

Vorlesung Systemnahe Informatik

Sommersemester 2023

Prof. Dr. Peter Martini, Dr. Matthias Frank, Lennart Buhl M.Sc.

8. Übungszettel

Ausgabe: Montag, 05. Juni 2023.
Abgabe: Sonntag, 11. Juni 2023
Besprechung: In den Übungen ab Montag, 12. Juni 2023.
Hinweis: Abgabe erfolgt freiwillig per PDF über eCampus, siehe Hinweise in Aufgabe 1 auf dem 1. Übungszettel
Sie können Kontakt zu Ihrer Tutor/in aufnehmen durch E-mail an `cs4+ueb-si-XX@cs.uni-bonn.de` mit *XX* als Gruppennummer.

Aufgabe 1: Endlicher Automat - mit Ausgabe

Es gibt endliche Automaten, die bei einem Zustandsübergang eine Ausgabe erzeugen können (*endliche Automaten mit Ausgabe*). Bei jedem Zustandsübergang gibt man dabei die Ausgabe mit an, z.B. im Zustandsübergangsdiagramm als Paar (a, b) . Gibt es keine Ausgabe, schreibt man dies dann als $(a,)$. Ein Zustandsübergang (c, c) liest ein Zeichen c und gibt dieses sofort wieder aus.

In Java (und anderen Programmiersprachen) gibt es die Möglichkeit einzeilige Kommentare, die mit „//“ eingeleitet werden, zu verwenden. Solche Kommentare muss der Compiler aus dem Quelltext entfernen.

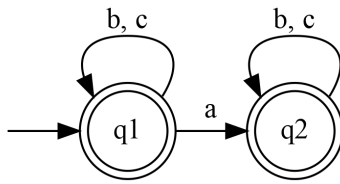
Betrachten Sie das Alphabet $A = \{c, '\backslash n', '/', eof\}$. Dabei steht $\backslash n$ für den Zeilentrenner, *eof* für das Dateiende und c für alle anderen Zeichen. Sonderfälle, bei denen $//$ als Bestandteil eines Strings auftreten kann, kommen nicht vor!

- Zeichnen Sie einen *endlichen Automaten mit Ausgabe*, der bei Eingabe des Quelltextes diesen ohne die einzeiligen, oben beschriebenen Kommentare ausgibt.
Geben Sie dabei eine Legende der Symbole für die vier, verschiedenen möglichen Typen von Zuständen eines *endlichen Automaten* an. Sie können gerne auch andere Notationen für die Zustände verwenden, diese müssen jedoch einheitlich zur Legende sein.
- Geben Sie ein Programm (in C, Java, Pseudocode, ...) an, welches dieselbe Aufgabe erfüllt. Dem Programm wird ein String „input“ übergeben, in dem der Quellcode des Programms zeichenweise (auch *eof* sei ein Zeichen) gespeichert ist. Es kommen nur Zeichen aus dem Alphabet A vor. Den Quellcode ohne Kommentarzeilen können sie auf der Standardausgabe ausgeben oder in einen String „output“ zurückgeben.

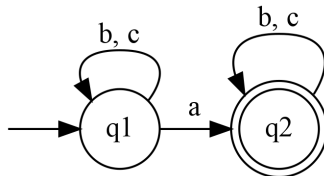
Aufgabe 2: Endlicher Automat, Spracherkennung

Welche Sprachen werden von den folgenden endlichen Automaten erkannt? Geben Sie wahlweise eine präzise umgangssprachliche Beschreibung oder einen regulären Ausdruck als Lösung an. Konzentrieren Sie sich bei der umgangssprachlichen Beschreibung auf das Symbol a . Das zugrunde liegende Alphabet ist $\{a, b, c\}$.

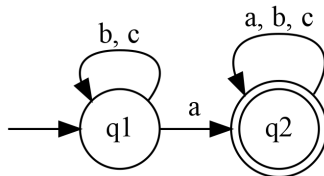
a)



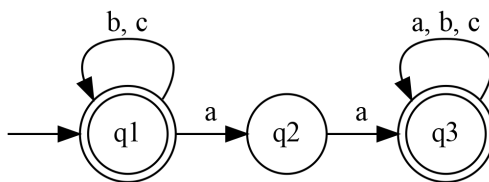
b)



c)



d)



Aufgabe 3: EA, CFG, Grammatik ergänzen, ...

- a) Geben Sie die vollständige Definition eines endlichen Automaten und die vollständige Definition einer kontextfreien Grammatik gemäß der Vorlesung an. Versuchen Sie die Aufgabe zunächst ohne Blick ins Vorlesungsmaterial zu lösen. Wie wird die Kontextfreiheit formal ausgedrückt?
- b) Gegeben seien (nur) die folgenden Produktionen einer Grammatik:

$$\begin{aligned}
 \langle E \rangle &::= \langle A \rangle \\
 \langle A \rangle &::= 0\langle B \rangle \mid 1\langle C \rangle \mid \epsilon \\
 \langle B \rangle &::= \langle A \rangle 0 \\
 \langle C \rangle &::= \langle A \rangle 1
 \end{aligned}$$

Ergänzen Sie die vollständige Definition für diese Grammatik, d.h. geben Sie sämtliche Mengen an und benennen das Startsymbol.

- c) Welche Sprache erzeugt diese Grammatik? Beschreiben Sie in Worten, oder (pseudo-) formal.
- d) Gibt es zu dieser Grammatik einen endlichen Automaten, der dieselbe Sprache akzeptiert? (vgl. Vorlesung Kapitel 2.5). Begründen Sie ihre Antwort!

Aufgabe 4: IT-Sicherheit - Dritte Zusatzaufgabe

Seit einigen Jahren gibt es zusätzlich zu den normalen Übungsaufgaben für Systemnahe Informatik einige zusätzliche Aufgaben mit dem Schwerpunkt IT-Sicherheit. Diese Aufgaben gehen über den Inhalt der Vorlesung hinaus und sind somit nicht prüfungsrelevant. Vielmehr möchten wir Anwendungsgebiete vermitteln, wofür Sie die Grundlagen benötigen, die Sie in der Systemnahen Informatik lernen.

Wir versuchen, praxisnahe Aufgaben zu formulieren, an denen Sie tüfteln und herumprobieren können. Manchmal gibt es dafür auch mehrere Lösungen. Wenn Sie bei einer Aufgabe keinen Ansatz finden, fragen Sie uns gern per E-Mail oder sprechen uns persönlich an. Wir freuen uns auf viele Ergebnisse und Ideen.

Nun zur Aufgabe:

In der ersten Zusatzaufgabe haben Sie ROT13 verwendet, um das Chiffre in Klartext zu überführen. Die Ver- und Entschlüsselung von ROT13 sind identisch, ein verschlüsselter Text lässt sich also mit demselben Algorithmus wieder in lesbaren umwandeln.

Entwickeln und beschreiben Sie einen ROT13-Algorithmus und programmieren diesen in Pseudocode oder einer Programmiersprache Ihrer Wahl. Es sollen dabei mindestens die Funktionen `encode` und `decode` vorhanden sein.

Schicken Sie Ihre Lösung wieder per E-Mail an die TutorInnen Ihrer Übungsgruppe.