Automaten und Formale Sprachen

SoSe 2013 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

7. Mai 2013

Organisatorisches

Vorlesung FR 10.15-11.45 im F 59
Verschiebungsangebot: FR 10.10-11.40

Übungsbetrieb in Form von einer Übungsgruppe
BEGINN: in der zweiten Semesterwoche MI 09.10-09.55 im HS 13
Ab dann finden sie (im Wesentlichen) **vierzehntägig** statt.
Also ist die zweite Übung am 08.05., 08.20-9.50 im H 7.
Bitte beachten Sie Stud.IP bzw. und unsere Institutsseite.

Dozentensprechstunde DO 14-15 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Markus Schmid) jederzeit

Tutorensprechstunde DO 10-12 H 407

Zulassungskriterien AFS gehört zusammen mit der "Schwesterveranstaltung" Berechenbarkeit und Komplexität im Wintersemester zu einem Modul.

Um zur Modulprüfung zugelassen zu werden, muss man für den AFS-Teil folgende Leistungen erbringen:

- 40% der Übungsaufgabenpunkte
- Bearbeitung jeder Aufgabe
- Wird eine Aufgabe eines Übungsblattes nicht bearbeitet, so wird das gesamte Blatt mit 0 Punkten bewertet.

Bearbeitung einer Aufgabe

Eine Aufgabe kann auf zweierlei Art bearbeitet werden:

1. Sie präsentieren eine Lösung der Aufgabe (wie üblich).
2. Sie formulieren wenigstens eine Frage oder Aussage, die erklärt, weshalb Sie keine Lösung präsentiert haben oder präsentieren konnten.
   Dieser Kommentar muss sich auf den Inhalt der Vorlesung oder Aufgabenstellung beziehen und sollte nicht bloß allgemeiner Natur sein.

Abgabe von Übungsaufgaben

- Abgaben sollten einzeln erfolgen.
- Abgabe 1: Kommentare zu einzelnen Aufgaben bis DI 7:30 Uhr.
- Abgabe 2: Lösungspräsentationen zu den (übrigen) Aufgaben: bis MI 8 Uhr.
- Abgabeort: mit AFS beschrifteter Kasten im 4. Stock vor dem Sekretariat von Prof. Näher. Alternativ unmittelbar vor der anschließenden Übung.
- Es dürfen auch Kommentare zu Aufgaben abgegeben werden, zu denen auch noch eine Lösung präsentiert wird.
- Es dürfe darüber hinaus auch Kommentare zur Übungs- oder Vorlesungsteilen abgegeben werden, die nicht unmittelbar in den Übungen abgefragt werden.

Weitere Regeln

- Verspätete Abgaben gelten als nicht abgegeben und werden dementsprechend mit 0 Punkten bewertet.

- Die Lösungen sind handschriftlich anzufertigen; weder Schreibmaschinen- noch Computerausdrucke werden akzeptiert, erst recht keine Kopien.

- In der nächsten Übung werden die korrigierten und “bepunkteten” Übungsaufgaben wieder zurückgegeben (in den Übungen).

- Lösungen sind immer ausführlich zu erläutern.

Zum Übungsbetrieb

- Durch die abgegebenen Kommentare steuern Sie wesentliche Teile des Übungsverlaufs.
- Darin erklären Sie nämlich, welche Teile der Vorlesung näher erläutert werden müssen.
- Aufgabe der Übungen oder des Übungsleiters ist es nicht, die gestellten Aufgaben der Reihe nach vorzurechnen oder Musterlösungen herauszugeben.
- Musterlösungen helfen auch nur eingeschränkt für Prüfungsvorbereitungen.

Wo finde ich was bei AFS?

Vorlesungen (Foliensätze) liegen zum Runterladen **auf unserer Institutsseite** bereit.

Übungen finden Sie unter Stud.IP bzw. unserer Institutsseite.

Bitte melden Sie sich **sowohl für die VL als auch für die Übungen** im LSF-System an.

Probleme ? Fragen ?

Klären Sie bitte Schwierigkeiten mit Vorlesungen oder Übungen möglichst umgehend in den zur Verfügung gestellten Sprechzeiten.

In der Tutorensprechstunde steht Ihnen Xenia Klinge zu Rückfragen bereit.

Wir helfen Ihnen gerne!
. . . wir sind aber keine Hellseher, die Ihnen Ihre Schwierigkeiten an der Nasenspitze ansehen. . .

Modulprüfungen

werden bei mir

- in der ersten Runde als schriftliche Prüfungen, aber
- in der zweiten Runde ("Nachprüfungsrunde") als mündliche Prüfungen abgelegt.

Erscheinen Ihnen die Modulprüfungen am Ersttermin als zu gedrängt, so können Sie auch die zweite Prüfungsrunde zur Erstprüfung nutzen!

- Organisatorisches
- Einführung
- Endliche Automaten und reguläre Sprachen
- Kontextfreie Grammatiken und kontextfreie Sprachen
- Chomsky-Hierarchie

Endliche Automaten und reguläre Sprachen

1. Deterministische endliche Automaten
2. Nichtdeterministische endliche Automaten
3. Reguläre Ausdrücke
4. Nichtreguläre Sprachen
5. Algorithmen mit / für endliche Automaten

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär* gdw. es ein endliches Monoid $(M, \circ, e)$, einen Monoidmorphismus $h : (\Sigma^*, \cdot, \lambda) \to (M, \circ, e)$ sowie eine endliche Menge $F \subseteq M$ gibt mit

$$L = \{w \in \Sigma^* \mid h(w) \in F\}.$$

zu abstrakt ??

Def.: Ein $\boxed{\textbf{deterministischer endlicher Automat}}$ oder DEA wird beschrieben durch ein Quintupel

$$A = (Q, \Sigma, \delta, q_0, F)$$

wobei gilt:

Q: endliche Menge von *Zuständen* (Zustandsalphabet)

Σ: endliche Menge von *Eingabezeichen* (Eingabealphabet)

$\delta : Q \times \Sigma \to Q$: *Überführungsfunktion*

$q_0 \in Q$: *Anfangszustand*

$F \subseteq Q$: *Endzustände*

Überführungstafel

Ein endlicher Automat kann vollständig durch seine *Überführungstafel* beschrieben werden.

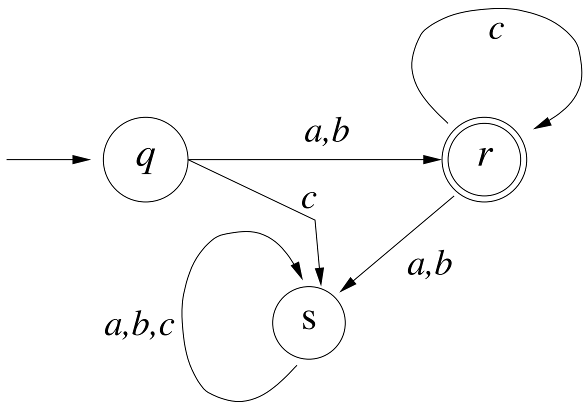
Beispiel: Betrachte:

δ	a	b	c
$\rightarrow q$	r	r	s
$r \rightarrow$	s	s	r
s	s	s	s

In der ersten Zeile ist das Eingabealphabet beschrieben, in der ersten Spalte das Zustandsalphabet; der eingehende Pfeil kennzeichnet den Anfangszustand und der ausgehende den Endzustand (es könnten auch mehrere sein).

Automatengraph

Ein gerichteter, Σ-kantenbeschrifteter Graph mit Knotenmenge Q:



“Eingangspfeile” kennzeichnen den Startzustand und doppelte Umkreisungen die Endzustände.

Wie arbeitet ein DEA ?

Es sei $w = a_1 \dots a_n \in \Sigma^n$ das *Eingabewort* von A.
Die Arbeit von A auf w kann wie folgt (informell) beschrieben werden:

1. Setze $q \leftarrow q_0$.
2. Für $x \leftarrow a_1$ bis a_n tue:
   Setze $q \leftarrow \delta(q, x)$
3. Akzeptiere w gdw. $q \in F$ gilt.

$L(A)$ bezeichnet die von A akzeptierte Sprache.
Beschreiben wir diese im Folgenden in formalerer Art und Weise.

Binärrelationen (bekannt aus DS)

Erinnerung: $R \subseteq X \times X$ heißt *Binärrelation* (auf X).

Die *Diagonale* $\Delta_X = \{(x, x) \mid x \in X\}$ ist eine spezielle Relation.
$R \subseteq X \times X$ heißt *reflexiv* gdw. $\Delta_X \subseteq R$.

Das *Produkt* zweiter Relationen R_1, R_2 auf X ist definiert durch:

$$R_1 \circ R_2 = \{(x, z) \in X \times X \mid \exists y \in X : (x, y) \in R_1 \wedge (y, z) \in R_2\}.$$

Satz: $(2^{X \times X}, \circ, \Delta_X)$ ist ein Monoid.

Eine Relation R auf X heißt *transitiv* gdw. $R \circ R \subseteq R$.

Binärrelationen (Forts.) (bekannt aus DS)

Wir können auch *Relationenpotenz*en induktiv definieren:
$R^0 := \Delta_X$ und $R^n := R^{n-1} \circ R$ für $n > 1$.

Satz: Die so genannte *transitive Hülle* $R^+ := \bigcup_{n \geq 1} R^n$ ist die kleinste R umfassende transitive Relation auf X für gegebenes $R \subseteq X \times X$.

Satz: Die so genannte *reflexive transitive Hülle* $R^* := \bigcup_{n \geq 0} R^n$ ist die kleinste R umfassende reflexive und transitive Relation auf X für gegebenes $R \subseteq X \times X$.

Def.: Die **von einem DEA $A = (Q, \Sigma, \delta, q_0, F)$ akzeptierte Sprache** kann man formal wie folgt beschreiben.

Def.: Ein Element aus $C = Q \times \Sigma^*$ heißt *Konfiguration* von A.
Definiere eine Binärrelation $\vdash_A$ auf C durch $(q, w) \vdash_A (q', w')$ gdw.
$\exists a \in \Sigma : w = aw'$ und $q' = \delta(q, a)$.
Die zweite Komponente einer Konfiguration ist die "übrige Eingabe".
$\vdash_A$ beschreibt den *Konfigurationsübergang in einem Schritt*.
Entsprechend beschreibt $\vdash_A^n$ "n Schritte von A".
Daher können wir definieren:

$$L(A) = \{w \in \Sigma^* \mid \exists q \in F : (q_0, w) \vdash_A^* (q, \lambda)\}.$$

Exkurs: Zustandsraum allgemein

Zustandsraum / Konfigurationsraum $\equiv$ Landkarte

Konfigurationsübergänge $\equiv$ erlaubte “Einzelschritte”

Konfigurationsübergangsfolge $\equiv$ Weg auf der Karte $\equiv$
reflexiv-transitive Hülle der “Einzelschrittübergangsrelation”

Diese Bezüge sind essentiell für das Verständnis vieler Konzepte der Informatik!

Beispiel: Konfiguration im Hornlogik-Kalkül: Menge von Klauselmengen
Konfigurationsübergang: Resolutionsschritt

Ein bekanntes Beispiel als Spiel mit Lösung

Drei Missionare und drei Kannibalen wollen einen Fluss queren.
Das verfügbare Boot befördert mindestens eine, höchstens zwei Personen.
Nebenbedingung: Die Kannibalen dürfen auf keinem Ufer in der Mehrheit gegenüber einer nicht-leeren Menge von Missionaren sein.

Zustandsraum: $\{(i, j) \mid 0 \leq i, j \leq 3\}$ i Missionare / j Kannibalen am linken Ufer
Dazu kämen noch Bootsposition und evtl. Bootsinhalt.

Selbst ohne diese Verfeinerung ist der Zustandsraum unübersichtlich.
Rechts sind “verbotenene Konfigurationen” durch X gekennzeichnet.

	3	2	1	0
3	Start			
2	X		X	X
1	X	X		X
0				Ziel

Alternative Sicht (für DEAs)

Es sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA.
Definiere induktiv:

$$\hat{\delta} : Q \times \Sigma^* \to Q, (q, w) \mapsto \begin{cases} q, & w = \lambda \\ \hat{\delta}(\delta(q, a), w'), & w = aw', a \in \Sigma \end{cases}$$

Lemma: Es sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Seien $p, q \in Q$ und $w \in \Sigma^*$ beliebig.
Dann gilt:

$$(p, w) \vdash_A^* (q, \lambda) \iff \hat{\delta}(p, w) = q.$$

Noch eine induktive Definition

Es sei $(M, \circ, e)$ ein Monoid. Für $n \in \mathbb{N}$ und $a \in M$ definiere:

$$a^n = \begin{cases} e, & n = 0 \\ a^{n-1} \circ a, & n > 0 \end{cases}$$

Sprechweise: *n-te Potenz von a.*

Beispiel: Potenz einer Binärrelation (aus DS).

Anwendung: Potenz eines Wortes.

Konkret: Ist $\Sigma = \{A, D, L, S\}$, so ist $LASSDAS \in \Sigma^*$ und ebenso

$$(LASSDAS)^3 = LASSDASLASSDASLASSDAS$$

... alternativ andersherum ...

Es sei $(M, \circ, e)$ ein Monoid. Für $n \in \mathbb{N}$ und $a \in M$ definiere:

$$a^{[n]} = \begin{cases} e, & n = 0 \\ a \circ a^{[n-1]}, & n > 0 \end{cases}$$

Diese Hilfsschreibweise ist unnötig:

Lemma: Für alle $n \in \mathbb{N}$ gilt: $a^n = a^{[n]}$.

<u>Beweis:</u> Für $n = 0$ gilt die Gleichheit nach Def.

Für $n = 1$: $a^1 = a^0 \circ a = e \circ a = a = a \circ e = a \circ a^{[0]} = a^{[1]}$, da e neutrales Element.

Angenommen, $a^j = a^{[j]}$ gilt für alle $j < m$.

Betrachte den Fall $n = m > 1$.

$a^m = a^{m-1} \circ a = a^{[m-1]} \circ a = (a \circ a^{[m-2]}) \circ a = a \circ (a^{[m-2]} \circ a) = a \circ (a^{m-2} \circ a) = a \circ a^{m-1} =$
$a \circ a^{[m-1]} = a^{[m]}$. Machen Sie sich die einzelnen Schritte klar! $\square$

Erinnerung: Komplexprodukt (DS)

Ist $(M, \circ, e)$ ein Monoid, so kann der Menge 2^M durch das *Komplexprodukt* zu einem Monoid gemacht werden. Dazu definieren wir:

$$A \circ B := \{a \circ b \mid a \in A \wedge b \in B\}$$

Das zugehörige neutrale Element ist $\{e\}$.

Beispiel: $(\Sigma^*, \cdot, \lambda)$ ist ein Monoid, und so kann man auch $\cdot$ als Sprachoperation auffassen.
Also: Sind $L_1, L_2 \subseteq \Sigma^*$, so ist $L_1 \cdot L_2 = \{x_1 \cdot x_2 \mid x_1 \in L_1 \wedge x_2 \in L_2\}$.
Entsprechend: Sprachpotenzen.
$L^n = \{x_1 \cdot x_2 \cdots x_n \mid x_1, x_2, \ldots, x_n \in L\}$.

Ein hilfreiches Lemma

Schlingenlemma: Es sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Es sei $q \in Q$ und sei $X = \{a \in \Sigma : (q, a) \vdash_A (q, \lambda)\}$. Dann gilt: $X^* \subseteq \{w \in \Sigma^* \mid (q, w) \vdash_A^* (q, \lambda)\}$.

Beweis: Wir führen einen Induktionsbeweis, da $X^* = \bigcup_{n \in \mathbb{N}} X^n$.

Wir zeigen (stärker): $X^n \subseteq \{w \in \Sigma^* \mid (q, w) \vdash_A^n (q, \lambda)\}$ für alle $n \in \mathbb{N}$.

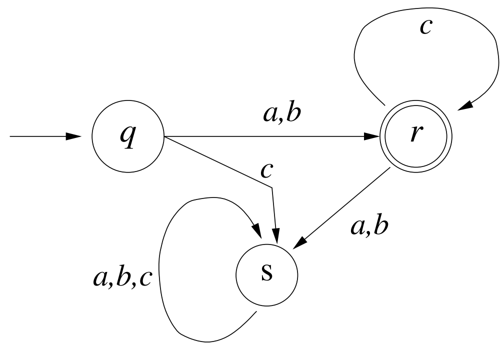
Für $n = 0$ gilt: $w \in X^0$ gdw. $w = \lambda$, und $(q, \lambda) \vdash_A^0 (q, \lambda)$ ✓

IH: Es sei die Behauptung für $n = m - 1$ bewiesen.

Betrachte $n = m$. Sei $w \in X^m = X^{m-1} \cdot X$, d.h. $w = va$ mit $v \in X^{m-1}$.

Da $\vdash_A^m = \vdash_A^{m-1} \circ \vdash_A$, liefert die IH:

$(q, w) \vdash_A^m (q, \lambda)$, da $(q, v) \vdash_A^{m-1} (q, \lambda)$, also $(q, w) \vdash_A^{m-1} (q, a)$ und $(q, a) \vdash_A (q, \lambda)$, falls $a \in X$. $\square$



Was tut also “unser” Automat ?

$$L(A) = \{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\}.$$

$$\{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\} = \{u \in \Sigma^* \mid (q, u) \vdash_A (r, \lambda)\} \cdot \{v \in \Sigma^* \mid (r, v) \vdash_A^* (r, \lambda)\}$$

Idee: Da es im Automatengraphen nur einen (gerichteten) Weg von q nach r gibt und da Wege von r nach r nur r als Zwischenknoten benutzen können, liefert das Schlingenlemma:

$$\{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\} = \{a, b\} \cdot \{c^n \mid n \in \mathbb{N}\}$$

Dualzahlen: ein ausführliches Beispiel für einen DEA

Zur Spezifikation: Was soll der DEA leisten?

Wir wollen einen Automaten A diskutieren, der Dualzahlen der Form 1010010 bzw. *Dualbrüche* der Form $10010 * 01110$ erkennt.
(Der $*$ soll der Deutlichkeit dienen für das "Dualzeichen.")
Wir wollen zur Erleichterung auch "leere Zahlen" akzeptieren und Zahlen der Formen $xxxxx*$ und $xxxxx * 0$.
Das heißt, der Automat soll alle Wörter über dem Alphabet $\{0, 1, *\}$ akzeptieren, die kein oder genau ein Dualzeichen enthalten.

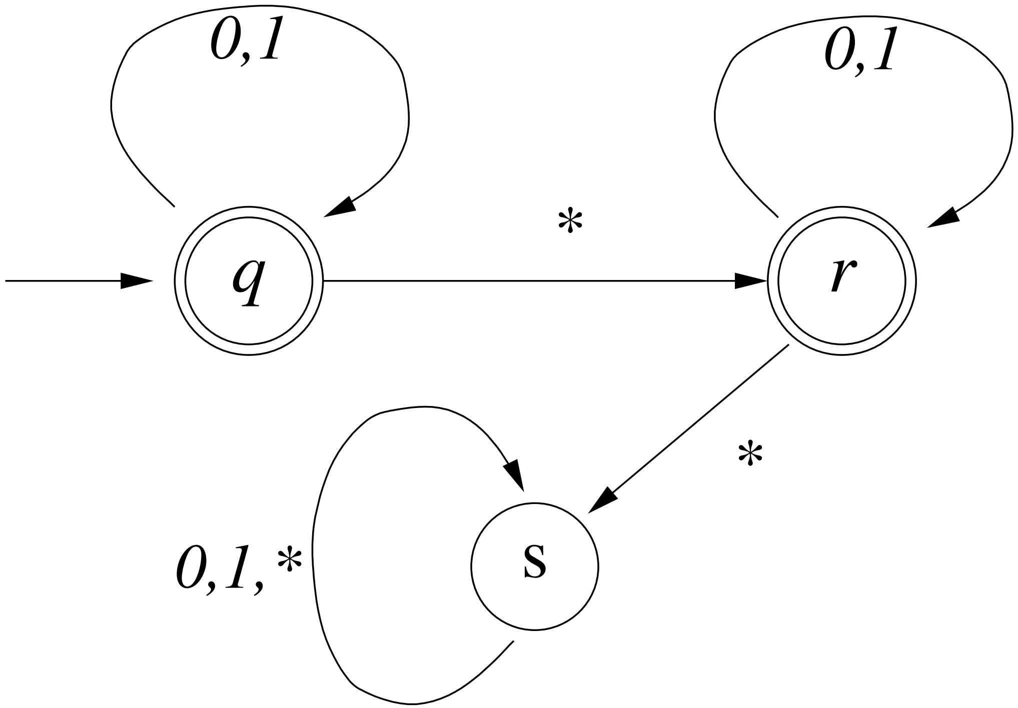
Zur Spezifikation des DEAs $A = (\{q, r, s\}, \{0, 1, *\}, \delta, q, \{q, r\})$:

$$
\begin{array}{rclrcl}
\delta(q,0) & = & q, & \delta(q,1) & = & q \\
\delta(q,*) & = & r, & \delta(r,0) & = & r \\
\delta(r,1) & = & r, & \delta(r,*) & = & s \\
\delta(s,0) & = & s, & \delta(s,1) & = & s \\
\delta(s,*) & = & s & & &
\end{array}
$$

Die Beschreibung mittels einer Überführungstafel hat das folgende Aussehen:

$\delta(\ ,\ )$	0	1	$*$
$\rightarrow q \rightarrow$	q	q	r
$r \rightarrow$	r	r	s
s	s	s	s

Beschreibung als gerichteter kantenbeschrifteter Graph

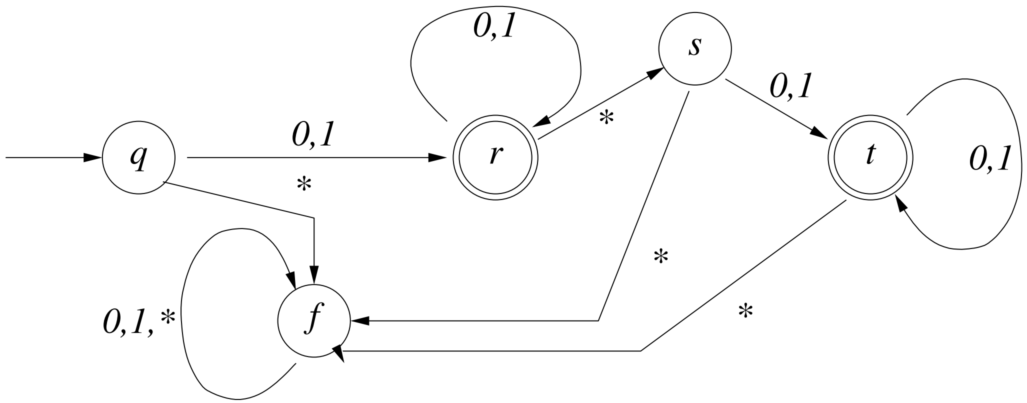


Hierbei haben wir uns erlaubt, Kanten zwischen den selben Knoten zusammenzufassen zu einer Kante mit einer Liste von Eingabezeichen.

Eigenschaften

Man erkennt mit dem Schlingenlemma, dass der Automat nur dann in einen Endzustand übergeht, wenn er eine ganze oder eine ‘real’-Dualzahl liest.

Wollen wir dagegen leere Zahlen und Zahlen der Formen $xxxxx*$ und $*xxxx$ **nicht** akzeptieren, so müssen wir den Automaten folgendermaßen modifizieren:



Wollen wir auch zulassen, dass Zahlen der Form $*xxxx$ möglich sind, oder führende Nullen vor einer Eins vor dem Dualzeichen unterdrücken, müssen wir den Automaten weiter modifizieren. Dies sei zur Übung überlassen.

Was “tut” der folgende Automat ?

δ	a	b
$\rightarrow s$	s	q
$q \rightarrow$	r	q
r	r	r

Wie sieht der Automatengraph aus ?

Wie kann man $L(A)$ beschreiben

- in Worten oder
- in Mengennotation?

Lemma: $L(A) = L$ mit $L := \{a^n b^m \mid n \geq 0, m \geq 1\}$.

Der Beweis von $L(A) = L$ hat in der Regel zwei Richtungen:
(a) $L(A) \subseteq L$ und
(b) $L \subseteq L(A)$.

Beweistechnik: Induktion.

- für (a) betrachtet das Induktionsargument i.d.R. n-Schritt-Konfigurationsübergänge $c \vdash^n c'$ für Konfigurationen c, c'

- für (b) erfolgt das Induktionsargument hingegen über die Wortlänge $n = |w|$ mit $w \in L$ (oder aus Σ^*).

Beweis von $L(A) \supseteq L$:

(a) Offensichtlich ist $b \in L$ das einzige Wort der Länge Eins oder kürzer in L und $b \in L(A)$, denn $\delta(s, b) = q$.

(b) Wir zeigen für bel. $n \in \mathbb{N}$ durch Induktion über $m \geq 1$: $a^n b^m \in L(A)$.

Gilt $m = 1$, so folgt mit Schlingenlemma und (a):

$$(s, a^n b) \vdash^* (s, b) \vdash (q, \lambda)$$

Für $m > 1$ folgt mit der IH und wegen $\delta(q, b) = q$:

$$(s, a^n b^{m-1} b) \vdash^* (q, b) \vdash (q, \lambda) .$$

Also gilt für jedes $w \in L$: $w \in L(A)$.

Zustand r ist eine “Falle” in folgendem Sinne:

Lemma A: $\forall v, w \in \Sigma^* : ((r, v) \vdash^* (x, w)) \Rightarrow x = r$.

Lemma B: $\forall w \in \Sigma^* : ((q, w) \vdash^* (q, \lambda)) \Rightarrow w \in \{b\}^*$.

Beweis durch Induktion über $|w|$. ✓ für $n = 0$.

Für $n > 0$ betrachte $w = a_1 a_2 \ldots a_n$ mit $\forall 1 \leq i \leq n : a_i \in \Sigma$.

Diskutiere $(q, a_1 a_2 \ldots a_n) \vdash (x, a_2 \ldots a_n) \vdash^* (q, \lambda)$.

Falls $a_1 = a$, so gilt $x = r$, und wegen Lemma A ist $(x, a_2 \ldots a_n) \vdash^* (q, \lambda)$ unmöglich.

Ist $a_1 = b$, so gilt $x = q$, und die IH liefert $a_2 = \cdots = a_n = b$.

Beweis von $L(A) \subseteq L$: Betrachte $(s, w) \vdash^n (q, \lambda)$.

Für $n = 1$, $w = b \in L$.

Für $n > 1$ erörtern wir $(s, w) \vdash (x, w') \vdash^{n-1} (q, \lambda)$.

Ist $w = aw'$, so $x = s$; die IH liefert die Behauptung.

Ist $w = bw'$, so $x = q$; dann folgt mit Lemma B die Behauptung. □

Endliche Automaten und Regularität 1

Satz: Wird $L \subseteq \Sigma^*$ von einem DEA erkannt, so ist L regulär.

Beweis: Sei $A = (Q, \Sigma, \delta, q_0, F)$ DEA mit $L(A) = L$.

δ definiert Abbildungen $f_a : Q \to Q$ mit $f_a(q) = \delta(q, a)$.

$h : (\Sigma^*, \cdot, \lambda) \to (Q^Q, \circ, \Delta_Q), a \mapsto f_a$ ist ein Monoidmorphismus.

Definiere als endliche Menge $E = \{f : Q \to Q \mid f(q_0) \in F\}$.

Dann gilt: $L = \{w \in \Sigma^* \mid h(w) \in E\}$. Denn:

Für $w = a_1 \cdots a_n \in \Sigma^n$ mit $a_i \in \Sigma$ für $i = 1, \ldots, n$ gilt:

$w \in L$ gdw.

$\exists q_1, \ldots, q_n$ mit $q_n \in F$, sodass $q_i = \delta(q_{i-1}, a_i)$ für $i = 1, \ldots, n$ gdw.

$\exists q_1, \ldots, q_n$ mit $q_n \in F$, sodass $f_{a_i}(q_{i-1}) = q_i$ für $i = 1, \ldots, n$ gdw.

$\exists q_n \in F$ mit $(f_{a_1} \circ f_{a_2} \circ \cdots \circ f_{a_n})(q_0) = q_n$ gdw.

$h(w) \in E$. $\square$

Endliche Automaten und Regularität 2

Satz: Ist $L \subseteq \Sigma^*$ regulär, so wird L von einem DEA erkannt.

Beweis: Es sei L regulär. Also gibt es ein endliches Monoid $(M, \circ, e)$, eine Menge $F \subseteq M$ und einen Morphismus $h : \Sigma^* \to M$ mit $h(w) \in F$ gdw. $w \in L$.

Definiere nun DEA $A = (Q, \Sigma, \delta, q_0, F)$ wie folgt:

$Q = M$, $q_0 = e$, $\delta(q, a) = q \circ h(a)$.

Unmittelbar aus dieser Definition folgt: $h(w) \in F$ gdw. $w \in L(A)$. $\square$

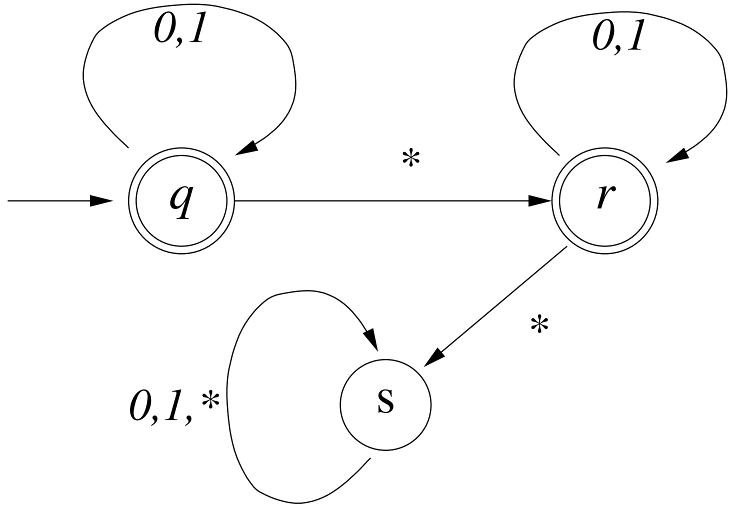
Ein großes Transformationsmonoid

Betrachte $\Sigma = \{a, b, c\}$, das Zustandsalphabet $Q = \{0, \ldots, n-1\}$ und die folgenden Abbildungen:

$$\begin{aligned}
f_a: &\quad 0 \mapsto 1, 1 \mapsto 0, k \mapsto k \text{ für } k \neq 0, 1\\
f_b: &\quad 0 \mapsto 1, 1 \mapsto 2, \ldots, (n-2) \mapsto (n-1), (n-1) \mapsto 0\\
f_c: &\quad (n-1) \mapsto 0, k \mapsto k \text{ für } k \neq n-1
\end{aligned}$$

Dies definiert zum einen die Überführungsfunktion eines DEA mit n Zuständen und Eingabealphabet Σ, zum anderen hat das zugehörige Transformationsmonoid n^n Elemente.
Genauer lassen sich mit f_a und f_b bereits alle $n!$ vielen Bijektionen $Q \to Q$ erzeugen (Idee wie bei Bubblesort durch wiederholtes Vertauschen von Nachbarelementen).

Ein “duales Beispiel” (mit lazy evaluation!)



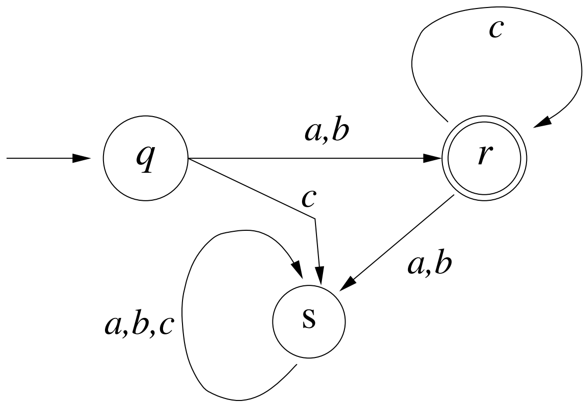
$f_\lambda = f_0 = f_1$	q	r	s
	q	r	s

f_*	q	r	s
	r	s	s

f_{**}	q	r	s
	s	s	s

$(M_A, \circ')$	f_λ	f_*	f_{**}
f_λ	f_λ	f_*	f_{**}
f_*	f_*	f_{**}	f_{**}
f_{**}	f_{**}	f_{**}	f_{**}

Noch ein Beispiel (Das Monoid ist nicht-kommutativ.)



f_λ	q	r	s
	q	r	s

$f_a = f_b$	q	r	s
	r	s	s

$f_c = f_{cc}$	q	r	s
	s	r	s

$f_x, x \in \{a, b\}^2$	q	r	s
	s	s	s

$f_{ac} = f_{bc} = f_a$ $f_{ca} = f_{cb} = f_{aa}$

$(M_A, \circ')$	f_λ	f_a	f_c	f_{aa}
f_λ	f_λ	f_a	f_c	f_{aa}
f_a	f_a	f_{aa}	f_a	f_{aa}
f_c	f_c	f_{aa}	f_c	f_{aa}
f_{aa}	f_{aa}	f_{aa}	f_{aa}	f_{aa}

Zusammenfassung

Wir haben den Begriff eines deterministischen endlichen Automaten (kurz DEA) kennengelernt.

Dieser ist zentral in vielen Bereichen der Informatik, z.B. auch in der Technischen Informatik.

Abschließend haben wir gesehen, dass DEAs Regularität kennzeichnen.

Tatsächlich ist Regularität ein sehr robuster Begriff mit sehr vielen verschiedenen Charakterisierungen.