

Blatt 1

Automaten und formale Sprachen Praktikum

Teammitglieder: Luis Staudt
Dominik Meurer

1 Wortmengen durch Aufzählen aller Element

1. $\{a, 0\}_1^* = \{a; 0\}$
2. $\{a, 0\}_2^* = \{aa; a0; 0a; 00\}$
3. $\{a, b, c\}_3^+ =$
 $\{aaa; aab; aac; aba; abb; abc; aca; acb; acc;$
 $baa; bab; bac; bba; bbb; bbc; bcb; bcc;$
 $caa; cab; cac; cba; cbb; cca; ccb; ccc\}$
4. $\{a\}^* = \{a; aa; aaa; aaaa; \dots\}$
5. $\{b\}^* = \{b; bb; bbb; bbbb; \dots\}$
6. $\{a, b\}^* = \{\epsilon; a; b; aa; bb; ab; ba; aab; \dots\}$

2 Richtig oder falsch?

1. $\{a\}^* \cup \{b\}^* = \{a, b\}^*$ Das ist falsch. Links wäre es $\{a, aa, \dots, b, bb, \dots\}$ und rechts $\{a, b, aa, ab, ba, bb, \dots\}$
2. $\{a\}^* \cup \{b\}^* \subseteq \{a, b\}^*$ Das ist richtig, da auch jedes $e \in \{a, aa, \dots, b, bb, \dots\}$ auch in $\{a, b, aa, ab, ba, bb, \dots\}$ vorkommt.
3. $\{a\}^* \cap \{b\}^* \cap \{c\}^* = \{\}$ Das stimmt nicht, die richtige Lösung wäre $\{\epsilon\}$
4. $\{a, b\}^* \cap \{b, c\}^* = \{b\}^*$ Das stimmt, da nur diese Elemente übrig bleiben.

3 Behauptungen überprüfen

1. Nein, gilt nicht für $v = 'abbbb', w = 'aaa'$

2. Nein, gilt nicht für $v = 'aab'$, $w = 'abb'$
3. Ja, die Wörter werden aneinander gehängt, und somit die Länge um die entsprechende Zahl erhöht.
4. Ja, macht nur Sinn. Wenn $'aa'$ ein Teilwort von $'aaa'$ ist und das wiederum von $'aaab'$, dann muss $'aa'$ auch ein Teilwort von $'aaab'$ sein.
5. Nein, denn $(ab)^4 = abababab$.
6. Ja, wenn u mehr $'a'$ hat, dann wäre es kein Teilwort.
7. Ja, da sie beide echte Teilwörter oder gleich sind. Da sie nicht beide gleichzeitig echte Teilwörter von einander sein können, müssen sie gleich sein.
8. Ja, wenn das Wort ϵ . Das ist das neutrale Element.
9. Ja, in ihrer Vorlesung hieß es, die Operation sei assoziativ.
10. Nein, bei $v = 'a'$ und $u = 'b'$ wäre das Ergebnis $ab = ba$, was nicht stimmt.

4 Programmieraufgabe

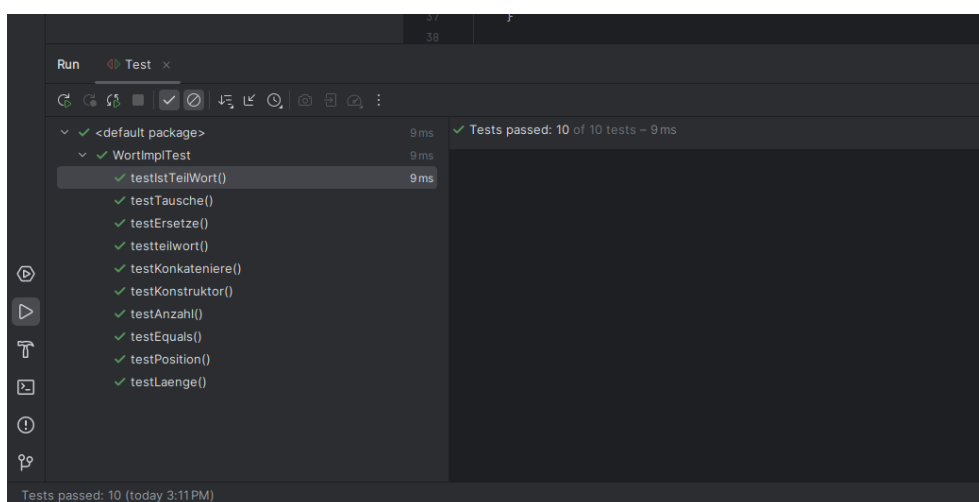


Abbildung 1: Screenshot Testprotokoll

Unten gibts noch den Code dazu:

```
1 package task1;
2
3 import java.util.Objects;
4
5 public class WortImpl implements Wort {
6
7     private final String internal;
8
9     public WortImpl(char[] internal) {
10         this.internal = new String(internal);
11     }
12
13     public WortImpl(String internal) {
14         this.internal = internal;
15     }
16
17     @Override
18     public char position(int k) {
19         if (k <= 0) {
20             throw new IllegalArgumentException("Index must not be greater than 0");
21         }
22         if (k > this.internal.length()) {
23             throw new IndexOutOfBoundsException("Index must not be greater than the length of the word");
24         }
25         return this.internal.charAt(k - 1);
26     }
27
28     @Override
29     public int laenge() {
30         return this.internal.length();
31     }
32
33     @Override
34     public Wort concat(Wort w) {
35         return new WortImpl(this.internal + w.toString());
36     }
37
38     @Override
39     public int anzahl(char c) {
40         return (int) this.internal.chars().filter(ch -> ch == c).count();
41     }
42
43     @Override
44     public Wort tausche(char c1, char c2) {
45         return new WortImpl(this.internal.replace(c1, c2));
46     }
47
48     @Override
49     public int istTeilwortVon(Wort w) {
50         return this.internal.indexOf(w.toString()) + 1;
51     }
52
53     @Override
54     public Wort teilwort(int start, int laenge) {
55         return new WortImpl(this.internal.substring(start - 1, start - 1 + laenge));
56     }
57
58     @Override
```

```
59 public Wort ersetze(Wort w1, Wort w2) {
60     return new WortImpl(this.internal.replace(w1.toString(), w2.toString()));
61 }
62
63 @Override
64 public boolean equals(Object o) {
65     if (this == o) return true;
66     if (!(o instanceof WortImpl wort)) return false;
67
68     return this.internal.equals(wort.internal);
69 }
70
71 @Override
72 public int hashCode() {
73     return Objects.hash(this.internal);
74 }
75
76 @Override
77 public String toString() {
78     return this.internal;
79 }
80 }
```

Code Listing 1: Implementierung