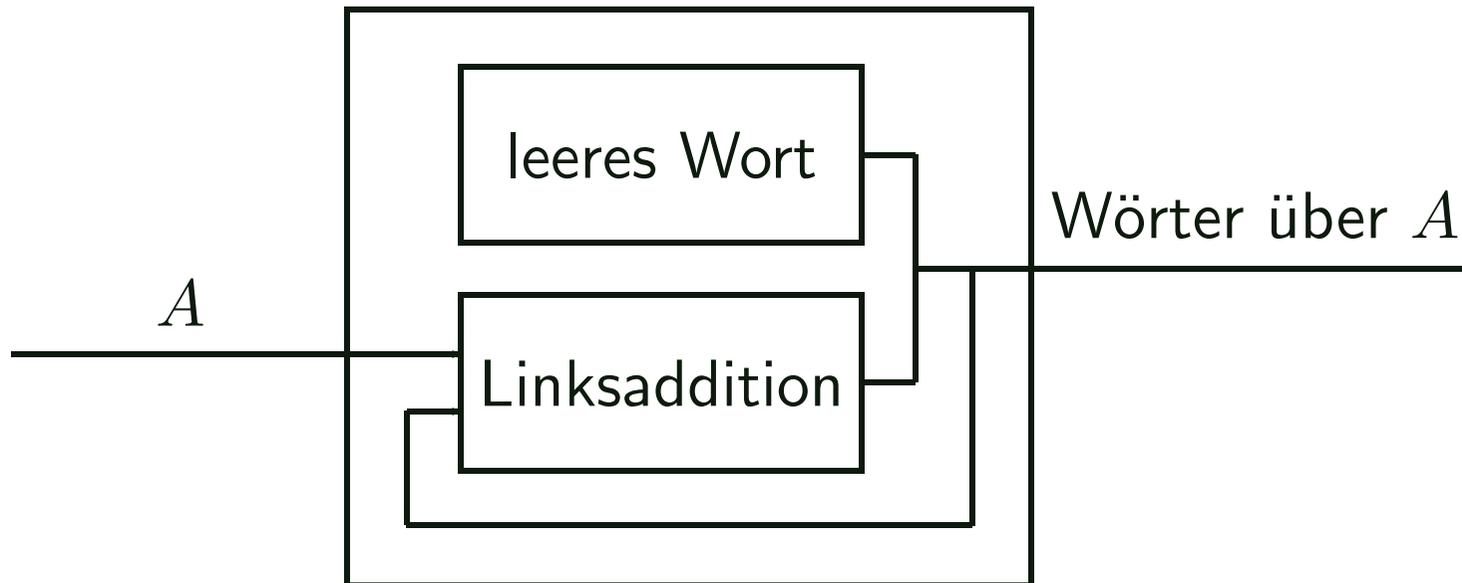


Erzeugung von Wörtern



Eindeutige Zerlegbarkeit von Wörtern

- ▶ Für alle $w \in A^*$:
 - entweder $w = \lambda$, oder
 - $w = xv$ für genau ein $x \in A$ und genau ein $v \in A^*$
- ▶ $head(xv) = x$, $tail(xv) = v$
- ▶ Falls A selbst Wörter enthält, muss die eindeutige Zerlegbarkeit garantiert werden.

Beispiel Für $A = \{B, E, I, BEI, SPIEL\}$ gilt
 $head(B \circ E \circ I \circ SPIEL) = B \neq$
 $head(BEI \circ SPIEL) = BEI$

Bemerkungen über Wörter

- ▶ $A \subseteq A^*$
- ▶ Rechtsaddition kann analog zur Linksaddition definiert werden.
- ▶ **Iterative Darstellung von Wörtern:** $w = x_1 \cdots x_n$
mit $n \in \mathbb{N}$ und $x_i \in A$ ($i = 1, \dots, n$).
- ▶ Falls $A = \{|\}$ repräsentiert A^* die natürlichen Zahlen:

$\lambda = |^0$ repräsentiert 0

$||^n$ repräsentiert $n + 1$

Konkatenation von Wörtern

1. für $v = \lambda$ gilt $\lambda \cdot w = w$,
2. für $v = xu$ mit $x \in A$ gilt $(xu) \cdot w = x(u \cdot w)$.

Beispiel

$$\begin{aligned} aba \cdot cd &= a(ba \cdot cd) = a(b(a \cdot cd)) = a(b(a(\lambda \cdot cd))) = \\ &= a(b(a(cd))) = a(b(acd)) = a(bacd) = abacd \end{aligned}$$

Satz (Assoziativität)

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

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:

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

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:

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

Endliche Automaten

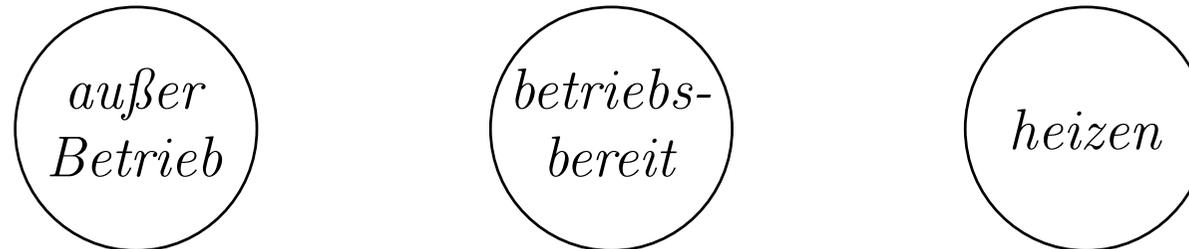
- Einfaches Modellierungswerkzeug (z.B. UML-Statecharts)
- Verarbeiten Wörter/Ereignisfolgen
- Erkennen Sprachen
- Erlauben schnelle Spracherkennung
- **Anwendungsbereiche:** Objektorientierte Modellierung, Model-Checking, Lexikalische Analyse, XML-Parser, Kontrollanweisungen, Spezifikation von Kommunikationsabläufen,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

Funktionsweise

- ▶ Taktweises Arbeiten
- ▶ Eingabe: ein Wort w
- ▶ Lesen von w Zeichen für Zeichen von links nach rechts
- ▶ In jedem Takt wird ein Zeichen gelesen.
- ▶ In jedem Takt befindet sich der endliche Automat in einem seiner endlich vielen *Zustände*.
- ▶ Das Eingabewort w wird akzeptiert, falls sich der Automat nach dem Lesen von w in einem *Endzustand* befindet.

Komponenten

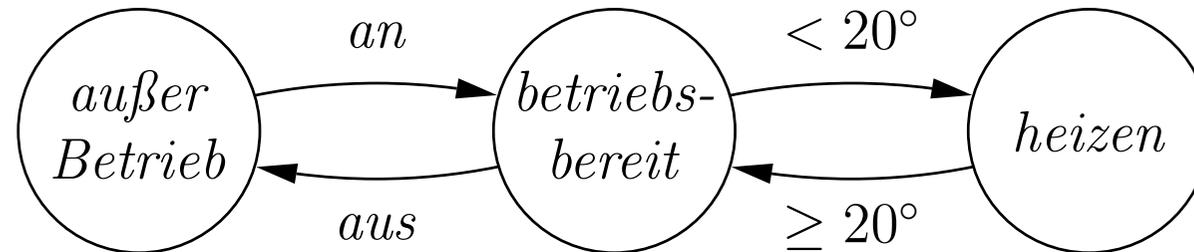
- Zustände



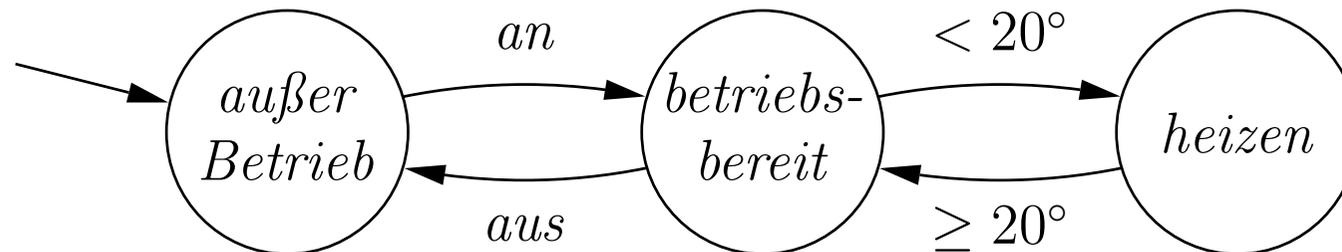
- Eingabealphabet (Menge von potenziellen Ereignissen)

$\{an, aus, < 20^\circ, \geq 20^\circ\}$

- Zustandsüberführungen



- Startzustand



- Endzustände

