

Informatik III



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer

Wintersemester 2006/07

1. Vorlesung

26.10.2006

schindel@informatik.uni-freiburg.de



Kapitel I Motivation

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Motivation



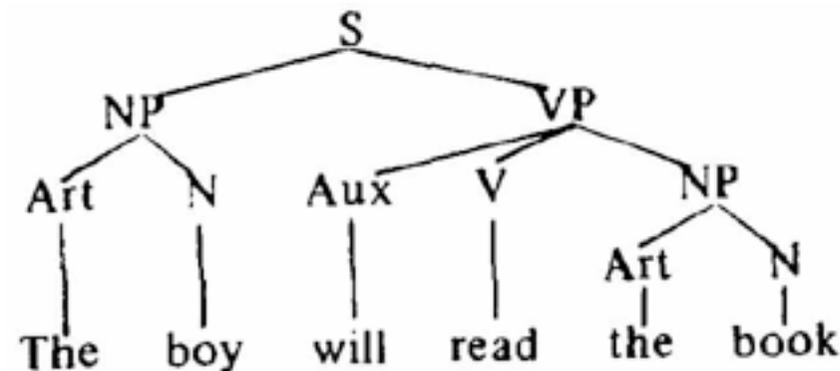
Automaten und Formale Sprachen

- Was kann man mit einem endlichen Speicher berechnen?
- Wie kann man diese Probleme charakterisieren?
- Was sind Automaten?



- Wie kann man die Grammatik einer Sprache formal fassen?

- Wie charakterisiert man die Komplexität einer Grammatik?





Berechenbarkeitstheorie: Richtig/Falsch?

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- Computer können alles berechnen
 - (wenn sie genügend schnell und groß sind).



Foto: Forschungszentrum Jülich



Berechenbarkeitstheorie Richtig/Falsch?

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- Manche Probleme kann man nur mit speziellen Programmiersprachen lösen.



– © Luis Guillermo



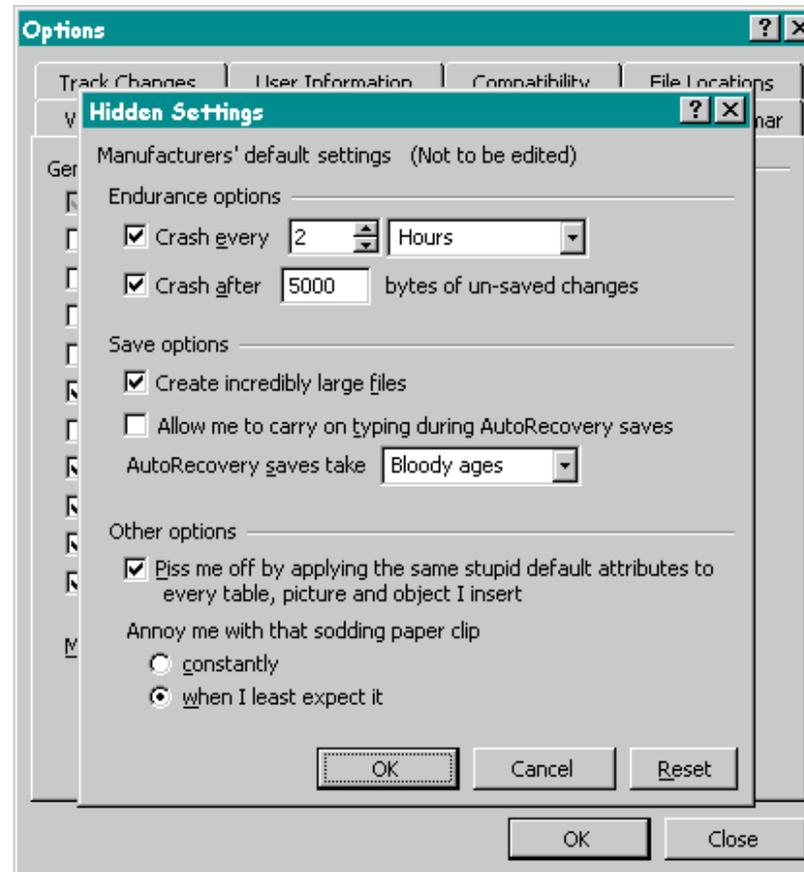


Berechenbarkeitstheorie

Richtig/Falsch?

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Durch genügend scharfes Hinsehen kann man jedes Programm verstehen.**



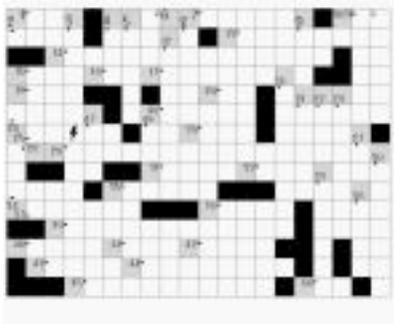


Komplexitätstheorie

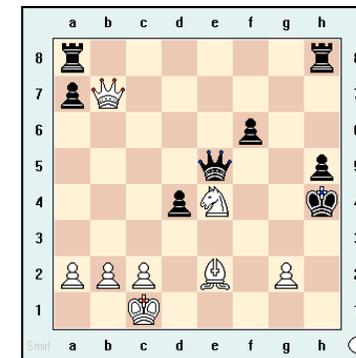
		3		5				
				1	3			6
5	7		6					
7	5						9	
1	2					5		3
	6						4	8
				4			1	9
8			9	7				
			3	2				

➤ Was ist schwieriger?

- Sudoku
- Kopfrechnen
- Kreuzworträtsel
- Schach
- Sokoban
- Puzzle



© Netzhaut





➤ **Kann ein Computer die Faktorisierung einer 500-stelligen Zahl berechnen?**

– $6 = 2 \cdot 3$

– $100 = 2 \cdot 2 \cdot 5 \cdot 5$

– 132948716395817361487361987436198357612351301305982613409860
193856109835764019386593856109357861039456203954871049238761
095348613209861304986134208967130298610398644314897109861092
385610938461605938612019034865701593610923586170349810561392
487103948163598163205916530193824710923561039561543761023985
613095661230413561359816301897356109235876103458976203574610
239678501394561832650193856710923576103957163509135861034576
190325861039586130958610234987123409861539086132509816350918
356103529861035986153 = ? · ? · ? · ? ... ?

➤ **Wenn das schwierig ist, wie schwierig ist es zu entscheiden, ob das eine Primzahl ist?**

➤ **Wie schwierig ist es, Koffer in einen Kofferraum zu packen?**



Kapitel II Organisation

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Organisation



Organisation

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

➤ Web-Seite

- <http://cone.informatik.uni-freiburg.de/teaching/vorlesung/informatik-III-w06/>

➤ Vorlesungen

- Donnerstag, 11-13 Uhr, HS 036, Geb. 101
- Freitag, 11-13 Uhr, HS 036, Geb. 101

➤ Übungen (Beginn: ab 30.10.2006)

- Montag 14-16 Uhr
 - SR 00-006, Geb. 051: Christoph Betz
 - SR 03-026, Geb. 051: Matthias Heizmann
 - SR 02-017, Geb. 052: Daniel Fader
- Dienstag 11-13 Uhr
 - SR 00-034, Geb. 051: Patrick Eyerich
- Dienstag 14-16 Uhr
 - SR 03-026, Geb. 051: Omar Hentate
 - SR 02-017, Geb. 052: Andreas Knab



Prüfung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

➤ Klausur

- Die Prüfung Informatik III findet statt am
- 21.03.2007, 10.00 - 12.00, Räume 026 & 036 (evtl. 010/14) in Geb.101

➤ Prüfungsanmeldung

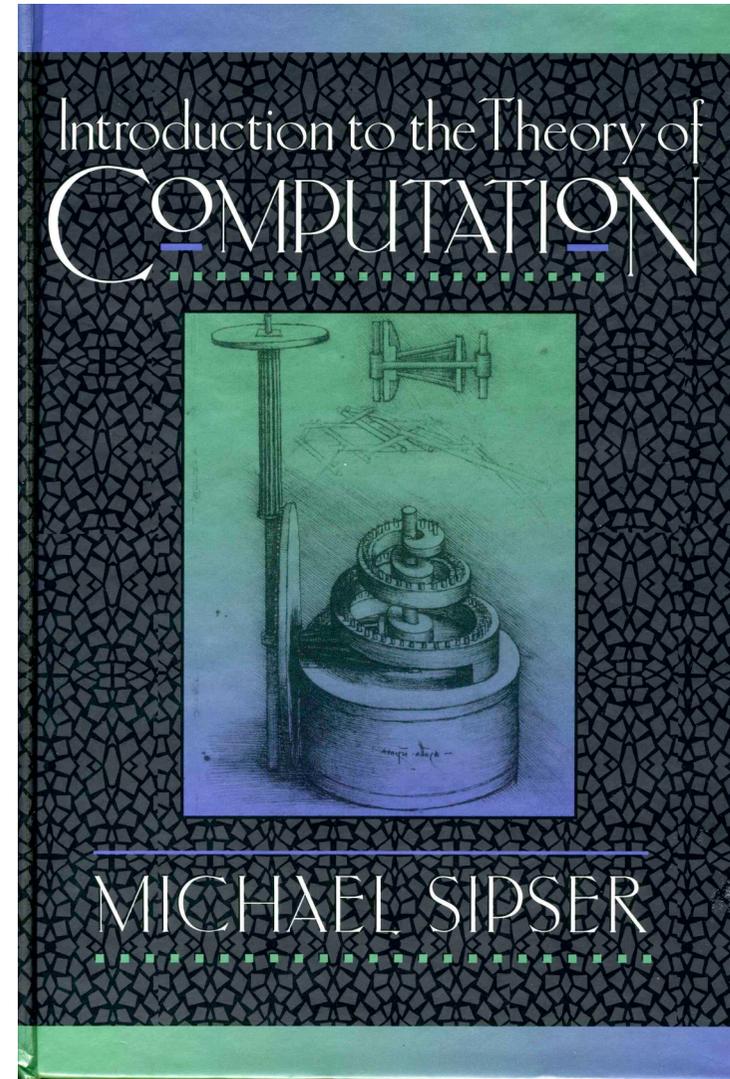
- für die Studierenden der Studiengänge Mikrosystemtechnik und Informatik erfolgt on-line über
- http://www.informatik.uni-freiburg.de/dekpaamt/allgemeines/online_anmeldung.htm
- An- und Abmeldungen sind von 01.10.2006 bis 21.01.2007 möglich.



Materialien

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Powerpoint/PDF-Foliensätze**
 - vor der Vorlesung auf der Web-Site
- **Lecturnity-Mitschnitte**
 - auf der Web-Site und im Vorlesungssystem
- **Literaturhinweis**
 - Vorlesung folgt dem Buch
 - Michael Sipser: *Introduction to the Theory of Computation*, PWS Publishing Company, 1997





Wie erreicht man uns?

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

1. Wie erreicht man mich?

- nach der Vorlesung
- per E-Mail: **schindel@informatik.uni-freiburg.de**
- per Telefon: 0761 - 203 8181
- in der Sprechstunde: Dienstag 11-12 Uhr (002-007, Geb. 051)
 - oder nach Vereinbarung

2. Wie erreicht man irgendeinen?

- Tutoren: E-Mail-Adressen werden in der Übung bekannt gegeben
- Auf der Webseite gibt es auch ein **Forum**,
 - um mit Leidensgenossen Kontakt aufzunehmen
 - um sich zur Übung anzumelden



Webseite

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

➤ URL

- <http://cone.informatik.uni-freiburg.de/teaching/vorlesung/informatik-III-w06/index.html> ([link](#))
- oder über Google: Schindelbauer III

➤ Inhalt

- Vorlesungsfolien als PPT/PDF vor der Vorlesung
- Vorlesungsaufzeichnungen mit Lecturnity
- Übungsaufgaben
- Literatur
- Klausur
- Termine
- Forum



Übungsbetrieb

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

➤ Gruppeneinteilung über das Forum

- Anmeldung bis Freitag möglich
- Siehe Web-Seite

➤ Übungsaufgaben

- Wochenübungsblatt
 - erscheint jeden Donnerstag auf der Webseite
 - Bearbeitung freiwillig
 - Vorrechnen ergibt Bonuspunkte
- Präsenzübungsblatt
 - in der Übung
 - Bearbeitung freiwillig
 - ergibt Bonuspunkte bis maximal einer Notenstufe
- Grundlage für mündliche Prüfung/schriftliche Klausur





Literatur

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Introduction to the Theory of Computation, von: Michael Sipser, PWS, 1997**
- **Introduction to Automata Theory, Languages, and Computation, von: John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Addison Wesley, 2001**
 - (auch übersetzt: Einführung in die Automatentheorie, Formale Sprachen und Komplexität, ... , Pearson Studium, 2002)
- **Computers and Intractability - A Guide to the Theory of NP-Completeness, von: Michael R. Garey, David S. Johnson, W.H. Freeman & Company, 1997**
- **Theoretische Informatik, von: Christel Baier, Alexander Asteroth, Pearson Studium, 2002**
- **Theoretische Informatik - Eine algorithmenorientierte Einführung, von: Ingo Wegener, Teubner, 1993**
- **The Theory of Computation, von Bernard M. Moret, Pearson Education, 1998**
- **Computational Complexity, von: Christos H. Papadimitriou, Addison-Wesley, 1994**
- **Theoretische Informatik - kurzgefaßt, von: Uwe Schöning, Spektrum, akad. Verlag, Heidelberg, 1997**
- **Elements of the Theory of Computation, von: Harry R. Lewis, Christos H. Papadimitriou, Prentice Hall, 1998**
- **Theory of Computing - A Gentle Introduction, von: Efim Kinber, Carl Smith, Prentice Hall, 2001**



Literatur Algorithmen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Algorithmen, von: Robert Sedgewick, Pearson Studium, 2002**
 - (übersetzt aus dem Englischen, gibt es in verschiedenen Ausgaben mit Schwerpunkten in Java, C, C++)
- **Algorithmen, von: Ellis Horowitz, Sartaj Sahni, Springer, 1981**
- **The Design and Analysis of Computer Algorithms, von: Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman Addison Wesley, 1974**
- **Data Structures and Algorithms, von: Alfred V. Aho, J. D. Ullman, J. E. Hopcroft, Addison Wesley, 1983**
- **Algorithmik - Theorie und Praxis, von: Gilles Brassard, Paul Bratley, Prentice Hall, 1993**
- **Approximation Algorithms for NP-Hard Problems, von: Dorit S. Hochbaum, Wadsworth Publishing Company, 1997**
- **Randomized Algorithms, von: Rajeev Motwani, Prabhakar Raghavan, Cambridge University Press, 1995**
- **The Art of Computer Programming (Vol.1-3), Donald Knuth, Addison Wesley 1997/1998**
 - 1: *Fundamental Algorithms*
 - 2: *Seminumerical Algorithms*
 - 3: *Sorting and Searching*



Literatur

Mathematik für Informatiker

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

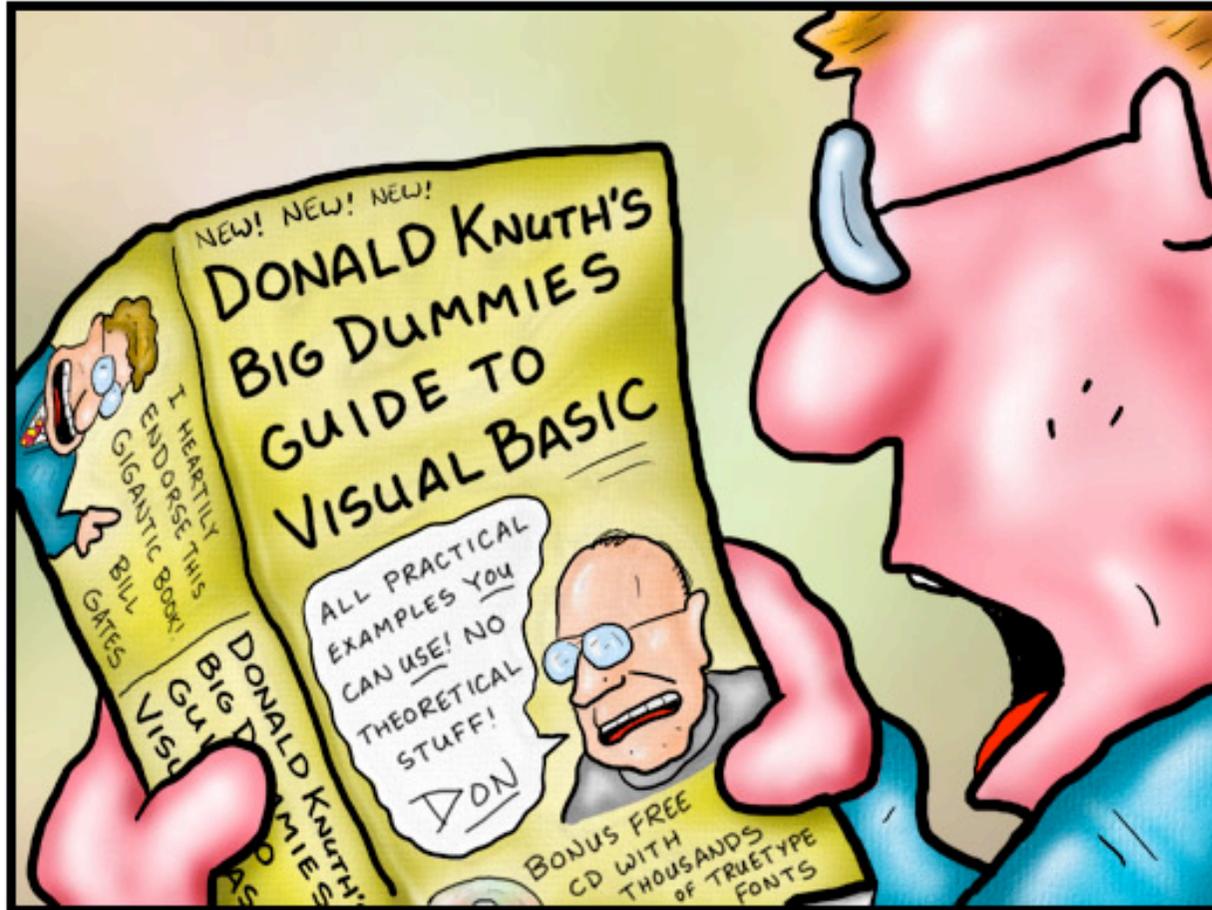
- **Concrete Mathematics, von: Ronald L. Graham, Donald E. Knuth, Oren Patashnik, Addison-Wesley, 1994**
- **Diskrete Mathematik für Informatiker, von: Rod Haggarty, Pearson Studium, 2004**



Noch ein Buch

DOCTOR FUN

10 Feb 2000



Copyright © 2000 David Farley, d-farley@metalab.unc.edu
<http://metalab.unc.edu/Dave/drfun.html>

This cartoon is made available on the Internet for personal viewing only. Opinions expressed herein are solely those of the author.

Don Knuth finally sells out.



Vorlesung

➤ **Orientiert sich stark an Sipers Buch**

- Introduction to the Theory of Computation, von: Michael Sipser, PWS, 1997
- Buch für die Vorlesung nicht notwendig

➤ **Gliederung**

- Endliche Automaten
- Formale Sprachen
 - Reguläre Sprachen
 - Kontextfreie Sprachen
- Berechenbarkeitstheorie
- Komplexitätstheorie

➤ **Skript:**

- Es wird kein Skript zur Verfügung gestellt

➤ **Folien:**

- Können von der Web-Seite heruntergeladen werden
- Kurz vor (!) der Veranstaltung verfügbar



Kapitel III Reguläre Sprachen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

Endliche Automaten



Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer



Ein einfacher Automat

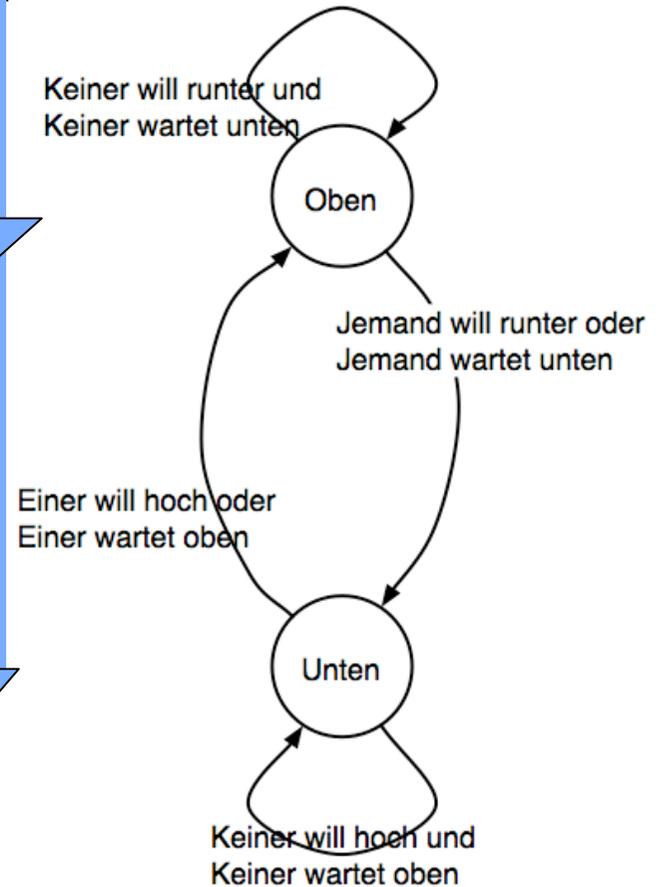
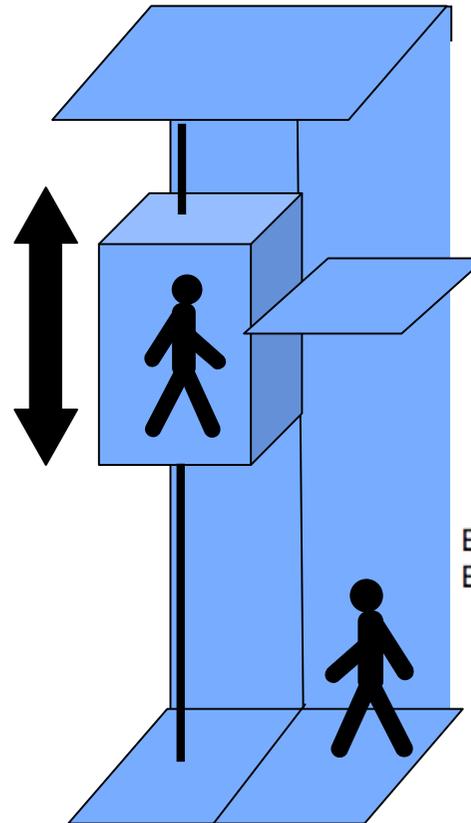
➤ Lift mit zwei Stockwerken

–Eingaben:

- Im Lift will jemand nach oben
- Im Lift will jemand nach unten
- Unten Bedarf
- Oben Bedarf

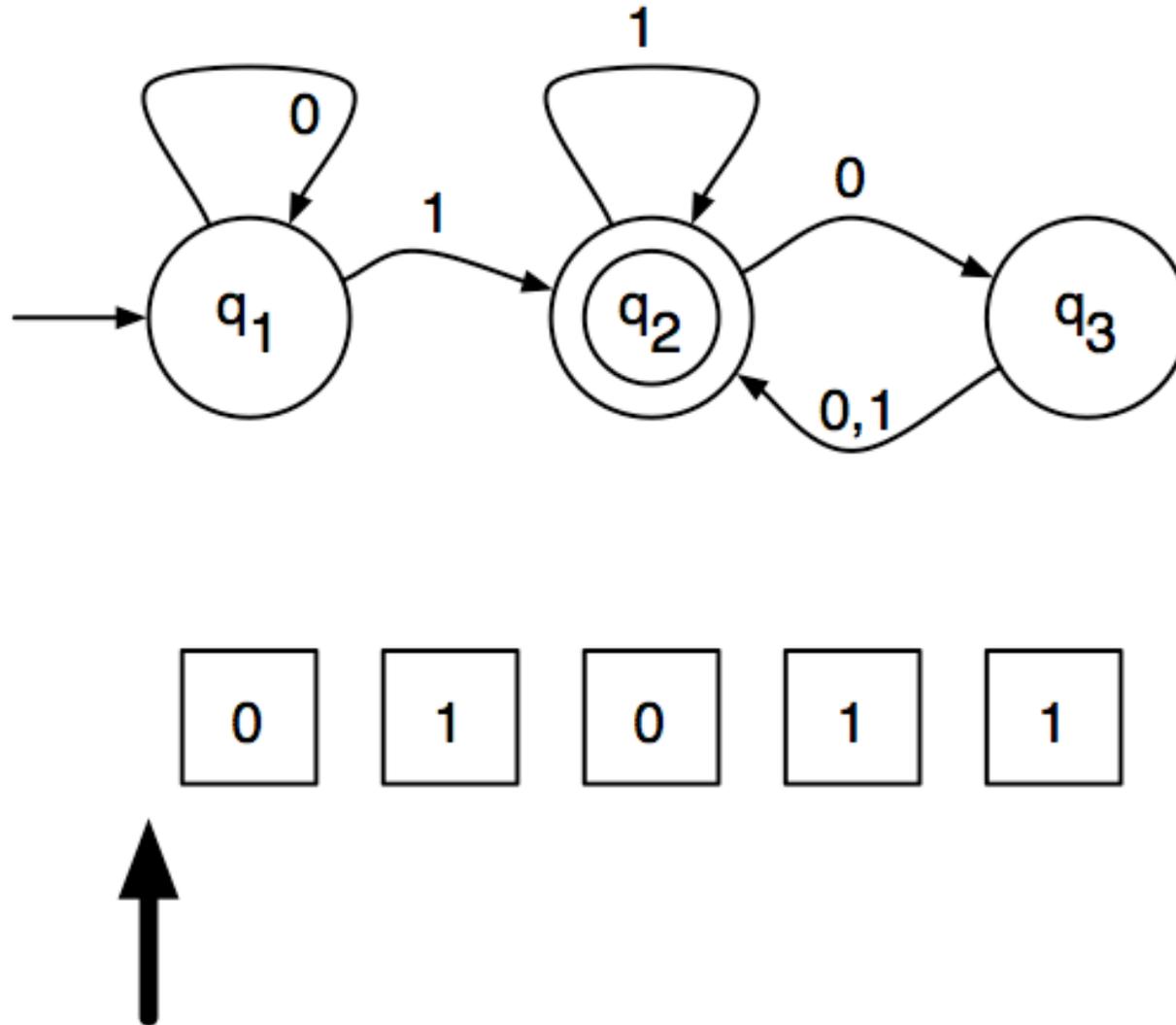
–Zwei Zustände:

- Lift oben
- Lift unten



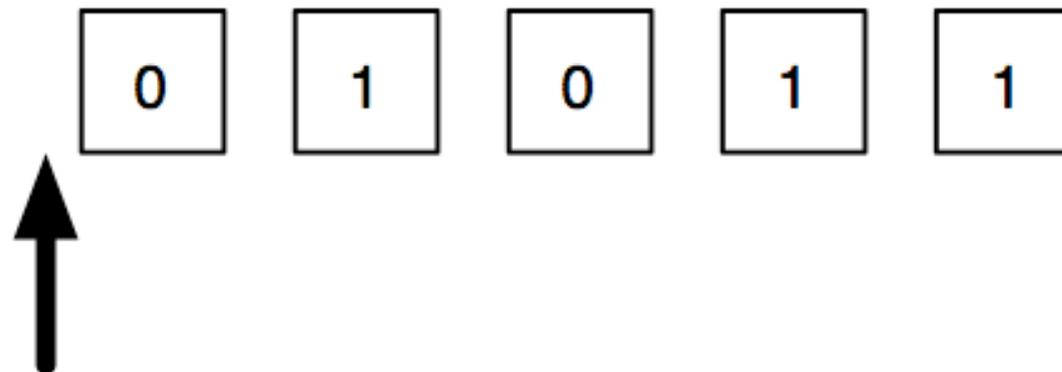
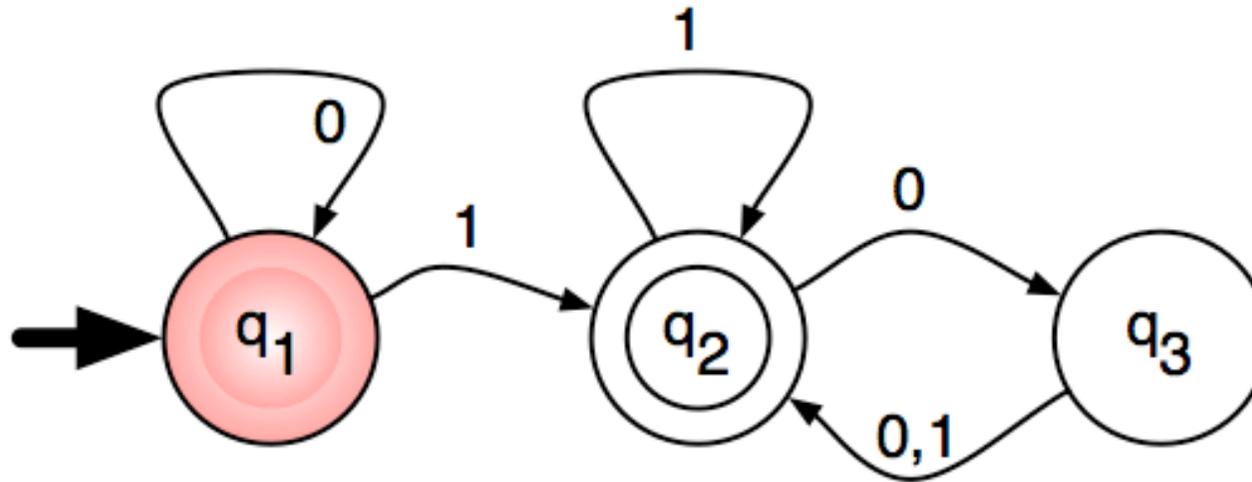


Ein Beispiel



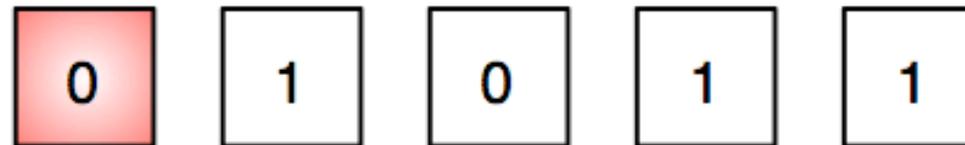
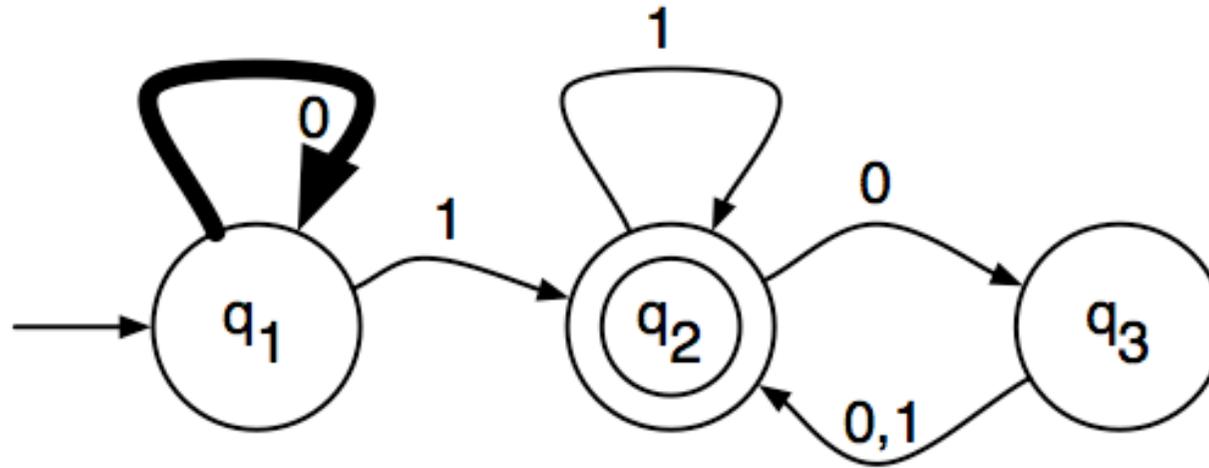


Startzustand q_1



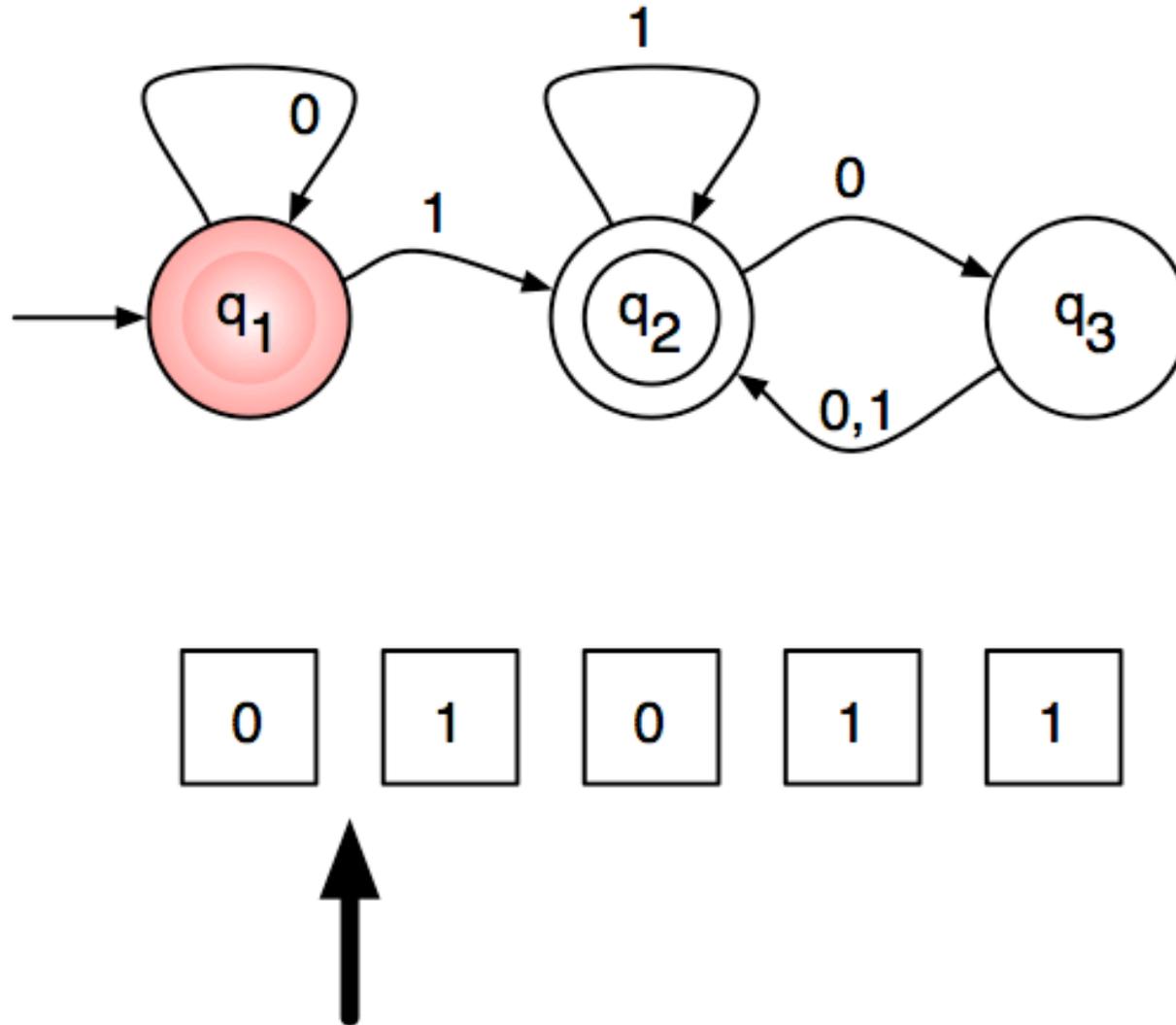


Lesen des ersten Zeichens: 0



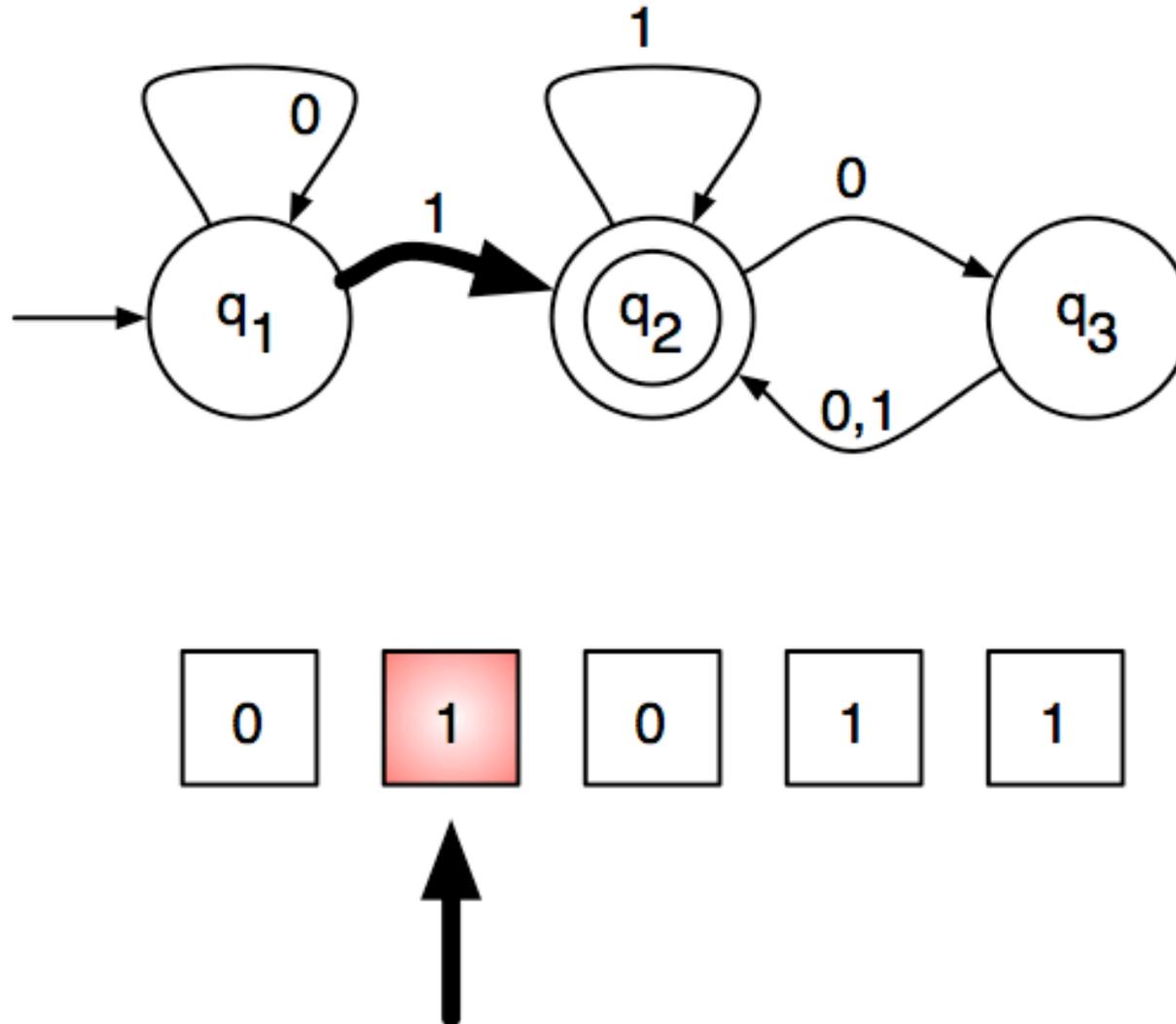


Zustandsübergang zu q_1



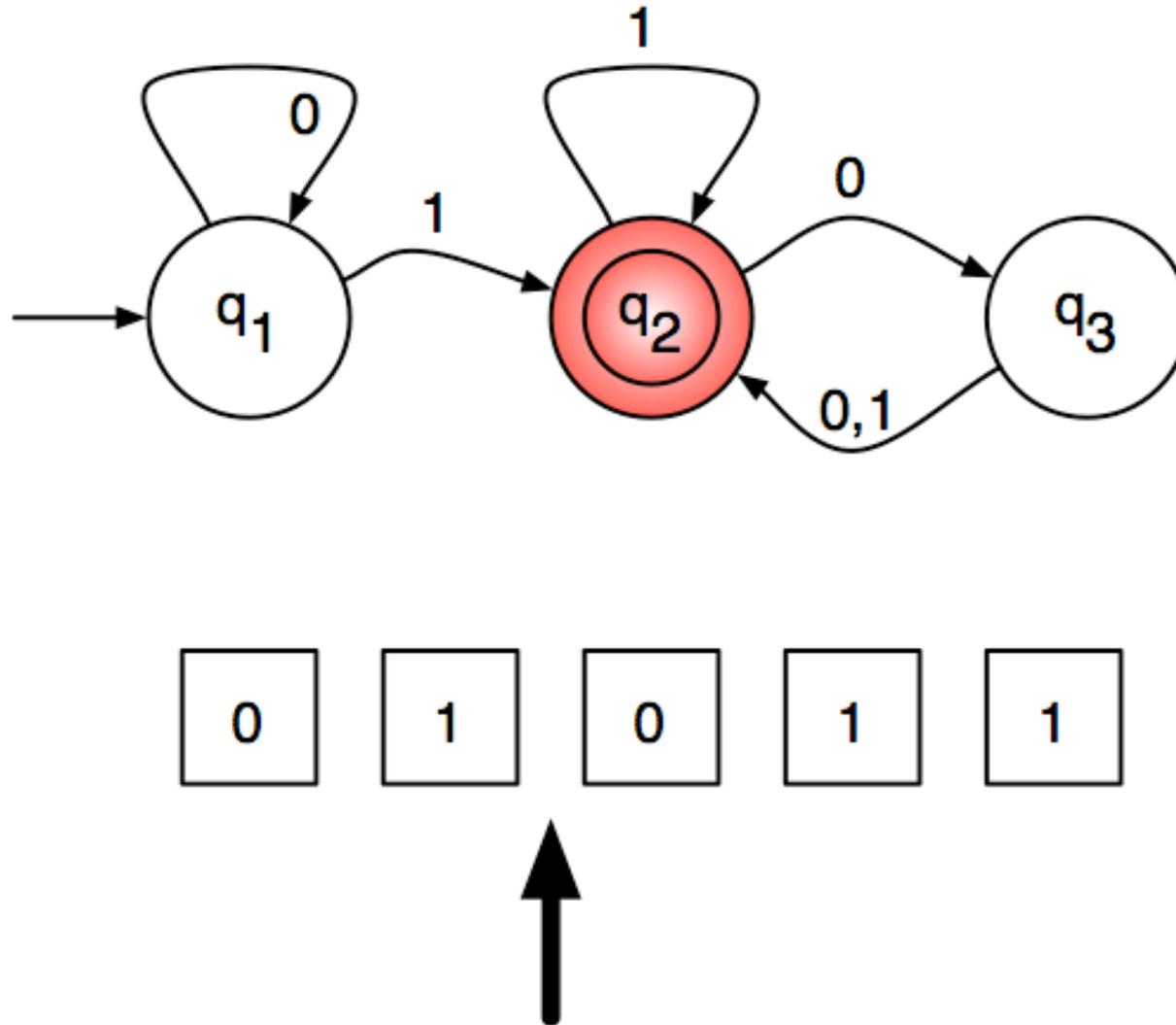


Lesen des 2. Zeichens: 1



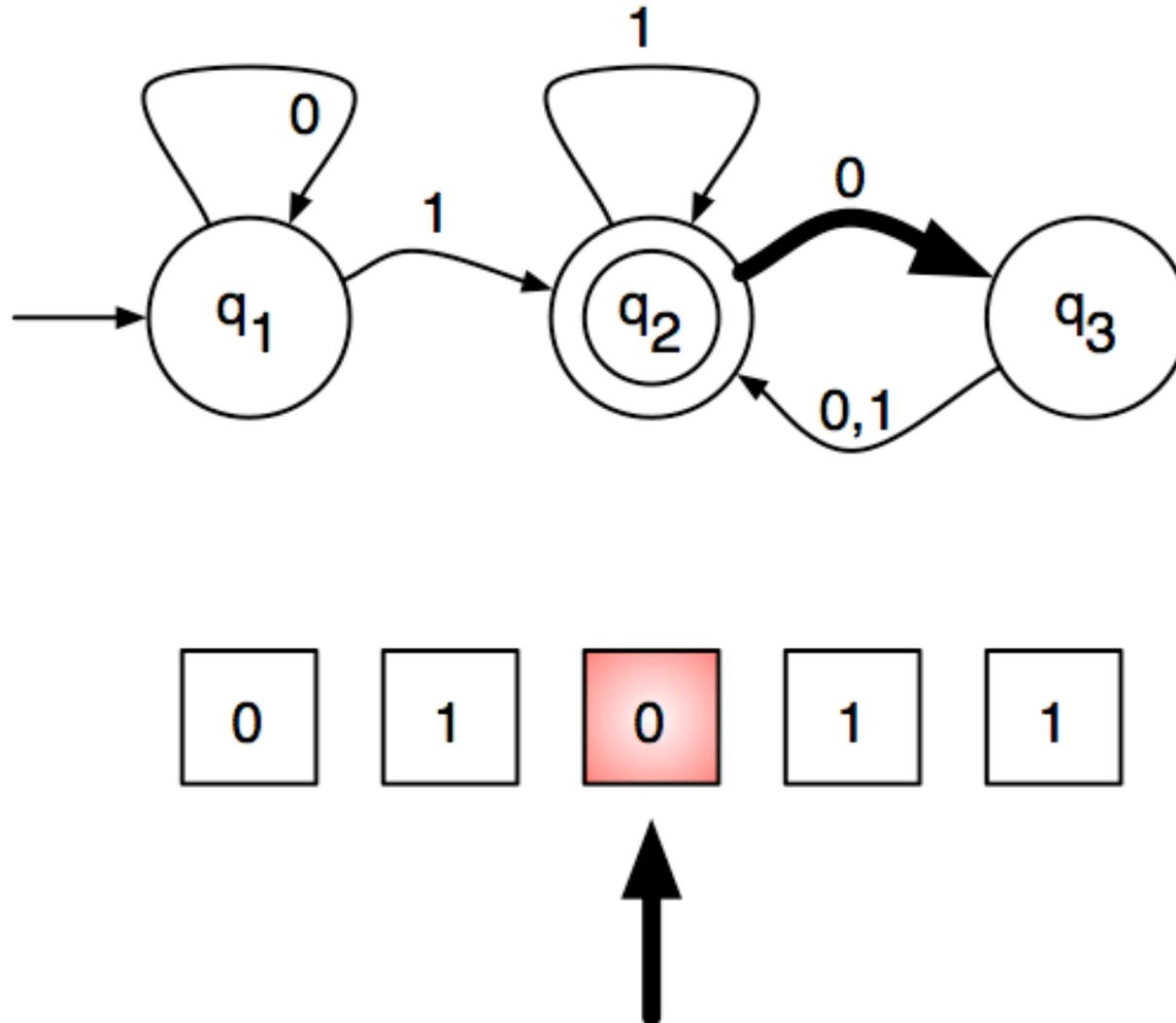


Neuer Zustand: q_2



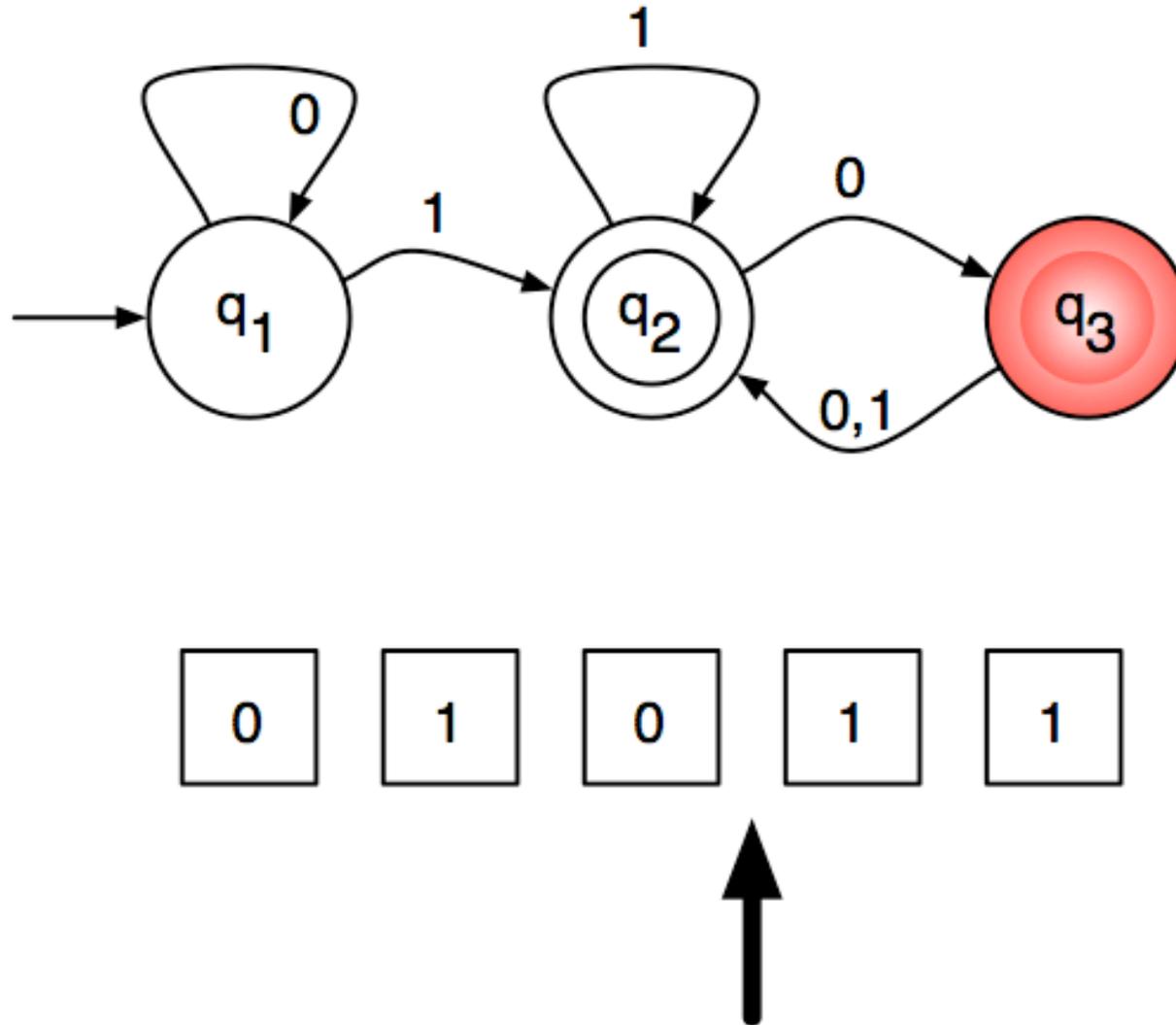


3. Zeichen: 0



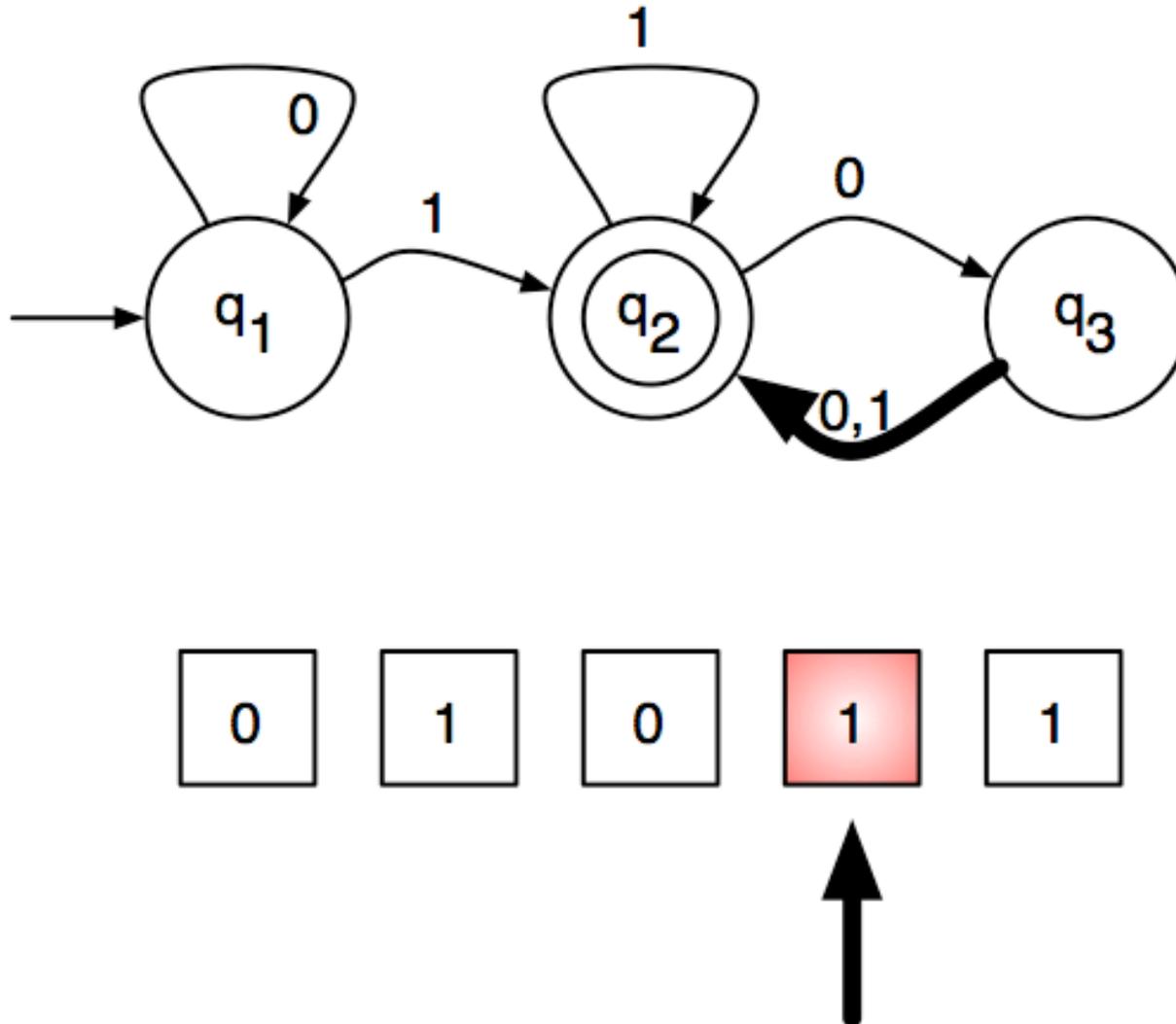


Neuer Zustand: q_3



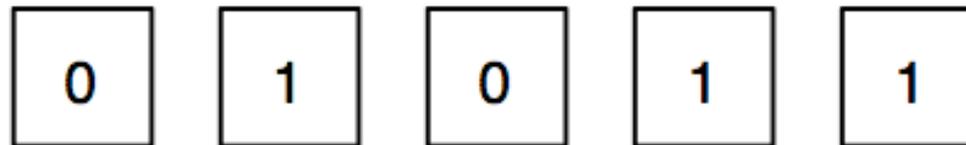
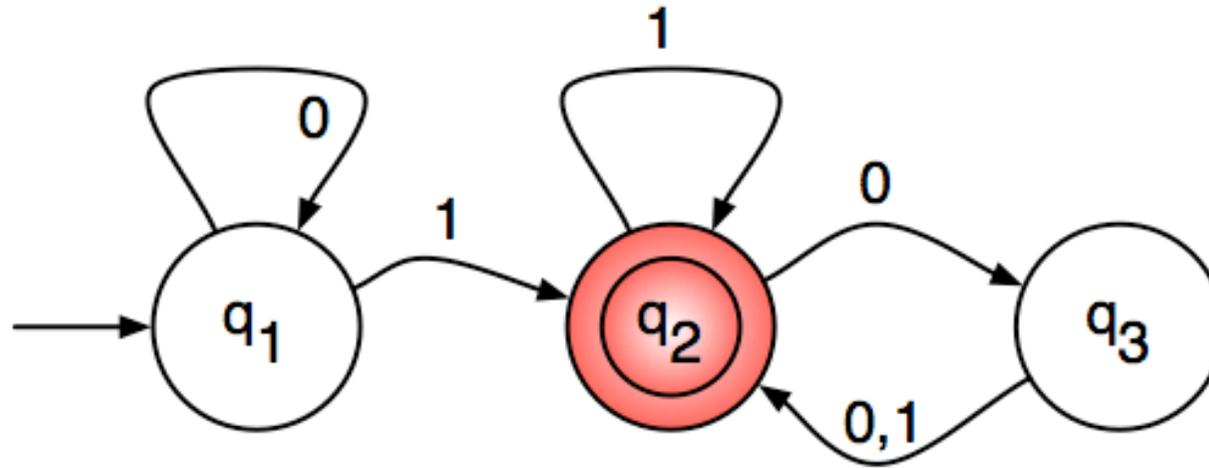


Lesen von "1"



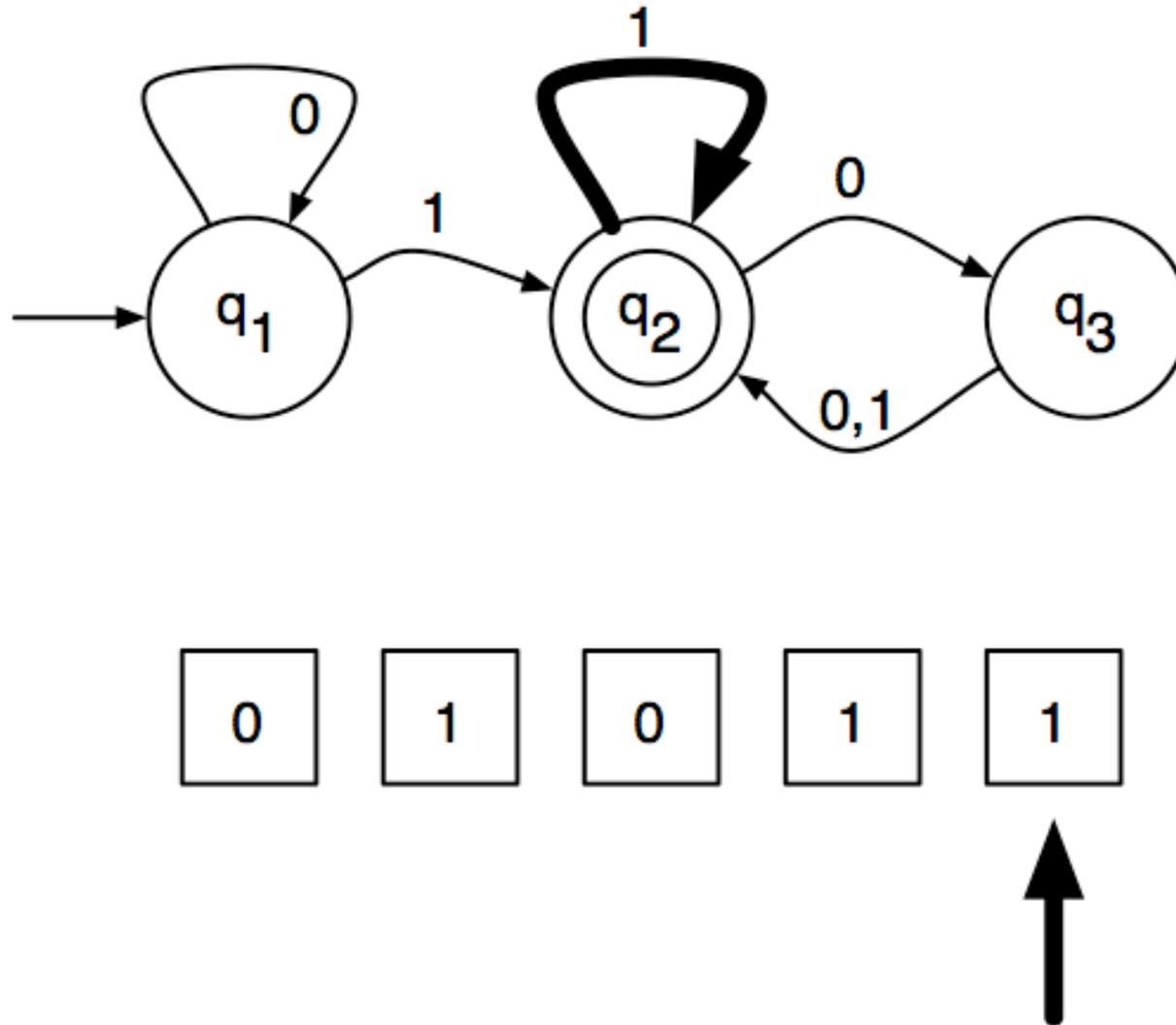


Zustandsübergang zu q_2



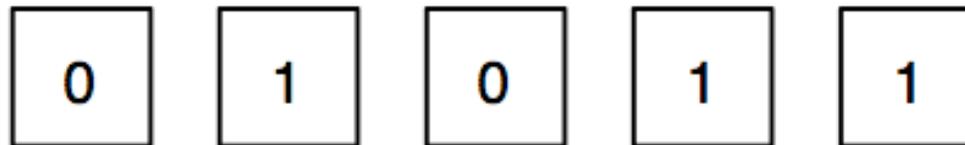
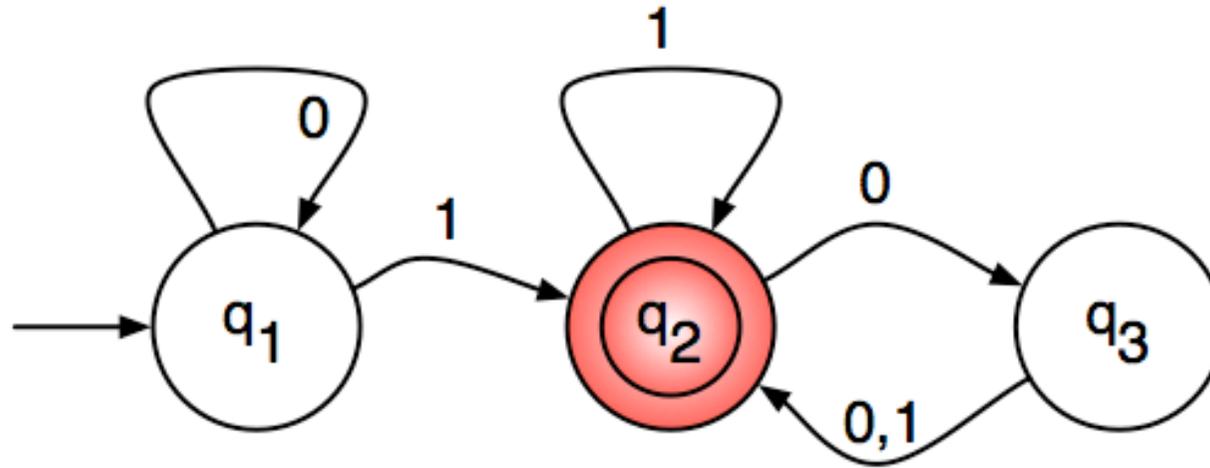


Letztes Zeichen: "1"





Wort "01011" wird akzeptiert





Automaten akzeptieren Sprachen

- **Ein Alphabet Σ besteht aus einer endlichen Menge von Zeichen, z.B.**
 - $\Sigma_1 = \{a,b,c\}$
 - $\Sigma_2 = \{0,1\}$
 - $\Gamma = \{0,1,x,y,z\}$
- **Eine Zeichenkette (String/Wort) ist eine endliche Folge (Tupel) von Zeichen, z.B.**
 - $w = abba$
 - Notation: $w_1=a, w_2=b, w_3=b, w_4=a$
 - Die Länge eines Worts wird mit $|w|$ beschrieben: $|w| = 4$
- **Σ^* bezeichnet die Menge aller Zeichenketten über Alphabet Σ**
 - z.B.: “abba” $\in \{a,b\}^*$
 - Die leere Zeichenkette wird mit ε bezeichnet.
 - Es gilt: $|\varepsilon| = 0$
- **Eine Menge von Zeichenketten wird als Sprache bezeichnet**



Worte und Sprachen

➤ Worte:

- "010010", "otto", "abba"

➤ Sprachen:

- {abba, bab, aa}

$$\begin{aligned} & - \{w \in \{0, 1\}^* \mid (\forall i \in \{1, \dots, |w|\} : w_i = 0) \vee (\forall i \in \{1, \dots, |w|\} : w_i = 1)\} \\ & = \{\epsilon, 0, 1, 00, 11, 000, 111, 0000, 1111, 00000, 11111, \dots\} \end{aligned}$$

$$\begin{aligned} \text{Palindrom} & = \{w \in \{a, b\}^* \mid \forall i \in \{1, \dots, |w|\} : w_i = w_{|w|-i+1}\} \\ & = \{\epsilon, a, b, aa, bb, aaa, aba, bab, bbb, \dots\} \end{aligned}$$



ACHTUNG, ACHTUNG !!!

➤ Verwechslungsgefahr:

- Die Sprache, die aus keinem Wort besteht, ist die leere Menge: \emptyset
 - \emptyset ist kein Wort!
- Das leere Wort ε ist keine Sprache, sondern eine Folge der Länge 0.
 - ε ist keine Sprache.
 - ε ist kein Zeichen.
 - Es gibt Sprachen, die ε beinhalten.
 - Es gibt Sprachen, die ε nicht beinhalten.



Komponenten des Beispielautomaten

1. Zustände $Q = \{q_1, q_2, q_3\}$

2. Alphabet $\Sigma = \{0,1\}$

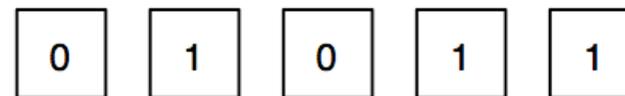
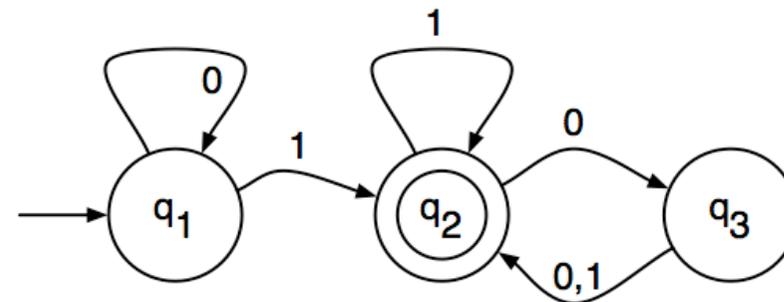
3. Übergangsfunktion:

- $\delta(q_1,0) = q_1$
- $\delta(q_1,1) = q_2$
- $\delta(q_2,0) = q_3$
- $\delta(q_2,1) = q_2$
- $\delta(q_3,0) = q_2$
- $\delta(q_3,1) = q_2$

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

4. Startzustand $q_0 = q_1$

5. $F = \{q_2\}$



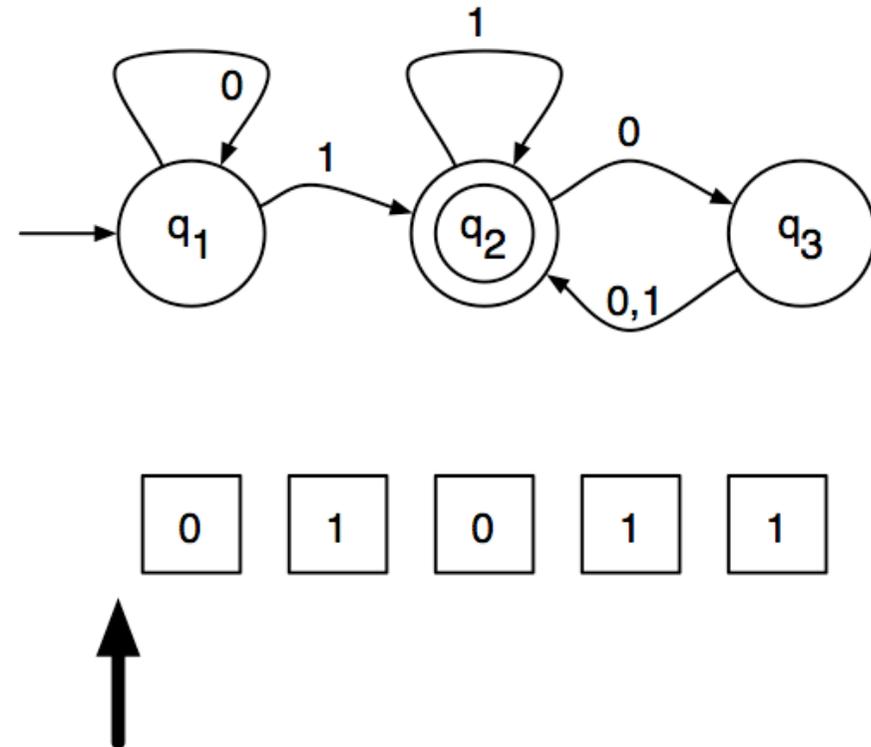


Komponenten eines endlichen Automats (DFA - Deterministic Finite Automaton)

➤ Definition

– Ein endlicher Automat wird durch die fünf Komponenten $(Q, \Sigma, \delta, q_0, F)$ beschrieben

1. Q : Eine endliche Menge von **Zuständen**
2. Σ : Eine endliche Menge; das **Alphabet**
3. $\delta: Q \times \Sigma \rightarrow Q$ ist die **Übergangsfunktion**
4. $q_0 \in Q$: ist der **Startzustand**
5. $F \subseteq Q$: ist die Menge der **akzeptierenden Zustände**



- Später wird statt von fünf Komponenten auch von einem 5-Tupel gesprochen



Formale Definition der Berechnung eines endlichen Automaten (DFA)

➤ Definition

- Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat und $w = w_1 w_2 \dots w_n$ eine Zeichenkette über dem Alphabet Σ . Dann **akzeptiert** M das Wort w genau dann, falls es ein Folge $r_0 r_1 \dots r_n$ von Zuständen aus Q gibt mit
 1. $r_0 = q_0$
 2. $\delta(r_i, w_{i+1}) = r_{i+1}$, für alle $i = 0, \dots, n-1$
 3. $r_n \in F$
- Wir sagen M **akzeptiert** die Sprache A falls
$$A = \{w \mid M \text{ akzeptiert } w\}$$

Ende der 1. Vorlesung



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Informatik III
Christian Schindelhauer
schindel@informatik.uni-freiburg.de