Formale Grundlagen der Informatik 1
Kapitel 3

Mit DFAs arbeiten

Frank Heitmann
heitmann@informatik.uni-hamburg.de

11. April 2016

Zusammenfassung - Begriffe

Wichtige Begriffe

Begriffe bisher zu endlichen Automaten:

- DFA
- Zustände, Startzustand, Endzustände
- Überführungsfunktion
- erweitere Überführungsfunktion
- vollständig, initial zusammenhängend
- Konfiguration, Konfigurationsübergang
- Rechnung, Erfolgsrechnung
- akzeptierte Sprache
- reguläre Menge, REG

Fragen...

Wie ist ein DFA definiert?

1. $A = (\Sigma, Z_0, \delta, Z, Z_{end})$
2. $A = (Z, z_0, \Sigma, Z_{end}, \delta)$
3. $A = (z_{end}, z_0, \delta, Z, \Sigma)$
4. $A = (Z, \Sigma, \delta, z_0, z_{end})$

Fragen...

Wann akzeptiert ein DFA A eine Eingabe w ?

1. Wenn $\delta(z_0, w) \in Z_{end}$
2. Wenn z_0 in Z_{end} überführt wird
3. Wenn $\hat{\delta}(z_0, w) \in Z_{end}$
4. Wenn $\hat{\delta}(z_0, w) = Z_{end}$

Fragen...

Wie ist die Überführungsfunktion definiert?

1. $\delta : Z \times \Sigma \to Z$
2. $\delta : Z \to \Sigma \times Z$
3. $\delta : \Sigma \times Z \to Z$
4. $\delta : Z \times \Sigma \to Z \times \Sigma$

Fragen...

Wie ist die erweiterte Überführungsfunktion definiert?

1. Das weiß ich!
2. Weiß ich nicht!

Fragen...

Sei c eine Konfiguration. Was ist richtig?

1. $c \in Z \times \Sigma$
2. $c \in Z \times \Sigma^+$
3. $c \in Z \times \Sigma^*$
4. etwas anderes ...

Fragen...

Sie A ein DFA. Was ist richtig?

1. $L(A) \subseteq REG$
2. $L(A) \in REG$
3. $L(A) \subsetneq REG$
4. $L(A) = REG$

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 2
2. 3
3. 1 (3 geht, ist aber ungewöhnlich)
4. -
5. 3
6. 2

Zusammenfassung - Technik

Wichtiges Vorgehen

Ermittelt man für einen DFA A seine akzeptierte Sprache oder konstruiert zu einer Menge M einen Automaten A, der M akzeptieren soll, so ist $L(A) = M$ zunächst eine Behauptung, die zu zeigen ist!

Wichtiges Vorgehen

Hierzu sind dann **zwei Richtungen** zu zeigen:

- $L(A) \subseteq M$. Jedes Wort, dass der Automat akzeptiert ist tatsächlich in M. Bei der Argumentation geht man von einem Wort w aus, das der Automat akzeptiert ($w \in L(A)$) und zeigt, dass dann auch $w \in M$ gilt.
- $M \subseteq L(A)$. Jedes Wort aus M wird auch von dem Automaten akzeptiert. Man geht von einem (beliebigen!) Wort aus M aus ("Sei $w \in M$, dann ...") und zeigt, dass dieses auch von A akzeptiert wird, also in $L(A)$ ist.

Konstruktion von DFAs

Technik

Techniken zum Konstruieren von DFAs

Methode 1: Was will ich speichern?
(Und wie sehen dann die Zustände aus?)

Technik

Techniken zum Konstruieren von DFAs

Methode 2: On-the-fly-Konstruktion ...

Ein komplizierteres Beispiel

Sei

$$\Sigma = \{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \mid x, y, z \in \{0, 1\}\}.$$

Ein Wort aus Σ^* kann also als drei Zeilen mit 0en und 1en gesehen werden. Wir interpretieren jede Zeile als Binärzahl und betrachten die Sprache

$$Sum = \{w \in \Sigma^* \mid \text{ die unterste Zeile ist Summe der oberen Zeilen}\}$$

Ein komplizierteres Beispiel

Sei

$$\Sigma = \{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \mid x, y, z \in \{0, 1\}\}.$$

Ein Wort aus Σ^* kann also als drei Zeilen mit 0en und 1en gesehen werden. Wir interpretieren jede Zeile als Binärzahl und betrachten die Sprache

$$Sum = \{w \in \Sigma^* \mid \text{ die unterste Zeile ist Summe der oberen Zeilen}\}$$

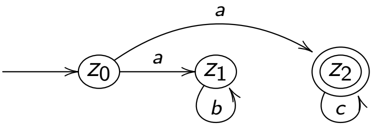
Ist *Sum* regulär?

Ein NFA...

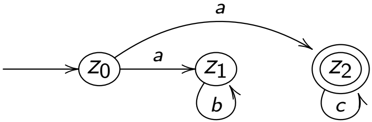
Beim DFA gibt es zu einem $z \in Z$ und einem $x \in \Sigma$ einen Nachfolgezustand $\delta(z, x)$. Der Nachfolgezustand ist also *determiniert*. Dies kann man aufweichen...

Ein NFA...

Beim DFA gibt es zu einem $z \in Z$ und einem $x \in \Sigma$ einen Nachfolgezustand $\delta(z, x)$. Der Nachfolgezustand ist also *determiniert*. Dies kann man aufweichen...



Ein NFA...

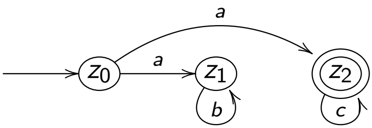


Wie würdet ihr die Überführungsfunktion definieren?

1. $\delta : Z \times \Sigma \rightarrow Z$
2. $\delta : Z \times \Sigma \rightarrow Z^*$
3. $\delta : Z \times \Sigma \rightarrow 2^Z$
4. $\delta : 2^{Z \times \Sigma} \rightarrow Z$
5. ganz anders ...

Nichtdeterministische Automaten

Beim DFA gibt es zu einem $z \in Z$ und einem $x \in \Sigma$ einen Nachfolgezustand $\delta(z, x)$. Der Nachfolgezustand ist also *determiniert*. Dies kann man aufweichen...

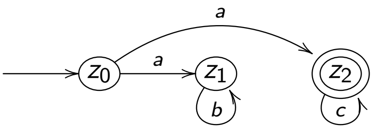


Nach Lesen von a ist der Automat *nichtdeterministisch* sowohl in z_1 als auch in z_2.

Informal akzeptiert der nichtdeterministische, endliche Automat ein Wort dann, wenn es *irgendeine Rechnung* gibt, in der er einen Endzustand erreicht.

Nichtdeterministische Automaten

Beim DFA gibt es zu einem $z \in Z$ und einem $x \in \Sigma$ einen Nachfolgezustand $\delta(z, x)$. Der Nachfolgezustand ist also *determiniert*. Dies kann man aufweichen...



Nach Lesen von a ist der Automat *nichtdeterministisch* sowohl in z_1 als auch in z_2.

Informal akzeptiert der nichtdeterministische, endliche Automat ein Wort dann, wenn es *irgendeine Rechnung* gibt, in der er einen Endzustand erreicht.

Ausblick

Morgen geht es dann mit der formalen Definition dieses nichtdeterministischen Automaten weiter und mit seiner Beziehung zum DFA.