Automaten über unendlichen Wörtern

Vorlesung im WS 2011/12 an der Universität Stuttgart

Manfred Kufleitner
(Basierend auf einem Mitschrieb von Stefan Bühler)

23. Januar 2014

1 Einführung

1.1 Presburger Arithmetik

[Mojzesz Presburger 1904-1943]
Definition 1.1.1: Syntax φ für Formeln in Presburger Arithmetik, x, y, z stehen für beliebige Variablen (über $\mathbb{N}$), n für eine beliebige Konstante aus $\mathbb{N}$:

$$\varphi ::= x = n \mid x + y = z \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x : \varphi$$

Makros: $\varphi \wedge \psi \equiv \neg((\neg\varphi) \vee (\neg\psi)), \forall x : \varphi \equiv \neg\exists x : \neg\varphi, x - y = z \equiv x = y + z$
Definition 1.1.2: Ein Satz ist eine Formel ohne freie Variablen.
Beispiel 1.1.3:

- $\varphi_1 = \forall x \exists y \exists z : 2x + 5y = 3z$
- $\varphi_2 = \forall x \exists y : 5x - 3y = 2$

Anmerkung: Trotz der Einschränkung auf $\mathbb{N}$ kann auch in $\mathbb{Z}$ gerechnet werden mit $z_1 = a_1 - b_1$:
$z_1 = z_2 \Leftrightarrow \exists i : (i + a_1 = b_1 \wedge i + a_2 = b_2) \vee (i + b_1 = a_2 \wedge i + b_2 = a_2)$

Problem:

- Eingabe: Satz φ
- Entscheide, ob φ wahr ist.

Wenn man als atomare Formel $x \cdot y = z$ zulässt, ist das Problem unentscheidbar (Gödel).

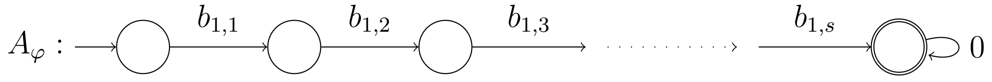
Ziel: Zeige, dass Presburger Arithmetik entscheidbar ist.

Idee: Konstruiere Sprachen und Automaten zu Formeln.
Definition 1.1.4: Sei $\varphi(x_1, \ldots, x_n)$ eine Formel mit freien Variablen $x_1, \ldots, x_n$.

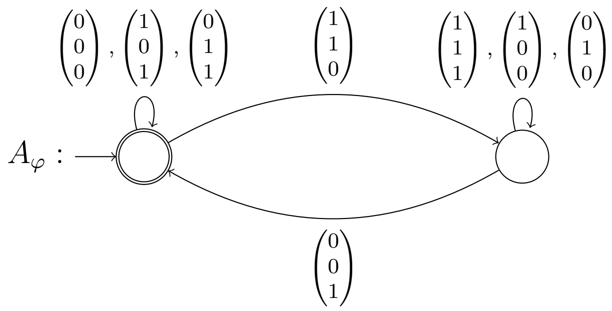
$$L(\varphi) := \left\{ \begin{pmatrix} b_{1,1} \\ \vdots \\ b_{n,1} \end{pmatrix} \cdots \begin{pmatrix} b_{1,s} \\ \vdots \\ b_{n,s} \end{pmatrix} \in (\{0,1\}^n)^* \mid l_i = \sum_{k=1}^{s} 2^{k-1} b_{i,k}, \varphi(l_1, \ldots, l_n) \right\}$$

Konstruiere nun induktiv einen Automaten A_φ mit $L(A_\varphi) = L(\varphi)$.

- $\varphi \equiv x = n$ und $n = \sum_{k=1}^{s} 2^{k-1} b_{1,k}$



- $\varphi(x, y, z) \equiv x + y = z$



  (Der rechte Zustand speichert einen Übertrag)

- $\neg\varphi$: $L(\neg\varphi) = \overline{L(\varphi)}$. Benötigt Algorithmus zum Negieren eines Automaten (traditionell: DFA, $F' = Q \backslash F$)

- $\varphi \vee \psi$: $L(\varphi \vee \psi) = L(\varphi) \cup L(\psi)$. Vereinigen von Automaten (besonders einfach mit NFA und mehreren Startzuständen)

- $\exists x_i : \varphi$: $L(\exists x_i : \varphi) = \pi_i(L(\varphi))$, wobei π_i die i-te Komponente entfernt:
  $\hat{\pi}_i(b_1, \ldots, b_n) = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$
  $\pi_i(v_1 v_2 \ldots v_s) = \hat{\pi}_i(v_1)\hat{\pi}_i(v_2) \ldots \hat{\pi}_i(v_s)$

Für einen Satz φ gilt nun:

$$L(A_\varphi) \neq \emptyset \Leftrightarrow \varphi \text{ wahr}$$

Achtung:

- Größe von $A_{\neg\varphi}$ kann exponentiell sein in Größe von A_φ
- Größe von $A_{\exists x_i : \varphi}$ kann exponentiell sein in Größe von A_φ
- $\Rightarrow$ doppelt exponentiell untere Platzschranke

Zusammenfassung:

- „Spurtechnik", funktioniert für beliebige automatische Strukturen, d.h. atomare Formeln lassen sich als Automaten darstellen.
- Automatenkonstruktion und Algorithmen auf Automaten liefern Entscheidungsalgo-

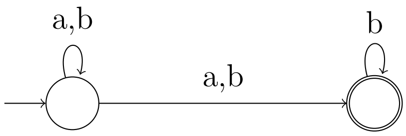
rithmus für Presburger Arithmetik

- Insbesondere Abschluss unter Booleschen Operationen und Projektionen

1.2 Büchi-Automaten (BA)

(Julius Richard Büchi 1924-1984, BA 1962 „erfunden“)
Definition 1.2.1: $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ mit endlicher Zustandsmenge Q, endlichem Alphabet Σ, Übergangsrelation $\delta \subseteq Q \times \Sigma \times Q$, Startzuständen Q_0 und Endzuständen F ist ein (nicht-deterministischer) Büchi-Automat (NBA, sieht also aus wie ein gewöhnlicher NFA).
Ein Büchi-Automat arbeitet auf unendlichen Wörtern $\Sigma^\omega (= \{\alpha : \mathbb{N} \to \Sigma\})$.
$r = q_0 q_1 q_2 \ldots \in Q^\omega$ heißt Lauf von $\mathcal{A}$ auf $\alpha \in \Sigma^\omega$, wenn $q_0 \in Q_0$ und $\forall i \in \mathbb{N} : (q_i, \alpha_i, q_{i+1}) \in \delta$.
$\mathcal{A}$ akzeptiert ein Wort $\alpha \in \Sigma^\omega :\Leftrightarrow \alpha \in L(\mathcal{A}) :\Leftrightarrow \exists$ ein Lauf von $\mathcal{A}$ auf α, der einen Endzustand unendlich oft durchläuft.
Beispiel 1.2.2:

$\mathcal{A}$:



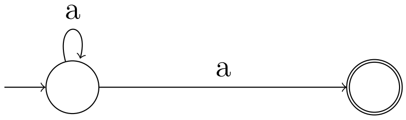
$L(\mathcal{A}) = \{\alpha \in \{a, b\}^\omega \mid |\alpha|_a \in \mathbb{N}\} = \{a, b\}^* \{b\}^\omega$

Angenommen, $L(\mathcal{A})$ wird von einem deterministischer Büchi-Automat $\mathcal{B}$ akzeptiert:
Für alle $w \in a, b^*$ gilt $wb^\omega \in L(\mathcal{B})$, d.h. $\exists n : q_0 \xrightarrow{wb^n} q_1, Q_0 = \{q_0\}, q_1 \in F$
Also lässt sich folgende Kette aufbauen:
$q_0 \xrightarrow{ab^{n_1}} q_1 \xrightarrow{ab^{n_2}} q_2 \ldots, q_1, q_2, \ldots \in F \Rightarrow \exists i < j : q_i = q_j$ (da Q und F endlich)
$\Rightarrow ab^{n_1} \ldots ab^{n_i} (ab^{n_{i+1}} \ldots ab^{n_j})^\omega \in L(\mathcal{B})$, Widerspruch!

Also existiert kein deterministischer Büchi-Automat, der $L(\mathcal{A})$ akzeptiert, und deterministische Büchi-Automaten erkennen echt weniger Sprachen als nicht-deterministische Büchi-Automaten.

Achtung: Es genügt nicht, dass alle (oder unendliche viele) endlichen Prefixe eines unendlichen Wortes akzeptiert werden! Zum Beispiel akzeptieren Büchi-Automaten, deren Endzustände „Sackgassen“ sind oder nur in solche führen, kein einziges Wort.
Beispiel 1.2.3:

$\mathcal{A}$:



$L(\mathcal{A}) = \emptyset$

Problem: Wie kann man das Komplement zu einem Büchi-Automaten finden? Und wie aufwändig?

1.3 Intermezzo: Der Satz von Ramsey

(Frank Plempton Ramsey 1903-1930)

Satz 1.3.1: (Ramsey) Sei $G = (V, E)$ ein unendlicher Graph, $E = \binom{V}{2}$ (alternativ: jeder Knoten braucht unendlich viele ausgehende Kanten). Sei $c : E \to C$ eine Kantenfärbung mit $|C| < \infty$. Dann existiert $V' \subseteq V$ mit

- $|V'| = \infty$
- $|c\left(E \cap \binom{V'}{2}\right)| = 1$, d.h. die Kanten in V' sind einfarbig.

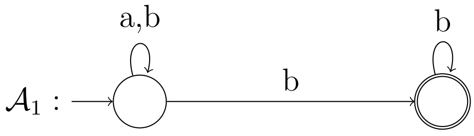
Beweis. Ohne Einschränkung: V ist abzählbar (sonst wähle abzählbare Teilmenge). Setze V_0 $:= V$. Sei $v_i \in V_{i-1}$. Es gibt eine Farbe k, so dass v_i unendliche viele ausgehende Kanten mit dieser Farbe hat.
Wähle unendliches $V_i \subseteq V_{i-1} \setminus \{v_i\}$ mit $c(\{\{v_i, w\} \mid w \in V_i\}) = k$.
Färbe v_i mit $c'(v_i) := k$.
Nun ist $c' : v_1, v_2, \ldots \to C$ eine Knotenfärbung.
Es existiert eine Farbe l, die als Färbung von $v_1, v_2, \ldots$ unendlich oft vorkommt.
$V' := \{v_i | c'(v_i) = l\}$
Alle Kanten in V' haben die Farbe l: $c(\{v_i, v_j\}) = l, v_i, v_j \in V'$. $\square$

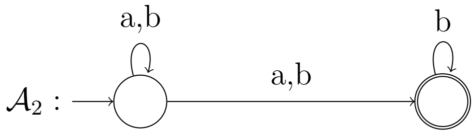
1.4 ω-reguläre Wörter

Definition 1.4.1:
- Σ^*: endliche Wörter, $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$
- $\Sigma^\omega = \{f : \mathbb{N} \to \Sigma\}$: unendliche Wörter
- $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$
- $L^\omega = \{w_1 w_2 \ldots \mid w_i \in L\} \subseteq L^\infty$ für $L \subseteq \Sigma^*, \epsilon^\omega = \epsilon [L^\omega \subseteq \Sigma^\omega$ für $L \subseteq \Sigma^+]$.

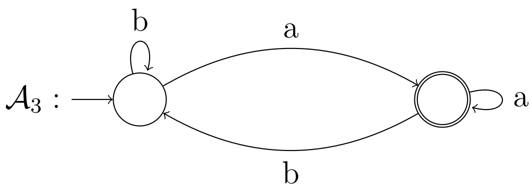
Definition 1.4.2: $L \subseteq \Sigma^\omega$ ist ω-regulär $:\Leftrightarrow \exists \text{NBA} \mathcal{A} : L(\mathcal{A}) = L$

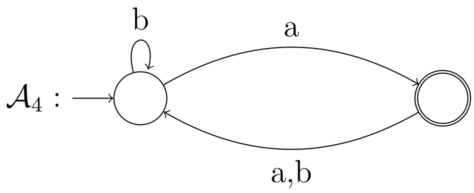
Beispiel 1.4.3:





$$L(\mathcal{A}_1) = L(\mathcal{A}_2) = \{a, b\}^* \, b^\omega$$





$$L(\mathcal{A}_3) = L(\mathcal{A}_4) = \Sigma^\omega \backslash L(\mathcal{A}_1)$$

Korollar 1.4.4: Deterministische Büchi-Automaten sind nicht abgeschlossen unter Komplement.

Korollar 1.4.5: Minimale (det. oder nicht-det.) Büchi-Automaten sind nicht eindeutig.

1.4.1 Einige Abschlusseigenschaften von ω-regulären Sprachen

Sei $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, Q_{0,i}, F_i), i = 1, 2.$

- $L(\mathcal{A}_1) \cup L(\mathcal{A}_2) = L(\mathcal{A}_1 \dot{\cup} \mathcal{A}_2)$
- $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = L(\mathcal{B})$ mit $\mathcal{B} = (Q_1 \times Q_2 \times \{1, 2\}, \Sigma, \delta, Q_0, F)$ mit

$$\delta = \left\{ (p_1, p_2, i) \xrightarrow{a} (q_1, q_2, j) \mid p_k \xrightarrow{a} q_k, p_i \in F_i \Leftrightarrow i \neq j \right\}$$

$$Q_0 = Q_{0,1} \times Q_{0,2} \times \{1, 2\}$$

$$F = Q_1 \times F_2 \times \{2\} \quad (\text{oder } F_1 \times Q_2 \times \{1\})$$

Proposition 1.4.6: $V \subseteq \Sigma^+$ regulär $\Rightarrow V^\omega \subseteq \Sigma^\omega$ ist ω-regulär.

Beweis. Sei A ein DFA für V. Ohne Einschränkung: q_0 ist von keinem anderen Zustand erreichbar.
Zu jeder Kante $p \xrightarrow{a} q, q \in F$ füge Kante $p \xrightarrow{a} q_0$ hinzu.
$F' := \{q_0\}$ □

Anmerkung: Wenn $V^+ = L_*(A)$, dann ist $V^\omega \subseteq L_\omega(A)$ - allerdings kann A als Büchi-Automat echt mehr Worte erkennen.

Proposition 1.4.7: $U \subseteq \Sigma^*$ regulär, $K \subseteq \Sigma^\omega$ ω-regulär.
$\Rightarrow UK \subseteq \Sigma^\omega$ ist ω-regulär.

Beweis. Zu jeder Kante $p \xrightarrow{a} q, q \in F_U$ füge Kanten $p \xrightarrow{a} Q_{0,K}$ hinzu, $Q_0 := Q_{0,U}, F := F_K$ □

Satz 1.4.8: (Büchli) L ω-regulär $\Leftrightarrow L = \bigcup_{i=1}^{n} U_i V_i^\omega$ für $U_i, V_i \subseteq \Sigma^+$ regulär.

Beweis. „$\Leftarrow$“: nach Abschlusseigenschaften
“$\Rightarrow$„: $W_{pq} = \left\{ w \in \Sigma^+ \mid p \xrightarrow{w} q \right\}$ regulär, $L = \bigcup_{\substack{q_0 \in Q_0 \\ q \in F}} W_{q_0 q} W_{qq}^\omega$ □

Korollar 1.4.9: $L(\mathcal{A}) \neq \emptyset \iff \exists u, v \in \Sigma^+ : uv^\omega \in L(\mathcal{A}) \iff \exists u, v \in \Sigma^{\leq |Q|} : uv^\omega \in L(\mathcal{A})$

Korollar 1.4.10: Es ist entscheidbar, ob $L(\mathcal{A}) \neq \emptyset$.

1.4.2 Komplementabschluss von ω-regulären Sprachen

Definition 1.4.11: Seien $u, v \in \Sigma^+, \mathcal{A}$ NBA.

$$u \equiv_\mathcal{A} v :\Leftrightarrow \forall p, q \in Q : p \xrightarrow{u} q \leftrightarrow p \xrightarrow{v} q, p \xrightarrow{u}_F q \leftrightarrow p \xrightarrow{v}_F q$$

$\equiv_\mathcal{A}$ hat Index $\leq e^{|Q|^2}$, alle Äquivalenzklassen $[u]_\mathcal{A} = \{v | v \equiv_\mathcal{A} u\}$ sind regulär (Schnitt von $2|Q|^2$ vielen Automaten).

Lemma 1.4.12:

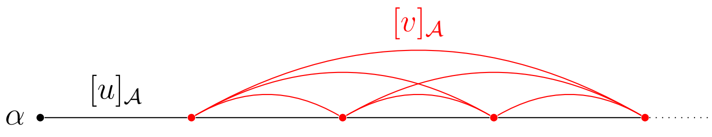
$$[u]_\mathcal{A}[v]_\mathcal{A}^\omega \cap L(\mathcal{A}) \neq \emptyset \Rightarrow [u]_\mathcal{A}[v]_\mathcal{A}^\omega \subseteq L(\mathcal{A})$$

Beweis. Sei $\alpha = u_0 v_1 v_2 \ldots \in L(\mathcal{A}), u_0 \in [u]_\mathcal{A}, v_i \in [v]_\mathcal{A}$. Sei $\beta = u_0' v_1' v_2' \ldots, u_0' \in [u]_\mathcal{A}, v_i' \in [v]_\mathcal{A}$.
Lauf $r = q_0 \xrightarrow{u_0} q_1 \xrightarrow{v_1} q_2 \xrightarrow{v_2}_F q_3 \ldots$
Es existieren unendliche viele $q_i \xrightarrow{v_i}_F q_{i+1}$
Lauf $r' = q_0 \xrightarrow{u_0'} q_1 \xrightarrow{v_1'} q_2 \xrightarrow{v_2'}_F q_3 \ldots$
Es existieren unendliche viele $q_i \xrightarrow{v_i'}_F q_{i+1}, \Rightarrow \beta \in L(\mathcal{A})$ □

Lemma 1.4.13: $\forall \alpha \in \Sigma^\omega \exists u, v \in \Sigma^+$ mit $\alpha \in [u]_\mathcal{A}[v]_\mathcal{A}^\omega$

Beweis. Betrachte Positionen $\{0, 1, \ldots\}$ in dem Wort $\alpha = a_1 a_2 a_3 \ldots$ als Knoten eines Graphen mit den Kanten $\{(i, j) \mid i < j\}$; eine Kante (i, j) wird dabei mit $[a_{i+1} a_{i+2} \ldots a_j]_\mathcal{A}$ gefärbt.

Nach Satz von Ramsey gibt es eine Farbe $[v]_{\mathcal{A}}$, die die Farbe aller Kanten zwischen unendlich vielen Knoten $V' \subseteq V$ ist; sei n der kleinste dieser Knoten V' und $u := a_1 a_2 \ldots a_n$, dann ist $\alpha \in [u]_{\mathcal{A}}[v]_{\mathcal{A}}$.



□

Korollar 1.4.14: $\Sigma^\omega \backslash L(\mathcal{A}) = \bigcup_{[u]_{\mathcal{A}}[v]_{\mathcal{A}} \cap L(\mathcal{A}) = \emptyset} [u]_{\mathcal{A}}[v]_{\mathcal{A}}$

$\Rightarrow$ Komplementierung ist effektiv!

2 MSO-Logik über Wörtern

Definition 2.15:

- „monadic secondary-order“ (MSO) $[<, +1]$:

$$\varphi ::= \top|\bot|\lambda(x) = a|x = y|x < y|x = y + 1|x \in Y|\neg\varphi|\varphi \vee \varphi|\exists x : \varphi|\exists Y : \varphi$$

(x steht für eine Position, Y für eine Menge)

- „first order“ (FO) $[<, +1]$: kein $x \in Y$, kein $\exists Y : \varphi$
- EMSO $[<, +1]$: $\exists Y_1 \dots \exists Y_k : \varphi(Y_1, \dots, Y_k)$ mit φ enthält kein „$\exists Y : \varphi$“
- FO $\subseteq$ EMSO $\subseteq$ MSO

Definition 2.16: Semantik auf Wörtern:

- Variablen laufen über Positionen
- Mengenvariablen laufen über Mengen von Positionen
- $\lambda(x) = a$ bedeutet, dass Position x mit a beschriftet ist („λ“ $\hat{=}$ „label“)

Definition 2.17: Übliche Makros:

- $\forall x : \varphi \equiv \neg\exists x : \neg\varphi, \forall Y : \varphi \equiv \neg\exists Y : \neg\varphi$
- $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$
- $x = y + 1 \equiv y < x \wedge \forall z : (z \leq y \vee x \leq z)$
  $\Rightarrow FO[<, +1] = FO[<], EMSO[<, +1] = EMSO[<], MSO[<, +1] = MSO[<]$
- $x < y \equiv \forall Y : (x + 1 \in Y \wedge (\forall z \in Y : z + 1 \in Y)) \to y \in Y$
  $\Rightarrow MSO[<] = MSO[+1]$
  Achtung: $FO[<] \neq FO[+1]$!

Beispiel 2.18: $L = (bb \mid a)^\omega$
$\varphi = \forall x \forall y : (\lambda(x) = a \wedge \lambda(y) = a \wedge x < y \wedge (\forall z : \lambda(z) = a \to z \leq x \vee y \leq z))$
$\to \exists Y : (x \in Y \wedge (\forall z : z \in Y \leftrightarrow z + 1 \notin Y) \wedge y \notin Y)$
(x und y aufeinanderfolgende a-Positionen)

$L = L(\varphi)$
Bemerkung: $\nexists \psi \in FO[<]$ mit $L(\psi) = L$

Satz 2.19: (Büchi, Elgot) Sei $L \subseteq \Sigma^\omega$. Dann sind äquivalent:

a) L ist ω-regulär

b) L ist definierbar in $EMSO[<]$

c) L ist definierbar in $MSO[<, +1]$

d) L ist definierbar in $MSO[+1]$

Beweis. „b) $\Rightarrow$ c)“, „c) $\Rightarrow$ d)“ schon gezeigt.
„a) $\Rightarrow$ b)“: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein NBA, $Q = \{q_o, \dots, q_m\}$

$$\begin{aligned}
\varphi \ := \ \exists Y_0, \dots, Y_m : & \left(\bigwedge_{i \neq j} \forall y : (y \notin Y_i \vee y \notin Y_j) \right) && \text{(nur ein Zustand pro Position)} \\
& \wedge \left(\forall x : \bigvee_{(q_i, a, q_j) \in \delta} (x \in Y_i \wedge \lambda(x) = a \wedge x + 1 \in Y_j) \right) && \text{(Zustandsübergänge)} \\
& \wedge 1 \in Y_0 && \text{(Startzustand)} \\
& \wedge \left(\bigvee_{q_i \in F} \forall x \exists y : (x < y \wedge y \in Y_i) \right) && \text{(Akzeptanz)}
\end{aligned}$$

„d) $\Rightarrow$ a)“: Spurtechnik:
Sei φ ein Satz in $MSO[<, +1]$. Ohne Einschränkung haben alle Variablen eindeutige Namen $x_1, \dots, x_k, Y_{k+1}, Y_l$.
Erweitertes Alphabet $\Sigma' = \Sigma \times \{0, 1\}^l$.

Idee:

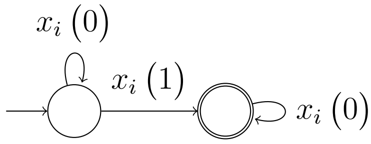
α	a_1	a_2	a_3	a_4	a_5	$\dots$
x_1	0	0	0	1	0	$\dots$
x_2	0	0	0	0	1	$\dots$
x_3	0	0	1	0	0	$\dots$
$\vdots$						$\dots$
x_k	0	1	0	0	0	$\dots$
Y_{k+1}	1	0	1	1	0	$\dots$
Y_{k+2}	1	0	0	1	1	$\dots$
$\vdots$						$\dots$
Y_l	1	1	0	1	0	$\dots$

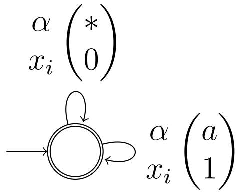
In den x_i-Reihen steht dabei nur eine 1 pro Zeile, und die Semantik ist wie folgt:

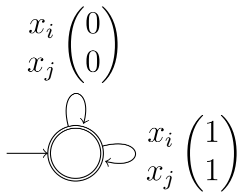
- In der j-ten Spalte von x_i steht eine 1 $\Leftrightarrow x_i = j$
- In der j-ten Spalte von Y_i steht eine 1 $\Leftrightarrow j \in Y_i$

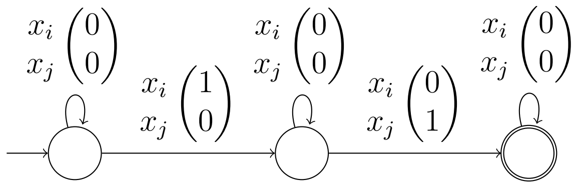
<u>Ziel</u>: Automat $\mathcal{A}_\psi$ mit $(\alpha, x_1, \ldots, x_k, Y_{k+1}, \ldots, Y_l) \in L(\mathcal{A}_\psi) \Leftrightarrow \alpha, x_1, \ldots, x_k, Y_{k+1}, \ldots, Y_l \vDash \psi$

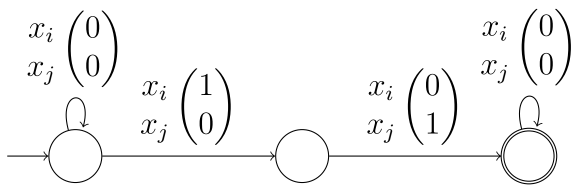
Für nicht angegeben Komponenten werden alle möglichen Kombinationen eingesetzt:

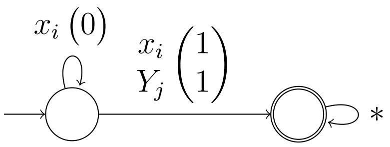
- Konsistenzcheck der x_i, jeweils: 

- $\lambda(x_i) = a$: 

- $x_i = x_j$: 

- $x_i < x_j$: 

- $x_i = x_j + 1$: 

- $x_i \in Y_j$: 

- $\psi_1 \wedge \psi_2$: $\mathcal{A}_{\psi_1} \cap \mathcal{A}_{\psi_2}$ (ohne Einschränkung selben freien Variablen)
- $\exists x_i : \psi, \exists Y_i : \psi$: Entfernen der i-ten Komponente durch Projektion.

- $\neg\psi$: Komplementbildung von $\mathcal{A}_\psi$, geschnitten mit den „Konsistenz“-Bedingungen.

$\square$

Korollar 2.20: Es ist entscheidbar, ob eine Formel φ in $MSO(\mathbb{N}, +1)$ erfüllbar ist.

Beweis. Umformung zu Büchi-Automat $\mathcal{A}$ mit $\Sigma = \{a\}$: φ erfüllbar $\Leftrightarrow L(\mathcal{A}) \neq \emptyset$. $\square$

2.1 Entscheidungsprobleme für Büchi-Automaten

Wortproblem

Eingabe: $u, v \in \Sigma^+$, Büchi-Automat $\mathcal{A}$.
Frage: Gilt $uv^\omega \in L(\mathcal{A})$?

Leerheitsproblem

Eingabe: Büchi-Automat $\mathcal{A}$
Frage: Gilt $L(\mathcal{A}) = \emptyset$?

Universalitätsproblem

Eingabe: Büchi-Automat $\mathcal{A}$
Frage: Gilt $L(\mathcal{A}) = \Sigma^\omega$?

Inklusionsproblem

Eingabe: Büchi-Automaten $\mathcal{A}, \mathcal{B}$
Frage: Gilt $L(\mathcal{A}) \subseteq L(\mathcal{B})$?

2.1.1 Wortproblem

a) $Q_u := \left\{q \mid q_0 \xrightarrow{u} q \text{ für } q_0 \in Q_0\right\}$

b) $E_v := \left\{(p, q) \in \mathbb{Q}^2 \mid p \xrightarrow{v} q\right\}$

c) $F_v := \left\{(p, q) \in \mathbb{Q}^2 \mid p \xrightarrow{v}_F q\right\}$

d) Graph $G = (Q, E_v)$, wobei Kanten in F_v markiert werden.

e) $\forall q_u \in Q_u$: „Lasso-Finding“ von q_u aus im Graph G, wobei Schlinge eine markierte Kante enthalten muss.

$\Rightarrow$ NL-Algorithmus, Aufwand: $\mathcal{O}(|Q|^2 \cdot |uv|)$.

2.1.2 (Nicht-)Leerheitsproblem

Rate „Lasso“ von q_0 aus mit einem Knoten in $q \in F$.
$\Rightarrow$ NL-Algorithmus, Aufwand: $\mathcal{O}(|Q|)$ mit „Tarjan“ (SCC)

2.1.3 (Nicht-)Universalitätsproblem

- Automat $\mathcal{B}$ für $\Sigma^\omega \backslash L(\mathcal{A})$
- $L(\mathcal{B}) \neq \emptyset \Leftrightarrow \mathcal{A}$ nicht universell. $\mathcal{B}$ hat $\leq 2^{c \cdot |Q|^2}$ viele Zustände.
- Ein einzelner Zustand von $\mathcal{B}$ lässt sich mit Platz $c' \cdot |Q|^2$ hinschreiben.
- Es lässt sich in Polynomialzeit testen, ob ein Übergang $p \xrightarrow{a} q$ in $\mathcal{B}$ existiert.
- Da $\mathcal{B}$ exponentiell groß ist, lässt sich in PSPACE testen (der NL-Alg. für Leerheit), ob $L(\mathcal{B}) = \emptyset$.

2.1.4 (Nicht-)Inklusionsproblem

- Automat für $L(\mathcal{A}) \backslash L(\mathcal{B})$ ist höchstens exponentiell groß, jeder einzelne Zustand benötigt höchstens polynomiell viel Platz, und Übergänge lassen sich in Polynomialzeit überprüfen.
- $L(\mathcal{A}) \backslash L(\mathcal{B}) \neq \emptyset$ lässt sich mit „Lasso-Finding“ in PSPACE lösen.

P1: Eingabe: NFA $\mathcal{A}$ über Alphabet $\Sigma = \{a, b\}$
Frage: Gilt $L(\mathcal{A}) = \{a, b\}^*$

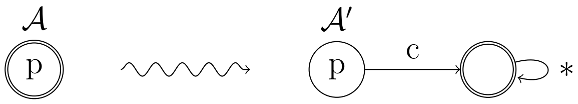
P2: Universalitätsproblem für Büchi-Automaten

Satz 2.1.1: P1 ist PSPACE-vollständig.

Satz 2.1.2: P2 ist PSPACE-hart.

Beweis. P1 $\leq$ P2: Sei $c \notin \{a, b\}$

Konstruiere $\mathcal{A}'$ aus $\mathcal{A}$, indem alle Endzustände folgendermaßen ersetzt werden:



$\mathcal{A}'' := \mathcal{A}' \cup \{a,b\}^\omega$

$L(\mathcal{A}'') = \{a,b,c\}^\omega \Leftrightarrow L(\mathcal{A}) = \{a,b\}^*$ $\square$

2.2 Klarlunds Konstruktion

zur Komplementierung von Büchi-Automaten.

Sei $\mathcal{A} = (Q, \Sigma, \delta, \{q_o\}, F), n := |Q|, L := L(\mathcal{A})$ ein Büchi-Automat mit zugehöriger Sprache.
Ziel: $\overline{\mathcal{A}} = (\overline{Q}, \Sigma, \overline{\delta}, \{\overline{q_o}\}, \overline{Q})$ mit $\overline{L} := L(\overline{\mathcal{A}}) = \Sigma^\omega \backslash L$

$\tilde{Q} := \{f : Q \nrightarrow \{0, 1, \ldots, 2n\} \mid f \text{ partiell } \wedge f(p) \text{ ungerade } \Rightarrow p \notin F\}$

$f \xrightarrow{a} g$ für $f, g \in \tilde{Q}, a \in \Sigma$ falls

a) $\mathrm{dom}(g) = \delta(\mathrm{dom}(f), a) = \left\{q \in Q \mid \exists p \in dom(f) : p \xrightarrow{a} q\right\}$

b) und $\mathrm{dom}(f) \ni p \xrightarrow{a} q \in \mathrm{dom}(g) \Rightarrow f(p) \geq g(q)$

$\overline{Q} := \tilde{Q} \times 2^Q$
$\overline{q_0} := (f_0, \emptyset)$ mit $\mathrm{dom}(f_0) = q_0, f_0(q_0) = 2n$
$(f, A) \xrightarrow{a} (g, B) :\in \overline{\delta}$ wenn

a) $f \xrightarrow{a} g$

b) $B = \begin{cases} \mathrm{dom}(g) & \text{für } A = \emptyset \\ \left\{q \in \mathrm{dom}(g) \mid \exists p \in A : p \xrightarrow{a} q \in \delta, f(p) = g(q) \text{ gerade}\right\} & \text{für } A \neq \emptyset \end{cases}$

$\overline{F} := \tilde{Q} \times \{\emptyset\}$
$|\overline{Q}| \leq (2n+2)^n \cdot 2^n \in 2^{n(\log n + \mathcal{O}(1)) + n} = 2^{n \log n + \mathcal{O}(n)}$

Zeige $L(\overline{\mathcal{A}}) \subseteq \overline{L}$:
Sei $\alpha = a_1 a_2 \ldots \in L(\overline{\mathcal{A}})$.
Sei $(f_0, \emptyset) \xrightarrow{a_1} (f_1, A_1) \xrightarrow{a_2} (f_2, A_2) \xrightarrow{a_3} \ldots$ ein akzeptierender Lauf in $\overline{\mathcal{A}}$
und $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \ldots$ ein beliebiger Lauf in $\mathcal{A}$.
Dann $q_i \in \mathrm{dom}(f_i) \forall i \geq 0$ und $f_i(q_i) \geq f_{i+1}(q_{i+1}) \forall i \geq 0$.

Folge $(f_i(q_i))_{i\geq 0}$ wird stationär $\Rightarrow \exists l, i_0 : \forall i \geq i_0 : f_i(q_i) = l$.
1. Fall: l ungerade $\Rightarrow \forall i \geq i_0 \colon q_i \notin F \Rightarrow$ Lauf auf $\mathcal{A}$ nicht akzeptierend.
2. Fall: l gerade $\Rightarrow \forall i \geq i_0 \colon q_i \in A_i \Rightarrow$ Widerspruch zu $\alpha \in L(\mathcal{A})$.

Zeige: $\overline{L} \subseteq L(\overline{\mathcal{A}})$:
Sei $\alpha = a_1 a_2 \ldots \in \overline{L}$.
Berechnungsgraph (alle Pfade für ein Wort): $G(\alpha) := (V, E), V \subseteq Q \times \mathbb{N}$ mit

a) $(q_0, 0) \in V$

b) $(p, k) \in V, p \xrightarrow{a_{k+1}} q \in \delta \Longrightarrow (q, k+q) \in V \wedge (p, k) \rightarrow (q, k+1) \in E$

ω_1: kleinste überabzählbare Ordinalzahl

Wir definieren Abbildung $\lambda : \omega_1 \rightarrow 2^V$ mit Eigenschaften:

a) $\forall i \neq j : \lambda(i) \cap \lambda(j) = \emptyset$

b) $\bigcup_{i\in\omega_1} \lambda(i) = V$

c) $i \leq j \wedge \lambda(i) = \emptyset \Rightarrow \lambda(j) = \emptyset$

d) $\bigcup_{i<j} \lambda(i)$ ist „rechtsabgeschlossen“, d.h. $v \in \bigcup_{i<j} \lambda(i) \wedge (v, w) \in E \Rightarrow w \in \bigcup_{i<j} \lambda(i)$.

e) $\lambda(i) \cap F \times \mathbb{N} \neq \emptyset \Rightarrow |\lambda(i)| = 1$

Sei $j \in \omega_1$ und sei $\lambda(i)$ schon definiert für alle $i < j$.
1. Fall: $V \setminus \bigcup_{i<j} \lambda(i) = \emptyset \Rightarrow \lambda(j) = \emptyset$
2. Fall: $V' := V \setminus \bigcup_{i<j} \lambda(i) \neq \emptyset$.
Dann existiert $v \in V'$ mit $NF(v) \cap (F \times \mathbb{N}) = \emptyset$ (NF alle transitiven Nachfolger in V'), da das Wort nicht akzeptiert wird.
2.1. Fall: $NF(v) = \emptyset$. Dann $\lambda(j) := \{v\}$.
2.2. Fall: $NF(v) \neq \emptyset$. Dann $\lambda(j) := NF(v)$.

Betrachte unendlichen Pfad in $G(\alpha)$. Dieser Pfad definiert eine unendliche monoton fallende Folge von Ordinalzahlen $\Rightarrow$ Folge wird stationär. Ordinalzahl i heißt *stationär*, falls sie die kleinste Ordinalzahl auf einem unendlichen Pfad ist.
Es gibt höchstens $n = |Q|$ viele stationäre Ordinalzahlen: Seien $i_1, \ldots, i_m$ verschiedene stationäre Ordinalzahlen, jeweils beginnend bei Knoten $(p_1, k_1), \ldots, (p_m, k_m)$, $l = max\,\{k_1, \ldots k_m\}$. Dann passieren alle Pfade den Level („Spalte“) l, aber jeder Level enthält höchstens n Knoten $\Rightarrow m \leq n$.

Bilde die stationären Ordinalzahlen ordnungserhaltend auf die ungeraden Zahlen $1, 3, \ldots, 2n-1$ ab und die übrigen Ordinalzahlen auf die geraden Zahlen $0, 2, \ldots, 2n$. Dies induziert Abbildung $\varphi : V \rightarrow \{0, \ldots, 2n\}$ mit

a) $\varphi(p,k)$ ungerade $\Rightarrow p \notin F$.

b) und $(p,k) \xrightarrow{a_{k+1}} (q,k+1) \Rightarrow \varphi(p,k) \geq \varphi(q,k+1)$

Definiere $f_k : Q \nrightarrow \{0,\dots,2n\}, \mathrm{dom}(f_k) := \{q \in Q \mid (q,k) \in V\}, f_k(q) := \varphi(q,k)$
$\forall k \geq 0 : f_k \in \tilde{Q}$
$\forall k \geq 0 : f_k \xrightarrow{a_{k+1}} f_{k+1}$

Betrachte Pfad in $\overline{\mathcal{A}}$: $(f_0,\emptyset) \xrightarrow{a_1} (f_1,A_1) \xrightarrow{a_2} (f_2,A_2) \xrightarrow{a_3} (f_3,A_3) \xrightarrow{a_4} \dots$
Die Mengen A_i sind durch die Funktionen f_j festgelegt.

Angenommen $\exists k_0 \in \mathbb{N} : \forall k \geq k_0 : A_k \neq \emptyset$:
Für alle $q \in A_{k+1}$ existiert $p \in A_k$ mit $((p,k),(q,k+1)) \in E$ (Kanten von $G(\alpha)$)
Lemma von König: Es existiert unendlicher Pfad in $(\bigcup_{k\geq k_0} A_k \times \{k\}, E)$:

$$(q_{k_0},k_0) \xrightarrow{a_{k_0+1}} (q_{k_0+1},k_0+1) \xrightarrow{a_{k_0+2}} (q_{k_0+2},k_0+2) \xrightarrow{a_{k_0+3}} \dots$$

$\forall k \geq k_0 : \varphi(q_k,k)$ gerade nach Konstruktion der A_i's.
Widerspruch - es gibt keine geraden stationären Zahlen.

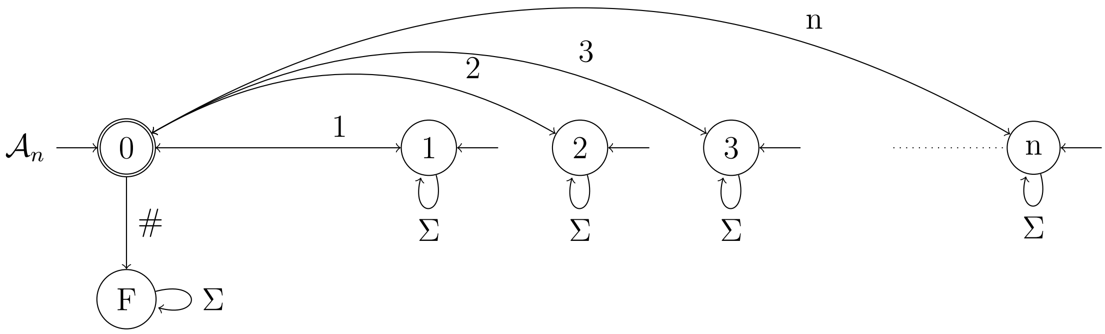
Also gilt $\alpha \in L(\overline{\mathcal{A}})$ und $\overline{L} \subseteq L(\overline{\mathcal{A}})$.

Bermerkung: Klarlunds Konstruktion ist auch ohne Auswahlaxiom korrekt:

- man kann alle Büchi-Automaten aufzählen: $\mathcal{A}_1, \mathcal{A}_2, \dots$
- für $\mathcal{A}_i$ berechne $\mathcal{B}_i$ mit $L(\mathcal{B}_i) = \Sigma_{A_i}^{\omega} \backslash L(\mathcal{A}_i)$
- sei $\mathcal{C}_i = \overline{\mathcal{A}_i}$ nach Klarlund
- teste ob $L(\mathcal{B}_i) = L(\mathcal{C}_i)$

Nach endlicher Zeit findet man Gegenbeispiel für Auswahlaxiom - Widerspruch, da Auswahlaxiom unabängig von der Mengenlehre.
Satz 2.2.1: (Max Michel 1988): $\forall n \in \mathbb{N} \exists \omega$-reguläre Sprache L_n, die von einem BA mit $n+2$ Zuständen erkannt wird, aber jeder BA, der das Komplement von L_n erkennt, benötigt mindestens $n!$ viele Zustände.

Beweis. $\Sigma := \{1,2,\dots,n,\#\}, L_n := L(\mathcal{A}_n)$



$\alpha \in L_n \Leftrightarrow$ es existiert ein Zyklus $(i_1 i_2)(i_2 i_3) \ldots (i_k i_1)$ von Buchstabenpaaren aus $\{1, 2, \ldots, n\}$, $s \neq t \Rightarrow i_s \neq i_t$, so dass jedes Paar unendlich oft als Faktor in α auftritt.

$\alpha = (i_1 \ldots i_n \#)^\omega \notin L_n$, wobei $i_1, \ldots, i_n$ eine Permutation von $\{1, \ldots, n\}$ ist.
Sei $\beta = (j_1 \ldots j_n \#)^\omega$ ein anderes Wort dieser Form.

Sei $\mathcal{B}$ ein Büchi-Automat für das Komplement von L_n, der also α und β akzeptiert.

Lauf von α auf $\mathcal{B}$ zykelt durch Zustände $R, \exists p \in R \cap F_{\mathcal{B}}$. Lauf von β auf $\mathcal{B}$ zykelt durch Zustände S.

Sei k minimal mit $i_k \neq j_k$, dann existieren eindeutig l mit $i_l = j_k (\Rightarrow l > k)$ und m mit $j_m = i_k (\Rightarrow m > k)$.

Angenommen $R \cap S \neq \emptyset$, d.h. $\exists q \in R \cap S$. Dann gibt es ein Wort γ und ein Lauf dazu auf $\mathcal{B}$, der durch die Zustände $R \cap S$ zykelt.

Die Paare $(i_k i_{k+1}), (i_{k+1} i_{k+2}), \ldots, (i_{l-1} i_l) = (i_{l-1} j_k), (j_k j_{k+1}), \ldots, (j_{m-1} j_m) = (j_{m-1} i_k)$ kommen in γ unendlich oft vor
$\Longrightarrow \gamma \in L_n$ Widerspruch!

Also $R \cap S = \emptyset$, und jede Permutation hat ihre eigenen Zustände $\Rightarrow$ mindestens $n!$ viele.

Klarlunds Konstruktion ist optimal bis auf einen Faktor in $2^{\mathcal{O}(n)}$. $\square$

2.3 Weitere Automatenmodelle für ω-reguläre Sprachen

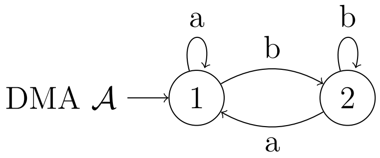
2.3.1 Muller-Automat (MA)

$\mathcal{A} = (Q, \Sigma, \delta, Q_0, \mathcal{T}), \mathcal{T} \subseteq 2^Q$

Lauf r wird Muller-akzeptiert vo $\mathcal{A}$, falls $\inf(r) \in \mathcal{T}$

(inf $r := \{q \in Q \mid q$ kommt unendlich oft in r vor$\}$)

Beispiel 2.3.1:



$\mathcal{T} := \{\{2\}\}$

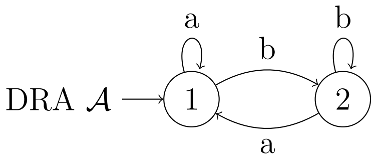
$L(\mathcal{A}) = \{a, b\}^* \, b^\omega$

2.3.2 Rabin-Automat (RA)

$\mathcal{A} = (Q, \Sigma, \delta, Q_0, R), R = \{(L_1, U_1), (L_2, U_2), \ldots\}, L_i, U_i \subseteq Q$

Lauf r wird Rabin-akzeptiert, falls $\exists i : \inf(r) \cap L_i \neq \emptyset \wedge \inf(r) \cap U_i = \emptyset$

Beispiel 2.3.2:



$R := \{(\{2\}, \{1\})\}$

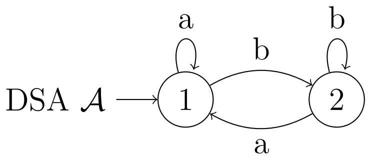
$L(\mathcal{A}) = \{a, b\}^* \, b^\omega$

2.3.3 Streett-Automat (SA)

$\mathcal{A} = (Q, \Sigma, \delta, Q_0, S), S = \{(L_1, U_1), (L_2, U_2), \ldots\}, L_i, U_i \subseteq Q$

Lauf r wird Streett-akzeptiert, falls $\forall i : \inf(r) \cap L_i = \emptyset \vee \inf(r) \cap U_i \neq \emptyset$ ($\forall i :$„$L_i \Rightarrow U_i$“)

Beispiel 2.3.3:



$S := \{(\{1\}, \{0\})\}$

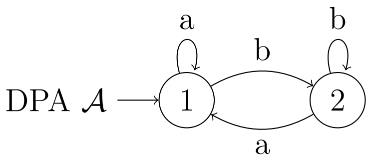
$L(\mathcal{A}) = \{a, b\}^* \, b^\omega$

2.3.4 Parity-Automat (PA)

$\mathcal{A} = (Q, \Sigma, \delta, Q_0), Q \subseteq \mathbb{N}$

Lauf r wird parity-akzeptiert, falls $\min(\inf(r))$ gerade ist.

Beispiel 2.3.4:



$L(\mathcal{A}) = \{a, b\}^* b^\omega$

2.3.5 Transformationen

- BA $\rightsquigarrow$ MA: $\mathcal{T} := \{T \subseteq Q \mid T \cap F \neq \emptyset\}$
- BA $\rightsquigarrow$ RA: $R := \{(F, \emptyset)\}$
- BA $\rightsquigarrow$ SA: $S := \{(Q, F)\}$
- BA $\rightsquigarrow$ PA: Umbenennen der Zustände mit $k \in \mathbb{N}$ und $f : Q \hookrightarrow \mathbb{N}$:
  $f(q)$ gerade $\Leftrightarrow q \in F \Leftrightarrow f(q) < k$
- RA $\rightsquigarrow$ MA: $T \in \mathcal{T} :\Leftrightarrow \exists i : T \cap U_i = \emptyset \wedge T \cap L_i \neq \emptyset$
- SA $\rightsquigarrow$ MA: $T \in \mathcal{T} :\Leftrightarrow \forall i : T \cap L_i = \emptyset \vee T \cap U_i \neq \emptyset$
- PA $\rightsquigarrow$ MA: $T \in \mathcal{T} :\Leftrightarrow \min(T)$ gerade

Alle diese Transformationen verändern nicht die Struktur der Automaten.

<u>MA $\rightsquigarrow$ NBA</u>

Für jedes $T \in \mathcal{T}$ konstruieren wir NBA $\mathcal{B}_T = (Q', \Sigma, \delta', q_0, \{q_1^1\})$ mit
$L(\mathcal{B}_T) = \{\alpha \mid \alpha$ besitzt Lauf in $\mathcal{A}$ mit $\inf(r) = T\}$.

Sei $T = \{q_1, \ldots, q_t\}$ und seien $T_1, \ldots, T_t$ disjunkte Kopien von T, also $T_i = \{q_1^i, \ldots, q_t^i\}$.
(T_i steht für „seit letztem Endzustand alle Zustände kleiner i gesehen“)

$Q' := Q \cup T_1 \cup \ldots \cup T_t$

$$\begin{aligned}
\delta' = \delta &\cup \left\{ q_r^i \xrightarrow{a} q_s^i \mid i \neq r, q_r \xrightarrow{a} q_s \in \delta \right\} \\
&\cup \left\{ q_r \xrightarrow{\epsilon} q_r^1 \mid 1 \leq r \leq t \right\} \\
&\cup \left\{ q_i^i \xrightarrow{\epsilon} q_i^{i+1} \mid 1 \leq i < t \right\} \\
&\cup \left\{ q_t^t \xrightarrow{\epsilon} q_t^1 \right\}
\end{aligned}$$

$$B := \bigcup_{T \in \mathcal{T}} \mathcal{B}_T$$

Satz von McNaughton: NBA = DMA
Zusammenfassung: DBA $\subsetneq NMA = NRA = NSA = NPA = NBA = DMA = DRA = DSA = DPA$

2.4 Determinisierung von Büchi-Automaten

Satz 2.4.1: (McNaughton) Zu jedem BA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F), n := |Q|$, existiert ein DRA $\mathcal{B}$ mit $2^{\mathcal{O}(n \log n)}$ vielen Zuständen und $\leq 2n$ vielen Akzeptanzpaaren.

Proof. (Safra Konstruktion): $\mathcal{B} := (S, \Sigma, \Delta, s_0, \mathcal{T})$

Zustände: $s = (T, \lambda, M) \in S$ mit

a) T ist Baum mit Knotenmenge $\subseteq \{1, \ldots, 2n\}$, Wurzel und angeordneten Kindknoten.

b) Knotenbeschriftung $\lambda : T \to 2^Q \setminus \{\emptyset\}$

c) Markierte Blätter M von T.

d) $\bigcup_{u \in succ(v)} \lambda(u) \subsetneq \lambda(v)$

e) $\lambda(v) \cap \lambda(u) = \emptyset$, falls weder $u \in succ^*(v)$ noch $v \in succ^*(u)$ (äquivalent zu $\forall u, v \in succ(w) : u \neq v \Rightarrow \lambda(u) \cap \lambda(v) = \emptyset$, d.h. Geschwisterknoten sind disjunkt beschriftet)

Aus d) und e) folgt |Knoten von $T| \leq n$.

Startzustand: s_0 : Nur Wurzel p (z.Bsp. $p = 1$), nicht markiert, $\lambda(p) = \{q_0\}$

Übergangsrelation: $\Delta : S \times \Sigma \to S : \big((T, \lambda, M), a\big) \mapsto \Delta(s, a)$
Verändere Zustand schrittweise:

1) Lösche alle Markierungen

2) Jeder Knoten $v \in T$ bekommt ganz rechts ein neues Kind $v' \in \{1, \ldots, 2n\}$, welches noch nicht verwendet wurde; $\lambda(v') := \lambda(v) \cap F$

3) $\lambda(v) := \delta(\lambda(v), a)$ (Potenzautomatenkonstruktion)

4) $\lambda(v) := \lambda(v) \setminus \{q \in Q \mid \exists$ Knoten $u : u$ links von $T, q \in \lambda(u)\}$

5) Entferne Knoten mit $\lambda(v) = \emptyset$

6) Falls $\lambda(v) = \bigcup_{u \in succ(v)} \lambda(u)$: Markiere v, entferne alle Kinder von v.

$\mathcal{R} := \{(L_v, U_v) \mid v \in \{1, \dots, 2n\}\}$
$L_v := \{(T, \lambda, M) \in S \mid v \in M\}$
$U_v := \{(T, \lambda, M) \in S \mid v \notin T\}$

Zeige $L(\mathcal{A}) \subseteq L(\mathcal{B})$: Sei $\alpha = a_1a_2a_3 \ldots \in L(\mathcal{A})$ mit akzeptierendem Lauf $r : q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} q_3 \ldots$
Dann existiert Berechnung von $\mathcal{B} : s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_3, q_i \in \lambda_i(\text{Wurzel})$.

- Alle s_i besitzen dieselbe Wurzel $p \in \{1, \dots, 2n\}$
- $T := \{v \in \{1, \dots, 2n\} \mid \exists j_0 : \forall j \geq j_0 : v \in T_j \wedge q_j \in \lambda_j(v)\}, p \in T \neq \emptyset$ (alle Knoten, die ab j_0 immer vorkommen, also nie gelöscht werden, d.h. ihre Tiefe ändert sich nicht mehr, und die den Lauf r verfolgen).
- Wähle $x \in T$ mit maximaler Tiefe (Tiefe, mit der $x \in T$ unendlich oft vorkommt). Sei j_0 Zeitpunkt mit $\forall j \geq j_0 : q_j \in \lambda_j(x)$. Nach j_0 kommt q_j nicht mehr links von x vor.
- Unendlich oft werden Kinder von x erzeugt, da unendlich oft Endzustände in r vorkommen. Nicht alle diese Kinder können über Regel 4 & 5 gelöscht werden.
- Wegen Maximalität der Tiefe von x werden diese Kinder unendlich oft nach Regel 6 gelöscht und x markiert.
- α wird durch (L_x, U_x) akzeptiert, d.h. $\alpha \in L(\mathcal{B})$

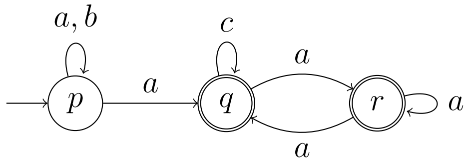
$L(\mathcal{B}) \subseteq L(\mathcal{A}) : \alpha = a_1a_2a_3 \ldots \in L(\mathcal{B})$ mit akzeptierender Berechnung
$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_3 \ldots, s_i := (T_i, \lambda_i, M_i)$ zu Akzeptanzpaar (U_x, L_x).
Sei $0 < i_1 < i_2 < i_3 < \ldots$ mit $\forall i \geq i_1 : x \in T_i$ und $\forall k : x \in M_{i_k}$.
$Q_k := \lambda_{i_k}(x)$
$\alpha[k] := a_{i_{k-1}+1} \ldots a_{i_k}$
$\forall q \in Q_k \exists p \in Q_{k-1} : p \xrightarrow{\alpha[k]}_F q$ in $\mathcal{A}$, da x an den Enden markiert ist.

Lemma von König:
Pfad $q_0 \xrightarrow{\alpha[1]} q_1 \xrightarrow{\alpha[2]}_F q_2 \xrightarrow{\alpha[3]}_F \ldots, q_1 \in Q_1, q_2 \in Q_2, \ldots$
$\Rightarrow \alpha \in L(\mathcal{A})$

Anzahl Zustände: $\leq 2n^{2n-2} \cdot (n+1)^n \cdot 2^n \in 2^{\mathcal{O}(n \cdot \log n)}$

$\square$

Aufgabe: Safra Konstruktion für



2.4.1 DMA → DRA und DPA

Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ein DMA. Sei $\nabla \notin Q$ ein neuer Zustand, Œ $Q = \{1, \ldots, n\}\,, q_0 = 1$.

$Q' := \{w \in (Q \cup \{\nabla\})^* \mid \text{jedes Zeichen aus } Q \cup \{\nabla\} \text{ kommt in } w \text{ genau einmal vor}\}$
$q_0' := \nabla n(n-1)(n-2)\ldots 1$

Idee: $w \in Q'$ repräsentiert den aktuellen Zustand (bezüglich Q) im letzten Zeichen, und davor die zuletzt besuchten Zustände (∇ steht also nie ganz hinten); beim Zustandsübergang wird also der neue Q-Zustand an die letzte Stelle geschoben, ∇ wird an die alte Stelle gesetzt:
Sei also $w = m_1 \ldots m_r \nabla m_{r+1} m_n \in Q'(r < n), \delta(m_n, a) = m_s : \delta'(w, a) := m_1 \ldots m_{s-1} \nabla m_{s+1} \ldots m_n m_s$.

Für $1 \leq i \leq n$:
$U_i := \{u\nabla v \in Q' \mid |u| < i\}$
$L_i := U_i \cup \{u\nabla v \in Q' \mid |u| = i \wedge \{q \in Q \mid |v|_q = 1\} \in \mathcal{T}\}$

Dann gilt $U_1 \subseteq L_1 \subseteq U_2 \subseteq L_2 \subseteq U_3 \subseteq \ldots \subseteq L_n$.

$\mathcal{A}' := (Q', \Sigma, \delta', q_0', \{(L_1, U_1), (L_2, U_2), \ldots, (L_n, U_n)\})$

Zeige: $L_{Muller}(\mathcal{A}) \subseteq L_{Rabin}(\mathcal{A}')$:

> Sei $\alpha \in L_{Muller}(\mathcal{A})$, und r der Run von α auf $\mathcal{A}$. mit $inf(r) = F \in \mathcal{T}$.
> Ab einem Punkt j werden in r nur noch Zustände aus F besucht (und alle treten unendlich oft auf); sei $j' > j$ so, dass von j bis j' jeder Zustand aus F mindestens zweimal von r besucht wurde.
> In $\mathcal{A}'$ gilt ab j' für alle besuchten Zustände $u\nabla v$ „$Q \backslash F \supseteq u$", d.h. $|u| \geq |Q \backslash F| =: i$
> Sei r' der Run von α auf $\mathcal{A}'$, dann gilt: $inf(r') \cap U_i = \emptyset$ und $inf(r') \cap L_i \neq \emptyset$ (der am weitesten links stehende Zustand aus F wird irgendwann wieder besucht).

Zeige: $L_{Rabin}(\mathcal{A}') \subseteq L_{Muller}(\mathcal{A})$:

> Sei $\alpha \in L_{Rabin}(\mathcal{A}')$ und r' der Run von α auf $\mathcal{A}'$, und r der Run auf $\mathcal{A}$.
> $\exists i, w = u\nabla v \in inf(r') \cap (L_i \backslash U_i) \wedge inf(r') \cap U_i = \emptyset \Rightarrow$ „$inf(r) = v \in \mathcal{T}$"

$|Q'| \leq (n+1)!$ (∇ steht nie hinten), $|R| \leq n$ (Einträge $U_i = L_i$ können entfernt werden).

Parity-Automat:

Nummeriere Zustände Q', so dass:

- Zustände aus $U_i \backslash L_{i-1}$ sind ungerade Zahlen ($L_0 = \emptyset, U_{n+1} = Q'$), d.h. akzeptieren nicht.
- Zustände aus $L_i \backslash U_i$ sind gerade Zahlen, d.h. akzeptieren.
- Zustände sind angeordnet: $U_1 \leq L_1 \leq U_2 \leq L_2 \leq \ldots \leq L_n \leq Q'$

$\Rightarrow$ DPA $\mathcal{A}'' := (Q', \Sigma, \delta', q_0')$ mit $L_{Parity}(\mathcal{A}'') = L_{Rabin}(\mathcal{A}') = L_{Muller}(\mathcal{A})$.

2.5 Wiederholung

Satz 2.5.1: Sei $L \subseteq \Sigma^\omega$. Dann sind äquivalent:

a) L wird von NBA akzeptiert, d.h. L ist ω-regulär

b) L^c ist ω-regulär

c) L wird von NRA/NSA/NMA/NPA erkannt

d) L wird von DRA/DSA/DMA/DPA erkannt

e) L ist definierbar in MSO/EMSO

f) $L = \bigcup_{i=1}^{n} U_i V_i^\omega$ für $U_i, V_i \subseteq \Sigma^+$ regulär und $V_i V_i \subseteq V_i, U_i V_i \subseteq U_i$ ($U_i' := U_i V_i$)

Satz 2.5.2: ω-reguläre Sprachen sind effetiv (berechenbar) abgeschlossen unter booleschen Operationen und Projektionen.

Außerdem:

- Leerheit für NBA in NL
- Äquivalenz udn Inklusion für NBA in PSPACE
- Klarlund für Komplementierung von NBA
- Safra für Determinisierung von NBA
- $L_{Muller}(Q, \Sigma, \delta, q_o, \mathcal{T})^c = L_{Muller}(Q, \Sigma, \delta, q_o, 2^Q \backslash \mathcal{T})$

- $L_{Rabin}(\mathcal{A})^c = L_{Streett}(\mathcal{A})$
- Komplementierung von DPA: $Q \to Q' : q \mapsto q+1$
- DBA $\subsetneq$ NBA

2.6 Typische Anwendung: Model-Checking

Sei $\mathcal{A}$ ein Syste, gegeben als ω-Automat. Sei $\mathcal{S}$ eine Spezifikation, z.Bsp. $\mathcal{S} \in$ MSO.
Frage: Erfüllt $\mathcal{A}$ die Spezifikation $\mathcal{S}$.
Lösung: Automatentheoretischer Ansatz:

Konstruiere ω-Automat $\mathcal{B}$ mit $L(\mathcal{B}) = \{\alpha \mid \alpha \text{ erfüllt } \mathcal{S}\}$
Teste ob $L(\mathcal{A}) \subseteq L(\mathcal{B})$, d.h. $L(\mathcal{A}) \backslash (\mathcal{B}) = L(\mathcal{A}) \cap L(\overline{\mathcal{B}}) = L(\mathcal{A} \cap \overline{\mathcal{B}}) \stackrel{?}{=} \emptyset$

Hohe Berechnungskomplexität wird in der Praxis durch Einschränkung von Spezifikationsmechanismen „praktikabel“, häufig aber PSPACE-hart.

Zu unterschiedlichen Spezifikationen passen unterschiedliche Automatenmodelle unterschiedlich gut.

2.7 Deterministische ω-reguläre Sprachen

Definition 2.7.1: Eine ω-reguläre Sprache heißt deterministisch, wenn sie von einem DBA erkannt wird.

Erinnerung: Det. ω-reguläre Sprachen sind abgeschlossen unter Schnitt und Vereinigung, aber nicht uner Komplementbildung.

Definition 2.7.2: Präfixvergleich: $u \leq \alpha :\Leftrightarrow \exists \beta : u\beta = \alpha :\Leftrightarrow$ „u ist Präfix von α".

Definition 2.7.3: Sei $W \subseteq \Sigma^*$. Pfeilsprache $\overrightarrow{W} := \{\alpha \in \Sigma^\omega \mid \alpha$ hat unendliche viele Präfixe in $W\}$ $= \{\alpha \in \Sigma^\omega \mid \forall u \leq \alpha \exists v \in \Sigma^* : uv \leq \alpha \wedge uv \in W\} = \bigcap_{n \geq 0} (W \cap \Sigma^{\geq n})\Sigma^\omega$

Beobachtung: Sei $\mathcal{A}$ ein DBA. Dann gilt $L(\mathcal{A}) = \overrightarrow{L_{fin}(\mathcal{A})}$.
Es gilt also auch: W regulär $\Rightarrow \overrightarrow{W}$ ω-regulär (interpretiere DFA als DBA).

Satz 2.7.4: L ist ω-regulär $\Leftrightarrow \exists V_i, W_i \subseteq \Sigma^*$ regulär: $L = \bigcup_{i=1}^{n} (\overrightarrow{W_i} \setminus \overrightarrow{V_i})$

Proof. „$\Leftarrow$": trivial
„$\Rightarrow$": Sei $L = L(\mathcal{A})$ für DRA $\mathcal{A} = (Q, \Sigma, \delta, q_0, R), R = \{(L_1, U_1), \ldots, (L_k, U_k)\}$
Erinnerung: $\mathcal{A}$ akzeptiert „$\exists i : L_i \wedge \neg U_i$"
Also gilt: $L = \bigcup_{i=1}^{k} L_{DBA}(Q, \Sigma, \delta, q_0, L_i) \setminus L_{DBA}(Q, \Sigma, \delta, q_0, U_i) = \bigcup_{i=1}^{k} (\overrightarrow{W_i} \setminus \overrightarrow{V_i})$
für $W_i := L_{fin}(Q, \Sigma, \delta, q_0, L_i), V_i := L_{fin}(Q, \Sigma, \delta, q_0, U_i)$ $\square$

Definition 2.7.5: Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ein Automat.
$E \subseteq Q$ heißt Schleife, falls E stark zusammenhängend bezüglich δ Kanten.
DMA $\mathcal{A}$ heißt bereinigt, falls alle Zustände erreichbar sind und alle $T \in \mathcal{T}$ Schleifen sind.
(Jede ω-reguläre Sprache wird von einem bereinigten DMA erkannt.)
Ein bereinigter DMA $\mathcal{A}$ ist unter Vereinigung mit nicht disjunkten Schleifen abgeschlossen, falls: $\forall T \in \mathcal{T}, E$ Schleife mit $T \cap E \neq \emptyset \Rightarrow T \cup E \in \mathcal{T}$.

Satz 2.7.6: (Landwehr) Sei L eine ω-reguläre Sprache. Dann sind äquivalent:

a) L ist deterministisch

b) $\exists W \subseteq \Sigma^*$ regulär: $L = \overrightarrow{W}$

c) $\exists W \subseteq \Sigma^*$: $L = \overrightarrow{W}$

d) Jeder bereinigte DMA, der L erkennt, ist unter Vereinigung mit nicht disjunkten Schleifen abgeschlossen.

e) Es existiert ein bereinigter DMA, der L erkennt und unter Vereinigung mit nicht disjunkten Schleifen abgeschlossen ist.

Proof. a) $\Rightarrow$ b) siehe oben, b) $\Rightarrow$ c) trivial, d) $\Rightarrow$ e) trivial.
c) $\Rightarrow$ d): Sei $W \subseteq \Sigma^*, L = \overrightarrow{(}W) = L(\mathcal{A})$ für einen bereinigten DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$.
Sei $T \in \mathcal{T}, E \subseteq Q$ eine Schleife mit $T \cap E \neq \emptyset$ und $q \in T \cap E$.

Es existieren $u, v, x \in \Sigma^*$ mit $q_0 \xrightarrow{u} q, q \xrightarrow{v} q, q \xrightarrow{x} q$, wobei v genau T durchläuft und x genau E durchläuft.

Da $\alpha_l := u(v^{n_i}x)_{i=1}^{l} v^\omega \in L(\mathcal{A}) = \overrightarrow{W} \Rightarrow \exists n_{l+1}, a \leq v^{n_{l+1}} : p_{l+1} := u(v^{n_i}x)_{i=1}^{l} a \in W$
Damit hat über induktiven Aufbau $\alpha = u(v^{n_i}x)_{i=1}^{\infty}$ unendlich viele Präfixe p_i in W, also $\alpha \in \overrightarrow{W} = L(\mathcal{A})$ und $T \cup E \in \mathcal{T}$.

e) $\Rightarrow$ a): Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T}), \mathcal{T} = \{T_1, \dots, T_m\}$ nach e).
$L = L(\mathcal{A}) = \bigcup_{i=1}^{m} L_{Muller}(Q, \Sigma, \delta, q_0, \{T_i\})$

Für ein $T \in \mathcal{T}$ sei $T = \{f_1, \dots, f_k\}$:
$L_{Muller}(Q, \Sigma, \delta, q_0, \{T\}) \subseteq \bigcap_{j=1}^{k} L_{DBA}(Q, \Sigma, \delta, q_0, \{f_j\}) =: L_T$ deterministisch.

Behauptung: $L_T \subseteq L(\mathcal{A})$
Sei $\alpha \in L_T$ und r ein Lauf von $\mathcal{A}$ auf α, also $T \subseteq inf(r)$.
D.h. $inf(r) = T \cup E$ für eine Schleife E mit $T \cap E \neq \emptyset$. Also $T \cup E \in \mathcal{T} \Rightarrow \alpha \in L(\mathcal{A})$.

$\Rightarrow \bigcup_{T \in \mathcal{T}} L_T = L$ deterministisch. $\square$

Korollar 2.7.7: Es ist entscheidbar, ob eine ω-reguläre Sprache deterministisch ist.

3 Cantor-Topologie

3.1 Borel-Hierarchie

Seien X, Y Mengen von Mengen, X abgeschlossen unter abzählbarer Vereinigung und endlichem Schnitt, $e \in X \Leftrightarrow e^c \in Y$.

Abzählbare Vereinigung („Summe“): $M_\sigma = \left\{ \bigcup\limits_{i=0}^{\infty} M_i \mid M_i \in M \right\}$
Abzählbarer Durchschnitt: $M_\delta = \left\{ \bigcap\limits_{i=0}^{\infty} M_i \mid M_i \in M \right\}$
Endliche Boolsche Ausdrücke: $\mathbb{B}(M) = \{m, m_1^c, m_1 \cap m_2, m_1 \cup m_2 \mid m \in M, m_1, m_2 \in \mathcal{B}(M)\}$

Beobachtung: $X, Y \subseteq \mathbb{B}(X) = \mathbb{B}(Y)$
Beobachtung: $\mathbb{B}(X \cap Y) = X \cap Y : m \in X \cap Y \Rightarrow m^c \in X \cap Y$

(Y_σ, X_δ) erfüllt die Voraussetzungen wie (X, Y), da $e \in Y_\sigma \Leftrightarrow e^c \in X_\delta$ usw, also auch $\mathbb{B}(X_\delta \cap Y_\sigma) = X_\delta \cap Y_\sigma$.

Gilt nun $X \subseteq Y_\sigma$, dann auch:

- $Y \subseteq X_\delta$
- $X_\delta \subseteq Y_{\sigma\delta}$
- usw.

Eine solche Struktur lässt sich als Borel-Hierarchie schreiben:

$$\begin{tikzcd}
 & X \arrow[dr, hook] & & & X_\delta \arrow[dr, hook] & & \\
X \cap Y \arrow[ur, hook] \arrow[dr, hook] & & \mathbb{B}(X) = \mathbb{B}(Y) \arrow[r, hook] & X_\delta \cap Y_\sigma \arrow[ur, hook] \arrow[dr, hook] & & \mathbb{B}(X_\delta) = \mathbb{B}(Y_\sigma) \arrow[r, hook] & X_{\delta\sigma} \cap Y_{\sigma\delta} \ \cdots \\
 & Y \arrow[ur, hook] & & & Y_\sigma \arrow[ur, hook] & &
\end{tikzcd}$$

3.2 Topologie

Definition 3.2.1: Eine ω-Sprache L heißt offen $:\Leftrightarrow \exists W \subseteq \Sigma^* : L = W\Sigma^\omega$.
G := offene Mengen („Gebiete“), F := geschlossene Mengen (Komplemente von offenen Mengen).
Dies definiert eine Topologie auf ω-Wörtern.
Satz 3.2.2: $\overline{L} = \{\alpha \in \Sigma^\omega \mid \forall p \leq \alpha \exists \beta : p\beta \in L\}$

Proof. $K := \{\alpha \in \Sigma^\omega \mid \forall p \leq \alpha \exists \beta : p\beta \in L\}, \overline{L} = \bigcap\limits_{\text{abg. } U \supseteq L} U$ nach Definition.
„$\supseteq$“: Sei U abgeschlossen, $U \supseteq L$, also U^c offen.
Sei $\alpha \in p\Sigma^\omega \subseteq U^c$, d.h. $p \leq \alpha, \nexists \beta : p\beta \in U \supseteq L \Rightarrow \alpha \notin K \Rightarrow K \subseteq U \Rightarrow K \subseteq \overline{L}$.

„$\subseteq$“: Sei $\alpha \notin K$, d.h. $\exists p \leq \alpha : \alpha \in p\Sigma^\omega \subseteq L^c$.
$U := (p\Sigma^\omega)^c$ abgeschlossen, $U \supseteq L, \alpha \notin U \Rightarrow \alpha \notin \overline{L}$. □

Lemma 3.2.3: Für $w \in \Sigma^*$: $w\Sigma^\omega \in F$, da $(w\Sigma^\omega)^c = (\Sigma^{|w|} \setminus \{w\})\Sigma^\omega \in G$.
Lemma 3.2.4: $G \subseteq F_\sigma$, da: $L \in G \Rightarrow \exists W \subseteq \Sigma^* : L = W\Sigma^\omega = \bigcup\limits_{w \in W} w\Sigma^\omega \in F_\sigma$.
Damit kann man aus der Topologie obige Hierarchie bauen.
Satz 3.2.5: $\exists W \subseteq \Sigma^* : L = \overrightarrow{W} \Leftrightarrow L \in G_\delta$
Insbesondere ist entscheidbar, ob eine ω-reguläre Sprache L in G_δ bzw. F_δ ist.

Proof. „$\Rightarrow$“: Trivial - $\overrightarrow{W} = \bigcap\limits_{n \geq 0} (W \cap \Sigma^{\geq n})\Sigma^\omega$

„$\Leftarrow$“: Sei $L = \bigcup\limits_{n \geq 0} W_n\Sigma^\omega$.
Da $(\bigcup\limits_{i \leq n} W_i\Sigma^*)\Sigma^\omega = \bigcup\limits_{i \leq n} W_i\Sigma^\omega$, kann W_n durch $\bigcup\limits_{i \leq n} W_i\Sigma^*$ ersetzt werden.
Also gilt Œ $W_0 \supseteq W_1 \supseteq W_2 \supseteq \ldots$.

Sei W_n' minimal mit $W_n'\Sigma^\omega = W_n\Sigma^\omega$; dann ist W_n' präfixfrei.

$W_n'' := W_n'\Sigma^n$. Es gilt $W_n''\Sigma^\omega = W_n\Sigma^\omega, W_n''$ präfixfrei.
Für $m < n$ sei $w \in W_m'' \cap W_n''$. Also ist $w = ab = cd$ für $a \in W_n', b \in \Sigma^n$, $c \in W_m', d \in \Sigma^m$.
Da $m < n$ ist $|c| > |a|$, also ist a ist echter Präfix von $c : a < c$.
Da nun $a\Sigma^\omega \subseteq W_n'\Sigma^\omega = W_n\Sigma^\omega \subseteq W_m\Sigma^\omega = W_m'\Sigma^\omega$, existiert ein $p \in W_m' : p \leq a < c$. Also wäre W_m' nicht präfixfrei $\Rightarrow$ Widerspruch - ein solches w kann nicht existieren.
$\Rightarrow \forall m < n : W_m'' \cap W_n'' = \emptyset$

$W := \bigcup\limits_{n \geq 0} W_n''$
Behauptung: $L = \overrightarrow{W}$
„$\supseteq$“: Sei $\alpha \in \overrightarrow{W}$, dann gibt es unendlich viele n so, dass α einen Präfix in W_n'' besitzt, d.h.

$\alpha \in W_n''\Sigma^\omega = W_n\Sigma^\omega = \bigcap_{i\leq n} W_i\Sigma^\omega \Rightarrow \alpha \in \bigcap_{n\geq 0} W_n\Sigma^\omega$.
„$\subseteq$“: Sei $\alpha \in \bigcap_{n\geq 0} W_n''\Sigma^\omega$, dann gibt es für jedes n einen Präfix $p \leq \alpha$ aus W_n'', die alle verschieden sind $\Rightarrow \alpha \in \overrightarrow{W}$. □

Korollar 3.2.6: Sei L ω-regulär, dann gilt: L ist deterministisch $\Leftrightarrow L \in G_\delta$
Korollar 3.2.7: Wenn L ω-regulär, dann $L \in \mathbb{B}(G_\delta) = \mathbb{B}(F_\sigma)$.
Satz 3.2.8: Sei $L \in F$ abgeschlossen. Dann ist $L = \overrightarrow{W}$ für $W := \{p \in \Sigma^* \mid \exists \alpha \in L : p \leq \alpha\}$.
Also gilt $F \subseteq G_\delta$ (dies folgt auch wegen Symmetrie aus $G \subseteq F_\sigma$).

Proof. „$\subseteq$“: klar
„$\supseteq$“: Sei $\alpha \in \overrightarrow{W}$, d.h. $\forall p \leq \alpha : \exists u : pu \leq \alpha, pu \in W$.
Nach Definition von W: $\exists \gamma : pu\gamma \in L \Rightarrow \alpha \in \overline{L} = L$. □

Definition 3.2.9: Sei r ein Run. $Occ(r) := \{q \in Q \mid q$ kommt in r vor$\}$.

3.3 Schwache Akzeptanzbedingungen

Schwache Akzeptanzbedingungen verwenden $Occ(r)$ an Stelle von $inf(r)$.
Definition 3.3.1: Ein Automat $\mathcal{A} = (Q, \Sigma, \delta, q_0, F), F \subseteq Q$, 1-akzeptiert $L_1(\mathcal{A}) := L = \{\alpha \in \Sigma^\omega \mid \exists \text{Run } r : Occ(r) \cap F \neq \emptyset\}$
Dies ist das „Occ-Büchi“ Äquivalent. Wenn $\mathcal{A}$ vollständig ist, ist $L = L_{fin}(\mathcal{A})\Sigma^\omega \in G$
Definition 3.3.2: Ein Automat $\mathcal{A} = (Q, \Sigma, \delta, q_0, F), F \subseteq Q$, 1'-akzeptiert $L_{1'}(\mathcal{A}) := L = \{\alpha \in \Sigma^\omega \mid \exists \text{Run } r : Occ(r) \subseteq F\}$
$\mathcal{A}$ kann deterministisch und vollständig gemacht werden:

- Entfernen aller nicht akzeptierenden Zustände $Q \backslash F$ samt Übergängen.
- Potenzautomat $\Rightarrow$ deterministisch
- Vervollständigung durch Fangzustand.

Dann 1-akzeptiert $(Q, \Sigma, \delta, q_0, Q\backslash F)$ das Komplement von L, also $L \in F$.
Definition 3.3.3: Ein Automat $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T}), \mathcal{T} \subseteq 2^Q$, staiger-wagner-akzeptiert $L_{sw}(\mathcal{A}) := L = \{\alpha \in \Sigma^\omega \mid \exists \text{Run } : Occ(r) \in \mathcal{T}\}$
Definition 3.3.4: Ein bereinigter DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ist
abgeschlossen unter erreichbaren Schleifen, falls
$\forall T \in \mathcal{T} : \forall$ Schleifen $E : (\exists t \xrightarrow{*} e \in \delta$ für $t \in T, e \in E) \Rightarrow E \in \mathcal{T}$
Satz 3.3.5: Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ein bereinigter DMA. Dann sind äquivalent:

a) $L(\mathcal{A}) \in G$

b) $\mathcal{A}$ ist abgeschlossen unter erreichbare Schleifen

c) $L(\mathcal{A})$ ist 1-akzeptierbar durch einen vollständigen Automaten

Proof. a) $\Rightarrow$ b): Sei E eine Schleife, $e \in E, t \in T \in \mathcal{T}, u, v, w, x \in \Sigma^* : q_0 \xrightarrow{u} t, t \xrightarrow{v} t, t \xrightarrow{w} e, e \xrightarrow{x} e \in \delta$, wobei v genau T durchläuft und x genau E.

$$\begin{tikzcd} q_0 \arrow[r, "u"] & t \arrow[loop above, "v"] \arrow[r, "w"] & e \arrow[loop right, "x"] \end{tikzcd}$$

$uv^\omega \in L(\mathcal{A}) \in G \Rightarrow \exists n : uv^n\Sigma^\omega \in L(\mathcal{A}) \Rightarrow uv^nwx^\omega \in L(\mathcal{A}) \Rightarrow E \in \mathcal{T}$

b) $\Rightarrow$ c): $F := \bigcup\limits_{T \in \mathcal{T}} T$

c) $\Rightarrow$ a): Siehe oben $\square$

Korollar 3.3.6: Es ist entscheidbar, ob eine ω-reguläre Sprache offen bzw. abgeschlossen ist.

Definition 3.3.7: Ein bereinigter DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ist abgeschlossen unter kleineren Schleifen, falls
$\forall T \in \mathcal{T} : \forall$ Schleifen $E \subseteq T : E \in \mathcal{T}$. Dies ist genau dann der Fall, wenn $\overline{\mathcal{A}} = (Q, \Sigma, \delta, q_0, 2^Q \setminus \mathcal{T})$ unter Vereinigung mit nicht disjunkten Schleifen abgeschlossen ist, d.h. $L(\mathcal{A})^c \in G_\delta \Leftrightarrow L(\mathcal{A}) \in F_\sigma$.

Satz 3.3.8: Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ein bereinigter DMA. Dann sind äquivalent:

a) $L(\mathcal{A}) \in F_\sigma$

b) $L(\mathcal{A})$ ist 1-akzeptierbar durch nicht-deterministischen Automaten (nicht notwendigerweise vollständig!)

c) $L(\mathcal{A})$ ist staiger-wagner-akzeptierbar durch nicht-deterministischen Automaten

Proof. a) $\Rightarrow$ b): Sei $T \subseteq \mathcal{T}$ und T' eine Kopie on T (mit $T' \cap Q = \emptyset$).
$\mathcal{B}_T := (Q \cup T', \Sigma, \delta', q_0, T')$ mit
$\delta' := \delta \cup \{(p', a, q') \mid (p, a, q) \in \delta, p, q \in T\} \cup \{(q, \epsilon, q') \mid q \in T\}$

Behauptung: $L_{Muller}(Q, \Sigma, \delta, q_0, \{T\}) \subseteq L_1(\mathcal{B}) \subseteq L(\mathcal{A})$

Erstes „$\subseteq$“: Ein Lauf r in $\mathcal{A}$ mit $inf(r) = T$ kann in $\mathcal{B}$ nach endlich vielen Schritten nach T' wechseln.
Zweites „$\subseteq$“: Für einen Lauf r in $\mathcal{B}$ mit $occ(r) \cap T' \neq \emptyset$ gilt insbesondere $inf(r) \subseteq T'$, in $\mathcal{A}$ also $inf(r) \subseteq T$. Da $\mathcal{A}$ abgeschlossen unter kleineren Schleifen $inf(r) \in \mathcal{T}$.

$\mathcal{B} := \bigcup_{T \in \mathcal{T}} \mathcal{B}_T \Rightarrow L_1(\mathcal{B}) = L(\mathcal{A})$ ($\mathcal{B}$ i.A. weder vollständig noch deterministisch)

b) $\Rightarrow$ c): Sei $\mathcal{B} = (Q, \Sigma, \delta, q_0, F)$ ein Automat mit $L_1(\mathcal{B}) = L(\mathcal{A})$.
Dann $L_{sw}(Q, \Sigma, \delta, q_0, \{G \subseteq Q \mid G \cap F \neq \emptyset\}) = L_1(\mathcal{B})$.

c) $\Rightarrow$ a): Sei $L = L_{sw}(\mathcal{B})$ für $\mathcal{B} = (Q, \Sigma, \delta, q_0, \mathcal{F})$. Zu zeigen: L^c ist Pfeilsprache.
Œ: $\mathcal{B}$ vollständig.
$W := \{w \in \Sigma^* \mid \forall$ Runs r auf $w : Occ(r) \notin \mathcal{F}\}$
Behauptung: $\overrightarrow{W} = L^c$

„$\subseteq$“: Sei $\alpha \in L \Rightarrow \exists$ Run r mit $Occ(r) \in \mathcal{F}$, d.h. für alle genügend langen Präfixe r' von r gilt $Occ(r) = Occ(r') \in \mathcal{F} \Rightarrow \alpha \notin \overrightarrow{W}$.
„$\supseteq$“: Sei $\alpha \notin L$, $G := \{Occ(r) \mid \text{Run } r \text{ auf } \alpha\}$.
Da $\alpha \notin L \Rightarrow G \cap F = \emptyset$.
Für jeden Run r auf α gibt es einen genügend langen Präfix r' von r mit $Occ(r') = Occ(r) \in G \Rightarrow Occ(r') \notin \mathcal{F} \Rightarrow \alpha \in \overrightarrow{W}$ □

Satz 3.3.9: Sei $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{T})$ ein bereinigter DMA. Dann sind äquivalent:

a) $L(\mathcal{A}) \in \mathbb{B}(G) = \mathbb{B}(F)$

b) $L(\mathcal{A}) \in F_\sigma \cap G_\delta$

c) $L(\mathcal{A})$ ist staiger-wagner-akzeptierbar durch deterministischen Automaten

Proof. a) $\Rightarrow$ b): trivial
b) $\Rightarrow$ c): $\mathcal{T}$ ist abgeschlossen unter „Vergrößerung“ (G_δ) und „Verkleinerung“ (F_σ) von Schleifen, d.h. falls $C \subseteq Q$ starke Zusammenhangskomponente ist, dann ist entweder jede Schleife in C in $\mathcal{T}$ oder keine.
$\Rightarrow$ Akzeptanz hängt nur von zuletzt besuchter starken Zusammenhangskomponente ab. Dies lässt sich als Staiger-Wagner-Akzeptanz formulieren:
$\mathcal{F} := \{F \subseteq Q \mid \exists \text{ endlicher Run } r \text{ auf } w \in \Sigma^* : Occ(r) = F \wedge \delta(q_0, w) \in T, T \in \mathcal{T}\}$

c) $\Rightarrow$ a): Sei $L = L_{sw}(\mathcal{B})$ für $\mathcal{B} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ deterministischer Automat.
$L_q := L_{fin}(Q, \Sigma, \delta, q_0, \{q\})$
$L_T := (\bigcap_{q \in T} L_q \Sigma^\omega) \backslash (\bigcup_{q \notin T} L_q \Sigma^\omega) \in \mathbb{B}(G)$
Es gilt: $L_T = L_{sw}(Q, \Sigma, \delta, q_0, \{T\})$, da $\mathcal{B}$ deterministisch ist.
$L = \bigcup_{T \in \mathcal{T}} L_T$ □

Bemerkung: $\mathbb{B}(G) \cap \omega$-regulär $= \mathbb{B}(G \cap \omega\text{-regulär}) = F_\sigma \cap G_\delta \cap \omega$-regulär.
Korollar 3.3.10: Innerhalb der ω-regulären Sprachen ist $\mathcal{B}(G)$ enscheidbar.

4 Übersicht

Die roten Mengen sind jeweils der Schnitt der angegeben Menge mit den ω-regulären Sprachen.

Innerhalb der ω-regulären Sprachen ist die Einordnung jeweils entscheidbar (anhand der Schleifen im bereinigten DMA).

$$
\begin{tikzcd}
\text{vollst. 1-Akzeptanz} & & & \text{det. Büchi, Pfeilsprachen} & \\
F \arrow[dr, hook] & & & F_\delta \arrow[dr, hook] & \\
& \mathbb{B}(F) = \mathbb{B}(G) \arrow[r, hook] & F_\delta \cap G_\sigma \arrow[ur, hook] \arrow[dr, hook] & & \mathbb{B}(F_\delta) = \mathbb{B}(G_\sigma) \\
G \arrow[ur, hook] & & & G_\sigma \arrow[ur, hook] & \omega\text{-regulär} \\
& \text{det. Staiger-Wagner} & & & \\
\text{1'-Akzeptanz} & & & \text{ndet. Staiger-Wagner, ndet. 1-Akzeptanz} &
\end{tikzcd}
$$

5 Syntaktische Monoide

5.1 Erkennung durch endliche Monoide

Definition 5.1.1: $L \subseteq \Sigma^*$ heißt erkennbar, falls $\exists$ Homomorphismus $\varphi : \Sigma^* \to M$, M endliches Monoid, mit $\varphi^{-1}\varphi(L) = L$, d.h. $w \in L \Rightarrow \varphi(w) \in \varphi(L)$.

Sei $\varphi : \Sigma^* \to M$ ein Homomorphismus, M endliches Monoid.
Lemma 5.1.2: $\forall \alpha \in \Sigma^\omega \exists$ Zerlegung $\alpha = x_0 y_1 y_2 \ldots$ und $s, e \in M$ mit $x_0, y_i \in \Sigma^+, \varphi(x_0) = s, \varphi(y_i) = e, se = s, e^2 = e$

Proof. Betrachte vollständigen Graph auf Positionen $\{0, 1, 2, \ldots\}$ von α, gefärbte Kanten $\{i, j\}$ für $i < j$ mit Farben $c(i, j) := \varphi(\alpha_{i..j-1})$ aus M.
Dann gibt es nach Satz von Ramsey einen unendlichen einfarbigen vollständigen Graph mit den Knoten $\{l_0, l_1, l_2, l_3, \ldots\}, l_i < l_{i+1}$ und der Kantenfarbe e.
$x' := \alpha_{0..l_0-1}, y_i := \alpha_{l_i..l_{i+1}-1} \Rightarrow \varphi(y_i) = c(l_i, l_{i+1}) = e$
$e^2 = \varphi(y_1)\varphi(y_2) = \varphi(y_1 y_2) = c(l_1, l_3) = e$
$x_0 := x' y_0, s := \varphi(x_0) = \varphi(x' y_0) = \varphi(x')e \Rightarrow se = \varphi(x')ee = \varphi(x')e = s$

□

Definition 5.1.3: Ein Paar $(s, e) \in M^2$ mit $se = s, e^2 = e$ heißt linked pair (l.p.)
Definition 5.1.4: $[m] := \varphi^{-1}(m) \cap \Sigma^+$
Aus φ kann ein endlicher Automat gebastelt werden: $\delta(m, a) := m \cdot \varphi(a)$.
Dann ist $[m] = L_{fin}(M, \Sigma, \delta, e, \{m\}) \cap \Sigma^+$, also ist $[m]$ regulär.
Definition 5.1.5:

- φ erkennt $L \subseteq \Sigma^\omega$ schwach, falls

$$L = \bigcup \{[s][e]^\omega \mid (s, e) \text{ linked pair mit } [s][e]^\omega \subseteq L\}$$

- φ erkennt $L \subseteq \Sigma^\omega$ (stark), falls

$$L = \bigcup \{[s][e]^\omega \mid (s, e) \text{ linked pair mit } [s][e]^\omega \cap L \neq \emptyset\}$$

- $\alpha \approx_\varphi \beta :\Leftrightarrow \exists x_i, y_i \in \Sigma^+ : \alpha = x_0 x_1 \ldots, \beta = y_0 y_1 \ldots, \varphi(x_i) = \varphi(y_i)$

- $\approx_\varphi$ sei der transitive Abschluss von $\hat{\approx}_\varphi$, $[\alpha]_{\approx_\varphi}$ die Äquivalenzklassen
- $\approx_\varphi$ <u>saturiert</u> $L \subseteq \Sigma^\omega$, falls $L = \bigcup \{[\alpha]_{\approx_\varphi} \mid \alpha \in L\}$

Lemma 5.1.6: φ erkennt $L \subseteq \Sigma^\omega \Leftrightarrow \approx_\varphi$ saturiert L.

Proof. „$\Rightarrow$“: Sei $\alpha \in L$ und $\alpha \approx_\varphi \beta$. Zu Zeigen: $\beta \in L$.

Konstruiere vollständigen Graph mit Knoten $\{0, 1, 2, \ldots\}$ und Kantenfarben (für $i < j$) $c(i, j) := \varphi(x_i x_{i+1} \ldots x_{j-1}) = \varphi(y_i y_{i+1} \ldots y_{j-1})$.
Wie oben gilt nach Satz vom Ramsey: $\exists$ linked pair $(s, e) : \alpha, \beta \in [s][e]^\omega$
Also $[s][e]^\omega \cap L \neq \emptyset \Rightarrow \beta \in [s][e]^\omega \subseteq L$.

„$\Leftarrow$“: Sei $\alpha \in [s][e]^\omega \cap L$ für ein linked pair (s, e). Zu Zeigen: $[s][e]^\omega \subseteq L$
Sei $\beta \in [s][e]^\omega$. Dann $\exists x_i, y_i \in \Sigma^+ : \alpha = x_0 x_1 x_2 \ldots, \beta = y_0 y_1 y_2 \ldots,$
$\varphi(x_0) = \varphi(y_0) = s, \forall i > 1 : \varphi(x_i) = \varphi(y_i) = e \Rightarrow \alpha \approx_\varphi \beta \Rightarrow \beta \in L$ $\square$

Wenn φL (stark) erkennt, dann auch schwach. Die Umkehrung gilt nicht:

Beispiel 5.1.7: $M := \{1, a, b\}$ Monoid mit $aa = ba = a, bb = ab = b$
$\Sigma = a, b, \varphi : \Sigma^* \to M$ kanonisch ($\epsilon \mapsto 1, a \mapsto a, b \mapsto b$).
linked pair $(s, e) := (a, a), L := [a][a]^\omega = (\Sigma^* a)^\omega = \{\alpha \mid a$ kommt in α unendlich oft vor$\}$.
Also $(ab)^\omega \in L \cap [b][b]^\omega$, aber $b^\omega \in [b][b]^\omega \backslash L$
$\Rightarrow \varphi$ erkennt L <u>nicht</u> stark.

Bemerkung. Wenn φ die Sprache $L \subseteq \Sigma^\omega$ erkennt, dann auch L^c - jedes Wort hat ein linked pair, dass entweder ganz oder gar nicht in L ist.
Gilt im Allgemeinen nicht für schwache Erkennung; von obigem Beispiel:
Es gibt nur die linked pairs $(1, 1)$, (a, a) und (b, b). $[1][1]^\omega = \emptyset, [a][a]^\omega \cap [b][b]^\omega \neq \emptyset$, also kann L^c von φ nicht schwach erkannt werden.

Satz 5.1.8: $\varphi : \Sigma^* \to M$ erkenne L, $\psi : \Sigma^* \to N$ erkenne K, dann gilt:
$\varphi \times \psi : \Sigma^* \to M \times L$ erkennt sowohl $L \cap K$ als auch $L \cup K$.
($L \cup K$ gilt auch für jeweils schwache Erkennung)

Satz 5.1.9: Sei $L \subseteq \Sigma^\omega$. Dann sind äquivalent:

a) L ist ω-regulär

b) $\exists \varphi : \varphi$ erkennt L

c) $\exists \varphi : \varphi$ erkennt L schwach

Proof. a) $\Rightarrow$ b) Sei $\mathcal{A}$ ein NBA mit $L(\mathcal{A}) = L$.
Definiere für $u, v \in \Sigma^+ : u \equiv_\mathcal{A} v :\Leftrightarrow \forall p, q \in Q : (p \xrightarrow{u} q \leftrightarrow p \xrightarrow{v} q) \wedge (p \xrightarrow{u}_F q \leftrightarrow p \xrightarrow{v}_F q)$.
$\equiv_\mathcal{A}$ ist eine Äquivalenzrelation mit endlichem Index (d.h. endlich vielen Äquivalenzklassen, siehe **??**)

Außerdem ist $\equiv_\mathcal{A}$ eine Kongruenzrelation, d.h. $u \equiv_\mathcal{A} v \Rightarrow \forall x, y : xuy \equiv_\mathcal{A} xvy$
$\Rightarrow M := \Sigma^*/\equiv_\mathcal{A}, \varphi : \Sigma^* \to M : u \mapsto [u]_{\equiv_\mathcal{A}}$ ist Monoidhomomorphismus.

Es gilt: $[u]_{\equiv_\mathcal{A}}[v]^\omega_{\equiv_\mathcal{A}} \cap L \neq \emptyset \Rightarrow [u]_{\equiv_\mathcal{A}}[v]^\omega_{\equiv_\mathcal{A}} \subseteq L$
(Runs mit unendlich vielen Endzuständen für ein Wort entsprechen „ähnlichen“ Runs für „äquivalente“ Wörter, die auch unendlich viele Endzustände durchlaufen.)

$\Rightarrow \varphi$ erkennt L.

b) $\Rightarrow$ c): trivial
c) $\Rightarrow$ a): Endliche Vereinigung von Sprachen der Form $[s][e]^\omega \Rightarrow \omega$-regulär (da $[m]$ regulär)
$\square$

Definition 5.1.10: Sei $\mathcal{A}$ ein NBA. Definiere für $u, v \in \Sigma^*$:
$u \equiv'_\mathcal{A} v :\Leftrightarrow \forall p, q \in Q : (p \xrightarrow{u} q \leftrightarrow p \xrightarrow{v} q)$.
$\Sigma^*/\equiv'_\mathcal{A}$ heißt Transitionsmonoid von $\mathcal{A}$.
$\varphi : \Sigma^* \to \Sigma^*/\equiv'_\mathcal{A}: u \mapsto [u]_{\equiv'_\mathcal{A}}$ erkennt $L(\mathcal{A})$ schwach.

5.2 Das syntaktische Monoid nach Arnold

Definition 5.2.1: Sei $L \subseteq \Sigma^\omega, u, v \in \Sigma^*$: definiere syntaktische Kongruenz:

$$u \equiv_L v :\Leftrightarrow \forall x, y, z \in \Sigma^* : (xuyz^\omega \in L \leftrightarrow xvyz^\omega \in L) \wedge (x(uy)^\omega \in L \leftrightarrow x(vy)^\omega \in L)$$

$\equiv_L$ ist eine Kongruenzrelation, da Äquivalenzrelation und $u \equiv_L v \Rightarrow \forall p, q \in \Sigma^* : puq \equiv_L pvq$.
Syntaktisches Monoid: $Synt(L) := \Sigma^*/\equiv_L$
Syntaktischer Homomorphismus: $\varphi_L : \Sigma^* \to Synt(L) : u \mapsto [u]_{\equiv_L}$
Satz 5.2.2: L ω-regulär $\Leftrightarrow \varphi_L$ erkennt L.
Satz 5.2.3: Sei $\varphi : \Sigma^* \twoheadrightarrow M$ ein surjektiver Homomorphismus, der L erkennt. Dann existiert ein Homomorphismus $\psi : M \to Synt(L)$ mit $\varphi_L = \psi \circ \varphi$

Proof. Behauptung: Da φ L erkennt: $\varphi(u) = \varphi(v) \Rightarrow u \equiv_L v$
Also ist $\psi : M \to Synt(L) : \varphi(u) \mapsto [u]_{\equiv_L}$ wohldefiniert.

Zu Zeigen: $\varphi(u) = \varphi(v) \Rightarrow u \equiv_L v$

Zerlege die Wortpaare $(xuyz^\omega, xvyz^\omega)$ und $(x(uy)^\omega, x(vy)^\omega)$ an den Teilwortgrenzen von u, v, x, y, z (da u nicht notwendigerweise gleichlang wie v ist, trifft die Zerlegung eventuell verschiedene Positionen). Auf den jeweilig entsprechenden Teilen der Zerlegungen stimmt φ überein, da $\varphi(u) = \varphi(v)$. Es lässt sich also wieder ein Graph konstruieren mit $\varphi(w)$ als Kantenbeschriftung zwischen den Auftrennungen, der für jeweils beide Wörter übereinstimmt. Mit Satz von Ramsey existiert also ein gemeinsames linked pair, mit der (starken) Erkennung von φ folgt die Aussage.

□

Korollar 5.2.4: $Synt(L)$ ist das bis auf Isomorphie eindeutige kleinste Monoid, das eine ω-reguläre Sprache erkennt.

Beispiel. Sei $L = \{\alpha\}$ für ein nicht ultimativ periodisches Wort, d.h. $\nexists u, v : \alpha = uv^\omega$, z.Bsp. $1, \alpha = \sqrt{2}$.
$\Rightarrow L$ nicht ω-regulär
$Synt(L) = \{1\}$ trivial, aber erkennt L nicht.

Bemerkung. $Synt(L)$ und φ_L sind effektiv z.Bsp. aus einem NBA berechenbar (als Vergröberung von $\equiv_\mathcal{A}$).

5.3 Anwendung des algebraischen Ansatzes

Resultate der Form:
Eine ω-reguläre Sprache L hat die Eigenschaft $\mathcal{P} \Leftrightarrow Synt(L)$ hat die Eigenschaft $\mathcal{P}'$.

Wenn $\mathcal{P}'$ entscheidbar ist, liefert dies die Entscheidbarkeit von $\mathcal{P}$.

Beispiel 5.3.1: Sei $L \subseteq \Sigma^\omega$ in MSO[<]-Logik definierbar. Dann sind äquivalent:

a) L ist in FO[<] definierbar

b) $Synt(L)$ ist aperiodisch, d.h. alle Unterhalbgruppen von $Synt(L)$, die Gruppen sind, haben nur ein Element („Untergruppen sind trivial", $Synt(L)$ ist „group-free").