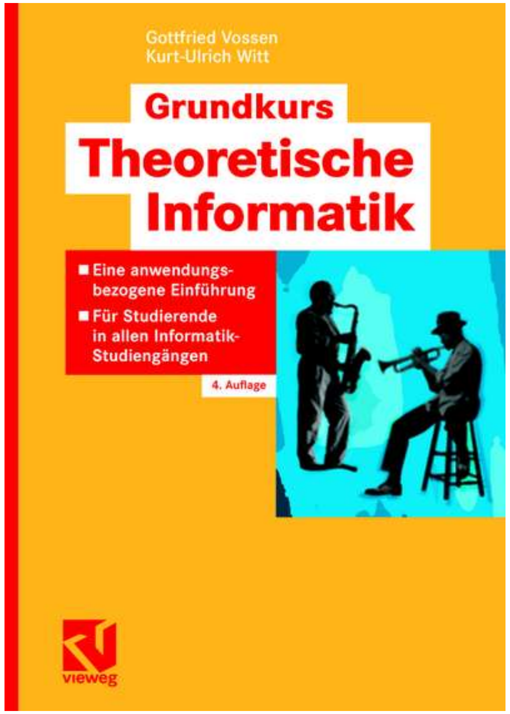
Automaten und Formale Sprachen

Nichtdeterministische Endliche Automaten

Ralf Möller
Hamburg Univ. of Technology

Literatur

- **Gottfried Vossen, Kurt-Ulrich Witt:** Grundkurs Theoretische Informatik, Vieweg Verlag



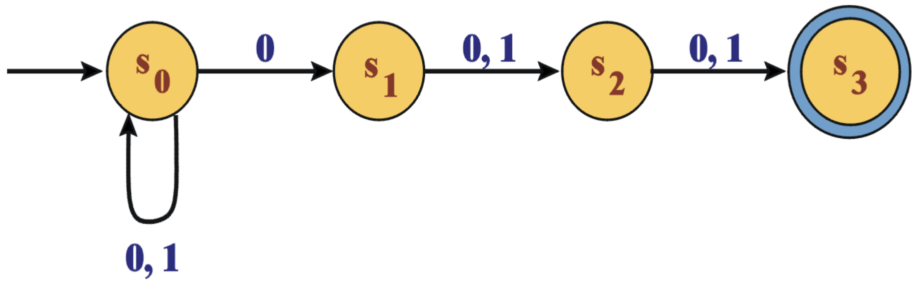
Danksagung

- Kurs basiert auf Präsentationsmaterial von
  - G. Vossen (Uni Münster), K.-U. Witt (Hochschule Bonn-Rhein-Sieg)
  - Christian Sohler (TU Dortmund)
  - Thomas Ottmann (Uni Freiburg)
  - Lenore Blum (CMU)

Motivation für nichtdeterministische Automaten

- Bei DFA ist der Nachfolgezustand immer eindeutig bestimmt, δ ist Funktion
- Für bestimmte Aufgaben lassen sich die Automaten einfacher entwerfen, wenn mehr als ein Nachfolgezustand zur Verfügung steht.
- Bsp.: Entwerfe A_{r3n1} für die Sprache

  $L_{r3n1} = \{w \in \{0, 1\}^*;\ w = u0v,\ u \in \{0, 1\}^*,\ v \in \{0, 1\}^2\}$
- Problem: A_{r3n1} kann nicht voraussehen, wie viele Zeichen des Wortes noch folgen.
- Lösung: A_{r3n1} darf raten, wann der drittletzte Übergang stattfindet. Ein Wort wird akzeptiert, wenn es eine Folge von Zustandsübergängen gibt, die zu einem Finalzustand führen.



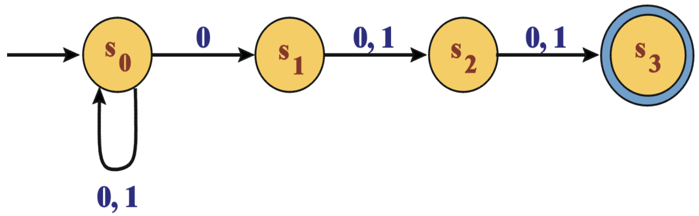
Nichtdeterministische endliche Automaten

Ein Nichtdeterministischer endlicher Automat (NFA) besteht aus

- einer endlichen Menge S von Zuständen
- einer endlichen Menge Σ von Eingabezeichen
- Einer Menge von Anfangszuständen $S_0 \subseteq S$
- einer Endzustandsmenge $F \subseteq S$
- einer Zustandsübergangsrelation $\delta \subseteq S \times \Sigma \times S$

Kurz: $A = (S, \Sigma, \delta, S_0, F)$

δ kann als Menge von Tripeln (s, a, t) oder als Tabelle mit Mengen-Einträgen notiert werden, Bsp.:

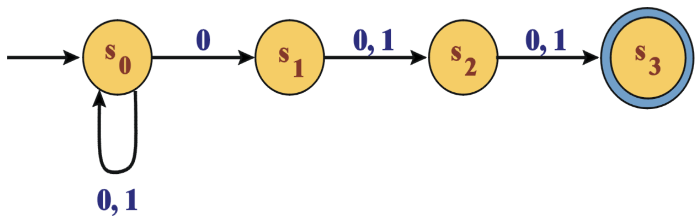


$\delta = \{(s_0, 0, s_0), (s_0, 0, s_1), (s_0, 1, s_0), (s_1, 1, s_2), (s_2, 0, s_3), (s_2, 1, s_3)\}$

Übergangsrelation

$\delta = \{(s_0, 0, s_0), (s_0, 0, s_1), (s_0, 1, s_0), (s_1, 1, s_2), (s_2, 0, s_3), (s_2, 1, s_3)\}$

δ	s_0	s_1	s_2	s_3
0				
1				



Fasst man die Folgezustände je Paar (Zustand, Eingabezeichen) zu einer Menge zusammen, kann man δ auch als mengenwertige Funktion auffassen:

$$\delta : S \times \Sigma \rightarrow 2^S$$

Konfigurationen, Übergänge, Sprache eines NFA

Konfigurationen $k = (s, v)$ und Konfigurationsübergänge $(s, v) \vdash (t, w)$ werden für einen NFA analog zu DFAs definiert:

$K = (s, v)$ mit $s \in S$, $v \in \Sigma^*$ ist der aktuelle Verarbeitungszustand.

Der Übergang $(s, v) \vdash (t, w)$ kann erfolgen, wenn $v = aw$ und $t \in \delta(s, a)$

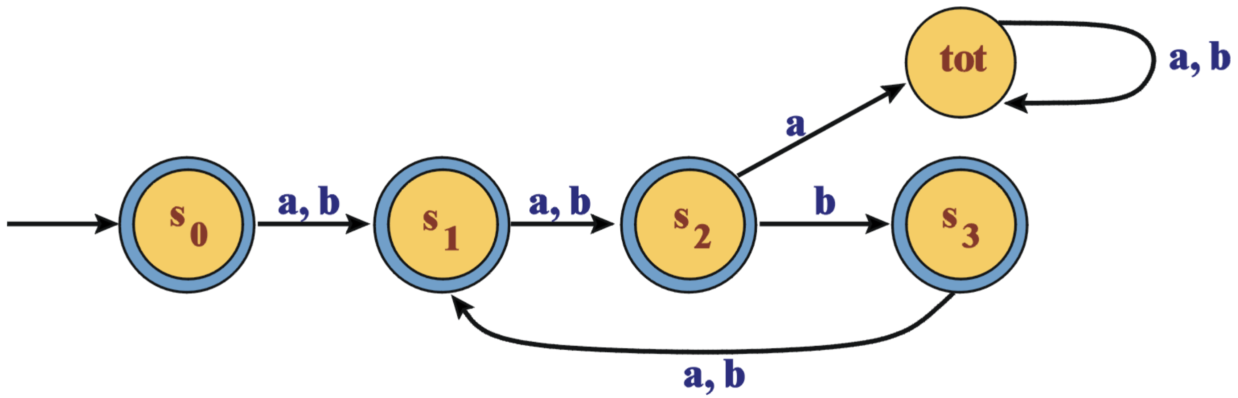
Die von einem NFA $A = (S, \Sigma, \delta, S_0, F)$ akzeptierte Sprache ist

$L(A) = \{w \in \Sigma^* ; (s_0, w) \vdash^* (s, \varepsilon), s_0 \in S_0, s \in F\}$

DFAs sind spezielle NFAs

Jeder DFA $A = (S, \Sigma, \delta, s_0, F)$ kann als spezieller NFA aufgefasst werden, der dieselbe Menge von Worten akzeptiert.

Beispiel:



Erweiterung der Zustandsüberführung bei NFA

Die mengenwertige Zustandsüberführung $\delta : S \times \Sigma \rightarrow 2^S$ kann leicht erweitert werden zu

$\delta^* : 2^S \times \Sigma^* \rightarrow 2^S$ definiert durch

$\delta^* (R, \varepsilon) = R$ für alle $R \subseteq S$

$\delta^* (R, aw) = \delta^* (\cup_{s \in R} \delta (s, a) , w)$ für alle $R \subseteq S$, $a \in \Sigma$, $w \in \Sigma^*$

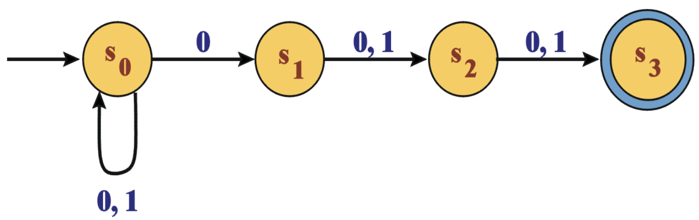
Für einen NFA A kann die Sprache L(A) daher auch so definiert werden:

$$L(A) = \{w \in \Sigma^*; \delta^* (S_0, w) \cap F \neq \emptyset\}$$

Beispiel

$\delta = \{(s_0, 0, s_0), (s_0, 0, s_1), (s_0, 1, s_0), (s_1, 1, s_2), (s_2, 0, s_3), (s_2, 1, s_3)\}$

δ	s_0	s_1	s_2	s_3
0	$\{s_0, s_1\}$	$\{s_2\}$	$\{s_3\}$	$\{\}$
1	$\{s_0\}$	$\{s_2\}$	$\{s_3\}$	$\{\}$



$\delta^*(\{s_0\}, 00011) =$

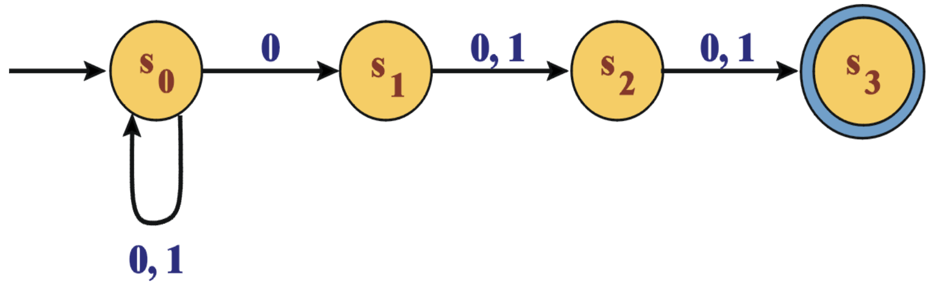
Äquivalenz von DFA und NFA

Satz: Zu jedem NFA A = (S, Σ, δ, S_0, F) kann man einen DFA A‘ konstruieren, so dass L(A) = L(A‘).

Beweis: Mit Hilfe der Potenzmengenkonstruktion.

[Rabin/Scott (1959)]

Potenzautomat für A_{r3n1}



2^S	$\delta(.\,,0)$	$\delta(.\,,1)$	$\in F_d$	letzte 3 Zeichen
$\{s_0\}$				
$\{s_0, s_1\}$				
$\{s_0, s_2\}$				
$\{s_0, s_3\}$				
$\{s_0, s_1, s_2\}$				
$\{s_0, s_1, s_3\}$				
$\{s_0, s_2, s_3\}$				
$\{s_0, s_1, s_2, s_3\}$				

Nicht erreichbare Zustände im Potenzautomaten:

$\{\}$, $\{s_1\}$, $\{s_2\}$, $\{s_3\}$, $\{s_1, s_2\}$ $\{s_1, s_3\}$, $\{s_2, s_3\}$, $\{ s_1, s_2, s_3\}$

Deterministischer Automat für $L(A_{r3n1})$