GTI Uebungsblatt 1

Max Springenberg, 177792

1.1

1.1.1 **Seien $\beta = (ab)^*$ und $\alpha_1, ..., \alpha_8$ die folgenden erweiterten regularen Ausdrucke. Beurteilen Sie fur alle $i \in \{3, ..., 8\}$, ob $L(\alpha_i) \subseteq L(\beta)$ gilt. Vervollstandigen Sie dazu die folgende Tabelle analog zu den Beispiel-Ausdrucken α_1, α_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
$\alpha_5 = (a + \epsilon)(b + \epsilon)(a * b*)^*$	✗	aa
$\alpha_6 = (ab * ab*)^*$	✓	
$\alpha_7 = (abab)^*$	✓	
$\alpha_8 = a(ba)^+b$	✓	

1.1.2 **Das Verhalten eines Netzwerk-Controllers soll anhand protokollierter Ausgaben analysiert wer- den. Der Controller schreibt, abhangig von der Eingabe, beliebig lange Bitfolgen. Eine Bitfolge wird als Wort uber dem Alphabet $\{0, 1\}$ reprasentiert. Die Menge der gultigen Ausgaben wird im Handbuch des Controllers formal als Sprache $L = \{0, 1\}^* - L(\gamma)$ fur den erweiterten regularen Ausdruck**

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

spezifiziert. Beschreiben Sie L naturlichsprachlich kurz in einem Satz.

In γ enthaltene Woerter haben mindestens eine der folgenden Eigenschaften:

- Das Wort enthaelt drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort beginnt mit 10 gefolgt von einer 1 oder ϵ und endet
- Das Wort beginnt mit 01 gefolgt von einer 0 oder ϵ und endet

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

- Das Wort enthaelt keine drei aufeinanderfolgenede Nullen oder Einsen
- Das Wort beginnt nicht mit 10 gefolgt von einer 1 oder ϵ und endet
- Das Wort beginnt nicht mit 01 gefolgt von einer 0 oder ϵ und endet

1.2 Geben Sie im Folgenden regulare Ausdrucke bzw. erweiterte regulare Ausdrucke an. Beschreiben Sie fur jede Konstruktion kurz, warum Ihr Ausdruck die Sprache beschreibt (warum er alle Worter 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 regulaeren Ausdruck α ueber Σ, der genau die gueltigen Temparaturangaben mit zwei nachkommastellen in Grad Celsius beschreibt.

Eine Solche Temperaturangabe ist gueltig, wenn sie :

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

Positive Temperaturangaben beginnen mit $\oplus$ oder keinem Vorzeichen, negative Temperaturangaben mit $\ominus$.
Der ganzzahlige Bereich darf keine fuehrenden Nullen aufweisen, wenn dieser nicht genau 0 ist. Ausserdem darf der Ganzzahlige Bereich nicht leer sein.
Der regulaere Ausdruck

$$u = 0 + (1-9)^{+}(0-9)^{*}$$

erfuellt genau diese Kriterien.
Nachkommastellen koennen beliebige Ziffern sein, aber es muessen genau zwei vorkommen. Nachkommstellen erfolgen unmittelbar nach dem Zeichen '.'.
Der regulare Ausdruck

$$v = .(0-9)^{2}$$

erfuellt genau diese Kriterien.
Waehrend Temperaturangaben unendlich gross werden duerfen wird eine untere Grenze $-273.15°C$ angegeben. Das bedeutet, dass im negativen Bereich sicher gestellt werden muss, dass keine Zahl kleiner ist.
Der regulaere Ausdruck

$$n = \ominus(0v+(1-9)((1-9)?)v+1(0-9)^2v+2(1-6)(0-9)v+27(1-2)v+273.(0+1)(0-5))$$

erfuellt die Kriterien fuer den numerischen Anteil von Temperaturangaben im negativen Bereich.
Der Suffix der Temperaturangabe muss genau $°C$ sein.
Der regulaere Ausdruck

$$c =^\circ C$$

erfuellt genau dieses Kriterium.

Aus dieser Regel und den oben angegebenen Teilausdruecken ergibt sich α mit

$$\alpha = (\oplus?uv + \ominus n)c$$

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 regulaere Ausdruecke $\alpha_1, \alpha_2, \alpha$ an, mit $L_1 = L(\alpha_1), L_2 = L(\alpha_2), L = L_1 \cap L_2$ an.

L_1

In L_1 enthaltene Woerter besizten mindest zwei beliebigen Zeichen aus Σ beginnen und koennen mit darauf folgend mit beliebig vielen Zeichen aus Σ konkatiniert werden.
Der entsprechende regulaere Ausdruck lautet:

$$\alpha_1 = (a + b)^2(a + b)^*$$

L_2

In L_2 enthaltene Woerter fangen mit einem beliebigem Zeichen aus Σ an und enden mit einem beliebigem Zeichen aus Σ. Zwischen diesen Zeichen koennen beliebige Zeichen aus Sigma vorkommen.
Der entsprechende regulaere Ausdruck lautet:

$$\alpha_2 = (a + b)(a + b)^*(a + b)$$

L

Im folgendem wird die Aeuquivalenz zwischen α_1 und α_2 und ferner die Aequivalenz von L_1 und L_2, sowie dann auch L bewiesen.

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

Fuer den regulaeren Ausdruck α muss gelten, dass er nur Woerter beschreibt, die auch α_1, α_2 beschreiben. Durch die Aequivalenz von α_1, α_2 muss damit dann auch gelten

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

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

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

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 reprasentiert. Konstruieren Sie einen erweiterten regularen Ausdruck uber dem Alphabet $\Sigma = \{0, 1, ..., 9, \ominus\}$, der alle gultigen Daten des Jahres 2018 beschreibt.

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

Der regulaere Ausdruck

$$\beta = 2018\ominus$$

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 zwei Ziffern und koennen nicht kleiner als 01 sein.

$m_{31} = (01 + 04 + 06 + 08 + 10 + 12), m_{30} = (03 + 05 + 07 + 09 + 11), m_{28} = 02$ seien die regulaeren Ausdruecke fuer Monate mit 31, 30 oder 28 Tagen.
Darueber ergibt sich fuer die Tage und Monate der regulaere Ausdruck

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

Der aus den Regeln resultierende regulaere Ausdruck ist

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