

Übungen zu regulären Ausdrücken

(Vorlesungsaufgaben)

Stichwörter: Reguläre Ausdrücke

- (a) Gegeben ist eine Sprache $L \subset \Sigma^*$ mit $\Sigma = \{a, b\}$. Zu der Sprache L gehören alle Wörter, die die Zeichenfolge `abba` beinhalten.

Geben Sie einen regulären Ausdruck für diese Sprache an („klassischer“ regulärer Ausdruck).

Lösungsvorschlag

$$(a|b)^*abba(a|b)^*$$

```
static String regexABBA = "(a|b)*abba(a|b)*";
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)

- (b) Gebe möglichst einfache reguläre Ausdrücke für die folgenden Sprachen $L_x \subset \Sigma^*$ mit $\Sigma = \{a, b\}$ und $x \in \{1, 2, 3\}$ („klassischer“ regulärer Ausdruck).

$$L_1 = \{x \mid x \text{ beinhaltet eine gerade Anzahl von } a\}$$

Lösungsvorschlag

$$b^*(ab^*ab^*)^*$$

```
static String regexGeradeA = "b*(ab*ab*)*"; // Epsilon aa aaaa abba bbaa
↪ aabb bbb
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)

$$L_2 = \{x \mid x \text{ beinhaltet eine ungerade Anzahl von } b\}$$

Lösungsvorschlag

$$a^*ba^*(ba^*ba^*)^*$$

```
static String regexUngeradeB = "a*ba*(ba*ba*)*"; // Epsilon b bbb abababa
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)

$$L_3 = \{x \mid x \text{ beinhaltet an seinen geradzahligen Positionen ausschließlich } a\}$$

Lösungsvorschlag

$$((a|b)a)^*(a^*|b)$$

```
static String regexGeradzahligA = "((a|b)a)^*(a^*|b)"; // aa ba aab b
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)

- (c) Geben Sie einen regulären Ausdruck an, der eine syntaktisch gültige E-Mail-Adresse erkennt. (mindestens 1 Zeichen (Groß-/Kleinbuchstabe oder Zahl) vor dem @; mindestens 1 Zeichen (Groß-/Kleinbuchstabe oder Zahl) nach dem @; alle E-Mail-Adressen sollen auf `.de` oder `.com` enden).

Lösungsvorschlag

```
static String regexEMAIL = "\\w+@\\w+\\. (de|com)";
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)

Lösungsvorschlag

```
public class TestRegularExpressions {
    // Hier bitte Lösungen der Aufgaben eintragen.
    static String regexABBA = "(a|b)*abba(a|b)*";
    static String regexGeradeA = "b*(ab*ab)*"; // Epsilon aa aaaa abba bbaa aabb bbb
    static String regexUngeradeB = "a*ba*(ba*ba)*"; // Epsilon b bbb abababa
    static String regexGeradzahligA = "((a|b)a)*(a*|b)"; // aa ba aab b
    static String regexEMAIL = "\\w+@\\w+\\. (de|com)";
    // Wenn die Lösungen stimmen, geben alle Tests true aus

    // Alternativen:
    // static String regexGeradzahligA = "((a|b)a)*((a|b)|)"; // aa ba aab b

    public static void main(String[] args) {
        testregexABBA();
        testregexGeradeA();
        testregexUngeradeB();
        testregexGeradzahligA();
        testregexEMAIL();
    }

    public static void testregexABBA() {
        boolean[] b = new boolean[7];
        b[0] = "abba".matches(regexABBA);
        b[1] = !"aba".matches(regexABBA);
        b[2] = "abbaabbaabba".matches(regexABBA);
        b[3] = "abababbaaabaabaaba".matches(regexABBA);
        b[4] = !"ab".matches(regexABBA);
        b[5] = !"bbb".matches(regexABBA);
        b[6] = !"".matches(regexABBA);
        if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
            System.out.println("Alle ABBA-Tests bestanden. Dein RegEx stimmt!");
        } else {
            for (int i = 0; i < b.length; i++) {
                if (!b[i]) {
                    System.out.println("Test mit dem Index" + i + " leider nicht bestanden.");
                }
            }
        }
    }

    public static void testregexGeradeA() {
        boolean[] b = new boolean[5];
        b[0] = "aa".matches(regexGeradeA);
        b[1] = !"aaa".matches(regexGeradeA);
        b[2] = "abbaabbaabba".matches(regexGeradeA);
        b[3] = !"abababbaaabaabaaba".matches(regexGeradeA);
        b[4] = !"ab".matches(regexGeradeA);
    }
}
```

```

b[5] = "bbb".matches(regexGeradeA);
b[6] = "".matches(regexGeradeA);
if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
    System.out.println("Alle GeradeA-Tests bestanden. Dein RegEx stimmt!");
} else {
    for (int i = 0; i < b.length; i++) {
        if (!b[i]) {
            System.out.println("Test mit dem Index" + i + " leider nicht bestanden.");
        }
    }
}
}

public static void testregexUngeradeB() {
    boolean[] b = new boolean[7];
    b[0] = "b".matches(regexUngeradeB);
    b[1] = !"bb".matches(regexUngeradeB);
    b[2] = !"abbaabbaabba".matches(regexUngeradeB);
    b[3] = "abababbaaabaabaaba".matches(regexUngeradeB);
    b[4] = "ab".matches(regexUngeradeB);
    b[5] = "bbb".matches(regexUngeradeB);
    b[6] = !"".matches(regexUngeradeB);
    if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
        System.out.println("Alle UngeradeB-Tests bestanden. Dein RegEx stimmt!");
    } else {
        for (int i = 0; i < b.length; i++) {
            if (!b[i]) {
                System.out.println("Test mit dem Index" + i + " leider nicht bestanden.");
            }
        }
    }
}

public static void testregexGeradzahligA() {
    boolean[] b = new boolean[7];
    b[0] = !"ab".matches(regexGeradzahligA);
    b[1] = "b".matches(regexGeradzahligA);
    b[2] = "babab".matches(regexGeradzahligA);
    b[3] = !"bababaab".matches(regexGeradzahligA);
    b[4] = "ba".matches(regexGeradzahligA);
    b[5] = "aaa".matches(regexGeradzahligA);
    b[6] = "".matches(regexGeradzahligA);
    if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
        System.out.println("Alle GeradzahligeA-Tests bestanden. Dein RegEx stimmt!");
    } else {
        for (int i = 0; i < b.length; i++) {
            if (!b[i]) {
                System.out.println("Test mit dem Index" + i + " leider nicht bestanden.");
            }
        }
    }
}

public static void testregexEMAIL() {

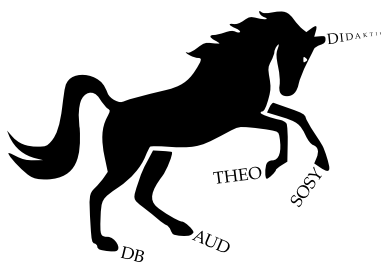
```

```

boolean[] b = new boolean[7];
b[0] = "3@s.de".matches(regexEMAIL);
b[1] = !"@0.de".matches(regexEMAIL);
b[2] = "asdf@asdf.com".matches(regexEMAIL);
b[3] = !"@.de".matches(regexEMAIL);
b[4] = "s@1.com".matches(regexEMAIL);
b[5] = !"a@a".matches(regexEMAIL);
b[6] = !"".matches(regexEMAIL);
if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
    System.out.println("Alle Email-Tests bestanden. Dein RegEx stimmt!");
} else {
    for (int i = 0; i < b.length; i++) {
        if (!b[i]) {
            System.out.println("Test mit dem Index" + i + " leider nicht bestanden.");
        }
    }
}
}
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java](https://github.com/src/main/java/org/bschlangaul/aufgaben/theo_inf/regulaere_ausdruecke/TestRegularExpressions.java)



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/70_THEO/10_Formale-Sprachen/10_Typ-3_Regulaer/Regulaere-Ausdruecke/Aufgabe_Vorlesungsaufgaben-Regulaere-Ausdruecke.tex