Theoretische Informatik 1

Sabine Kuske
Linzer Str. 9a, OAS 3005
Tel.: 2335, 8794
kuske@informatik.uni-bremen.de
www.informatik.uni-bremen.de/theorie

25. Oktober 2006

Gliederung

- Themen dieses Kurses
- Motivation
- Zeichenketten

Themen dieses Kurses

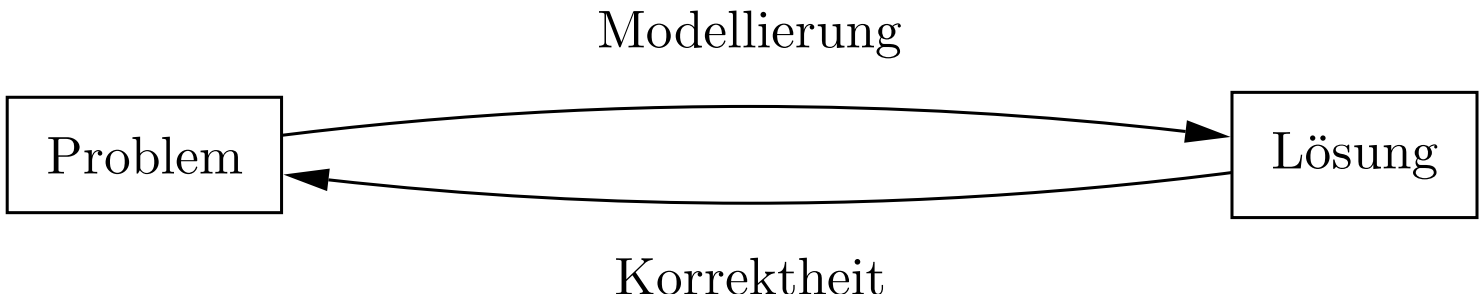
1. **Automatentheorie**
   - Welche Automatentypen gibt es? Welche Probleme können mit Automaten (nicht) gelöst werden? Wie lange dauert es, bis man eine Lösung erhält? Ist die Lösung korrekt? Wie löst man überhaupt Probleme mit Automaten?
2. **Formale Sprachen**
   - Welche Sprache versteht ein Computer? Welcher Automat versteht welche Sprache? Können alle Sprachen das Gleiche ausdrücken?
3. **Berechenbarkeit**
   - Welche Probleme kann ein Computer (nicht) lösen? Was bedeutet berechenbar? Woher weiss man, dass ein Probelm mit einem Computer lösbar ist?

Literatur

1. Skript (`www.informatik.uni-bremen.de/theorie/teach/thi1/`)
2. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Addison-Wesley, 2002. (Automatentheorie, Formale Sprachen)
3. A.J. Kfoury, Robert N. Moll, and Michael A. Arbib. A Programming Approach to Computability. Springer, 1982. (Berechenbarkeit)

Motivation

Informatik beinhaltet die Modellierung von Problemen und deren (korrekte) Lösungen.



Einige wichtige Fragen

- Welche Probleme lassen sich überhaupt mit dem Computer lösen?
- Wie kann die Korrektheit von Lösungen (möglichst automatisch) gezeigt werden?
- Wie lange muss man auf Lösungen warten?
- Mit welcher formalen Sprache kann man was modellieren?

Beispiel: Modellierung einer Heizung

Modelliere eine Heizung,

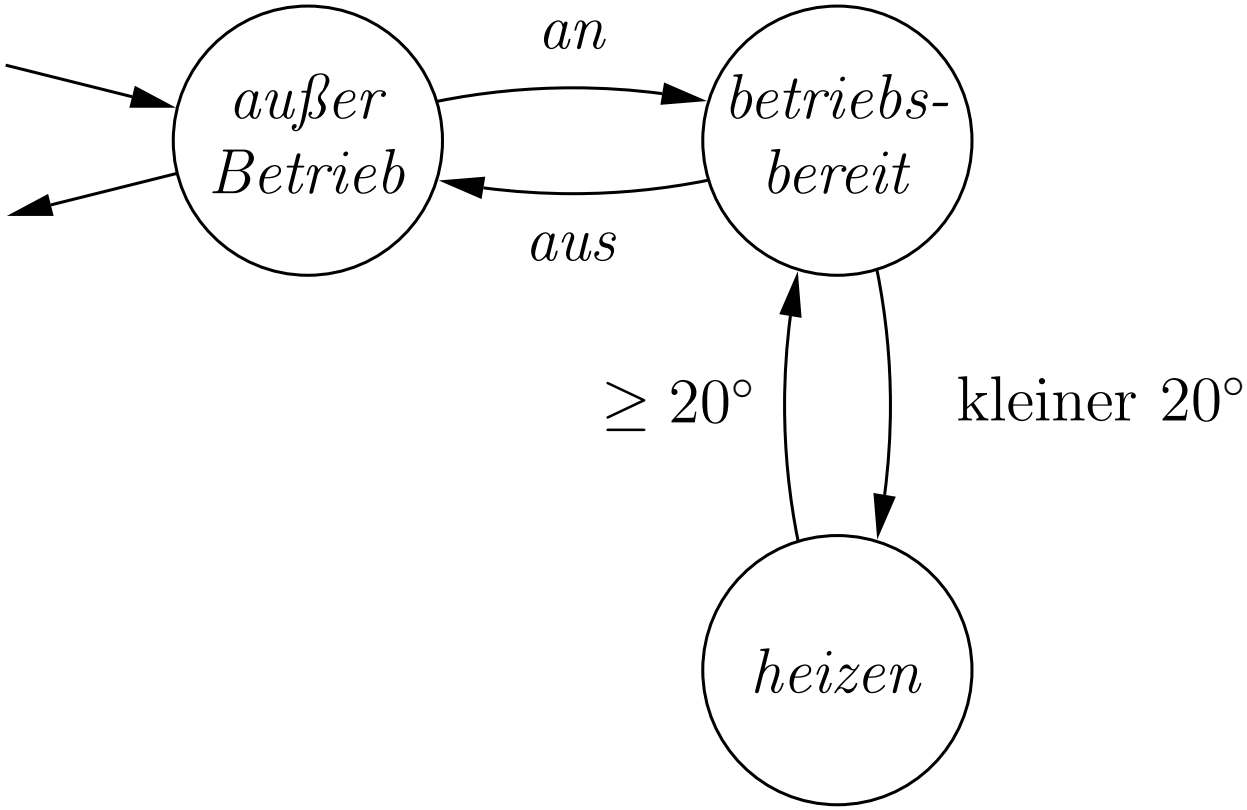
(1) die angeschaltet werden kann und dann betriebsbereit ist,

(2) die bei einer Temperatur unter 20° heizt,

(3) die wieder betriebsbereit wird, wenn die Temperatur mindestens 20° erreicht hat, und

(4) die ausgeschaltet werden kann, wenn sie nicht gerade heizt.

Zustandsgraph Heating



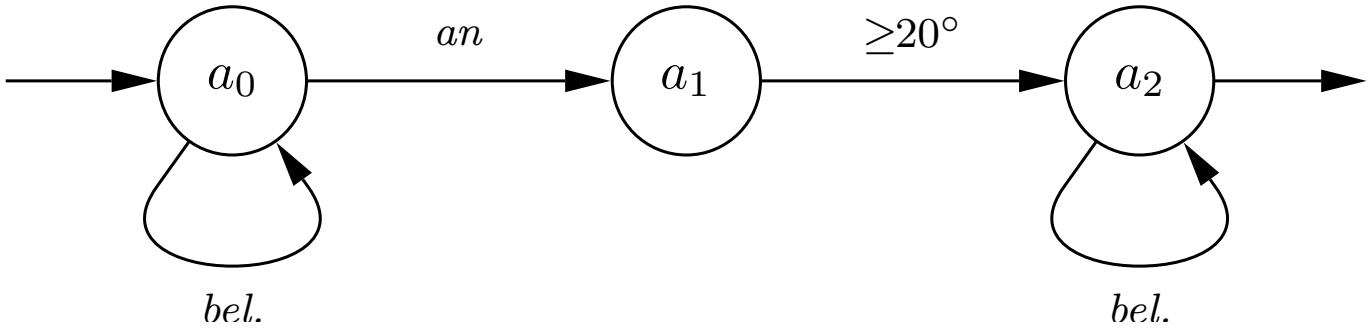
Abläufe (Ereignisfolgen)

- $an\ aus\ an\ aus\ldots$
- $an < 20° \geq 20° < 20° \geq 20°\ aus$
- usw.

$L(heating)$: Menge aller möglichen Abläufe

Verbotene Teilfolgen

(a) $an \geq 20°$

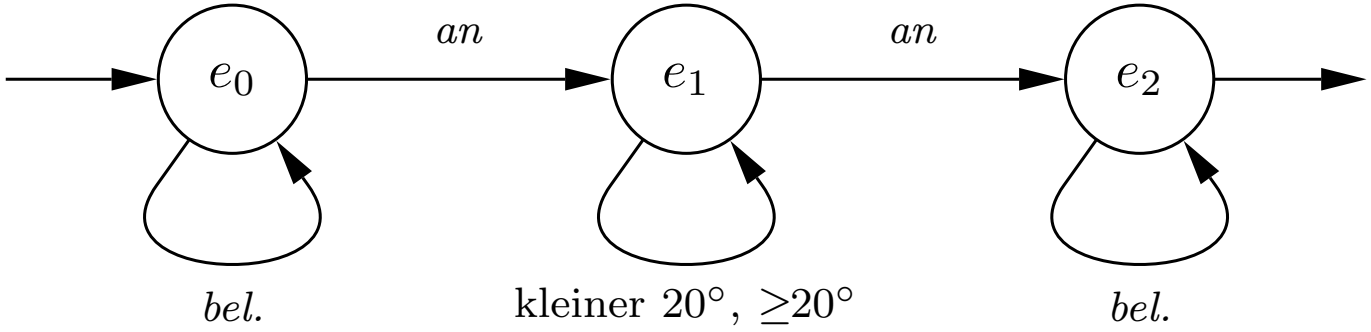


(b) $< 20°$ aus (Zustandsgraph analog)

(c) $< 20°$ $< 20°$ (Zustandsgraph analog)

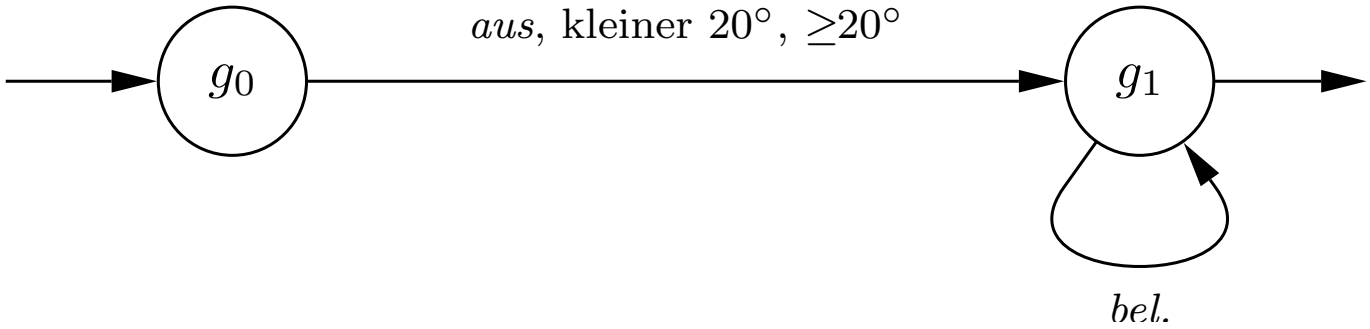
(d) $\geq 20°$ $\geq 20°$ (Zustandsgraph analog)

(e) $an\ u\ an$ (u: Ablauf, der nur $<20°$ und $\geq 20°$ enthält.)



(f) $aus\ u\ aus$ (Zustandsgraph analog)

(g) Abläufe, die nicht mit an beginnen



(h) Abläufe, die nicht mit aus enden (Zustandsgraph analog)

$L_{forbidden}$: Alle verbotenen Abläufe.

Korrektheit

Heating ist korrekt bezüglich $L_{forbidden}$, falls

$$L(heating) \cap L_{forbidden} = \emptyset.$$

Wörter

Wörter sind Zeichenketten

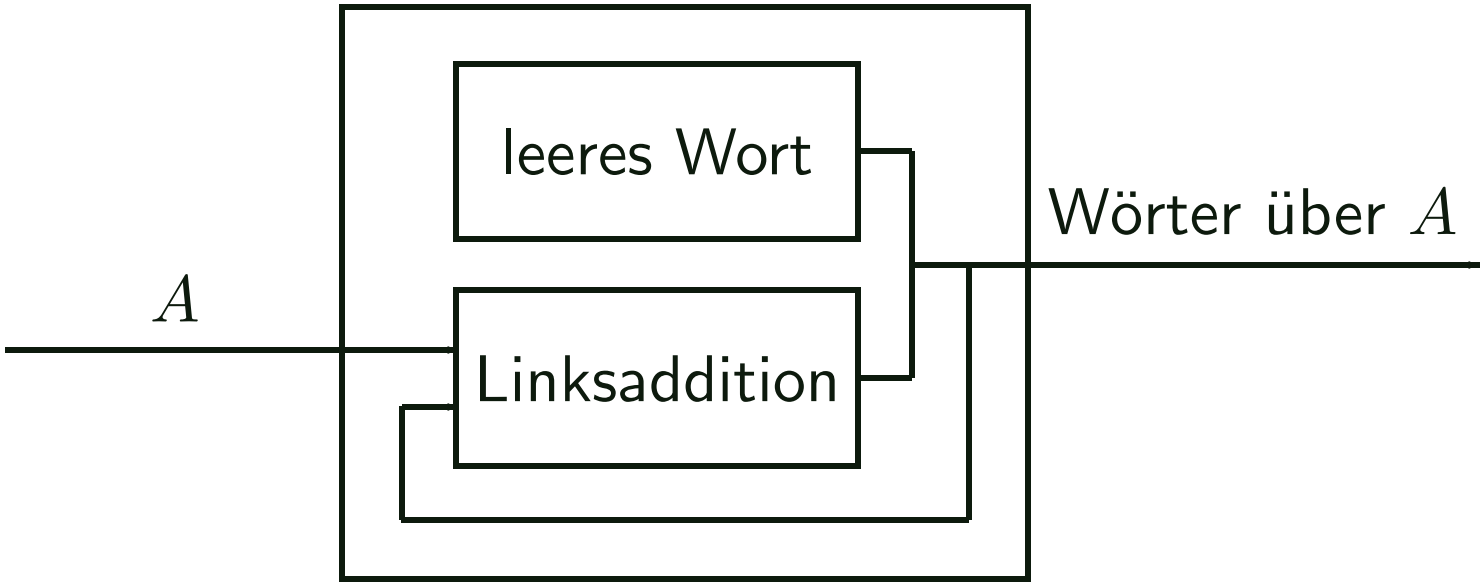
- Beispiele
  - Programmiersprachen:
    Namen, Ausdrücke, Datentypen, Programme,. . .
  - Systemmodellierung:
    mögliche Abläufe, Ereignisfolgen, verbotene Folgen,. . .
  - Natürlche Sprachen:
    Wörter, Sätze, Texte, . . .
  - Textsysteme:
    Schreiben und Lesen, Verändern und analysieren: Cut & Paste,
    Zeichen zählen, Vergleichen, Zusammensetzen, . . .
- Eine der wichtigsten Datenstrukturen

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
Wörter über A
A
Linksaddition

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 $(xu) \cdot w = x(u \cdot w)$.

Länge

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

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