# GTI Übungsblatt 1
# Tutor: Marko Schmellenkamp
# ID: MS1
# Übung: Mi 16-18

Max Springenberg, 177792

## 1.1

### 1.1.1 Seien $\beta = (ab)^*$ und $\alpha_1, ..., \alpha_8$ die folgenden erweiterten regulären Ausdrücke. Beurteilen Sie für alle $i \in \{3, ..., 8\}$, ob $L(\alpha_i) \subseteq L(\beta)$ gilt. Vervollständigen Sie dazu die folgende Tabelle analog zu den Beispiel-Ausdrücken $\alpha_1, \alpha_2$: Falls $L(\alpha_i \nsubseteq L(\beta)$ gilt, geben Sie ein Wort $w_i \in L(\alpha_i) - L(\beta)$ an.

| RE | $L(\alpha_i) \subseteq L(\beta)$ | Gegenbeispiel |
|---|---|---|
| $\alpha_1 = (ab)^*(ab)^*$ | ✓ | |
| $\alpha_2 = (ba)^*$ | ✗ | ba |
| $\alpha_3 = (a^*b^*)$ | ✗ | aa |
| $\alpha_4 = (b?a?)^*$ | ✗ | aa |
| $\alpha_5 = (a+\epsilon)(b+\epsilon)(ab)^*$ | ✗ | aab |
| $\alpha_6 = (ab^*ab^*)^*$ | ✗ | aab |
| $\alpha_7 = (abab)^*$ | ✓ | |
| $\alpha_8 = a(ba)^+b$ | ✓ | |

### 1.1.2 Das Verhalten eines Netzwerk-Controllers soll anhand protokollierter Ausgaben analysiert werden. Der Controller schreibt, abhängig von der Eingabe, beliebig lange Bitfolgen. Eine Bitfolge wird als Wort über dem Alphabet $\{0, 1\}$ repräsentiert. Die Menge der gültigen Ausgaben wird im Handbuch des Controllers formal als Sprache $L = \{0, 1\}^* - L(\gamma)$ für den erweiterten regulären Ausdruck

$$\gamma = (0+1)^*(000+111)(0+1)^* + (10)*1? + (01)^*0?$$

**spezifiziert. Beschreiben Sie L natürlichsprachlich kurz in einem Satz.**

**FEHLER:**

In $\gamma$ enthaltene Wörter haben mindestens eine der folgenden Eigenschaften:

- Das Wort enthält drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort beginnt mit 10 gefolgt von einer 1 oder $\epsilon$ und endet
- Das Wort beginnt mit 01 gefolgt von einer 0 oder $\epsilon$ und endet

, L enthaelt nur Wörter über $\{0, 1\}$, fuer die das nicht der Fall ist, also Wörter, für die alle der folgenden Eigenschaften gelten:

- Das Wort enthält keine drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort beginnt nicht mit 10 gefolgt von einer 1 oder $\epsilon$ und endet
- Das Wort beginnt nicht mit 01 gefolgt von einer 0 oder $\epsilon$ und endet

**KORREKTUR:**

In $\gamma$ enthaltene Wörter haben mindestens eine der folgenden Eigenschaften:

- Das Wort enthält drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort besteht aus eine beliebigen Folge von 10 endend auf 1 oder $\epsilon$
- Das Wort besteht aus eine beliebigen Folge von 01 endend auf 0 oder $\epsilon$

, L enthaelt nur Wörter über $\{0, 1\}$, fuer die das nicht der Fall ist, also Wörter, für die alle der folgenden Eigenschaften gelten:

- Das Wort enthält keine drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort besteht nicht aus eine beliebigen Folge von 10 endend auf 1 oder $\epsilon$
- Das Wort besteht nicht aus eine beliebigen Folge von 01 endend auf 0 oder $\epsilon$

## 1.2 Geben Sie im Folgenden reguläre Ausdrücke bzw. erweiterte reguläre Ausdrücke an. Beschreiben Sie für jede Konstruktion kurz, warum Ihr Ausdruck die Sprache beschreibt (warum er alle Wörter der Sprache erzeugt und warum er kein Wort auerhalb der Sprache erzeugt).

### 1.2.1 Sei $\Sigma = \{0, \ldots, 9, \oplus, \ominus, ., \circ, C\}$ .

**Konstruieren Sie einen erweiterten regulären Ausdruck $\alpha$ über $\Sigma$, der genau die gültigen Temparaturangaben mit zwei nachkommastellen in Grad Celsius beschreibt.**

**Eine Solche Temperaturangabe ist gültig, wenn sie :**

- **genau zwei Nachkommastellen besitzt**
- **keine überfluessigen führenden Nullen im ganzzahligen Anteil ausfweisst**
- **den Minimalwert von $-273.15°C$ nicht unterschreitet**
- **mit dem Zusatz $°C$ endet**

**FEHLER:**

Positive Temperaturangaben beginnen mit $\oplus$ oder keinem Vorzeichen, negative Temperaturangaben mit $\ominus$.

Der ganzzahlige Bereich darf keine führenden Nullen aufweisen, wenn dieser nicht genau 0 ist. Außerdem darf der Ganzzahlige Bereich nicht leer sein.
Der reguläre Ausdruck
$$u = 0 + (1 - 9)(0 - 9)^*$$
erfüllt genau diese Kriterien.

Nachkommastellen können beliebige Ziffern sein, aber es muessen genau zwei vorkommen. Nachkommstellen erfolgen unmittelbar nach dem Zeichen '.'.
Der reguläre Ausdruck
$$v = .(0 - 9)^2$$
erfüllt genau diese Kriterien.

Während Temperaturangaben unendlich gross werden dürfen, wird eine untere Grenze $-273.15°C$ angegeben. Das bedeutet, dass im negativen Bereich sicher gestellt werden muss, dass keine Zahl kleiner ist.
Der reguläre Ausdruck
$$n = \ominus(0v + (1 - 9)((1 - 9)?)v + 1(0 - 9)^2 v + 2(1 - 6)(0 - 9)v + 27(1 - 2)v + 273.(0 + 1)(0 - 5))$$
erfüllt die Kriterien für den numerischen Anteil von Temperaturangaben im negativen Bereich.

Der Suffix der Temperaturangabe muss genau $°C$ sein.
Der reguläre Ausdruck
$$c = °C$$
erfüllt genau dieses Kriterium.

Aus diesen Regel und den oben angegebenen Teilausdrücken ergibt sich $\alpha$ mit
$$\alpha = (\oplus?uv + n)c$$

**KORREKTUR:**

Positive Temperaturangaben beginnen mit $\oplus$ oder keinem Vorzeichen, negative Temperaturangaben mit $\ominus$.

Der ganzzahlige Bereich darf keine führenden Nullen aufweisen, wenn dieser nicht genau 0 ist. Außerdem darf der Ganzzahlige Bereich nicht leer sein.
Der reguläre Ausdruck
$$u \stackrel{\text{def}}{=} 0 + [1 - 9][0 - 9]^*$$
erfüllt genau diese Kriterien.

Nachkommastellen können beliebige Ziffern sein, aber es muessen genau zwei vorkommen. Nachkommstellen erfolgen unmittelbar nach dem Zeichen '.'.
Der reguläre Ausdruck

$$v \stackrel{\text{def}}{=} .[0-9]^2$$

erfüllt genau diese Kriterien.

Während Temperaturangaben unendlich gross werden dürfen, wird eine untere Grenze $-273.15°C$ angegeben. Das bedeutet, dass im negativen Bereich sicher gestellt werden muss, dass keine Zahl kleiner ist.
Der reguläre Ausdruck

$$n \stackrel{\text{def}}{=} \ominus(0v + [1-9]([0-9]?)v + 1[0-9]^2v + 2[1-6][0-9]v + 27[1-2]v + 273.(0[0-9] + 1[0-5]))$$

erfüllt die Kriterien für den numerischen Anteil von Temperaturangaben im negativen Bereich, da die erste Veroderung die Zahlen bis 99.99, die Zweite die Zahlen bis 199.99, die Dritte die Zahlen bis 269.99, die Vierte die Zahlen bis 272.99 und die Fünfte die Zahlen bis 273.15 abdeckt.

Der Suffix der Temperaturangabe muss genau $°C$ sein.
Der reguläre Ausdruck

$$c \stackrel{\text{def}}{=} °C$$

erfüllt genau dieses Kriterium.

Aus diesen Regel und den oben angegebenen Teilausdrücken ergibt sich $\alpha$ mit

$$\alpha \stackrel{\text{def}}{=} (\oplus?uv + n)c$$

**1.2.2** **Seien** $\Sigma = \{a, b\}, L_1 = \{\sigma\sigma u | \sigma \in \Sigma, u \in \Sigma^*\}, L_2 = \{\tau w \tau | \tau \in \Sigma, w \in \Sigma^*\}$
**Geben Sie reguläre Ausdrücke** $\alpha_1, \alpha_2, \alpha$ **an, mit** $L_1 = L(\alpha_1), L_2 = L(\alpha_2), L = L_1 \cap L_2$ **an.**

**FEHLER:**

$L_1$

In $L_1$ enthaltene Wörter müssen mit mindest zwei beliebigen Zeichen aus $\Sigma$ beginnen und können darauf folgend mit beliebig vielen Zeichen aus $\Sigma$ konkatiniert werden.
Der entsprechende reguläre Ausdruck lautet:

$$\alpha_1 \overset{\text{def}}{=} (a+b)^2(a+b)^*$$

$L_2$

In $L_2$ enthaltene Wörter fangen mit einem beliebigem Zeichen aus $\Sigma$ an und enden mit einem gleichem Zeichen aus $\Sigma$. Zwischen diesen Zeichen können beliebige Zeichen aus Sigma vorkommen.
Der entsprechende erweiterte reguläre Ausdruck lautet:

$$\alpha_2 \overset{\text{def}}{=} a(a+b)^*a + b(a+b)^*b$$

**TODO: Beweiß ohne Äquivalenz-Kette**

$L$

Im Folgendem wird die Äuquivalenz zwischen $\alpha_1$ und $\alpha_2$ und ferner die Gleichheit von $L_1$ und $L_2$, sowie dann auch $L$ gezeigt.

$$\alpha_1 = (a+b)^2(a+b)^* \equiv (a+b)(a+b)(a+b)^* \overset{(\beta\beta^* \equiv \beta^*\beta)}{\equiv} (a+b)(a+b)^*(a+b) \equiv \alpha_2$$

Da $\alpha_1, \alpha_2$ äquivalent und damit der Schnitt der durch sie erzeugten Sprachen gleich den Sprachen selbst ist, muss fuer den regulären Ausdruck $\alpha$ gelten, dass er nur Wörter beschreibt, die auch $\alpha_1, \alpha_2$ beschreiben.

$$\alpha \equiv \alpha_1 \equiv \alpha_2$$

Dies ist etwa mit $\alpha = \alpha_1$ gegeben.
Damit gilt

$$L(\alpha) = L(\alpha_1) = L(\alpha_2); L(\alpha) = L(\alpha) \cap L(\alpha) = L(\alpha_1) \cap L(\alpha_1) = L(\alpha_1) \cap L(\alpha_2)$$

**KORREKTUR:**

$L_1$

In $L_1$ enthaltene Wörter müssen mit mindest zwei gleiche Zeichen aus $\Sigma$ beginnen und können darauf folgend mit beliebig vielen Zeichen aus $\Sigma$ konkatiniert werden.
Der entsprechende reguläre Ausdruck lautet:

$$\alpha_1 \overset{\text{def}}{=} a^2(a+b)^* + b^2(a+b)^*$$

$L_2$

In $L_2$ enthaltene Wörter fangen mit einem beliebigem Zeichen aus $\Sigma$ an und enden mit einem beliebigem Zeichen aus $\Sigma$. Zwischen diesen Zeichen können beliebige Zeichen aus Sigma vorkommen.
Der entsprechende erweiterte reguläre Ausdruck lautet:

$$\alpha_2 \overset{\text{def}}{=} (a+b)(a+b)^*(a+b)$$

$L$

Im folgendem wird die Äuquivalenz zwischen $\alpha_1$ und $\alpha_2$ und ferner die Gleichheit von $L_1$ und $L_2$, sowie dann auch $L$ gezeigt.

$$\alpha_1 = (a+b)^2(a+b)^* \equiv (a+b)(a+b)(a+b)^* \equiv^{(\beta\beta^* \equiv \beta^*\beta)} (a+b)(a+b)^*(a+b) \equiv \alpha_2$$

Da $\alpha_1, \alpha_2$ äquivalent und damit der Schnitt der durch sie erzeugten Sprachen gleich den Sprachen selbst ist, muss fuer den regulären Ausdruck $\alpha$ gelten, dass er nur Wörter beschreibt, die auch $\alpha_1, \alpha_2$ beschreiben.

$$\alpha \equiv \alpha_1 \equiv \alpha_2$$

Dies ist etwa mit $\alpha \overset{\text{def}}{=} \alpha_1$ gegeben.
Damit gilt

$$L(\alpha) = L(\alpha_1) = L(\alpha_2); L(\alpha) = L(\alpha) \cap L(\alpha) = L(\alpha_1) \cap L(\alpha_1) = L(\alpha_1) \cap L(\alpha_2)$$

### 1.2.3 Nach dem Standard ISO 8601 wird ein Datum in der Form JJJJ-MM-TT notiert. Beispielsweise wird der Geburtstag Alan Turings, der 23. Juni 1912, durch 1912-06-23 repräsentiert. Konstruieren Sie einen erweiterten regulären Ausdruck über dem Alphabet $\Sigma = \{0, 1, ..., 9, \ominus\}$, der alle gültigen Daten des Jahres 2018 beschreibt.

Es sind folgende Regeln zu beachten:
(i) Das Jahr ist 2018 und wird von einem $\ominus$ gefolgt

Der reguläre Ausdruck

$$\beta \stackrel{\text{def}}{=} 2018\ominus$$

erfuellt genau diese Regel.

(ii) Ein Monat besteht aus genau zwei Ziffern und darf nicht kleiner als 01 oder groeßer als 12 sein. Der Februar hat nur 28 Tage , die Monate $\{01, 04, 06, 08, 10, 12\}$ koennen bis zu 31 und die Monate $\{03, 05, 07, 09, 11\}$ koennen bis zu 30 Tage haben. Monate und Tage sind mit einem $\ominus$ konkatiniert. Tage bestehen aus zwei Ziffern und können nicht kleiner als 01 sein.

$m_{31} \stackrel{\text{def}}{=} (01 + 04 + 06 + 08 + 10 + 12), m_{30} \stackrel{\text{def}}{=} (03 + 05 + 07 + 09 + 11), m_{28} \stackrel{\text{def}}{=} 02$ seien die regulären Ausdruecke für Monate mit 31, 30 oder 28 Tagen.
Daraus ergibt sich fuer die Tage und Monate der reguläre Ausdruck

**FEHLER:**

$$\begin{aligned}\gamma = {} & m_{31} \ominus (0(1-9) + (1+2)(0-9) + 3(0+1)) \\ & + m_{30} \ominus (0(1-9) + (1+2)(0-9) + 30)) \\ & + m_{28} \ominus (0(1-9) + 1(0-9) + 2(0-8))\end{aligned}$$

Der aus den Regeln resultierende reguläre Ausdruck ist

$$\alpha = \beta(\gamma)$$

**KORREKTUR:**

$$\begin{aligned}\gamma \stackrel{\text{def}}{=} {} & m_{31} \ominus (0[1-9] + (1+2)[0-9] + 3(0+1)) \\ & + m_{30} \ominus (0[1-9] + (1+2)[0-9] + 30)) \\ & + m_{28} \ominus (0[1-9] + 1[0-9] + 2[0-8])\end{aligned}$$

Der aus den Regeln resultierende reguläre Ausdruck ist

$$\alpha \stackrel{\text{def}}{=} \beta(\gamma)$$