Automaten und Formale Sprachen

SoSe 2017 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

24. Mai 2017

Automaten und Formale Sprachen Gesamtübersicht

- 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. **Logik und endliche Automaten**

Logik und Formale Sprachen

Formale Logik erlaubt die Beschreibung unterschiedlichster Sachverhalte.

Es muss dabei streng zwischen Syntax und Semantik unterschieden werden.

Zusammenhänge mit Formalen Sprachen:

- Die Syntax selbst lässt sich als formale Sprache begreifen.
- Wir können Logik benutzen, um formale Sprachen zu beschreiben.

Im Folgenden einige Folien aus dem Vorkurs zur Wiederholung.

Grundüberlegungen zur formalen Logik

Wir gehen davon aus, es gäbe eine Menge $\mathfrak{A}$ von *atomaren Formeln*.

Über einen Teil $\mathfrak{D} \subseteq \mathfrak{A}$ dieser Formeln "wissen wir Bescheid", d.h., wir können eine Abbildung $\beta : \mathfrak{D} \to \{0, 1\}$ angeben mit der Bedeutung:

- $\beta(\mathfrak{a}) = 0$, falls $\mathfrak{a}$ falsch ist;
- $\beta(\mathfrak{a}) = 1$, falls $\mathfrak{a}$ wahr ist.

$\mathfrak{D}$ ist also eine Menge definierter Aussagen, und β ist eine *Belegungsfunktion*.
$\mathfrak{A} \setminus \mathfrak{D}$: *logische Variablen* oder *Unbestimmte*.
Wir wollen dann *zusammengesetzte Aussagen* untersuchen.
Wir beschreiben nun, was das formal bedeutet.

Formalitäten: Die Syntax der Aussagenlogik

(Aussagenlogische) Formeln werden durch einen induktiven Prozess definiert:

- Jede atomare Formel ist eine Formel.
  Formaler: Ist $F \in \mathfrak{A}$, so ist F eine Formel.
- Ist F eine Formel, so auch $\neg F$. *Negation von* F
- Sind F und G Formeln, so auch $(F \wedge G)$. *Konjunktion von* F *und* G

Eine Formel, die als Teil einer Formel in F auftritt, heißt *Teilformel* von F.
$\mathfrak{F}$ bezeichne die Gesamtheit aller aussagenlogischen Formeln (bzgl. $\mathfrak{A}$).

Beispiel: Sind A, B, C Formeln, so auch $F = \neg((\neg(A \wedge B) \wedge C) \wedge \neg C)$.
$(A \wedge B)$ ist eine Teilformel von F. $(A \wedge B) \wedge C$ ist keine Teilformel von F, ebensowenig $(B \wedge A)$.
Aufgabe: Wie sieht die Menge aller Teilformeln von F aus?
Muss eine Teilformel notgedrungen als Formel im induktiven Aufbau einer Formel auftreten?

Übliche Abkürzungen

Wir werden im Folgenden zwei weitere Elemente von Formeln als Abkürzungen bekannter Formeln kennenlernen.

- $(F \vee G)$ steht für $\neg(\neg F \wedge \neg G)$. *Disjunktion von F und G*
- $(F \rightarrow G)$ steht für $(\neg F \vee G)$. *Implikation*
- $(F \leftrightarrow G)$ steht für $((F \rightarrow G) \wedge (G \rightarrow F))$. *Äquivalenz*

Die Objekte $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ heißen auch *Junktoren*.
Sie dienen dazu, Teilformeln zu verbinden.
Aufgabe: Wofür steht $(F \leftrightarrow G)$ gemäß unserer ursprünglichen Formeldefinition?

Wir haben bislang nicht festgelegt, was diese Objekte bedeuten sollen.
Die Semantik der Aussagenlogik betrachten wir im Folgenden.

Wozu Formeln formal?

- Wir können einfach feststellen, ob eine Folge von Zeichen eine aussagenlogische Formel darstellt.
- Wir können dies auch programmieren.
- Daher kann ein Computer etwas mit Formeln als Eingabe anfangen.
- Ganz ähnlich kann man z.B. arithmetische Formeln (Ausdrücke, Terme) maschinell verarbeiten.
- Mehr dazu in Veranstaltungen wie “Automaten und Formale Sprachen” oder “Compilerbau”.

Die Semantik der Aussagenlogik: Was sollen die Formeln bedeuten?

$\{0, 1\}$ ist die Menge der *Wahrheitswerte*.
Ggb.: Menge $\mathfrak{A}$ atomarer Formeln, $\mathfrak{F}$ von Formeln (über den atomaren Formeln $\mathfrak{A}$);
Teilmenge $\mathfrak{D} \subseteq \mathfrak{A}$ mit Belegungsfunktion $\beta : \mathfrak{D} \to \{0, 1\}$.
$\mathfrak{E}$ bezeichne die aus $\mathfrak{D}$ aufbaubaren Formeln, also diejenigen Formeln aus $\mathfrak{F}$, die nur Elemente aus $\mathfrak{D}$ als atomare Formeln enthalten.
Wir erweitern β induktiv zu einer Belegungsfunktion $\widehat{\beta} : \mathfrak{E} \to \{0, 1\}$ wie folgt:
Ist $F \in \mathfrak{E}$ atomar, so setze $\widehat{\beta}(F) := \beta(F)$.
Andernfalls unterscheide zwei Fälle:
(a) $F = \neg G$. Setze $\widehat{\beta}(F) := \begin{cases} 0, & \widehat{\beta}(G) = 1 \\ 1, & \widehat{\beta}(G) = 0 \end{cases}$
(b) $F = (G \wedge H)$. Setze $\widehat{\beta}(F) := \begin{cases} 1, & \widehat{\beta}(G) = 1 \text{ und } \widehat{\beta}(H) = 1 \\ 0, & \text{sonst.} \end{cases}$

Nach dieser sauberen Definition werden wir auch für $\widehat{\beta}$ vereinfachend β schreiben.

Wahrheitstafeln für die Grund-Junktoren: Semantik im Überblick

- Negation:

$\beta(G)$	$\beta(\neg G)$
0	1
1	0

für $F = \neg G$

- Konjunktion:

$\beta(G)$	$\beta(H)$	$\beta(G \wedge H)$
0	0	0
0	1	0
1	0	0
1	1	1

für $F = (G \wedge H)$.

Die Verwandtschaft zur Schaltkreislogik ist offenbar.

`http://www.dietrichgrude.de/informatik/schaltlogik.htm`

Wahrheitstafeln für abgeleitete Junktoren

- Disjunktion:

$\beta(F)$	$\beta(G)$	$\beta(F \vee G)$
0	0	0
0	1	1
1	0	1
1	1	1

, denn

$\beta(F)$	$\beta(G)$	$\beta(\neg F)$	$\beta(\neg G)$	$\beta((\neg F \wedge \neg G))$	$\beta(\neg(\neg F \wedge \neg G))$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Aufgabe: Leiten Sie entsprechend die Wahrheitstafeln für $\rightarrow$ und $\leftrightarrow$ her.

Ein Beispiel: Formelauswertung bei gegebener Belegung

Es sei $\mathfrak{D} = \{p, q\}$ vorgegeben mit $\beta(p) = 1$ und $\beta(q) = 0$.
Betrachte die Formel $F = ((p \to q) \vee p)$. Was liefert $\beta(F)$?
1. Rückführen auf die ursprüngliche Definition:
F steht für $((\neg p \vee q) \vee p)$ oder $\neg(\neg\neg(\neg\neg p \wedge \neg q) \wedge \neg p)$.
2. Benutze die induktive Definition als rekursive Berechnungsvorschrift:
Um $\beta(F)$ zu berechnen, benötigen wir nach (a) $\beta(F')$ für $F' = (\neg\neg(\neg\neg p \wedge \neg q) \wedge \neg p)$.
Dazu bestimme wegen (b): $\beta(G')$ und $\beta(H')$ mit $G' = \neg\neg(\neg\neg p \wedge \neg q)$ und $H' = \neg p$.
Mit (a) und wegen $\beta(p) = 1$ folgt: $\beta(H') = 0$.
Die Definition der Semantik der Konjunktion zeigt, dass dann (unabh. von $\beta(G')$) $\beta(F') = 0$ gilt.
Wegen (a) folgt also: $\beta(F) = 1$.

Aufgabe: Berechnen Sie $\beta(G')$ (was wir uns ja “gespart” hatten).
Berechnen Sie mit obiger Belegungsfunktion $\beta(J)$.

Semantische Begriffe

Ist F eine Formel, so bezeichne $\mathfrak{A}(F)$ die in F vorkommenden atomaren Formeln.
Mit $\mathfrak{D} \subseteq \mathfrak{A}$ heißt $\beta : \mathfrak{D} \to \{0, 1\}$ *passend* zu F, falls $\mathfrak{A}(F) \subseteq \mathfrak{D}$.
Ist β eine zu F passende Belegungsfunktion, so heißt β ein *Modell* für F, falls $\beta(F) = 1$. Man schreibt dann auch: $\beta \models F$.
Zwei Formeln F und G heißen *(semantisch) äquivalent* gdw. für jede Belegungsfunktion β, die sowohl für F als auch für G passend ist, gilt: $\beta(F) = \beta(G)$.
Man schreibt dafür auch: $F \equiv G$.

Satz: (*Idempotenz*) Für jede Formel F gilt: $F \equiv (F \wedge F)$.

<u>Beweis:</u> Da $\mathfrak{A}(F) = \mathfrak{A}((F \wedge F))$, können wir uns auf Belegungsfunktionen β mit $\mathfrak{D} = \mathfrak{A}(F)$ beschränken. Sei β solch eine passende Belegungsfunktion.

Gilt $\beta(F) = 1$, so ist nach Def. der Semantik der Konjunktion $\beta((F \wedge F)) = 1$.

Gilt $\beta(F) = 0$, so ist nach Def. der Semantik der Konjunktion $\beta((F \wedge F)) = 0$. $\square$

Rechenregeln für Junktoren

Lemma: (*Absorption*) $F \equiv (F \wedge (F \vee G))$.

Beweis: Nach Definition der Disjunktion gilt:

$$(F \wedge (F \vee G)) \quad = \quad (F \wedge \neg(\neg F \wedge \neg G))$$

Sei nun β eine zu F und zu $H := (F \wedge \neg(\neg F \wedge \neg G))$ passende Belegung.

Fall 1.: Ist $\beta(F) = 1$, so gilt $\beta(\neg F) = 0$ und daher $\beta((\neg F \wedge \neg G)) = 0$, unabh. von $\beta(G)$.

Mithin ist $\beta(\neg(\neg F \wedge \neg G)) = 1$ und somit $\beta(H) = 1$.

Fall 2.: Ist $\beta(F) = 0$, so gilt $\beta(H) = 0$ unabh. von $\beta(\neg(\neg F \wedge \neg G))$.

Kein anderer Fall kann eintreten. $\square$

Die Art der soeben durchgeführten Fallunterscheidung kann man systematisieren durch die Betrachtung von *Wahrheitstafeln*.
Beobachte hierbei, dass nur die Belegung der in den Formeln vorkommenden atomaren Aussagen von Interesse für den Wahrheitsgehalt einer Formel ist.
Enthält also eine Formel n solche verschiedenen atomaren Aussagen, so sind 2^n viele Fälle zu unterscheiden.

Prädikatenlogik und Anwendungen

Aussagenlogik gestattet Formulierung (und Nachweis) einfacher Aussagen. Allgemeinere Aussagen lassen sich mit Hilfe von Quantifizierungen wie $\forall$ und $\exists$ niederschreiben. $\rightsquigarrow$ Prädikatenlogik

Für konkrete Anwendungen müssen die atomaren Aussagen angereichert werden, um überhaupt Aussagen über Eigenschaften in Universen mit gewisser Struktur zu erlauben.

Im Folgenden: Universum mit Struktur "lineare Ordnung". $\rightsquigarrow$ Neues Symbol $\leq$.
Klar: Falls Universum endlich (aber sonst beliebig), ergeben sich "Ketten".
Weitere Symbole R_a (unär) zur Angabe "Zeichen a an dieser Stelle".
Damit lassen sich Zeichenketten modellieren.

Syntax von Büchis Logik erster Stufe über Alphabet Σ

Atomare Formeln: **wahr**, $x \leq y$ und $R_a x$, wobei x und y (Positions-)Variablen sind und $a \in \Sigma$. Daraus induktiv *Formeln erster Stufe*:

- Atomare Formeln sind Formeln erster Stufe.
- Sind ϕ und ψ Formeln erster Stufe, so auch $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$.
- Ist ϕ eine Formel erster Stufe und ist x eine Variable, so sind auch $(\exists x \phi)$ und $(\forall x \phi)$ Formeln erster Stufe.

Wieder Begriffe wie *gebundene Variablen(vorkommen)*, *freie Variablen*, *Aussagenform* usf.
Die Menge der freien Variablen $FV(\phi)$ von Formel ϕ könnte man auch einfach induktiv definieren:

- $FV(x \leq y) = \{x, y\}$, $FV(R_a x) = \{x\}$.
- Sind ϕ und ψ Formeln erster Stufe, so gilt:
  $FV(\neg\phi) = FV(\phi)$, $FV(\phi \wedge \psi) = FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
- Ist ϕ eine Formel erster Stufe und x eine Variable, so ist $FV(\exists x \phi) = FV(\forall x \phi) = FV(\phi) \setminus \{x\}$.

Büchis Semantik

Erinnere: Elemente aus Σ^* sind Wörter über Σ oder auch Abbildungen $[n] \to \Sigma$.
Speziell: $\lambda : [0] \to \Sigma$ mit $[0] = \emptyset$.
Formeln sollen Mengen von Wörtern über Σ beschreiben (als Modelle).
"Mögliche Welten" sind im engeren Sinne $\{[n] \mid n \in \mathbb{N}\}$ mit Halbordnung $\leq$.
$\mathrm{Dom}(u)$ gibt dann den Definitionsbereich von $u \in \Sigma^*$ an.
R_a soll die Menge der Positionen von Vorkommen von Zeichen a angeben.
Bsp.: $\Sigma = \{a, b\}$, $u = abbaab$, $\mathrm{Dom}(u) = [6]$, $R_a = \{0, 3, 4\}$, $R_b = \{1, 2, 5\}$.

<u>Schreibweise</u>: Ist $f : X \to Y$ und $\xi \notin X$ sowie $\eta \in Y$ fest, so bezeichnet $f[\xi \mapsto \eta] : (X \cup \{\xi\}) \to Y$ mit $f[\xi \mapsto \eta] \mid_X = f$ und $f[\xi \mapsto \eta](\xi) = \eta$.

<u>Abkürzungen</u>: $x = y := ((x \leq y) \wedge (y \leq x))$; $x < y := ((x \leq y) \wedge (\neg(x = y)))$.
$S(x, y) := ((x < y \wedge \neg\exists z(x < z \wedge z < y)) \vee (\forall z(z \leq x) \wedge x = y))$.

Belegungen sind Abbildungen $V \to \mathrm{Dom}(u)$ für Variablenmenge V und $u \in \Sigma^*$.

(u, ν) ist *Modell* für Formel ϕ, kurz $(u, \nu) \models \phi$, falls folgende induktive Definition für u mit Belegung ν zutrifft. Hierbei ist $V = \mathrm{FV}(\phi)$.
$(u, \nu) \models (x \leq y)$ gdw. $\nu(x) \leq \nu(y)$ sowie $(u, \nu) \models R_a x$ gdw. $x \in R_a$;
Sind ϕ und ψ Formeln erster Stufe, so gilt:

- $(u, \nu) \models \neg\phi$ gdw. (u, ν) ist kein Modell für ϕ;
- $(u, \nu) \models (\phi \wedge \psi)$ gdw. $(u, \nu) \models \phi$ und $(u, \nu) \models \psi$;
- $(u, \nu) \models (\phi \vee \psi)$ gdw. $(u, \nu) \models \phi$ oder auch $(u, \nu) \models \psi$.

Ist ϕ eine Formel erster Stufe und x eine Variable, so ist:

- $(u, \nu) \models (\exists x \phi)$ gdw. es gibt $d \in \mathrm{Dom}(u)$, sodass $(u, \nu[x \mapsto d]) \models \phi$;
- $(u, \nu) \models (\forall x \phi)$ gdw. für alle $d \in \mathrm{Dom}(u)$ gilt $(u, \nu[x \mapsto d]) \models \phi$.

Ist $\mathrm{FV}(\phi) = \emptyset$, so sagen wir auch: u *erfüllt* ϕ, falls $(u, \emptyset) \models \phi$.

Sprachen von Formeln

Ist ϕ eine Formel über Σ ohne freie Variablen, so ist $L(\phi) = \{u \in \Sigma^* \mid u \text{ erfüllt } \phi\}$.

Beispiele:

- $\phi = \exists x((\forall y \neg(y < x)) \wedge R_a x)$;
- $\psi = \forall x((\forall y \neg(y < x)) \rightarrow R_a x)$;
- $\tau = \exists x(\textbf{wahr})$.

Dann gilt für beliebige Alphabete Σ mit $a \in \Sigma$:
$L(\phi) = \{a\}\Sigma^*$ und $L(\psi) = L(\phi) \cup \{\lambda\}$.
Welche Positionen x erfüllen denn überhaupt $(\forall y \neg(y < x))$?
Beachte: λ erfüllt alle allquantifizierten Aussagen.
Daher interessant: $L(\tau) = \Sigma^+$. Mithin $L(\psi \wedge \tau) = L(\psi) \cap L(\tau) = L(\phi)$.

Logik zweiter Stufe erweitert Syntax um atomare Formeln (Xx) sowie Formeln der Bauart $(\exists X\phi)$ und $(\forall X\phi)$;
erweitert Semantik so, dass X als Mengen von Positionen interpretiert werden; entsprechend wird ν (induktiv) erweitert; ν ordnet Mengenvariablen Mengen zu.
$(u, \nu) \models (Xx)$ gdw. $\nu(x) \in \nu(X)$;
$(u, \nu) \models (\exists X\phi)$ gdw. es gibt $D \subseteq \mathrm{Dom}(u)$, sodass $(u, \nu[X \mapsto D]) \models \phi$;
$(u, \nu) \models (\forall X\phi)$ gdw. für alle $D \subseteq \mathrm{Dom}(u)$ gilt $(u, \nu[X \mapsto D]) \models \phi$.
Damit kann dann schließlich wieder $L(\phi)$ definiert werden.

Beispiele:
$x < y \equiv \exists X(Xy \wedge \neg Xx \wedge (\forall z \forall t((Xz \wedge S(z,t)) \rightarrow Xt)))$.
$\phi = \exists X(\forall x(Xx \leftrightarrow ((\forall y \neg(x < y)) \vee (\forall y \neg(y < x)))) \wedge (\forall x(Xx \rightarrow R_a x) \wedge \exists Xx))$.
$L(\phi) = \{a\}\Sigma^* \cap \Sigma^*\{a\}$.
$\psi = \exists X((\forall x \forall y((x < y) \wedge (\forall z \neg((x < z) \wedge (z < y)))) \rightarrow (Xx \leftrightarrow \neg Xy)) \wedge (\forall x(\forall y \neg(y < x)) \rightarrow Xx) \wedge (\forall x(\forall y \neg(x < y)) \rightarrow \neg Xx))$.
$L(\psi) = \{w \in \Sigma^* \mid \ell_2(w) = 0\}$.

Satz von Kleene / Büchi I: Jede reguläre Sprache ist MSO-definierbar.

Idee: $L \subseteq \Sigma^*$ wird durch DEA beschrieben.
Für jedes $w \in \Sigma^*$ induziert Durchlauf der Zustandsmenge Q Partition von $\mathrm{Dom}(w)$ in $\leq |Q|$ Klassen. Wird die leere Menge als Klasse zugelassen, so sind es o.E. $|Q|$ viele Klassen; sei $Q = [n]$.
Formel für L soll also beschreiben:

- $X_0, \ldots, X_{n-1}$ ist Klasseneinteilung;
- die Zustandsübergänge werden befolgt;
- Anfangs- und Endzustände werden beachtet.

Klasseneinteilung:

$$(\bigwedge_{q \neq p} \neg \exists x (X_q x \wedge X_p x)) \wedge (\forall x \bigvee_q X_q x)$$

Zustandsübergänge (bis auf Wortende):

$$\forall x \forall y (S(x, y) \rightarrow \bigvee_{q \in Q} \bigvee_{a \in \Sigma} (X_q x \wedge R_a y \wedge X_{\delta(q,a)} y))$$

Zusammen Formel für L:

$$\exists X_0 \cdots \exists X_{n-1} (\text{Klasseneinteilung} \wedge \text{Zustandsübergänge} \wedge \text{Randbedingungen})$$

Zur Übung: Formalisieren Sie die Randbedingungen!

Hilfsüberlegungen

Satz: REG ist abgeschlossen gegenüber Homomorphismen.
Das meint: Ist $h : \Sigma^* \to \Gamma^*$ ein Homomorphismus und ist $L \subseteq \Sigma^*$ regulär, so ist $h(L) \subseteq \Gamma^*$ regulär.
Idee: Arbeite induktiv mit RA-Definition.

(p, q)-*erweiterte Alphabete* $\Sigma_{(p,q)} = \Sigma \times \{0, 1\}^p \times \{0, 1\}^q$.
Beachte natürliche Bijektion $f : \Sigma^n_{(p,q)} \to \Sigma^n \times (\{0, 1\}^n)^{p+q}$ für alle n.
$h : \{0, 1\}^* \to \{0, 1\}^*$ sei Morphismus, gegeben durch $h(1) = 1$ und $h(0) = \lambda$.
π_i: Projektion auf Komponente i. Hier: $\pi_0, \pi_1, \ldots, \pi_{p+q}$ sinnvoll.
$K_{p,q} = \{\lambda\} \cup \{w \in \Sigma^+_{(p,q)} \mid \forall i = 1, \ldots, p : \ell(h(\pi_i(f(w)))) = 1\}$.
Wegen Durchschnitts- und Homomorphismen-Abgeschlossenheit von REG folgt: $K_{p,q} \in$ REG.

Zusammenhang Logik zweiter Stufe und erweiterte Alphabete

Betrachte $(u_0, u_1, \ldots, u_p, u_{p+1}, \ldots, u_{p+q}) \in \Sigma^n \times (\{0,1\}^n)^{p+q}$.
Deute R_a als $\{i \in \mathrm{Dom}(u_0) \mid u_0(i) = a\}$;
Variable x_i belegt durch die eindeutig bestimmte Position j, für die $u_i(j) = 1$ gilt.
Variable X_i meint Menge der Positionen j, für die $u_{p+i}(j) = 1$ gilt.
So können wir davon sprechen, dass $u \in K_{p,q}$ eine Formel $\phi(x_1, \ldots, x_p, X_1, \ldots, X_q)$ erfüllt und umgekehrt ϕ die Sprache $L_{p,q}(\phi) \subseteq K_{p,q}$ zuordnen.

Hilfsalphabete:
$C_i = \{w \in \Sigma_{(p,q)} \mid \pi_i(w) = 1\}$
$C_{i,a} = \{w \in C_i \mid \pi_0(w) = a\}$

Satz von Kleene / Büchi II: Jede MSO-definierbare Sprache ist regulär.

Idee: Beschreibe für jede Formel ϕ die Sprache $L_{p,q}(\phi) \subseteq K_{p,q}$ induktiv.
$L_{p,q}(R_a x_i) = K_{p,q} \cap \Sigma^*_{(p,q)} C_{i,a} \Sigma^*_{(p,q)}$;
$L_{p,q}(x_i \leq x_j) = K_{p,q} \cap \Sigma^*_{(p,q)} (C_i \Sigma^*_{(p,q)} C_j \cup (C_i \cap C_j)) \Sigma^*_{(p,q)}$;
$L_{p,q}(X_j x_i) = K_{p,q} \cap \Sigma^*_{(p,q)} (C_i \cap C_{j+p}) \Sigma^*_{(p,q)}$;
$L_{p,q}(\phi \vee \psi) = L_{p,q}(\phi) \cup L_{p,q}(\psi)$;
$L_{p,q}(\phi \wedge \psi) = L_{p,q}(\phi) \cap L_{p,q}(\psi)$;
$L_{p,q}(\neg\phi) = K_{p,q} \setminus L_{p,q}(\phi)$;
$l_i(w)$: Löschen der i-ten Komponente von $w \in \Sigma^+_{(p,q)}$, interpretiert als $f(w)$.
Beachte: $l_i : \Sigma^*_{(p,q)} \to \Sigma^*_{(p-1,q)}$ falls $1 \leq i \leq p$ und $l_i : \Sigma^*_{(p,q)} \to \Sigma^*_{(p,q-1)}$, falls $i > p$, sind auffassbar als Morphismen.
Nach de Morgan genügt nun die Interpretation der Existenzquantoren:
$L_{p-1,q}(\exists x_i \phi) = l_i(L_{p,q}(\phi))$ und $L_{p,q-1}(\exists X_i \phi) = l_{i+p}(L_{p,q}(\phi))$.
Enthält schließlich ϕ keine freien Variablen mehr, so gilt: $L(\phi) = L_{0,0}(\phi)$ ist regulär wegen der bekannten Abschlusseigenschaften der regulären Sprachen.

Abschließende Hinweise

Die Darstellung im zweiten Teil dieser Folien folgt im Wesentlichen dem auch elektronisch erhältlichen Aufsatz von H. Straubing und P. Weil:
An Introduction to Finite Automata and their Connection to Logic, siehe

`http://www.worldscientific.com/worldscibooks/10.1142/7237`

Wichtige Anwendung: Model Checking / Temporale Logiken