Theoretische Informatik 1

Sabine Kuske
(in Vertretung von Hans-Jörg Kreowski)

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

22. Oktober 2008

Inhaltsverzeichnis

Termine

Vorlesung

Mo 10:00 - 12:00 GW2
Großer Studierraum (B3009)
Hans-Jörg Kreowski

Tutorien

- Mo 08:00 - 10:00 MZH 7250 (ab 3.11.),
- Mo 08:00 - 10:00 MZH 4194(ab 3.11.),
- Di 13:00 - 15:00 MZH 7250 (ab 28.10.),
- Mi 08:00 - 10:00 GW1 B2130(ab 29.10.),
- Mi 10:00 - 12:00 MZH 7250(ab 29.10.),
- Do 08:00 - 10:00 MZH 7220 (ab 30.10.),
- Do 10:00 - 12:00 MZH 7260 (ab 30.10.),

TutorInnen

- Marcus Ermler (maermler@informatik.uni-bremen.de)
- Sebastian Jauert (sjauert@informatik.uni-bremen.de)
- Hans-Jörg Kreowski (kreo@informatik.uni-bremen.de)
- Sabine Kuske (kuske@informatik.uni-bremen.de)

Scheinkriterien

Es gibt zwei Möglichkeiten, den Leistungsnachweis zu erwerben:

1. Regelmäßige Bearbeitung von Übungsaufgaben und Fachgespräch
   - Übungsblätter werden in Gruppen bearbeitet; die Gruppengröße soll 3 nicht überschreiten. Alle 14 Tage erscheint ein Übungsblatt, Bearbeitungszeit: ca. 14 Tage.
   - Jedes Blatt muss mindestens zu 50% richtig bearbeitet werden. Ein Blatt darf nachgebessert werden.
   - Das Fachgespräch dauert ca. 10 Minuten pro Person und dient der Überprüfung der individuellen Leistungsfähigkeit. Es findet i.d.R. gegen Ende der Vorlesungszeit statt.
2. Mündliche Prüfung
   - Eine mündliche Prüfung dauert 20-30 Minuten.

Weitere Infos unter

http://studienzentrum.informatik.uni-bremen.de/

Theoretische Informatik ist ...

wichtig
und
interessant,

denn sie beantwortet Fragen, wie z.B.

> Macht es Sinn, für ein gegebenes Problem eine Lösung zu entwickeln?

Theoretische Informatik ist...

wichtig
und
interessant,

denn sie beantwortet Fragen, wie z.B.

> Welches Modellierungswerkzeug ist für welchen Zweck geeignet?

Theoretische Informatik ist...

wichtig
und
interessant,

denn sie beantwortet Fragen, wie z.B.

> Wie findet man eine möglichst fehlerfreie Lösung?

Theoretische Informatik ist...

wichtig
und
interessant,

denn sie beantwortet Fragen, wie z.B.

> Wie lange muss man höchstens oder mindestens auf ein Ergebnis warten?

Theoretische Informatik ist...

wichtig
und
interessant,

denn sie beantwortet Fragen, wie z.B.

> Für welche Probleme existieren bisher nur viel zu langsame Lösungen?

Lernziele

- Grundlagen der Theoretischen Informatik
- Abstraktes Denken
- Formalisierung von Sachverhalten
- Beweisen

Themen dieses Kurses

- Automatentheorie
- Formale Sprachen
- Berechenbarkeit

Automatentheorie

- Was sind Automaten?
- Was haben Automaten mit Informatik zu tun?
- Welche Automaten gibt es?
- Wofür kann man Automaten (nicht) einsetzen?
- Wie entwirft man korrekt funktionierende Automaten?
- Wie “schnell” sind Automaten?

Formale Sprachen

- Was sind Formale Sprachen?
- Was haben Formale Sprachen mit Informatik zu tun?
- Welche (endlichen) Beschreibungsmittel gibt es für Formale Sprachen?
- Was kann man mit Formalen Sprachen ausdrücken?
- Wie hängen Automaten und Formale Sprachen zusammen?

Berechenbarkeit

- Was ist berechenbar (und was nicht)?
- Was hat Berechenbarkeit mit Computern zu tun?
- Wie zeigt man, dass etwas (nicht) berechenbar ist?

Literatur

Skript
(www.informatik.uni-bremen.de/theorie/teach/thi1/;
(dieser Kurs basiert auf dem Skript.)

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman.
Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.
Addison-Wesley, 2002. (Das Buch gibt es auch auf Englisch)

A.J. Kfoury, Robert N. Moll, and Michael A. Arbib.
A Programming Approach to Computability.
Springer, 1982. (Berechenbarkeit)

Literatur


Uwe Schöning.
Theoretische Informatik – kurz gefasst, 4. Auflage.
Spektrum Akademischer Verlag, 2003.


Gottfried Vossen und Kurt-Ulrich Witt.
Grundlagen der Theoretischen Informatik mit Anwendungen.
Vieweg, 2000.

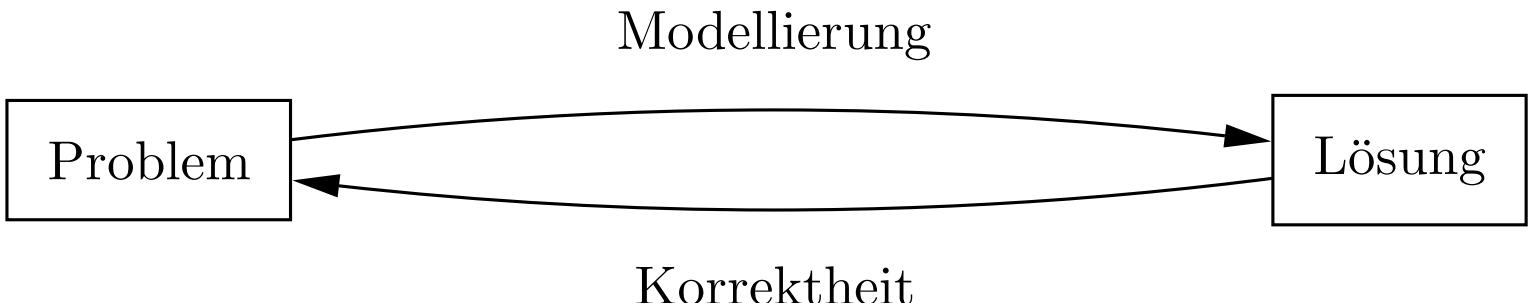

Elaine Rich.
Automata, Computability and Complexity: Theory and Applications.
Prentice Hall, 2007.

Weiteres Material:

- Folien zur Vorlesung
  (`www.informatik.uni-bremen.de/theorie/teach/thi1/`)
- Literaturhinweise im Skript

Informatik beinhaltet...

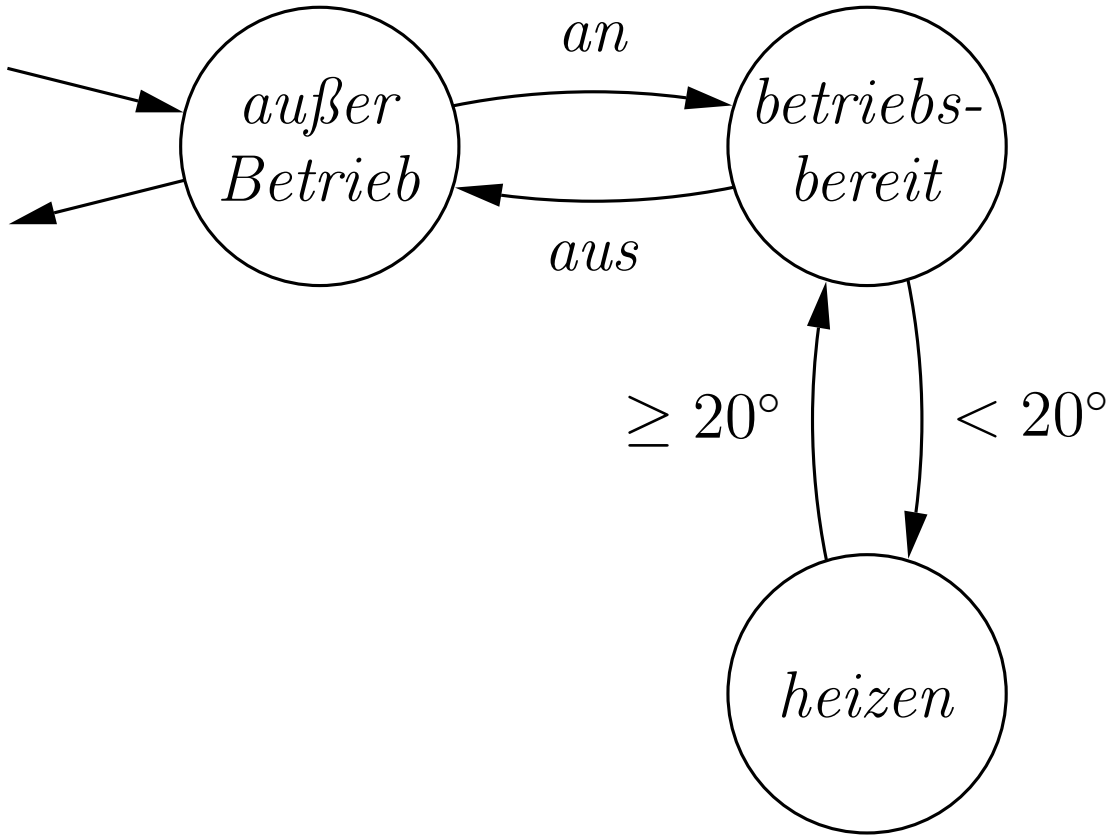
die Modellierung von Problemen und deren (korrekte) Lösungen.



Modelliere eine Heizung, die

1. angeschaltet werden kann und dann betriebsbereit ist,
2. bei einer Temperatur unter 20° heizt,
3. wieder betriebsbereit wird, wenn die Temperatur mindestens 20° erreicht hat, und
4. ausgeschaltet werden kann, wenn sie nicht gerade heizt.

Zustandsgraph Heating



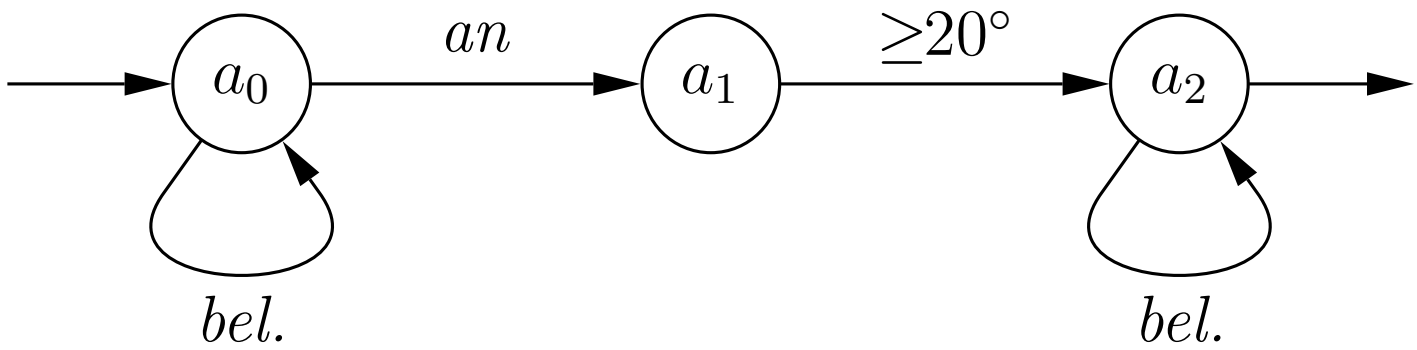
Abläufe (Ereignisfolgen)

- *an aus an aus*...
- *an* $< 20^\circ \geq 20^\circ < 20^\circ \geq 20^\circ$ *aus*
- usw.

$L(heating)$: Menge aller möglichen Abläufe

Verbotene Teilfolgen

(a) *an* $\geq 20°$

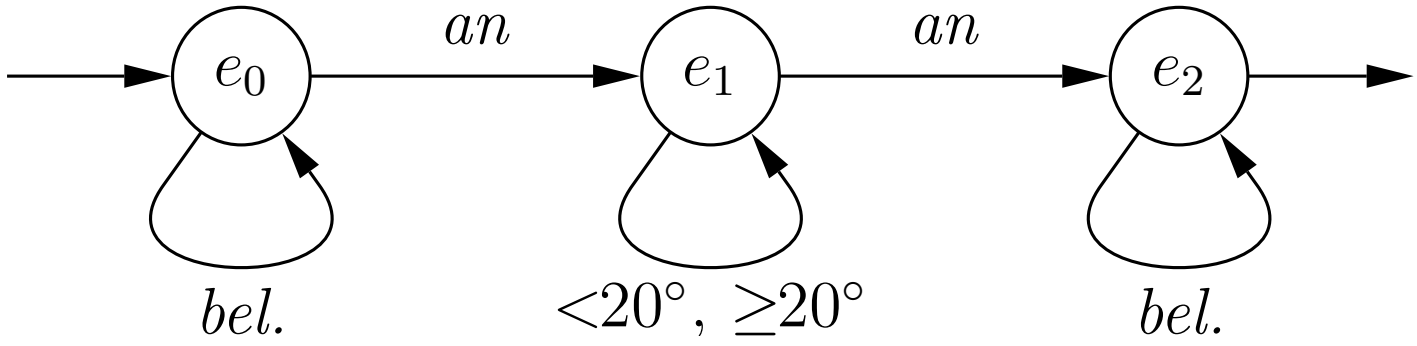


(b) $<20°$ *aus* (Zustandsgraph analog)

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

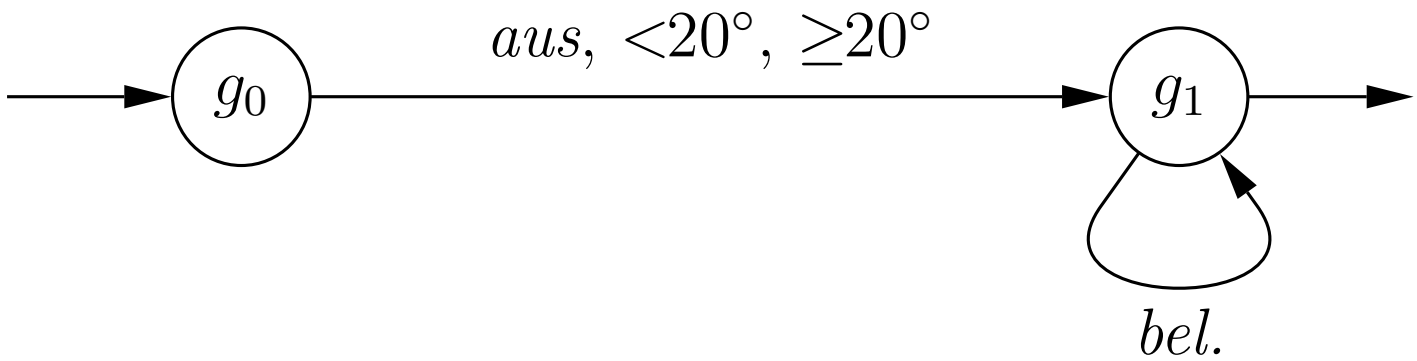
(d) $\geq 20°$ $\geq 20°$ (Zustandsgraph analog)

(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)

Korrektheit

$L_{forbidden}$: Alle verbotenen Abläufe.

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

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