennung
- **Anwendungsbereiche:** Objektorientierte Modellierung, Model-Checking, Lexikalische Analyse, XML-Parser, Kontrollbedingungen in regelbasierten Systemen, Spezifikation von Kommunikationsabläufen, Beschreibung von Rechnersystemen und deren Systemprogrammierung, . . .

Komponenten

- Zustände

Zum Beispiel:

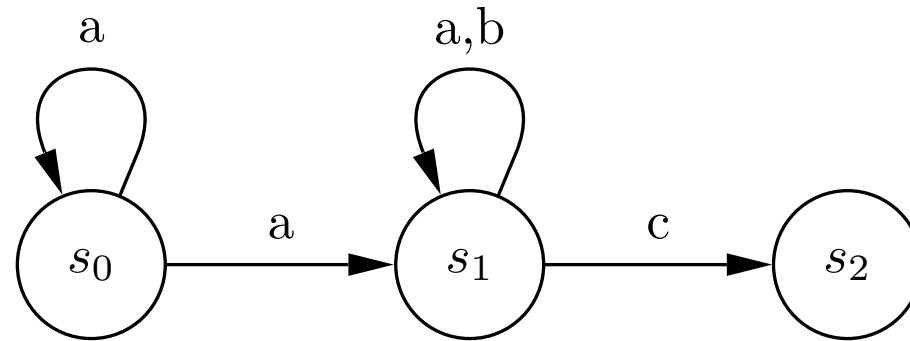


- Eingabealphabet (Menge von potenziellen Ereignissen)

Zum Beispiel: $\{a, b, c\}$

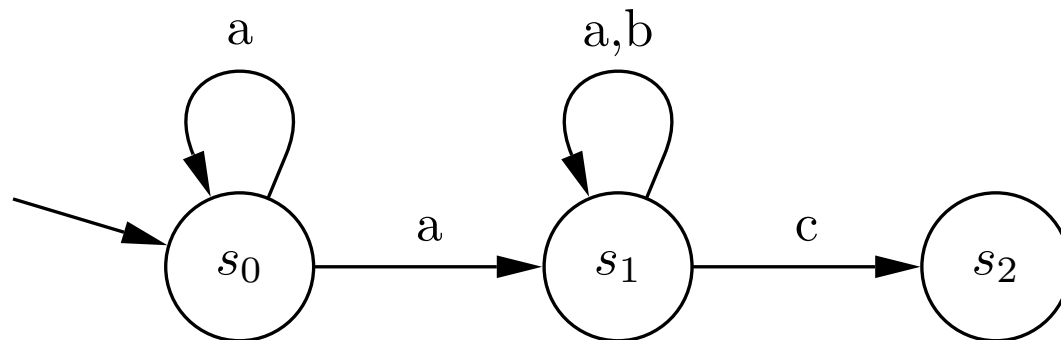
- Zustandsüberführungen

Zum Beispiel:



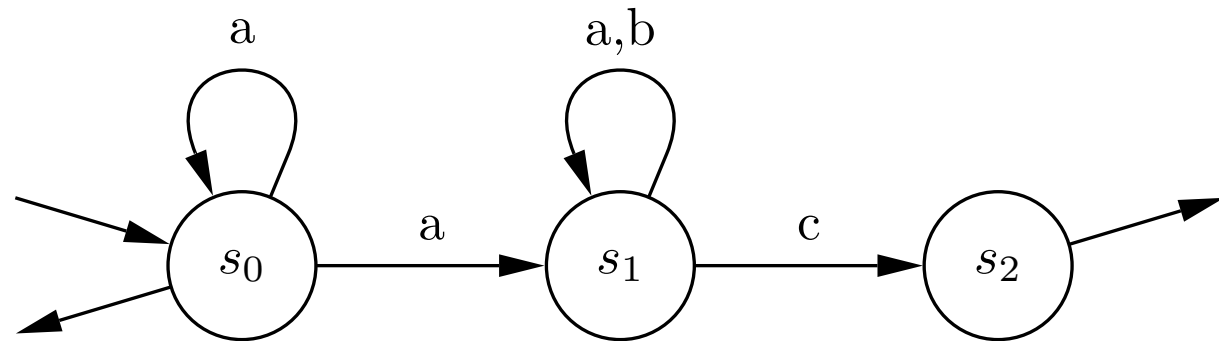
- Startzustand

Zum Beispiel:



- Endzustände

Zum Beispiel:



Definition

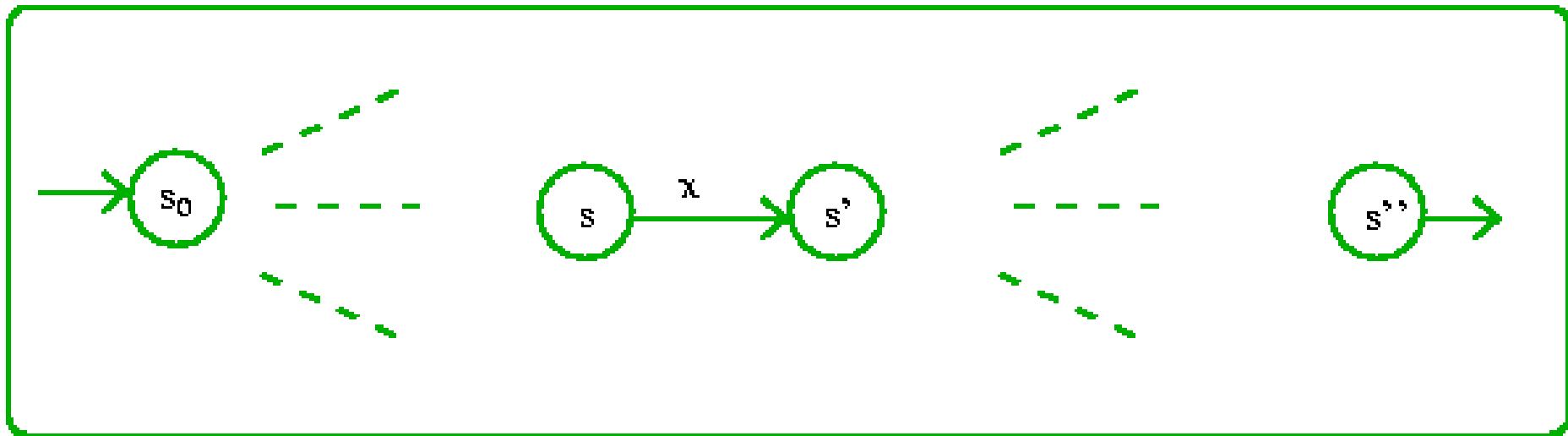
Endliche Automaten

Ein **endlicher Automat** ist ein System $A = (Z, I, d, s_0, F)$ mit

- Z : endliche Menge von **Zuständen**,
- I : endliches **Eingabealphabet**,
- d : **Zustandsüberführung** mit $d \subseteq Z \times I \times Z$
- $s_0 \in Z$: **Startzustand**,
- $F \subseteq Z$: **Endzustände**.

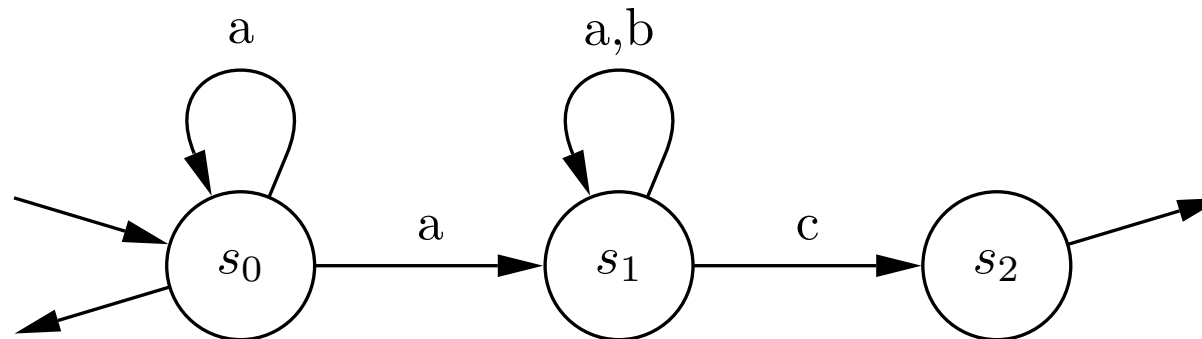
Graphische Darstellung

Gegeben: $A = (Z, I, d, s_0, F)$



$$s' \in d(s, x), \quad s'' \in F$$

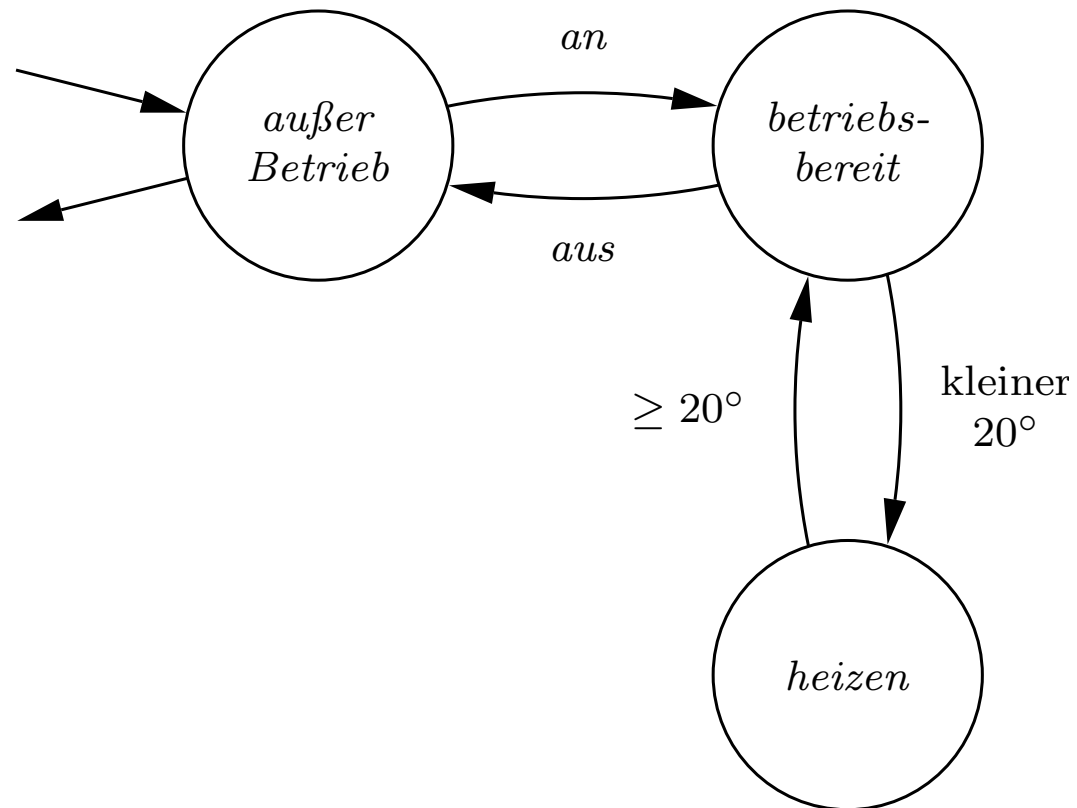
Beispiel 1



Akzeptiert u.a. das Wort $aabc$:

- $(s_0, aabc) \vdash (s_0, abc) \vdash (s_1, bc) \vdash (s_1, c) \vdash (s_2, \lambda)$
- $(s_0, aabc) \vdash (s_1, abc) \vdash (s_1, bc) \vdash (s_1, c) \vdash (s_2, \lambda)$

Beispiel 2



Fortgesetzte Zustandsüberführung

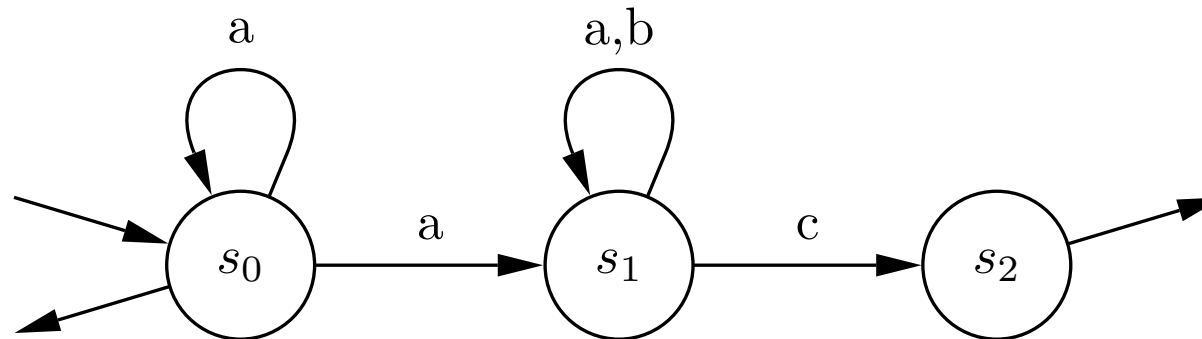
Die fortgesetzte Zustandsüberführung verarbeitet Wörter statt Zeichen.

Gegeben: $A = (Z, I, d, s_0, F)$

Für alle $s, s', s'' \in Z, x \in I, w \in I^*$:

- $d^*(s, \lambda) = \{s\}$,
- $d^*(s, wx) = \bigcup_{s' \in d^*(s, w)} d(s', x)$.

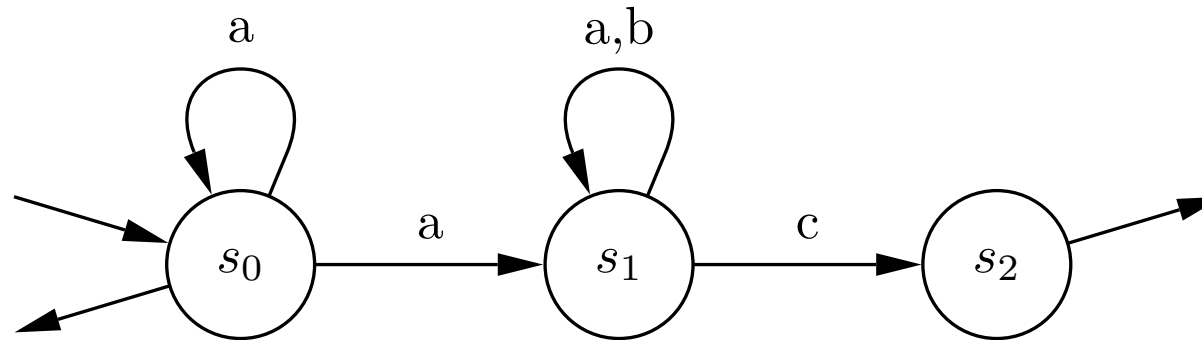
Beispiel



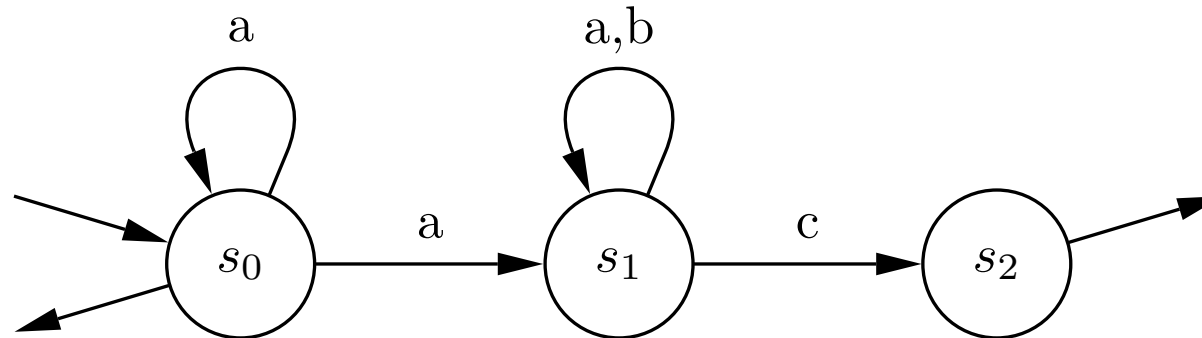
$$d^*(s_0, aabc) = \bigcup_{s' \in d^*(s_0, aab)} d(s', c) = \text{?????}$$

$$d^*(s_0, aab) = \bigcup_{s' \in d^*(s_0, aa)} d(s', b) = \text{?????}$$

$$d^*(s_0, aa) = \bigcup_{s' \in d^*(s_0, a)} d(s', a) = \text{?????}$$



$$d^*(s_0, a) = \bigcup_{s' \in d^*(s_0, \lambda)} d(s', a) = \bigcup_{s' \in \{s_0\}} d(s', a) = d(s_0, a) = \{s_0, s_1\}$$

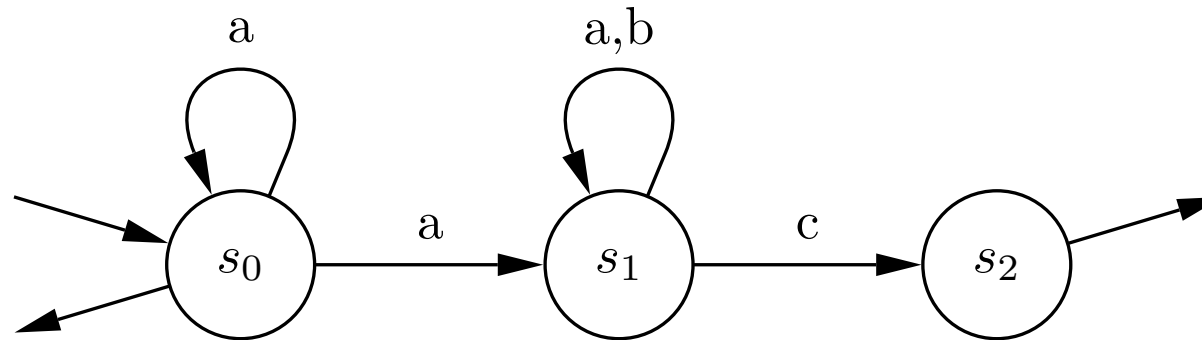


$$d^*(s_0, aabc) = \bigcup_{s' \in d^*(s_0, aab)} d(s', c) = \text{?????}$$

$$d^*(s_0, aab) = \bigcup_{s' \in d^*(s_0, aa)} d(s', b) = \text{?????}$$

$$d^*(s_0, aa) = \bigcup_{s' \in d^*(s_0, a)} d(s', a) = \bigcup_{s' \in \{s_0, s_1\}} d(s', a) = d(s_0, a) \cup d(s_1, a) = \{s_0, s_1\} \cup \{s_1\} = \{s_0, s_1\}$$

$$d^*(s_0, a) = \{s_0, s_1\}$$

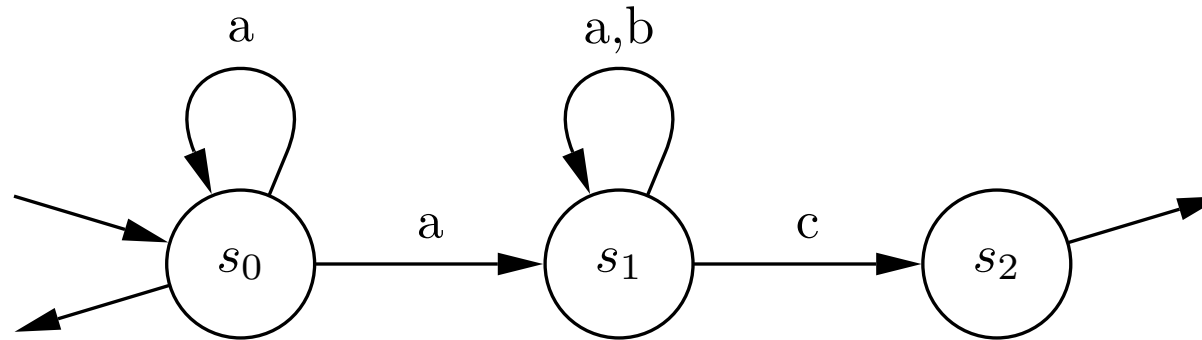


$$d^*(s_0, aabc) = \bigcup_{s' \in d^*(s_0, aab)} d(s', c) = \text{?????}$$

$$\begin{aligned} d^*(s_0, aab) &= \bigcup_{s' \in d^*(s_0, aa)} d(s', b) = \bigcup_{s' \in \{s_0, s_1\}} d(s', b) \\ &= d(s_0, b) \cup d(s_1, b) = \emptyset \cup \{s_1\} \end{aligned}$$

$$d^*(s_0, aa) = \{s_0, s_1\}$$

$$d^*(s_0, a) = \{s_0, s_1\}$$



$$d^*(s_0, aabc) = \bigcup_{s' \in d^*(s_0, aab)} d(s', c) = \bigcup_{s' \in \{s_1\}} d(s', c) \\ = d(s_1, c) = \{s_2\}$$

$$d^*(s_0, aab) = \{s_1\}$$

$$d^*(s_0, aa) = \{s_0, s_1\}$$

$$d^*(s_0, a) = \{s_0, s_1\}$$

Erkannte Sprache

Die **erkannte Sprache** besteht aus allen Wörtern, die der Automat ausgehend vom Startzustand einlesen kann, so dass nach dem Einlesen ein Endzustand erreicht wird.

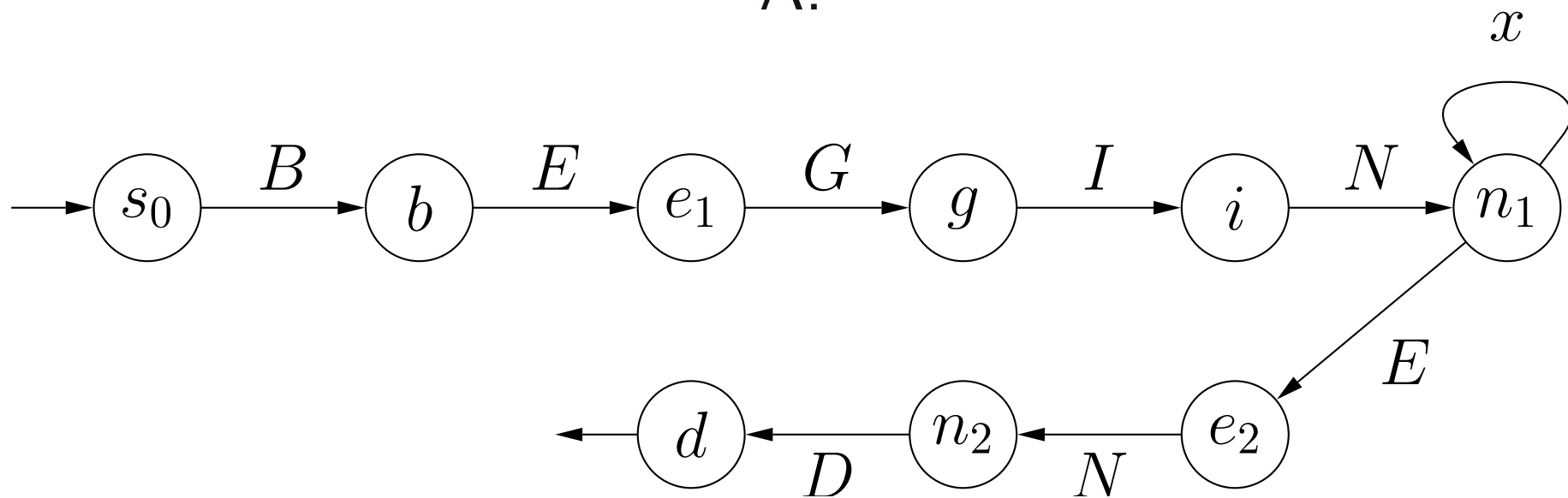
Erkannte Sprache

Gegeben: $A = (Z, I, d, s_0, F)$

$$L(A) = \{w \in I^* \mid d^*(s_0, w) \cap F \neq \emptyset\}$$

Beispiele (1)

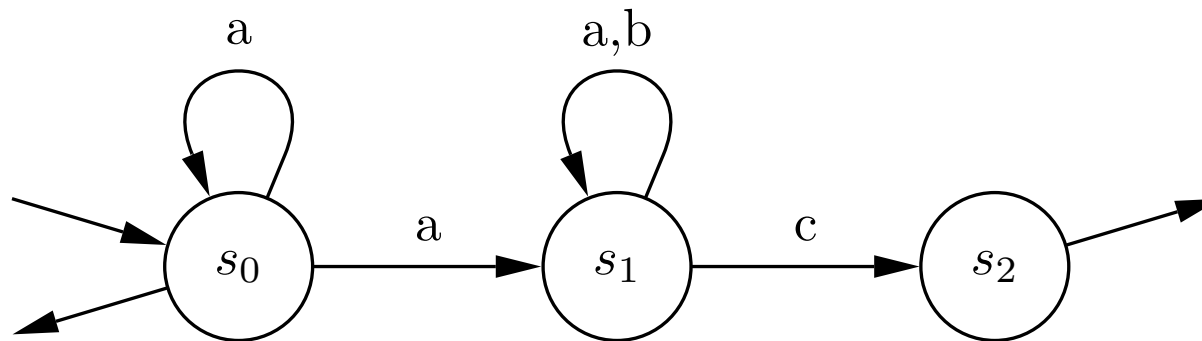
A:



$$L(A) = \{BEGINuEND \mid u \in I^*\}$$

Beispiele (2)

A:



$$L(A) = \{a^n wc \mid n \geq 1, w \in \{a, b\}^*\} \cup \{a^n \mid n \in \mathbb{N}\}$$

Beispiele (3)

Teilwortsuche

- ▶ **Eingabe:** $u, v \in A^*$
- ▶ **Ausgabe:** Alle Stellen in v , an denen u vorkommt.

Idee: Konstruiere einen endlichen Automaten, der alle Anfangswörter (Präfixe) von v erkennt, die mit u enden.

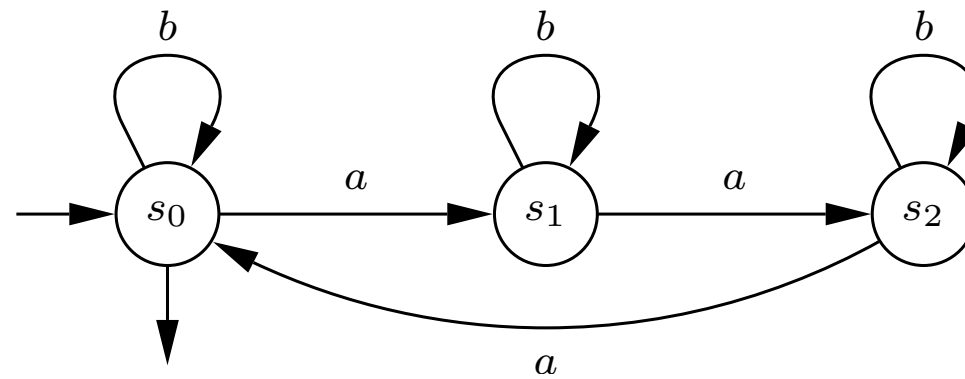
Deterministische Endliche Automaten

Definition

Ein **deterministischer endlicher Automat (DEA)** ist ein System $A = (Z, I, d, s_0, F)$ mit

- Z : endliche Menge von Zuständen,
- I : endliches Eingabealphabet,
- $d: Z \times I \rightarrow Z$: **Abbildung**
- $s_0 \in Z$: Startzustand,
- $F \subseteq Z$: Endzustände.

Beispiel



Gegeben: DEA $A = (Z, I, d, s_0, F)$

Fortgesetzte Zustandsüberführung

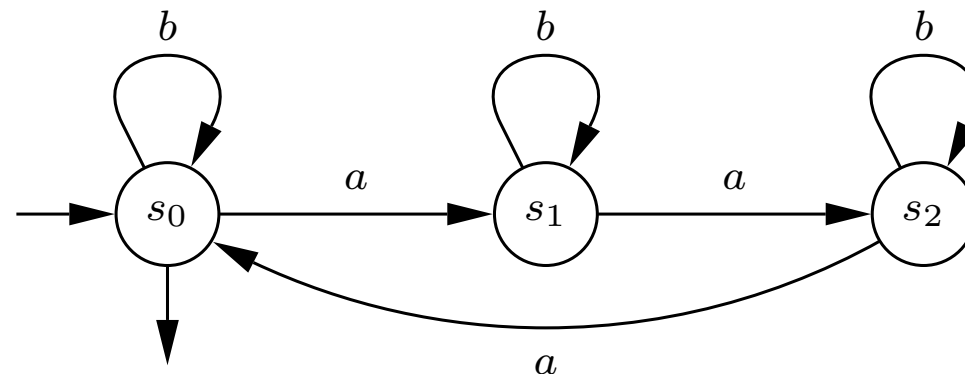
Für alle $s, s', s'' \in Z, x \in I, w \in I^*$:

- $d^*(s, \lambda) = s$;
- $d^*(s, wx) = d(d^*(s, w), x)$.

Erkannte Sprache

$$L(A) = \{w \in I^* \mid d^*(s_0, w) \in F\}$$

Beispiel



Erkannte Sprache:

$$\{w \in \{a, b\}^* \mid \text{count}(a, w) \bmod 3 = 0\}$$

Satz

Sei $A = (Z, I, d, s_0, F)$ ein DEA. Dann gilt für alle $u, v \in I^*$, $s \in Z$: $d^*(s, uv) = d^*(d^*(s, u), v)$.

Verarbeitung von Wörtern in iterativer Darstellung

Gegeben: $A = (Z, I, d, s_0, F)$

$w = a_1 \cdots a_n \in A^*$ mit $a_i \in A$ für $i = 1, \dots, n$

($n = 0$ impl. $w = \lambda$)

$s' \in d^*(s, w)$ für $s, s' \in Z$ genau dann, wenn es eine Folge von Zuständen t_0, \dots, t_n gibt, derart dass $s = t_0$, $s' = t_n$ und $t_i \in d(t_{i-1}, a_i)$ für $i = 1, \dots, n$.

Verarbeitung im Zustandsgraph



Wortproblem

Gegeben: eine Sprache $L \subseteq A^*$
(z.B. in Form eines endlichen Automaten).

Eingabe: Ein Wort $w \in A^*$.

Ausgabe: Ja, falls $w \in L$
Nein, sonst.

Satz (schnelle Worterkennung)

Für von deterministischen endlichen Automaten erkannte Sprachen ist das Wortproblem in linearer Zeit lösbar.



Lesedauer von $a_1 \dots a_n$: n Schritte, da Folgezustände eindeutig sind.

Potenzautomat

Gegeben: $A = (Z, I, d, s_0, F)$.

$$\mathcal{P}(A) = (\mathcal{P}(Z), I, D, \{s_0\}, F_{\mathcal{P}})$$

- $\mathcal{P}(Z)$: Potenzmenge von Z ;
- $D: \mathcal{P}(Z) \times I \rightarrow \mathcal{P}(Z)$ mit

$$D(S, x) = \bigcup_{s \in S} d(s, x)$$

für alle $S \in \mathcal{P}(Z)$, $x \in I$;

- $F_{\mathcal{P}} = \{S \in \mathcal{P}(Z) \mid S \cap F \neq \emptyset\}$.

Ein optimierter Potenzautomat

