

Kapitel 2

Reguläre Sprachen

2.5

Ausgabeautomaten

(Mealy und Moore)

Prof. Dr. Robert Preis
Fachbereich Informatik
Fachhochschule Dortmund
Robert.Preis@fh-dortmund.de

Alle Materialien (Folien, Übungsblätter, etc.) dieser Veranstaltung sind urheberrechtlich geschützt und nur von Teilnehmern dieser Veranstaltung und im Rahmen dieser zu verwenden. Eine anderweitige Verwendung oder Verbreitung ist nicht gestattet.

Weitere Automaten

Es gibt eine Vielzahl weiterer Automatenmodelle, insbesondere z.B. auch Automaten mit Ausgaben.

Automaten werden auch in weiteren Veranstaltungen behandelt:

- **Softwaretechnik** (z.B. Prof. Sachweh, Prof. Vollmer, Prof. Ecke-Schüth):
Harel- Automaten
- **Hardware Engineering** (z.B. Prof. Wolff):
Mealy-, Moore- und Medwedew-Automaten
- **Systems Engineering** (z.B. Prof. Künemund):
Mealy-, Moore-, Medwedew- und Harel-Automaten

Dies sind alles deterministische Automaten mit zusätzlichen Ausgaben.

Entscheidungsautomaten und Automaten mit Ausgabe

Die Automaten DEA, NEA und ϵ -NEA sind Entscheidungsautomaten für das Wortproblem, d.h. die Ausgabe ist immer eine Antwort JA oder Nein.

Es gibt aber auch Automaten, die eine Ausgabe erzeugen:

1. **Eingabe:** Wie bisher wird in jedem Schritt ein Buchstabe aus dem Eingabealphabet bearbeitet.
2. **Ausgabe:** In jedem Schritt wird ein Buchstabe aus dem Ausgabealphabet geschrieben.

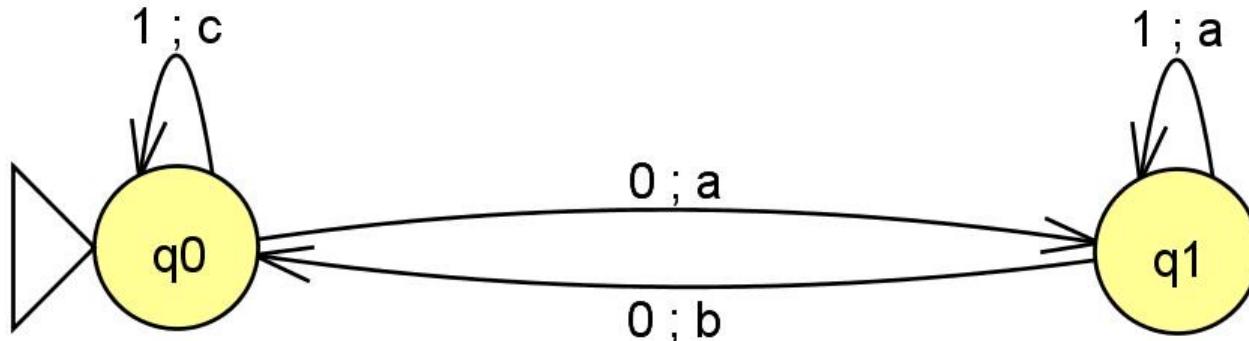
Es wird praktisch ein Eingabewort in ein Ausgabewort umgewandelt.

Es gibt verschiedene Ausgabeautomaten. Der Unterschied liegt darin, wie die Ausgabe produziert wird.

Mealy Automaten

Ein Mealy Automat ist wie ein DFA, nur mit

- zusätzlicher Ausgabe und
- ohne Endzuständen.



Bei Eingabe **1010** produziert der Automat die Ausgabe **caab**.

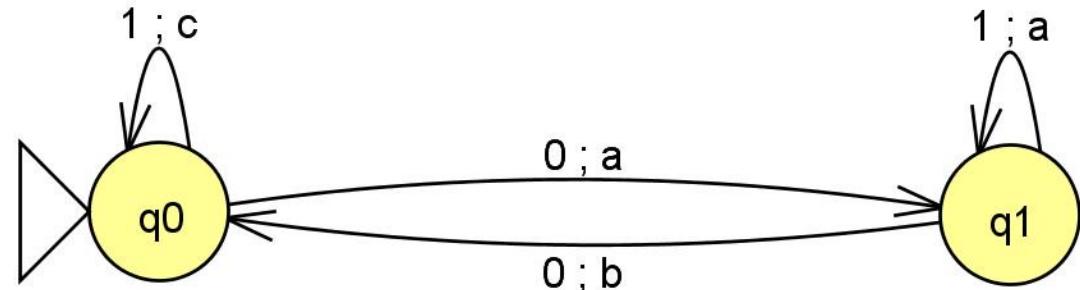
Welche Ausgabe produziert der Automat allgemein?

Definition eines Mealy Automaten (George H. Mealy, 1955)

Ein Mealy Automat ist ein 6-Tupel

$$A = (Q, \Sigma, \Gamma, \delta, \theta, q_0)$$

mit



Q : nichtleere endliche Zustandsmenge

Σ : endliches Eingabealphabet

Γ : endliches Ausgabealphabet

δ : $Q \times \Sigma \rightarrow Q$ Zustandsübergangsfunktion

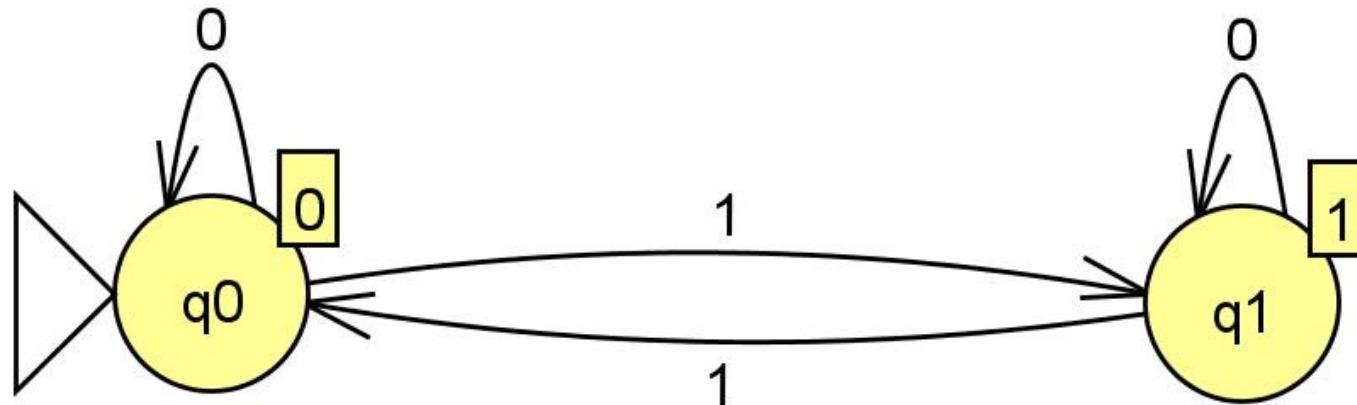
θ : $Q \times \Sigma \rightarrow \Gamma$ Ausgabefunktion

$q_0 \in Q$: Startzustand (Anfangszustand)

Mealy	0	1
$\rightarrow q_0$	q_1/a	q_0/c
q_1	q_0/b	q_1/a

Moore Automaten

Bei einem Moore Automaten wird die Ausgabe nicht durch die Übergänge, sondern durch die Zustände produziert.



D.h. jedes Mal, wenn ein Zustand besucht wird, wird die Ausgabe des Zustands getätigt (nur am Anfang beim Startzustand nicht).

Bei Eingabe **1001101** erzeugt der Moore Automat die Ausgabe **1110110**.

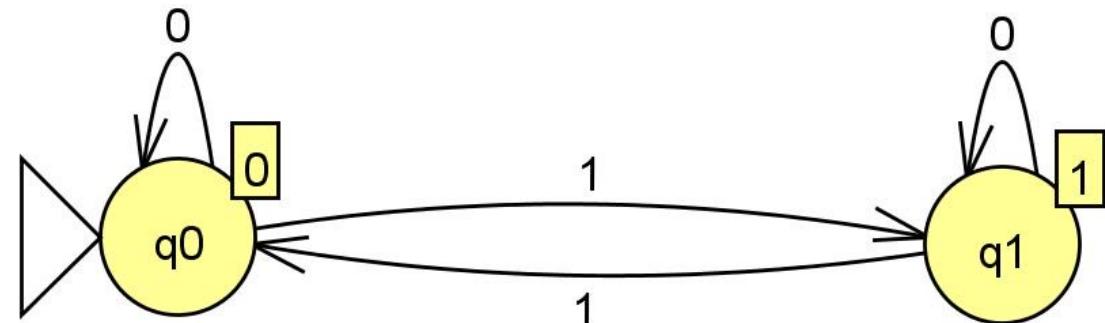
Welche Ausgabe produziert der Automat allgemein?

Definition eines Moore Automaten (Edward F. Moore, 1956)

Ein Moore Automat ist ein 6-Tupel

$$A = (Q, \Sigma, \Gamma, \delta, \theta, q_0)$$

mit



Q : nichtleere endliche Zustandsmenge

Σ : endliches Eingabealphabet

Γ : endliches Ausgabealphabet

δ : $Q \times \Sigma \rightarrow Q$ Zustandsüberführungsfunktion

θ : $Q \rightarrow \Gamma$ Ausgabefunktion

$q_0 \in Q$: Startzustand (Anfangszustand)

Moore	0	1	θ
$\rightarrow q_0$	q_0	q_1	0
	q_1	q_1	1

Eigenschaften von Mealy und Moore Automaten

- Die Mealy und Moore Automaten sind gleich mächtig.

*Jeden Mealy Automaten kann man
in einen Moore Automaten umwandeln.*

*Jeden Moore Automaten kann man
in einen Mealy Automaten umwandeln.*

- Man kann Mealy und Moore Automaten minimieren, d.h. man kann den kleinsten Mealy/Moore Automaten bestimmen, der dieselbe Ausgabe erzeugt.

Harel Automaten (David Harel, 1984)

Das Originalpaper:

Science of Computer Programming 8 (1987) 231-274
North-Holland

STATECHARTS: A VISUAL FORMALISM FOR COMPLEX SYSTEMS*

David HAREL

Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel

Communicated by A. Pnueli
Received December 1984
Revised July 1986

- Kombination von Mealy- und Moore
- Bedingte Zustandsübergänge
- Hierarchie
- Zustände mit Gedächtnis
- Nebenläufige Zustände

Science of Computer Programming 8 (1987) 231-274
North-Holland

STATECHARTS: A VISUAL FORMALISM FOR COMPLEX SYSTEMS*

David HAREL

Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel

Communicated by A. Pnueli
Received December 1984
Revised July 1986

Abstract. We present a broad extension of the conventional formalism of state machines and state diagrams, that is relevant to the specification and design of complex discrete-event systems, such as multi-computer real-time systems, communication protocols and digital control units. Our diagrams, which we call *statecharts*, extend conventional state-transition diagrams with essentially three elements, dealing, respectively, with the notions of hierarchy, concurrency and communication. These transform the language of state diagrams into a highly structured and economical description language. Statecharts are thus compact and expressive—small diagrams can express complex behavior—as well as compositional and modular. When coupled with the capabilities of computerized graphics, statecharts enable viewing the description at different levels of detail, and make even very large specifications manageable and comprehensible. In fact, we intend to demonstrate here that statecharts counter many of the objections raised against conventional state diagrams, and thus appear to render specification by diagrams an attractive and plausible approach. Statecharts can be used either as a stand-alone behavioral description or as part of a more general design methodology that deals also with the system's other aspects, such as functional decomposition and data-flow specification. We also discuss some practical experience that was gained over the last three years in applying the statechart formalism to the specification of a particularly complex system.

1. Introduction

The literature on software and systems engineering is almost unanimous in recognizing the existence of a major problem in the specification and design of large and complex *reactive systems*. A reactive system (see [14]), in contrast with a *transformational system*, is characterized by being, to a large extent, event-driven, continuously having to react to external and internal stimuli. Examples include telephones, automobiles, communication networks, computer operating systems, missile and avionics systems, and the man-machine interface of many kinds of ordinary software. The problem is rooted in the difficulty of describing reactive behavior in ways that are clear and realistic, and at the same time formal and

* The initial part of this research was carried out while the author was consulting for the Research and Development Division of the Israel Aircraft Industries (IAI), Lod, Israel. Later stages were supported in part by grants from IAI and AD CAD, Ltd.

Zusammenfassung

- Die bisherigen Automaten (DEA, NEA, ε -NEA) waren Entscheidungsautomaten, d.h. die Ausgabe ist JA/NEIN.
- Es gibt Automaten mit Ausgabe.
- Bei Mealy Automaten erzeugt jeder Übergang die Ausgabe eines Zeichens.
- Bei Moore Automaten erzeugt jeder Zustand beim Erreichen die Ausgabe eines Zeichens.
- Es gibt noch weitere Automaten wie z.B. Medwedew Automaten, die so ähnlich wie Moore Automaten sind, aber hierbei sind die Zustände selber die Ausgabe.
- Harel Automaten kombinieren Mealy und Moore Automaten und besitzen noch weitere Eigenschaften, die für viele Anwendungen in der Softwaretechnik von Bedeutung sind.