Script **generated by TTT**

Title: Seidl: Theoretische_Informatik (30.04.2012)

Date: Mon Apr 30 10:16:34 CEST 2012

Duration: 89:30 min

Pages: 103

Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für R.

Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für R.

Die logische Struktur des Satzes:

$\forall$*reg.* R.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = a^n b^n \in L$.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = a^n b^n \in L$.
Dann ist z zerlegbar in uvw mit
$u, v \in \{a\}^*$ (weil $|uv| \leq n$) und $v \neq \epsilon$.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = a^n b^n \in L$.
Dann ist z zerlegbar in uvw mit
$u, v \in \{a\}^*$ (weil $|uv| \leq n$) und $v \neq \epsilon$.
Damit müsste gelten $a^{n-|v|} b^n = uw \in L$. ↯ □

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.31
Die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = a^n b^n \in L$.
Dann ist z zerlegbar in uvw mit
$u, v \in \{a\}^*$ (weil $|uv| \leq n$) und $v \neq \epsilon$.
Damit müsste gelten $a^{n-|v|} b^n = uw \in L$. ↯ □

Endliche Automaten können nicht unbegrenzt zählen

Ist die Sprache $\{a^n b^n \mid n \leq 10^6\}$ regulär?

Ist die Sprache $\{a^n b^n \mid n \leq 10^6\}$ regulär?

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.32
$L = \{0^{m^2} \mid m \geq 0\}$ *ist nicht regulär.*

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.32
$L = \{0^{m^2} \mid m \geq 0\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = 0^{n^2} \in L$. Dann ist z zerlegbar in uvw mit

$$1 \leq |v| \leq |uv| \leq n$$

und $uv^l w \in L$ für alle $l \in \mathbb{N}$.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.32
$L = \{0^{m^2} \mid m \geq 0\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = 0^{n^2} \in L$. Dann ist z zerlegbar in uvw mit

$$1 \leq |v| \leq |uv| \leq n$$

und $uv^l w \in L$ für alle $l \in \mathbb{N}$. D.h. insb. $|uv^2w|$ ist Quadratzahl.

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.32
$L = \{0^{m^2} \mid m \geq 0\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = 0^{n^2} \in L$. Dann ist z zerlegbar in uvw mit

$$1 \leq |v| \leq |uv| \leq n$$

und $uv^l w \in L$ für alle $l \in \mathbb{N}$. D.h. insb. $|uv^2 w|$ ist Quadratzahl.
Dies führt zu einem Widerspruch:

$$n^2 = |z| = |uvw| < |uv^2 w|$$

Beispiel für die Anwendung des Pumping Lemmas:

Satz 2.32
$L = \{0^{m^2} \mid m \geq 0\}$ *ist nicht regulär.*

Beweis:
Angenommen, L sei doch regulär.
Sei n eine Pumping-Lemma-Zahl für L.
Wähle $z = 0^{n^2} \in L$. Dann ist z zerlegbar in uvw mit

$$1 \leq |v| \leq |uv| \leq n$$

und $uv^l w \in L$ für alle $l \in \mathbb{N}$. D.h. insb. $|uv^2 w|$ ist Quadratzahl.
Dies führt zu einem Widerspruch:

$$n^2 = |z| = |uvw| < |uv^2 w| \leq n^2 + n < n^2 + 2n + 1 = (n+1)^2$$

Übungsaufgabe:

$\{1^p \mid p \text{ prim}\}$ ist nicht regulär.

Pumping-Lemma-Zahl für $L(ab^*c)$?

Pumping-Lemma-Zahl für $L(ab^*c)$?

Pumping-Lemma-Zahl für $\{aaaaaa\}$?

Erinnerung:

n ist Pumping-Lemma-Zahl für L
gdw
alle $z \in L$ mit $|z| \geq n$ lassen sich aufpumpen.

Bemerkung
Es gibt nicht-reguläre Sprachen, für die das Pumping-Lemma gilt!

Bemerkung
Es gibt nicht-reguläre Sprachen, für die das Pumping-Lemma gilt!

$\Rightarrow$ Pumpin Lemma hinreichend aber nicht notwendig um Nicht-Regularität zu zeigen.

Bemerkung
Es gibt nicht-reguläre Sprachen, für die das Pumping-Lemma gilt!

$\Rightarrow$ Pumpin Lemma hinreichend aber nicht notwendig um Nicht-Regularität zu zeigen.

regulär $\subset$ Pumping-Lemma gilt $\subset$ alle Sprachen

2.9 Entscheidbarkeit

- Welche Probleme sind für reguläre Sprachen entscheidbar?

2.9 Entscheidbarkeit

- Welche Probleme sind für reguläre Sprachen entscheidbar?
Sehr viele!

2.9 Entscheidbarkeit

- Welche Probleme sind für reguläre Sprachen entscheidbar?
  Sehr viele!
- Wie hängt die Komplexität mit der Repräsentation zusammen: DFA , NFA und RE?

Statt Mengen verwendet man oft Eigenschaften oder Probleme.
Bsp: "ist prim" statt "ist Element der Primzahlen".

Statt Mengen verwendet man oft Eigenschaften oder Probleme.
Bsp: "ist prim" statt "ist Element der Primzahlen".

Eine Eigenschaft nennt man entscheidbar gdw die zugehörige Menge entscheidbar ist.
Bsp:

„Es ist entscheidbar, ob eine Zahl prim ist"

$$\equiv$$

„Die Menge der Primzahlen ist entscheidbar"

Definition 2.33

Sei D eine Beschreibung einer Sprache (DFA, NFA, RE, ~~Grammatik,~~ etc).

Wortproblem: Gegeben w, gilt $w \in L(D)$?

Definition 2.33

Sei D eine Beschreibung einer Sprache (DFA, NFA, RE, Grammatik, etc).

Wortproblem: Gegeben w, gilt $w \in L(D)$?

Leerheitsproblem: Gilt $L(D) = \emptyset$?

Definition 2.33

Sei D eine Beschreibung einer Sprache (DFA, NFA, RE, Grammatik, etc).

Wortproblem: Gegeben w, gilt $w \in L(D)$?

Leerheitsproblem: Gilt $L(D) = \emptyset$?

Endlichkeitsproblem: Ist $L(D)$ endlich?

uniformes Kostenmaß

Das Wortproblem für DFAs ist in linearer Zeit entscheidbar:

Das Wortproblem für DFAs ist in linearer Zeit entscheidbar:

Das Wortproblem für DFAs ist in linearer Zeit entscheidbar:

Fakt 2.34
Sei M ein DFA. Das Problem $w \in L(M)$ ist in Zeit $O(|w|)$ entscheidbar.

Lemma 2.35
Jede reguläre Sprache ist in linearer Zeit entscheidbar.

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Jetzt: Wort *und* Automat Eingabe.

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Jetzt: Wort *und* Automat Eingabe.

Lemma 2.36
Das Problem $w \in L(N)$ ist für beliebiges Wort w und NFA N in Zeit $O(|Q|^2|w|)$ entscheidbar.

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Jetzt: Wort *und* Automat Eingabe.

Lemma 2.36
Das Problem $w \in L(N)$ ist für beliebiges Wort w und NFA N in Zeit $O(|Q|^2|w|)$ entscheidbar.

Beweis:
Sei $Q = \{1, \dots, s\}$, $q_0 = 1$ und $w = a_1 \dots a_n$.

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Jetzt: Wort *und* Automat Eingabe.

Lemma 2.36
Das Problem $w \in L(N)$ ist für beliebiges Wort w und NFA N in Zeit $O(|Q|^2|w|)$ entscheidbar.

Beweis:
Sei $Q = \{1, \dots, s\}$, $q_0 = 1$ und $w = a_1 \dots a_n$.

$S := \{1\}$
for $i := 1$ **to** n **do** $S := \bigcup_{j \in S} \delta(j, a_i)$

Bisher Automat bzw Sprache fix. Die Eingabe besteht nur aus w.

Jetzt: Wort *und* Automat Eingabe.

Lemma 2.36
Das Problem $w \in L(N)$ ist für beliebiges Wort w und NFA N in Zeit $O(|Q|^2|w|)$ entscheidbar.

Beweis:
Sei $Q = \{1, \ldots, s\}$, $q_0 = 1$ und $w = a_1 \ldots a_n$.

$S := \{1\}$
for $i := 1$ **to** n **do** $S := \bigcup_{j \in S} \delta(j, a_i)$
return $(S \cap F \neq \emptyset)$

□

Lemma 2.37
Das Leerheitsproblem ist für NFAs und DFAs entscheidbar

Lemma 2.37
Das Leerheitsproblem ist für NFAs und DFAs entscheidbar (in Zeit $O(|Q|^2|\Sigma|)$ bzw $O(|Q||\Sigma|)$).

Lemma 2.37
Das Leerheitsproblem ist für NFAs und DFAs entscheidbar (in Zeit $O(|Q|^2|\Sigma|)$ bzw $O(|Q||\Sigma|)$).

Beweis:
$L(M) = \emptyset$ gdw kein Endzustand von q_0 erreichbar ist.

Lemma 2.37

Das Leerheitsproblem ist für NFAs und DFAs entscheidbar (in Zeit $O(|Q|^2|\Sigma|)$ bzw $O(|Q||\Sigma|)$).

Beweis:

$L(M) = \emptyset$ gdw kein Endzustand von q_0 erreichbar ist.
Dies ist eine einfache Suche in einem Graphen, die jede Kante maximal ein Mal benutzen muss.

Lemma 2.37

Das Leerheitsproblem ist für NFAs und DFAs entscheidbar (in Zeit $O(|Q|^2|\Sigma|)$ bzw $O(|Q||\Sigma|)$).

Beweis:

$L(M) = \emptyset$ gdw kein Endzustand von q_0 erreichbar ist.
Dies ist eine einfache Suche in einem Graphen, die jede Kante maximal ein Mal benutzen muss.
Ein NFA hat $\leq |Q|^2|\Sigma|$ Kanten, ein DFA hat $\leq |Q||\Sigma|$ Kanten. □

Ist Σ fix, z.B. ASCII, so wird daraus $O(|Q|^2)$ bzw $O(|Q|)$.

Global: $\delta : Q \times \Sigma \to \mathcal{P}(Q)$

$Reach(K) =$

Global: $\delta : Q \times \Sigma \to \mathcal{P}(Q)$

$Reach(K) =$
$R := \emptyset$
$W := K$

Global: $\delta : Q \times \Sigma \to \mathcal{P}(Q)$

$Reach(K) =$
    $R := \emptyset$
    $W := K$
    while $W \neq \emptyset$ **do**
        pick and remove some $p \in W$
        if $p \notin R$ **then**
            $R := R \cup \{p\}$
            $W := W \cup \bigcup_{a\in\Sigma} \delta(p,a)$

Global: $\delta : Q \times \Sigma \to \mathcal{P}(Q)$

$Reach(K) =$
    $R := \emptyset$
    $W := K$
    while $W \neq \emptyset$ **do**
        pick and remove some $p \in W$
        if $p \notin R$ **then**
            $R := R \cup \{p\}$
            $W := W \cup \bigcup_{a\in\Sigma} \delta(p,a)$
    return R

Lemma 2.38
Für endliche Automaten ist das Endlichkeitsproblem entscheidbar.

Lemma 2.38
Für endliche Automaten ist das Endlichkeitsproblem entscheidbar.

Beweis:
$|L(M)| = \infty$ gdw von q_0 aus eine nicht-leere Schleife erreichbar ist, von der aus F erreichbar ist:

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

Lemma 2.38

Für endliche Automaten ist das Endlichkeitsproblem entscheidbar.

Beweis:

$|L(M)| = \infty$ gdw von q_0 aus eine nicht-leere Schleife erreichbar ist, von der aus F erreichbar ist:

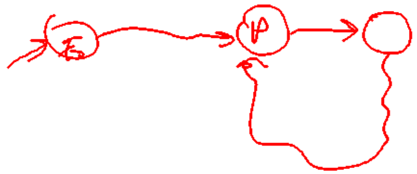
$finite(Q, \Sigma, \delta, q_0, F) =$
$\quad R := Reach(\{q_0\})$

Lemma 2.38

Für endliche Automaten ist das Endlichkeitsproblem entscheidbar.

Beweis:

$|L(M)| = \infty$ gdw von q_0 aus eine nicht-leere Schleife erreichbar ist, von der aus F erreichbar ist:



$finite(Q, \Sigma, \delta, q_0, F) =$
$\quad R := Reach(\{q_0\})$
$\quad C := \{p \in R \mid p \in Reach(\bigcup_{a \in \Sigma} \delta(p, a))\}$

Lemma 2.38

Für endliche Automaten ist das Endlichkeitsproblem entscheidbar.

Beweis:

$|L(M)| = \infty$ gdw von q_0 aus eine nicht-leere Schleife erreichbar ist, von der aus F erreichbar ist:

$finite(Q, \Sigma, \delta, q_0, F) =$
$\quad R := Reach(\{q_0\})$
$\quad C := \{p \in R \mid p \in Reach(\bigcup_{a \in \Sigma} \delta(p, a))\}$
$\quad$ **return** $(Reach(C) \cap F = \emptyset)$

$\square$

Definition 2.39

Seien D_1 und D_2 Sprachbeschreibungen (DFAs, NFAs, REs, Grammatiken, etc).

Definition 2.39
Seien D_1 und D_2 Sprachbeschreibungen (DFAs, NFAs, REs, Grammatiken, etc).

Äquivalenzproblem: Gilt $L(D_1) = L(D_2)$?

Definition 2.39
Seien D_1 und D_2 Sprachbeschreibungen (DFAs, NFAs, REs, Grammatiken, etc).

Äquivalenzproblem: Gilt $L(D_1) = L(D_2)$?

Lemma 2.40
Das Äquivalenzproblem ist für DFAs entscheidbar.

Beweis:
Folgt direkt aus

$$L_1 \subseteq L_2 \quad \Leftrightarrow \quad L_1 \cap \overline{L_2} = \emptyset$$

Definition 2.39
Seien D_1 und D_2 Sprachbeschreibungen (DFAs, NFAs, REs, Grammatiken, etc).

Äquivalenzproblem: Gilt $L(D_1) = L(D_2)$?

Lemma 2.40
Das Äquivalenzproblem ist für DFAs entscheidbar.

Beweis:
Folgt direkt aus

$$\begin{aligned} L_1 \subseteq L_2 \quad &\Leftrightarrow \quad L_1 \cap \overline{L_2} = \emptyset \\ L_1 = L_2 \quad &\Leftrightarrow \quad L_1 \subseteq L_2 \wedge L_2 \subseteq L_1 \end{aligned}$$

Satz 2.41
Das Äquivalenzproblem für DFAs ist in Zeit $O(|Q_1||Q_2|)$ entscheidbar

Satz 2.41

Das Äquivalenzproblem für DFAs ist in Zeit $O(|Q_1||Q_2|)$ entscheidbar (bei fixem Σ).

Beweis:

Gegeben: DFAs M_1 mit m und M_2 mit n Zuständen.
Mit Hilfe der Produkt-Konstruktion für $\cap$ folgt:

	Anzahl der Zustände
$L(M_1) \cap \overline{L(M_2)}$	mn
$\overline{L(M_1)} \cap L(M_2)$	mn

Das Leerheitsproblem ist jeweils in Zeit $O(mn)$ entscheidbar. □

Korollar 2.42

Das Äquivalenzproblem für NFAs ist in Zeit $O(2^{|Q_1|+|Q_2|})$ entscheidbar (bei fixem Σ).

Korollar 2.42

Das Äquivalenzproblem für NFAs ist in Zeit $O(2^{|Q_1|+|Q_2|})$ entscheidbar (bei fixem Σ).

Beweis: 2 NFAs mit m und n Zuständen $\rightsquigarrow$
2 DFAs mit 2^m und 2^n Zuständen $\rightsquigarrow$
Äquivalenztest in Zeit $O(2^m 2^n)$

Korollar 2.42

Das Äquivalenzproblem für NFAs ist in Zeit $O(2^{|Q_1|+|Q_2|})$ entscheidbar (bei fixem Σ).

Beweis: 2 NFAs mit m und n Zuständen $\rightsquigarrow$
2 DFAs mit 2^m und 2^n Zuständen $\rightsquigarrow$
Äquivalenztest in Zeit $O(2^m 2^n)$

Korollar 2.43

Das Äquivalenzproblem für reguläre Ausdrücke ist entscheidbar.

Fazit:

Die Kodierung der Eingabe (DFA, NFA, RE, ...) kann entscheidend für die Komplexität eines Problems sein.

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

Es gilt z.B.

$$(0^* \mid 1^*)^* \quad \equiv \quad (0 \mid 1)^*$$

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

Es gilt z.B.

$$(0^* \mid 1^*)^* \quad \equiv \quad (0 \mid 1)^*$$

Beweis z.B. über Automaten und Leerheit (s.o.) — automatisch!

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

Es gilt z.B.

$$(0^* \mid 1^*)^* \quad \equiv \quad (0 \mid 1)^*$$

Beweis z.B. über Automaten und Leerheit (s.o.) — automatisch!

Aber wie zeig man, dass

$$(\alpha^* \mid \beta^*)^* \quad \equiv \quad (\alpha \mid \beta)^*$$

für beliebige reguläre Ausdrücke α und β gilt?

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

Es gilt z.B.

$$(0^* \mid 1^*)^* \quad \equiv \quad (0 \mid 1)^*$$

Beweis z.B. über Automaten und Leerheit (s.o.) — automatisch!

Aber wie zeig man, dass

$$(\alpha^* \mid \beta^*)^* \quad \equiv \quad (\alpha \mid \beta)^*$$

für beliebige reguläre Ausdrücke α und β gilt?
Beweis per Hand mit Hilfe von

- $L(.)$ und Mengenlehre
- oder mit bereits bewiesenen Gesetzen (siehe §7).

Wir werden zeigen:

> Das Äquivalenzproblem für reguläre Ausdrücke *mit* Variablen ist reduzierbar auf das Äquivalenzproblem für reguläre Ausdrücke *ohne* Variablen.

Wir werden zeigen:

> Das Äquivalenzproblem für reguläre Ausdrücke *mit* Variablen ist reduzierbar auf das Äquivalenzproblem für reguläre Ausdrücke *ohne* Variablen.

Methode:

> Ersetze Variablen durch Konstanten und entscheide das Äquivalenzproblem.

2.10 Äquivalenz regulärer Ausdrücke

Erinnerung: $\alpha \equiv \beta$ bedeutet $L(\alpha) = L(\beta)$.

Es gilt z.B.

$$(0^* \mid 1^*)^* \quad \equiv \quad (0 \mid 1)^*$$

Beweis z.B. über Automaten und Leerheit (s.o.) — automatisch!

Aber wie zeig man, dass

$$(\alpha^* \mid \beta^*)^* \quad \equiv \quad (\alpha \mid \beta)^*$$

für beliebige reguläre Ausdrücke α und β gilt?
Beweis per Hand mit Hilfe von

- $L(.)$ und Mengenlehre
- oder mit bereits bewiesenen Gesetzen (siehe §7).

Automatisierbar?

Wir werden zeigen:

> Das Äquivalenzproblem für reguläre Ausdrücke *mit* Variablen ist reduzierbar auf das Äquivalenzproblem für reguläre Ausdrücke *ohne* Variablen.

Methode:

> Ersetze Variablen durch Konstanten und entscheide das Äquivalenzproblem.

Beispiel:

> $\alpha\alpha^* \equiv \alpha^*\alpha$ gilt für alle regulären Ausrücke α weil $11^* \equiv 1^*1$ gilt.

Funktioniert leider bei Zahlen nicht:

$$1 + 2 = 3 \quad \not\Rightarrow \quad x_1 + x_2 = x_3$$

Funktioniert leider bei Zahlen nicht:

$$1 + 2 = 3 \quad \not\Rightarrow \quad x_1 + x_2 = x_3$$

Funktioniert auch nicht auf regulären Ausdrücken mit *Durchschnitt*: Neuer Operator $\sqcap$ für reguläre Ausdrücke mit

$$L(\alpha \sqcap \beta) = L(\alpha) \cap L(\beta)$$

Funktioniert leider bei Zahlen nicht:

$$1 + 2 = 3 \quad \not\Rightarrow \quad x_1 + x_2 = x_3$$

Funktioniert auch nicht auf regulären Ausdrücken mit *Durchschnitt*: Neuer Operator $\sqcap$ für reguläre Ausdrücke mit

$$L(\alpha \sqcap \beta) = L(\alpha) \cap L(\beta)$$

Auch hier gibt es ein Problem:

$$a \sqcap b \equiv \emptyset \quad \not\Rightarrow \quad \alpha \sqcap \beta \equiv \emptyset$$

Funktioniert leider bei Zahlen nicht:

$$1 + 2 = 3 \quad \not\Rightarrow \quad x_1 + x_2 = x_3$$

Funktioniert auch nicht auf regulären Ausdrücken mit *Durchschnitt*: Neuer Operator $\sqcap$ für reguläre Ausdrücke mit

$$L(\alpha \sqcap \beta) = L(\alpha) \cap L(\beta)$$

Auch hier gibt es ein Problem:

$$a \sqcap b \equiv \emptyset \quad \not\Rightarrow \quad \alpha \sqcap \beta \equiv \emptyset$$

Setze $\alpha = \beta = a$: $a \sqcap a \equiv a \not\equiv \emptyset$

Aber für „normale“ REs funktioniert es!

Funktioniert leider bei Zahlen nicht:

$$1 + 2 = 3 \quad \not\Rightarrow \quad x_1 + x_2 = x_3$$

Funktioniert auch nicht auf regulären Ausdrücken mit *Durchschnitt*: Neuer Operator $\sqcap$ für reguläre Ausdrücke mit

$$L(\alpha \sqcap \beta) = L(\alpha) \cap L(\beta)$$

Auch hier gibt es ein Problem:

$$a \sqcap b \equiv \emptyset \quad \not\Rightarrow \quad \alpha \sqcap \beta \equiv \emptyset$$

Setze $\alpha = \beta = a$: $a \sqcap a \equiv a \not\equiv \emptyset$

Aber für „normale“ REs funktioniert es!

Notwendig: Präzisierung von Aussagen wie

$$\alpha \mid \beta \equiv \beta \mid \alpha \quad \text{für alle REs } \alpha, \beta$$

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Definition 2.44

Eine Substitution σ ist eine Funktion von V nach REs.

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Definition 2.44

Eine Substitution σ ist eine Funktion von V nach REs.
Substitutionen können auf Wörter und REs mit Variablen angwandt werden und ersetzen dort $X_1, X_2, \dots$ durch $\sigma(X_1), \sigma(X_2), \dots$.

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Definition 2.44

Eine Substitution σ ist eine Funktion von V nach REs.
Substitutionen können auf Wörter und REs mit Variablen angwandt werden und ersetzen dort $X_1, X_2, \dots$ durch $\sigma(X_1), \sigma(X_2), \dots$.

Beispiel 2.45

Für $\sigma = \{X_1 \mapsto a^*|b,\ X_3 \mapsto c\}$ gilt $\sigma(X_3 d X_1) = cd(a^*|b)$.

c $\quad$ a^*b

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Definition 2.44

Eine Substitution σ ist eine Funktion von V nach REs.
Substitutionen können auf Wörter und REs mit Variablen angwandt werden und ersetzen dort $X_1, X_2, \dots$ durch $\sigma(X_1), \sigma(X_2), \dots$.

Beispiel 2.45

Für $\sigma = \{X_1 \mapsto a^*|b,\ X_3 \mapsto c\}$ gilt $\sigma(X_3 d X_1) = cd(a^*|b)$.

Im folgenden ist E ein RE über Σ, der auch Variablen enthalten darf.

Sei Σ ein Alphabet und $V = \{X_1, \dots\}$ eine davon disjunkte Menge von Variablen.

Definition 2.44
Eine Substitution σ ist eine Funktion von V nach REs. Substitutionen können auf Wörter und REs mit Variablen angwandt werden und ersetzen dort $X_1, X_2, \dots$ durch $\sigma(X_1), \sigma(X_2), \dots$.

Beispiel 2.45
Für $\sigma = \{X_1 \mapsto a^*|b,\ X_3 \mapsto c\}$ gilt $\sigma(X_3 d X_1) = cd(a^*|b)$.

Im folgenden ist E ein RE über Σ, der auch Variablen enthalten darf.

Beim Äquivalenztest ersetzen wir nicht Variablen durch Konstanten (X durch a) sondern betrachten Variablen als Konstanten, d.h. E als RE über $\Sigma_V := \Sigma \cup V$.

Ziel:

$$E_1 \equiv E_2 \implies \sigma(E_1) \equiv \sigma(E_2)$$

Ziel:

$$E_1 \equiv E_2 \implies \sigma(E_1) \equiv \sigma(E_2)$$

Beweisidee:
Mit $L(E_1) = L(E_2)$ zeige

$$L(\sigma(E_1)) \qquad = \qquad L(\sigma(E_2))$$

Ziel:

$$E_1 \equiv E_2 \implies \sigma(E_1) \equiv \sigma(E_2)$$

Beweisidee:
Mit $L(E_1) = L(E_2)$ zeige

$$L(\sigma(E_1)) = \sigma(L(E_1)) = \sigma(L(E_2)) = L(\sigma(E_2))$$

Ziel:

$$E_1 \equiv E_2 \implies \sigma(E_1) \equiv \sigma(E_2)$$

Beweisidee:
Mit $L(E_1) = L(E_2)$ zeige

$$L(\sigma(E_1)) = \boxed{\sigma}(L(E_1)) = \sigma(L(E_2)) = L(\sigma(E_2))$$

Noch nicht ganz:

$$L(\sigma(X_1X_2)) = L(\alpha_1\alpha_2)$$

NB: $L(E) \subseteq \Sigma_V^*$.

NB: $L(E) \subseteq \Sigma_V^*$.

Definition 2.46
Die Erweiterung von σ auf $A \subseteq \Sigma_V^*$ ist elementweise definiert:

$$\sigma(A) := \{\sigma(w) \mid w \in A\}$$

NB: $L(E) \subseteq \Sigma_V^*$.

Definition 2.46
Die Erweiterung von σ auf $A \subseteq \Sigma_V^*$ ist elementweise definiert:

$$\sigma(A) := \{\sigma(w) \mid w \in A\}$$

NB: $\sigma(w)$ ist wieder ein RE.
Die Erweiterung von L auf Mengen von REs ist definiert durch:

$$L[M] := \bigcup_{\alpha \in M} L(\alpha)$$

NB: $L(E) \subseteq \Sigma_V^*$.

Definition 2.46
Die Erweiterung von σ auf $A \subseteq \Sigma_V^*$ ist elementweise definiert:

$$\sigma(A) := \{\sigma(w) \mid w \in A\}$$

NB: $\sigma(w)$ ist wieder ein RE.
Die Erweiterung von L auf Mengen von REs ist definiert durch:

$$L[M] := \bigcup_{\alpha \in M} L(\alpha)$$

Beispiel 2.47
Sei $\sigma = \{X_1 \mapsto a, X_2 \mapsto b|c\}$.

$$\sigma(\{X_1X_1, X_1X_2d\}) = \{aa, a(b|c)d\}$$

Lemma 2.48
Seien M, M_1, M_2 Mengen von REs.

- $L[M_1M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

NB: $L(E) \subseteq \Sigma_V^*$.

Definition 2.46
Die Erweiterung von σ auf $A \subseteq \Sigma_V^*$ ist elementweise definiert:

$$\sigma(A) := \{\sigma(w) \mid w \in A\}$$

NB: $\sigma(w)$ ist wieder ein RE.
Die Erweiterung von L auf Mengen von REs ist definiert durch:

$$L[M] := \bigcup_{\alpha \in M} L(\alpha)$$

Beispiel 2.47
Sei $\sigma = \{X_1 \mapsto a, X_2 \mapsto b|c\}$.

$$\sigma(\{X_1X_1, X_1X_2d\}) = \{aa, a(b|c)d\}$$
$$L[\{aa, a(b|c)d\}] = \{aa\} \cup \{abd, acd\}$$

Lemma 2.48
Seien M, M_1, M_2 Mengen von REs.

- $L[M_1M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$L[M_1M_2] \quad = \quad \bigcup_{\alpha \in M_1M_2} L(\alpha)$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$
\begin{aligned}
L[M_1 M_2] &= \bigcup_{\alpha \in M_1 M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1 \alpha_2)
\end{aligned}
$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$
\begin{aligned}
L[M_1 M_2] &= \bigcup_{\alpha \in M_1 M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1 \alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1) L(\alpha_2)
\end{aligned}
$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$
\begin{aligned}
L[M_1 M_2] &= \bigcup_{\alpha \in M_1 M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1 \alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1) L(\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} \{w_1 w_2 \mid w_1 \in L(\alpha_1) \wedge w_2 \in L(\alpha_2)\}
\end{aligned}
$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$
\begin{aligned}
L[M_1 M_2] &= \bigcup_{\alpha \in M_1 M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1 \alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1) L(\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} \{w_1 w_2 \mid w_1 \in L(\alpha_1) \wedge w_2 \in L(\alpha_2)\} \\
&= \{w_1 w_2 \mid w_1 \in \bigcup_{\alpha_1 \in M_1} L(\alpha_1) \wedge w_2 \in \bigcup_{\alpha_2 \in M_2} L(\alpha_2)\}
\end{aligned}
$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$\begin{aligned}
L[M_1M_2] &= \bigcup_{\alpha \in M_1M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1)L(\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} \{w_1w_2 \mid w_1 \in L(\alpha_1) \wedge w_2 \in L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in \bigcup_{\alpha_1 \in M_1} L(\alpha_1) \wedge w_2 \in \bigcup_{\alpha_2 \in M_2} L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in L[M_1] \wedge w_2 \in L[M_2]\}
\end{aligned}$$

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$\begin{aligned}
L[M_1M_2] &= \bigcup_{\alpha \in M_1M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1)L(\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} \{w_1w_2 \mid w_1 \in L(\alpha_1) \wedge w_2 \in L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in \bigcup_{\alpha_1 \in M_1} L(\alpha_1) \wedge w_2 \in \bigcup_{\alpha_2 \in M_2} L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in L[M_1] \wedge w_2 \in L[M_2]\} \\
&= L[M_1]L[M_2]
\end{aligned}$$

□

Lemma 2.48

Seien M, M_1, M_2 Mengen von REs.

- $L[M_1 M_2] = L[M_1]L[M_2]$
- $L[M_1 \cup M_2] = L[M_1] \cup L[M_2]$
- $L[M^*] = (L[M])^*$

Beweis:

$$\begin{aligned}
L[M_1M_2] &= \bigcup_{\alpha \in M_1M_2} L(\alpha) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} L(\alpha_1)L(\alpha_2) \\
&= \bigcup_{\alpha_1 \in M_1, \alpha_2 \in M_2} \{w_1w_2 \mid w_1 \in L(\alpha_1) \wedge w_2 \in L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in \bigcup_{\alpha_1 \in M_1} L(\alpha_1) \wedge w_2 \in \bigcup_{\alpha_2 \in M_2} L(\alpha_2)\} \\
&= \{w_1w_2 \mid w_1 \in L[M_1] \wedge w_2 \in L[M_2]\} \\
&= L[M_1]L[M_2]
\end{aligned}$$

□

Gilt auch $L[M_1 \cap M_2] = L[M_1] \cap L[M_2]$?

Satz 2.49 (Substitutionslemma)

$$L[\sigma(L(E))] = L(\sigma(E))$$