

Automaten und Formale Sprache

Aufgabenblatt 1

Tim Zolleis & Julian Bertol

30. September 2024

Aufgabe 1

1. $\{a, 0\}$
2. $\{aa, 00, a0, 0a\}$
3. $\{aaa, aab, aac, aba, abb, abc, baa, bba, bbb, bca, bcb, bcc, caa, cba, cbb, cbc, cca, ccb, ccc\}$
4. $\{\epsilon, a, aa, aaa, aaaa, \dots\}$
5. $\{\epsilon, b, bb, bbb, bbbb, \dots\}$
6. $\{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$

Aufgabe 2

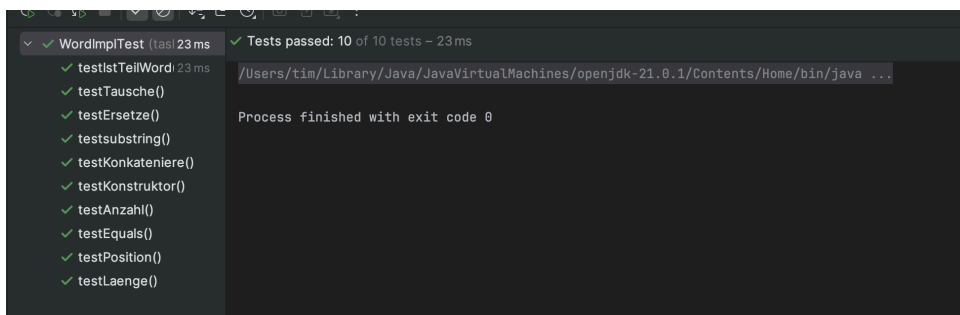
1. Nicht korrekt, da $\{a\}^* \cup \{b\}^*$ nur Zeichenfolgen entweder aus ausschließlich a oder ausschließlich b enthalten kann, $\{a, b\}^*$ aber auch z.B. $\{ab, aabb\}$
2. Korrekt, da $\{a\}^* \cup \{b\}^*$ eine Teilmenge von $\{a, b\}^*$ darstellt (da diese auch aa, bb etc. enthalten kann)
3. Dies ist nicht korrekt, da die drei Mengen jeweils ein leeres Wort beinhalten, somit ist die Schnittmenge $\{\epsilon\}$
4. Dies ist nicht korrekt, da nur noch $\{b\}$ übrig bleibt, nicht $\{b\}^*$

Aufgabe 3

1. Dies ist falsch, da man beliebige Teile aus $\{a, b\}$ verwenden kann
2. Dies ist falsch, da bei der Folgerung nur Längen, nicht Inhalte verglichen werden.
3. Dies ist korrekt, da z.B. $|a, aa, aaa| = 6$ und $|aaaaaa| = 6$ (die drei Wörter zusammengehängt)
4. Dies ist korrekt, da die Teilwortbeziehung transitiv ist, somit a ein Teilwort von C sein muss
5. Dies ist nicht korrekt, da $(ab)^4 \rightarrow (abababab)$ und nicht $(aaaa,bbbb)$

6. Dies ist korrekt, da v vollständig in w vorhanden sein muss, und somit die Anzahl der A's in v immer kleiner gleich der Anzahl der A's in W sein muss
7. Dies ist korrekt. Wenn u vollständig in v enthalten ist, und v vollständig in u müssen die beiden Wörter gleich sein
8. Dies ist korrekt, das geht mit dem leeren Wort
9. Dies ist korrekt, da die Reihenfolge der Konkatenation hier keine Rolle spielt, es kommt immer das gleiche Wort heraus
10. Dies ist nicht korrekt, da z.B wenn $u = ab$ und $v = ba$, dann wäre $u * v = abba$ und $v * u = baab$

Aufgabe 4



Code (siehe nächste Seite) kann zusätzlich hier eingesehen werden:

<https://github.com/timzolleis/hfu-ain-afs/tree/main/src>

<https://github.com/JulianBertol/HFU/tree/master/3.Semester/AFS/src/main/java/org/Task1>

```
package task01;

import lombok.Getter;

@Getter
public class WordImpl implements Word {

    public final String contents;

    public WordImpl(final String contents) {
        this.contents = contents;
    }

    public final char position(final int k) {
        final int index = k - 1;
        if (index < 0 || index > this.contents.length()) {
            throw new IndexOutOfBoundsException("Index" + index + " out of bounds");
        }
        return this.contents.charAt(index);
    }

    public final int length() {
        return this.contents.length();
    }

    public final Word concat(Word word) {
        final String newContents = this.contents.concat(word.toString());
        return new WordImpl(newContents);
    }

    public final int count(char c) {
        return (int) this.contents.chars().filter(character -> character == c).count();
    }

    public final Word replaceChar(char char1, char char2) {
        final String newContents = this.contents.replace(char1, char2);
        return new WordImpl(newContents);
    }

    public final int isSubstringOf(Word word) {
        final int index = this.contents.indexOf(word.toString());
        return index == -1 ? 0 : index + 1;
    }

    public final Word substring(int start, int length) {
        final int startIndex = start - 1;
        final int endIndex = start + length - 1;
        if (startIndex < 0 || endIndex > this.contents.length()) {
            throw new IndexOutOfBoundsException("Index out of bounds");
        }
        return new WordImpl(this.contents.substring(startIndex, endIndex));
    }

    public final Word replace(Word word1, Word word2) {
        final String contentToReplace = word1.toString();
        final String replacementContent = word2.toString();
        final String newContents = this.contents.replace(contentToReplace,
replacementContent);
        return new WordImpl(newContents);
    }

    public final boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (!(obj instanceof WordImpl word)) {
            return false;
        }
        return this.contents.equals(word.contents);
    }

    public final String toString() {
        return this.contents;
    }
}
```

