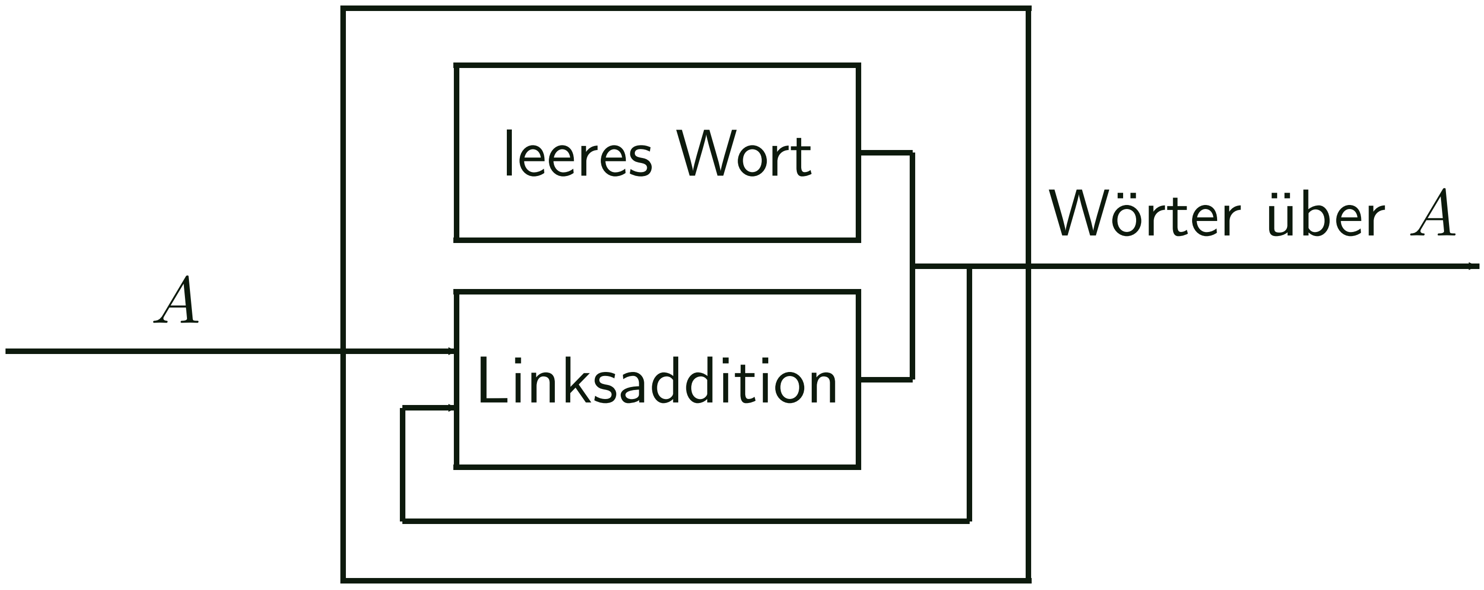
Wörter

Erzeugung von Wörtern

- A : Alphabet (Menge von Zeichen)
- A^*: Menge aller Wörter über A.

Rekursive Definition von A^*

- $\lambda \in A^*$ (leeres Wort, enthält keine Zeichen)
- mit $x \in A$ und $v \in A^*$ ist auch $xv \in A^*$ (Linksaddition)


leeres Wort
Linksaddition
A
Wörter über A

Bemerkungen über Wörter

- Schreibweise: $x\lambda$ wird mit x abgekürzt.
- Es gilt $A \subseteq A^*$
- Wörter sind eindeutig zerlegbar:

  Für alle $w \in A^*$ gilt entweder $w = \lambda$, oder es existieren eindeutig ein $x \in A$ und ein $v \in A^*$, so dass $w = xv$.

  Bezeichnung: $x = head(w)$, $v = tail(w)$
- Falls A selbst Wörter enthält, muss die eindeutige Zerlegbarkeit garantiert werden.

  Zum Beispiel gilt für $A = \{B, E, I, BEI, SPIEL\}$

  $head(B \circ E \circ I \circ SPIEL) = B \neq$

  $head(BEI \circ SPIEL) = BEI$

Bemerkungen über Wörter (2)

- Rechtsaddition kann anolog zur Linksaddition definiert werden.
- Iterative Darstellung von Wörtern: Für jedes Wort w existieren eindeutig ein $n \in \mathbb{N}$ und $x_i \in A$ für $i = 1, \ldots, n$ mit $w = x_1 \cdots x_n$.
- Falls $A = \{|\}$ repräsentiert A^* die natürlichen Zahlen.

$$\lambda = |^0 \text{ repräsentiert } 0$$
$$||^n \text{ repräsentiert } n + 1$$

Induktionsprinzip

- Induktionsanfang (IA):
  Zeige THEOREM für $v = \lambda$.
- Induktionsvoraussetzung (IV):
  Nimm THEOREM für v an.
- Induktionsschluss (IS):
  Zeige THEOREM für xv mit $x \in A$.

Konkatenation von Wörtern

(a) für $v = \lambda$ gilt $\lambda \cdot w = w$,

(b) für $v = xu$ mit $x \in A$ gilt $(xu) \cdot w = x(u \cdot w)$.

$$(aba) \cdot cd = a(ba \cdot cd) = a(b(a \cdot cd)) = a(b(a(\lambda \cdot cd))) = a(b(a(cd))) = a(b(acd)) = a(bacd) = abacd$$

Satz (Assoziativität)
Für alle Wörter u, v, w gilt $(uv)w = u(vw)$.

Länge

1. $length(\lambda) = 0$
2. $length(xv) = length(v) + 1$ für $x \in A, v \in A^*$

$length(aba) = 1 + length(ba) = 1 + 1 + length(a) =$
$2 + 1 + length(\lambda) = 3 + 0 = 3$

Satz

Für alle $u, v \in A^*$ gilt:

$$length(uv) = length(u) + length(v).$$

Gleichheitstest

1. $\lambda \equiv \lambda$,
2. $xu \equiv yv$, falls $x \equiv y$ und $u \equiv v$ für alle $x, y \in A$, $u, v \in A^*$

(Voraussetzung: Gleichheitstest $\equiv$ auf A gegeben.)

Satz

Für alle $u, v, w \in A^*$ gilt:

$$u \equiv v \text{ impliziert } uw \equiv vw.$$

Zeichen zählen

1. $count(x, \lambda) = 0$
2. $count(x, yv) = \begin{cases} count(x, v) + 1, & \text{falls } x \equiv y \\ count(x, v) & \text{sonst} \end{cases}$

Satz

Für alle $x \in A$, $u, v \in A^*$ gilt:

$$count(x, uv) = count(x, u) + count(x, v).$$

Endliche Automaten

Gliederung

- Endliche Automaten
- Deterministische endliche Automaten
- Zusammenhang zwischen nichtdeterministischen und deterministischen endlichen Automaten (Potenzautomat)
- Komposition von Automaten (Produktautomat, Vereinigungsautomat)
- Lösbarkeit des Wortproblems und des Leerheitsproblems
- Pumping-Lemma

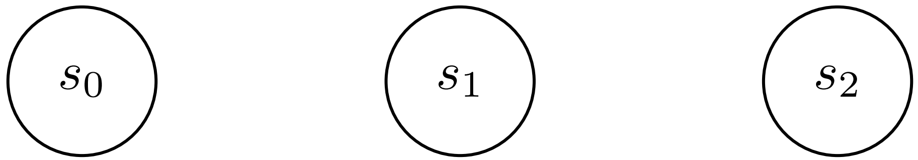
Endliche Automaten

- Einfaches Modellierungswekzeug (z.B. UML-Statecharts)
- Verarbeiten Wörter/Ereignisfolgen
- Erkennen Sprachen
- Erlauben schnelle Spracherkennung

Komponenten

- Zustände

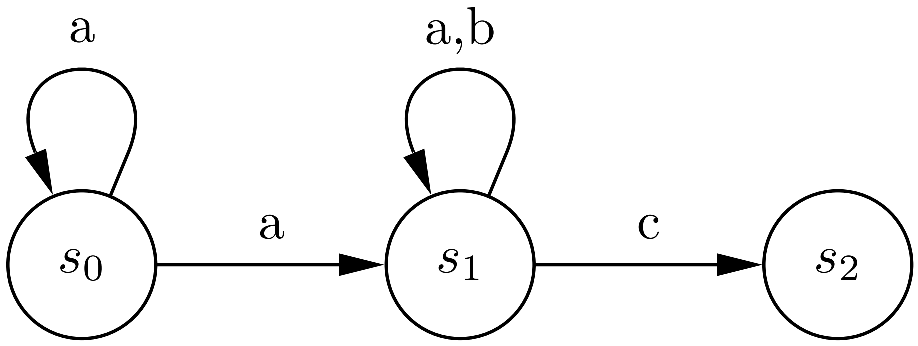
Zum Beispiel:



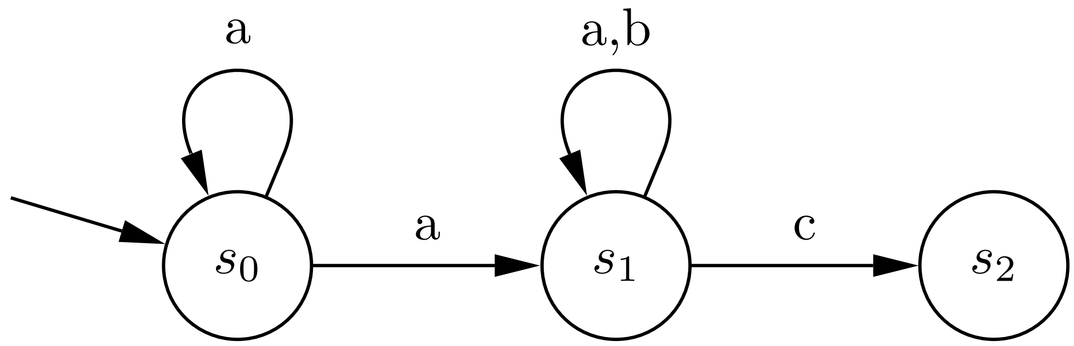
- Eingabealphabet (Menge von potenziellen Ereignissen)

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

- Zustandsüberführungen

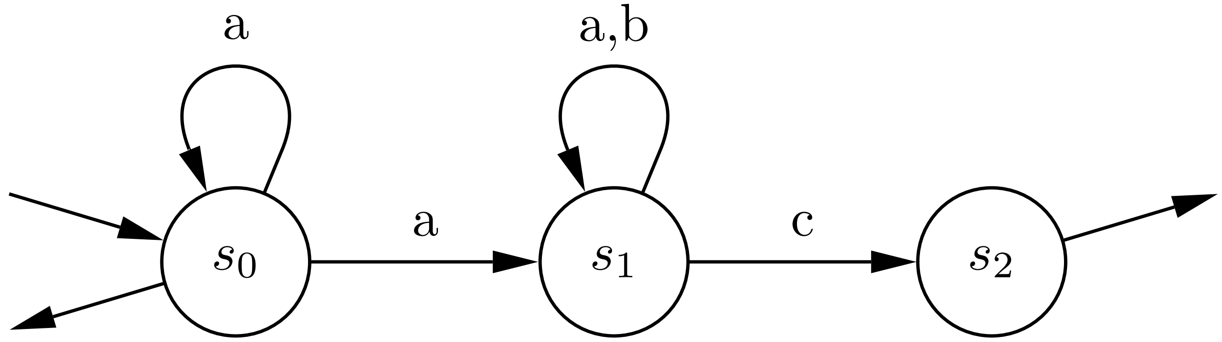
Zum Beispiel: 

- 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.