

Aufgabe 1: Störung

Team-ID: 00534

Team: Monads

Bearbeiter/-innen dieser Aufgabe:
Leopold Jofer, Benjamin Hantschel, Robert Zeitter

20. November 2022

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	2

Lösungsidee

Zunächst muss man sich das Problem bewusst machen: man besitzt einen beliebigen Auszug eines Buches, bei dem jedoch einzelne Wörter beziehungsweise durch ein „_“ ersetzt wurden. Nun ist das Ziel die entsprechende Stelle im Buch zu finden und die fehlenden Wörter wieder einzutragen. Grundsätzlich benötigt man also die Länge des Satzes/Auszuges (in Wörtern) und den Ausschnitt. Da nicht bekannt ist, ob der Auszug der vollständige Satz ist, ist es nicht nötig, alle Sätze, welche die Anzahl an Wörter überschreitet zu filtern. Man kann aber das Buch nach Sätzen trennen. Anschließend wird jeder Satz kontrolliert: hierbei wird zunächst überprüft, ob die bereits bekannten Wörter miteinander übereinstimmen. Ist dies der Fall kann inspiziert werden, ob die Lücken übereinstimmen. D. h. für jeden Unterstrich, welcher zwischen den bekannten Wörtern steht, muss ein Wort gefunden werden, welches die gleichen Vor- und Nachfolger hat. Befinden sich n viele Unterstriche hintereinander muss zunächst überprüft werden, ob es einen Satz gibt, welche die bekannten Wörter beinhaltet und zwischen diesen n viele weitere Wörter sind. Dadurch, dass jedoch theoretisch mehrere Kombinationen beziehungsweise Sätze in Frage kommen könnten, darf nicht bereits nach der ersten Übereinstimmung die Suche abgebrochen werden. Diese analysiert zunächst das gesamte Buch auf Treffer und gibt diese dann aus.

Umsetzung

Das Programm wurde in Rust geschrieben.

Zunächst muss das gesamte Buch aus der Datei ausgelesen und als ein String gespeichert werden:

```
„let book = fs::read_to_string(\"./buch.txt\")?;“
```

Um ebenfalls den Auszug zu erhalten wird das gleiche verfahren angewendet, lediglich der Pfad ändert sich.

Als nächstes wird jeder Unterstrich mit einem regulärem Ausdruck (Regex) ersetzt und zu einem

String zusammengeführt: „let query = format!(\"(?mi){}\", sentence.replace('_', \"[:alpha:]+\"));“
 Nun müssen wir mithilfe einer for-loop das gesamte Buch analysieren und überprüfen, ob sich unsere Ausschnitt darin befindet: „for (mat) in re.find_iter(&book)“
 „mat“ beschreibt einen jeweiligen Treffer, welcher durch „re.find_iter(&book)“ gesucht wird.
 Sobald ein Treffer erfolgt, kann dieser innerhalb der jeweiligen Wiederholung entweder direkt ausgegeben werden oder in einer weiteren Variable gespeichert werden.

Beispiele

Beispiel Nummer	Eingabe	Ausgabe
1	das _ mir _ _ _ vor	Das kommt mir gar nicht richtig vor
2	ich muß _ clara _	1. Ich muß in Clara verwandelt 2. Ich muß doch Clara sein
3	_ _ _ nicht vorhanden	----- (keine Ausgabe)
4	_	Alle Wörter des Buches

Quellcode

```
use regex::Regex;
use std::{error::Error, fs};
fn main() -> Result<(), Box<dyn Error>> {
    let book = fs::read_to_string(\"./buch.txt\")?;
    let sentence = shared::file_picker::to_string(\"Stoerung\")?;
    let query = format!(\"(?mi){}\", sentence.replace('_', \"[:alpha:]+\"));
    let re = Regex::new(&query).unwrap();
    for (idx, mat) in re.find_iter(&book).enumerate() {
        println!(\"{idx}: {}\", mat.as_str());
    }
    Ok(())
}
```