Algorithmen und Datenstrukturen

Praktische Einführung und Programmierung

Stefan Bosse

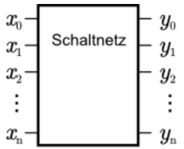
Universität Koblenz - FB Informatik

Endliche Zustandsautomaten

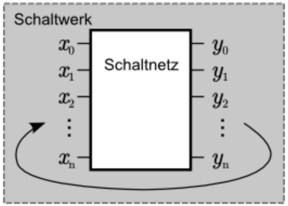
1. Logik
2. Mathematik
3. Graphen
4. Tabellen
5. Automaten (die Maschine)

`https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/ti_7_4_ger_web.html`

Schaltnetze und Schaltwerke



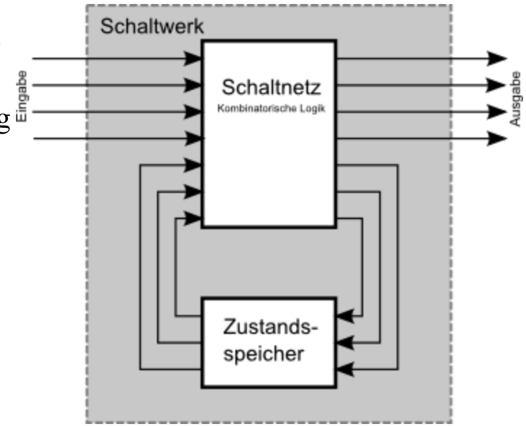
- Ohne Rückkopplung (Schaltnetz)
  - Werte an den Ausgängen sind nur abhängig von den Eingängen
  - Solche Schaltungen verhalten sich immer gleich (sind zustandslos) und sind durch ihre Schaltfunktion eindeutig beschrieben
  - Es ist jedoch nicht möglich, etwas zu speichern



- Mit Rückkopplung (Schaltwerk)
  - Werte an den Ausgängen sind abhängig von den Eingängen und den vorherigen Ausgangswerten
  - Das Zeitverhalten muss genau betrachtet werden
  - Die vorherigen Ausgangswerte können als Zustand der Gatter interpretiert werden. Abhängig vom Zustand verhalten sich die Gatter anders (zustandsabhängige Schaltfunktion)
  - Es wird möglich, Zustände zu speichern

Schaltwerke

- Ein Schaltwerk kann auch als eine Kombination aus einem Schaltnetz und einem Zustandsspeicher dargestellt werden
- Die Ausgabe eines Schaltwerks kann abhängig vom aktuellen Zustand sein
- Abhängig von den Eingangswerten kann sich der Zustand eines Schaltwerks ändern
- Zur Beschreibung der zustandsabhängigen Schaltfunktion können **endliche Automaten** verwendet werden



Endliche Automaten

- Ein endlicher Automat (engl. Finite State Machine, FSM) ist definiert durch
  - eine endliche Menge A von Eingabesymbolen $a_i \in A$ (Alphabet)
  - eine endliche Menge S von Zuständen $s_i \in S$
  - einen Anfangszustand $s_0 \in S$
  - eine Zustandsübergangsfunktion $\delta: S \times A \rightarrow S$
- Des Weiteren kann er umfassen
  - eine endliche Menge B von Ausgabesymbolen $b_i \in B$
  - eine Ausgabefunktion $\lambda: S \times A \rightarrow B$
- Bei deterministischen Automaten erfolgen die Zustandsübergänge deterministisch (nicht zufällig)
- Endliche Automaten können durch **Zustandsübergangsgraphen** dargestellt werden

Zustandsübergangsgraphen

- In einem Zustandsübergangsgraphen wird jeder Zustand si∈S als ein Kreis dargestellt
- Die möglichen Zustandsübergänge werden mit Pfeilen gekennzeichnet
- Jeder Pfeil wird mit dem zugehörigen Eingabesymbol am∈A beschriftet, für das dieser Zustandsübergang auftritt
- Außerdem (abgetrennt durch einen "/") kann jeweils das Ausgabesymbol oder eine Aktion bn∈B angegeben werden

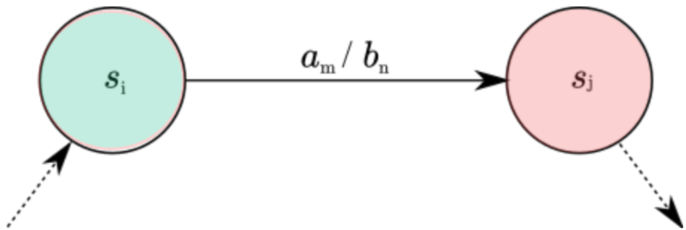


Abb. 1. Ein Zustandsübergang vom Zustand s_i nach s_j durch eine Bedingung, hier das Vorliegen eines Eingabesymbols a, mit der Aktion der Ausgabe eines Symbols b.

Zustandsübergangsgraphen

Beispiel: Ampelschaltung

- Es soll eine Schaltung für eine Fußgängerampelanlage erstellt werden. Es wird dabei die Ampel, die den Autoverkehr regelt, betrachtet (nicht die Fußgängerampel)
    - Die Ampel reagiert auf das Drücken eines Ampelknopfs durch einen Fußgänger:
    - a0 = 0 bedeutet der Ampelknopf wurde nicht gedrückt
    - a1 = 1 bedeutet der Ampelknopf wurde gedrückt
    - Im Anfangszustand s0∈S ist die Ampel grün
    - Wurde der Ampelknopf gedrückt, soll die Ampel zunächst auf gelb und dann auf rot schalten
    - Es wird weiterhin davon ausgegangen, dass das durch den Ampelknopf gesteuerte Eingabesymbol automatisch, nachdem der Fußgänger genug Zeit hatte die Straße zu überqueren, von a1 nach a0 wechselt
    - Die Ampel soll dann zunächst gelb-rot zeigen und schließlich wieder grün

Zustandsübergangsgraphen

- Menge der Eingabesymbole A={a0,a1} binär kodiert mit {0,1}
- Menge der Zustände S={s0,s1,s2,s3}
- Menge der Ausgabesymbole B={b0,b1,b2,b3} binär kodiert mit y2y1y0 zu {001,010,110,100}
- Es gibt |S|·|A|=4·2=8 mögliche Zustandsübergänge

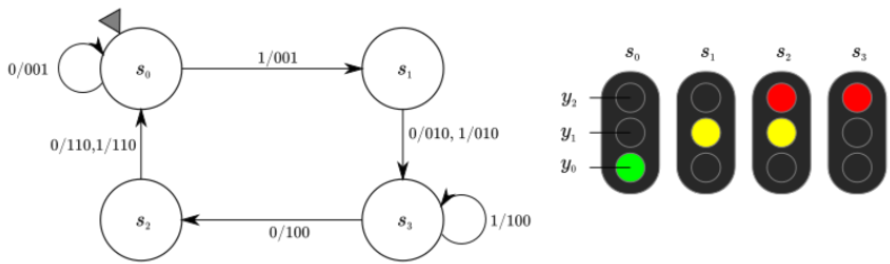


Abb. 2. Beispiel Ampelschaltung: (Links) Zustandsübergangsgraph (Rechts) Die Ausgabe der Zustände

Mealy und Moore-Automaten

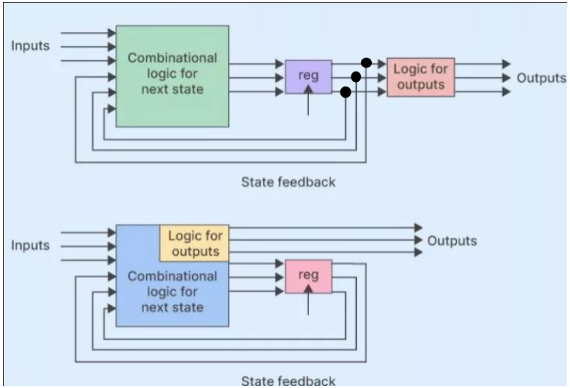
Moore-Automaten

Bei einem Moore-Automaten ist die Ausgabefunktion $\lambda : S \rightarrow B$ nur vom aktuellen Zustand abhängig

Mealy-Automaten

Bei einem Mealy-Automaten ist die Ausgabefunktion $\lambda : S \times A \rightarrow B$ vom aktuellen Zustand und der Eingabe abhängig

[https://unstop.com/blog/difference-between-mealy-and-moore-machine]



Vom Zustandsübergangsgraphen zum Automaten

- Der Zustandsübergangsgraph der Ampelschaltung soll in ein Schaltwerk umgesetzt werden
- Zur Erstellung des Schaltwerks kann die Ausgabefunktion, die Übergangsfunktion und der Zustandsspeicher separat betrachtet werden
- Für gewünschte Ausgabe- und Übergangsfunktion kann direkt aus dem Zustandsübergangsgraphen abgelesen und als Wahrheitstafel dargestellt werden
- Zur Minimierung der Schaltlogik (Boolesche Algebra) können anschließend z.B. BDD verwendet werden

> ? Handelt es sich bei Ampelbeispiel um einen Moore- oder Mealy-Automaten?

Vom Zustandsübergangsgraphen zum Automaten

- Der Zustandsübergangsgraph der Ampelschaltung soll in ein Schaltwerk umgesetzt werden
- Zur Erstellung des Schaltwerks kann die Ausgabefunktion, die Übergangsfunktion und der Zustandsspeicher separat betrachtet werden
- Für gewünschte Ausgabe- und Übergangsfunktion kann direkt aus dem Zustandsübergangsgraphen abgelesen und als Wahrheitstafel dargestellt werden
- Zur Minimierung der Schaltlogik (Boolesche Algebra) können anschließend z.B. BDD verwendet werden

Handelt es sich bei Ampelbeispiel um einen Moore- oder Mealy-Automaten?

Antwort: Moore-Automat, da die Ausgabefunktion nur abhängig vom aktuellen Zustand ist.

Funktionen und Tabellen für das Schaltwerk

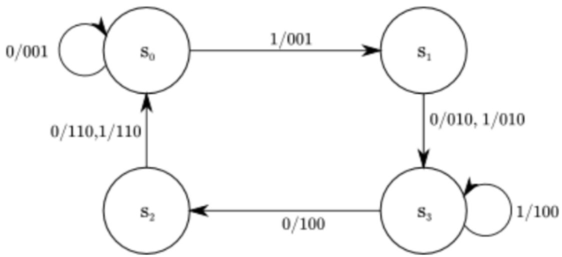
$$y2 = q1$$
$$y1 = q0 \nleftrightarrow q1$$
$$y0 = \neg q0 \wedge \neg q1 = \neg(q0 \vee q1)$$

Zustand	q_1	q_0	y_2	y_1	y_0
s_0	0	0	0	0	1
s_1	0	1	0	1	0
s_2	1	0	1	1	0
s_3	1	1	1	0	0

Abb. 3. Ausgabefunktion als Wahrheitstabelle

Übergangsfunktion

- Es gibt neben einem Schaltnetz, also der Zustandsübergangsfunktion, einen Zustandsspeicher, hier **q**=(q_0,q_1) mit den Dateneingängen **d**=(d_0,d_1).



Zustand	q_1	q_0	x_0	d_1	d_0
s_0	0	0	0	0	0
s_0	0	0	1	0	1
s_1	0	1	0	1	1
s_1	0	1	1	1	1
s_2	1	0	0	0	0
s_2	1	0	1	0	0
s_3	1	1	0	1	0
s_3	1	1	1	1	1

Abb. 4. (Links) Zustandsübergangsgraph (Rechts) Zustandsübergangstabelle mit d_1=q_0

Aus der Zustandsübergangstabelle kann die Übergangsfunktion berechnet werden und z.B. mit BDD Verfahten verinfacht werden.

- Normalformen von Booleschen Funktionen können direkt aus der Tabelle (automatisch) abgeleitet werden.

$$d_0 = (\neg q_1 x_0) \lor (\neg q_1 q_0) \lor (q_0 x_0)$$

Vom Zustandsübergangsgraphen zum Schaltwerk

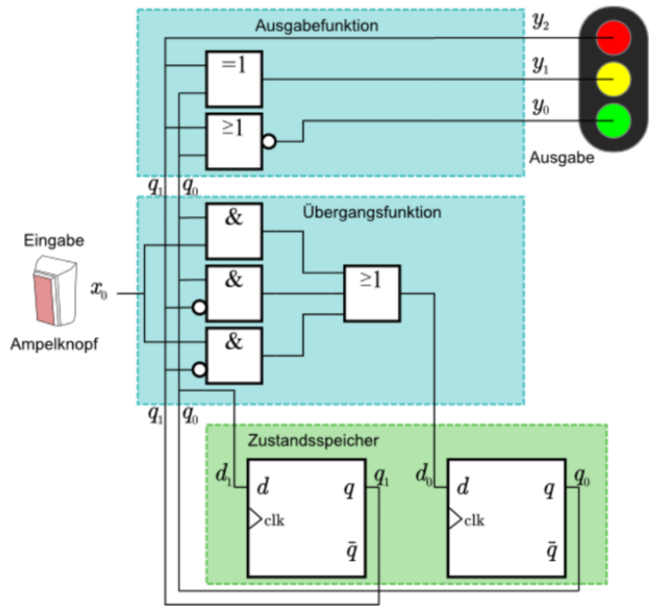


Abb. 5. Die Maschinen mit Digitallogik (Kombinatorische und sequenzielle Logik)