Aufgabe 1

a) $L_1 = aab^*$ $\quad L_2 = ab^*a^*$ $\quad L_3 = \varepsilon$

$L_1 / L_2 = a + \varepsilon$ $\quad aabb^* \in L_1$, $abb^* \notin L_2$ $\quad aa \in L_1$, $aa \in L_2$

$L_1 / L_3 = L_1 = aab^*$ $\quad aab^*$, ε

$L_2 / L_1 = ab^*a^* + \varepsilon$ $\quad ab^*a^*aa \in L_2$, $aa \in L_1$ $\quad aa$ L_2, $aa \in L_1$

b) $L_1 = \{ w \in L \mid |w| \text{ gerade} \}$

$L_1 = \underbrace{L \cap \underbrace{(\underbrace{\Sigma\Sigma}_{\text{regulär}})^*}_{\text{regulär}}}_{\text{regulär}}$

effektiv, effektiv, effektiv

$L_2 = \{ v \in \Sigma^* \mid u \in \Sigma^{10},\ uv \in L \}$ $\quad$ d.h. löschen der ersten 10 Zeichen der Wörter aus L

Wir definieren $\Sigma' = \{ a' \mid a \in \Sigma \}$

$\Sigma' \cap \Sigma = \emptyset$

und $h: \Sigma \cup \Sigma' \to \Sigma$ mit $h(a) = a$ für $a \in \Sigma$

und $h(a') = a$ für $a' \in \Sigma'$

$L_2' = h^{-1}(L)$ $\quad w = a_1 a_2 \dots a_n \in L$

$\Downarrow$

$L_2'' = h^{-1}(L) \cap (\Sigma'^{10} \circ \Sigma^*)$ $\quad w' = a_1' a_2 \dots a_i' \dots a_j \dots a_n \in h^{-1}(L)$

Striche an den ersten 10 Positionen sonst keine $\quad$ Striche an beliebigen Positionen

diese werden durch g gelöscht

$L_2 = g(L_2'')$ mit $g(a) = a$ für $a \in \Sigma$

$g: \Sigma \cup \Sigma' \to \Sigma$ $\quad g(a') = \varepsilon$ für $a' \in \Sigma'$

Da reguläre Sprachen unter allen benutzten Operationen abgeschlossen ist, ist L_2 regulär.

Aufgabe 1 (Forts.)

c) Seien L_1 und L_2 regulär.

Wir definieren Σ' wie in Teilaufgabe b.
und h ebenso; $f: \Sigma \to \Sigma'$ mit $f(a) = a'$

$L' = h^{-1}(L_1) \cap \Sigma^* \circ f(L_2)$ $\quad w = \overbrace{a_1 \dots a_n \underbrace{a_{n+1}' \dots a_m'}_{\in f(L_2)}}^{\in h^{-1}(L)}$

$L'' = g(L')$ $\quad$ g wie in Teilaufgabe b.
(löscht Zeichen a')

Also gilt $L'' = L_1 / L_2$

Da reguläre Sprachen unter allen angewendeten Operationen effektiv abgeschlossen sind und L_1 und L_2 regulär sind, können wir $L'' = L_1 / L_2$ effektiv konstruieren.

Blatt 5 ③

Aufgabe 2

a) Entweder: Spezialfall von b.

Oder: ε ist in der von r bezeichneten Sprache, wenn r die LWE erfüllt. Die LWE wurde in der Vorlesung rein syntaktisch definiert (induktiv). Die LWE ist also syntaktisch entscheidbar.

b) " $x \in r$ ":

- Übersetze r in einen endlichen Automaten.
- Wandle den Automaten in einen deterministischen Automaten um.
- Vollziehe Schritt für Schritt die Übergänge $x = a_1 a_2 \dots a_n$.
- Wenn das Wort vollständig abgearbeitet werden kann und dann der Automat in einem Endzustand ist,

  akzeptiere Ja

  sonst akzeptiere nicht Nein

c) " $r = \emptyset$ ":

- Übersetze r in einen endlichen Automaten,
- Überprüfe, ob es wenigstens einen Pfad von einem Anfangszustand zu

Blatt 5 ④

Aufgabe 2 c. (Forts.)

einem Endzustand gibt.

Wenn ja: akzeptiere nicht Nein

Wenn nein: akzeptiere Ja

d) " $r = s$ ":

- Übersetze r und s in endliche Automaten A_r und A_s.
- Konstruiere die Automaten
  $A_r \cap \bar{A}_s \cup A_s \cap \bar{A}_r$
- Wenn kein Pfad von einem Anfangszustand zu einem Endzustand in diesem Automaten existiert, dann akzeptiere Ja

  sonst akzeptiere nicht. Nein

e) " $G = r$ ":

- Übersetze r in einen endlichen Automaten A_r
- Übersetze G in einen endlichen Automaten A_G
- Konstruiere $A_r \cap \bar{A}_G \cup A_G \cap \bar{A}_r$
- weiter wie in d)

Aufgabe 3

a)

$\oplus$	0	a	b	c	d
0	0	a	b	c	d
a	a	a	c	c	d
b	b	a	d	c	d
c	c	a	d	c	d
d	d	a	d	c	d

← (1) ✓

↑ (1) ✓

Um nachzuweisen, daß $M = (A, 0, \oplus)$ ein Monoid ist, müssen wir für jedes $x \in A$ zeigen $0 \oplus x = x \oplus 0 = x$ (1)

und für alle $x, y, z \in A$ zeigen $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (2)

ad (1): gilt offensichtlich für alle x

ad (2): Wir unterscheiden die folgenden Fälle:

$z = a$: $(x \oplus y) \oplus a = a \;=\; a = x \oplus \overbrace{(y \oplus a)}^{=a}$

analog zu a) { $z = c$: $(x \oplus y) \oplus c = c \;=\; c = x \oplus \overbrace{(y \oplus c)}^{=c}$

$z = d$: $(x \oplus y) \oplus d = d \;=\; d = x \oplus \overbrace{(y \oplus d)}^{=d}$ }

$z = b$:

$y = a$: $\overbrace{(x \oplus a)}^{=a} \oplus b = c \;=\; c = x \oplus \overbrace{(a \oplus b)}^{=c}$

$y = b$: $\overbrace{(x \oplus b)}^{=b/c} \oplus b = d \;=\; d = x \oplus \overbrace{(b \oplus b)}^{=d}$

$y = c$: $\overbrace{(x \oplus c)}^{=c} \oplus b = d \qquad d = x \oplus \overbrace{(c \oplus b)}^{=d}$

$y = d$: $\overbrace{(x \oplus d)}^{=d} \oplus b = d \qquad d = x \oplus \overbrace{(d \oplus b)}^{=d}$

$y = 0$: $(x \oplus 0) \oplus b = x \oplus b \qquad = x \oplus (0 \oplus b)$ ← (1)

$z = 0$: $(x \oplus y) \oplus 0 = x \oplus y = x \oplus (y \oplus 0)$ ← (1)

Aufgabe 3 (Forts.)

b) Sei w ein beliebiges Wort aus L, $\Sigma = \{a, b\}$

d.h. $w = xab$ für $x \in \Sigma^*$.

$h(w) = h(x) \oplus h(ab) = h(x) \oplus (h(a) \oplus h(b))$

$= h(x) \oplus (a \oplus b) = h(x) \oplus c = c$

Also $h(L) = \{c\} = A'$

$L' = h^{-1}(A') = \{ w \in \Sigma^* \mid h(w) = c \}$ $\qquad h(\varepsilon) = 0 \neq c$

$= \{ w \in \Sigma^* \mid w = a_1 a_2 \ldots a_n \qquad n \geq 1$

$\wedge\; h(a_1) \oplus h(a_2) \oplus \ldots \oplus h(a_n) = c \}$

$= \{ w \in \Sigma^* \mid w = a_1 a_2 \ldots a_n \;\wedge$

$h(a_n) = b \;\wedge$

$h(a_{n-1}) = a \}$ $\qquad h(\cancel{a}) = c$

$= (a+b)^* ab$ $\qquad h(\cancel{a}) = 0 \qquad \underline{h(a_n) = b}$

$h(a_1) \oplus \ldots \oplus \underbrace{h(a_{n-1})}_{=a} = a$

$\Rightarrow \quad L' = L$ $\qquad$ also $h^{-1}(h(L)) = L$

c) $h^{-1}(h(L)) = L$ gdw ein $A' \subseteq A$ ex. mit $h^{-1}(A') = L$

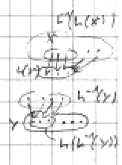
"⇒" wähle $A' = h(L)$ (vgl. Aufgabe b)

offensichtlich gilt $h^{-1}(A') = h^{-1}(h(L)) = L$

"⇐" sei nun $A' \subseteq A$ mit $h^{-1}(A') = L$ (*)

für jede Abbildung h gilt: $h^{-1}(h(X)) \supseteq X$

und $h(h^{-1}(Y)) \subseteq Y$



$L \overset{(*)}{=} h^{-1}(A') = \overbrace{h^{-1}(\underbrace{h(h^{-1}(A'))}_{\subseteq A'})}^{\supseteq h^{-1}(A')} \overset{(*)}{=} h^{-1}(h(L))$

$\subseteq h^{-1}(A')$

also $L = h^{-1}(h(L))$

Aufgabe 3 c. (Forts.)

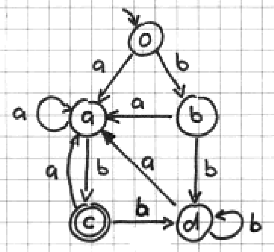
Bem: Eine Sprache L ist also genau dann erkennbar, wenn ein Monoid $M = (A, 0, \oplus)$, ein Homomorphismus $h: \Sigma^* \to M$ und eine Menge $A' \subseteq A$ existieren, so daß

A': "Endzustände"

$L = h^{-1}(A')$ ≙

$$w \in L \quad \text{gdw} \quad h(w) \in A'$$

d) Der Automat zum Monoid M könnte wie folgt aussehen:



Zustände: Elemente von A

Anfangszustand: 0

Endzustände: $A' = \{c\}$

Übergänge:

$q \in A$

$(q, a, q \oplus h(a))$

allgemein:

$q \xrightarrow{w} q'$ gdw $q \oplus h(w) = q'$

insbes.

$0 \xrightarrow{w} q'$ gdw $0 \oplus h(w) = q'$

gdw $h(w) = q'$

$$A_L = (A, \Sigma, \delta, 0, A')$$

$$\delta = \{ (q, a, q \oplus h(a)) \mid q \in A, a \in \Sigma \}$$

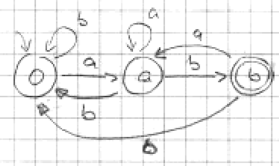
$w \in L$ gdw $0 \xrightarrow{w} q'$, $q' = h(w) \in A'$

e) Wir müssen aus dem Automaten für eine reguläre Sprache ein Monoid konstruieren!

Problem: Bei naiver Konstruktion entsteht kein Monoid

Aufgabe 3 e. (Forts.)

"Gegenbeispiel":



$\oplus$	0	a	b
0	0	a	0 ↯
a	a	a	b
b	b	a	0

- $0 \oplus b = 0 \neq b$ ↯

- $(b \oplus b) \oplus b = 0 \oplus b = 0$
  $\neq$ ↯
  $b \oplus (b \oplus b) = b \oplus 0 = b$

Abgesehen davon können wir bei den meisten Automaten gar nicht davon ausgehen, daß die Zustände "zum Alphabet passen"!

Wir gehen davon aus, daß unser Automat $A_L = (Q, \Sigma, \delta, I, F)$ buchstabierend ist. Wir wählen als Monoid:

$$M = (A, \delta^*_\varepsilon, \circ)$$

Relationskomposition

$$A = \{ \delta^*_w \mid w \in \Sigma^* \}$$

ist endlich, da $Q \times Q$ nur endlich viele Teilmengen besitzt

$$\text{wobei} \quad \delta^*_w = \{ (q, q') \mid (q, w, q') \in \delta^* \}$$

$\subseteq Q \times Q$

Für buchstabierende Automaten gilt:

- $\delta^*_\varepsilon = \{ (q, q) \mid q \in Q \} = id_Q$ gilt nicht für alphabetische Aut.
- $\delta^*_{uv} = \delta^*_u \circ \delta^*_v$

Komposition von Relationen

$R = R_1 \circ R_2 = \{ (q, r) \mid (q, p) \in R_1, (p, r) \in R_2 \}$

insbes. ist $\circ$ assoziativ und δ^*_ε ist neutrales Element!

Und es gilt:

$$w \in L \quad \text{gdw} \quad \delta^*_w \cap (I \times F) \neq \emptyset$$

Sei nun $h: \Sigma \to A$ mit $h(a) = \delta^*_a$

bzw. $h: \Sigma \to M$

Aufgabe 3 (Forts.)

$w = a_1 \dots a_n \quad h(w) = h(a_1) \circledast h(a_2) \circledast \dots \circledast h(a_n)$
$= \delta^*_{a_1} \circ \delta^*_{a_2} \circ \dots \circ \delta^*_{a_n} = \delta^*_{a_1 a_2 \dots a_n}$

Dann gilt $h(w) = \delta^*_w$

Wir wählen also $A' = \{ \delta^*_w \mid \delta^*_w \cap (I \times F) \neq \emptyset \}$

Dann gilt $w \in L$ gdw $h(w) \in A'$ bzw. $w \in h^{-1}(A')$

also $h^{-1}(A') = L$

Also wird L von M und h erkannt. (vgl. c.)

f.) $u \hat{R}_L v$ gdw für alle $w, w' \in \Sigma$: $wuw' \in L$ gdw $wvw' \in L$

$u \hat{R}_L v \overset{\text{insbes. } w=\varepsilon}{\Rightarrow}$ für alle $w' \in \Sigma$: $uw' \in L$ gdw $vw' \in L$

gdw $u R_L v$

Dies bedeutet insbesondere, daß der Index von R_L endlich ist, wenn der Index von $\hat{R}_L$ endlich ist.

Wir zeigen nun, daß umgekehrt der Index von $\hat{R}_L$ endlich ist, wenn der Index von R_L endlich ist:

Wenn der Index von R_L endlich ist, ist L regulär; gemäß c. und e. gibt es also ein endliches Monoid $M = (A, 0, \circledast)$, ein $A' \subseteq A$ und einen Homomorphismus $h: \Sigma^* \to M$ mit

$w \in L$ gdw $h(w) \in A'$

Wir beweisen nun, daß für alle Wörter $u, v \in \Sigma^*$ gilt $h(u) = h(v) \Rightarrow u \hat{R}_L v$.
Da das Monoid endlich ist hat $\hat{R}_L$ dann nur endlich viele Äquivalenzklassen.

Aufgabe 3 f, (Forts.)

Der Index von $\hat{R}_L$ ist also endlich.

Bleibt zu zeigen: $h(u) = h(v) \Rightarrow u \hat{R}_L v$

für alle $w, w' \in \Sigma$ gilt

$wuw' \in L$ gdw $h(wuw') \in A'$ gdw

$h(w) \circledast h(u) \circledast h(w') \in A'$
gdw
$h(w) \circledast h(v) \circledast h(w') \in A'$

$wvw' \in L$ gdw $h(wvw') \in A'$ gdw

also $u \hat{R}_L v$