

Vorlesung Systemnahe Informatik

Sommersemester 2023

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

7. Übungszettel

Ausgabe: Montag, 22. Mai 2023.
Abgabe: Sonntag, 04. Juni 2023
Besprechung: In den Übungen ab Montag, 05. 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.

Besonderheit: Die Bearbeitungszeit erstreckt sich über die vorlesungsfreie Woche nach Pfingsten. Am Donnerstag 08.06.2023 ist Feiertag. Teilnehmer/innen unserer Donnerstags-Übungen können in dieser Woche in eine der anderen Präsenz-Übungen gehen. Teilnehmer/innen der Übungsgruppen am Brücken-Freitag 09.06. sprechen bitte das Vorgehen mit der Tutorin ab.

Aufgabe 1: Lexikalische und syntaktische Analyse

Gegeben ist das folgende Syntaxdiagramm (Abb. 1), welches die Block-Struktur von Java vereinfacht darstellen soll. Die Block-Struktur in Java erlaubt es Anweisungen als Block zusammen zu fassen und zu klammern. Eine Anweisung kann aber ihrerseits auch wieder aus einer Block-Struktur bestehen.

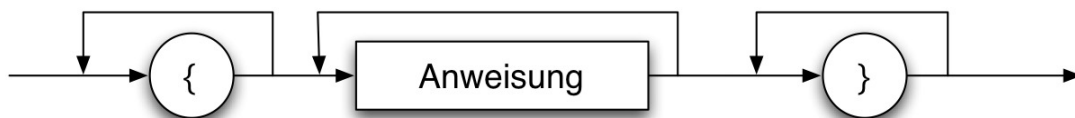


Abbildung 1: Gegebenes Syntaxdiagramm

Wandeln Sie das Syntaxdiagramm in einen deterministischen endlichen Automaten um. Verwenden Sie die bekannte Methode aus der Vorlesung. Geben Sie als Lösung jeweils

- das markierte Syntaxdiagramm (ergänzen Sie das gegebene Diagramm in Abbildung 1)
- die bekannte Tabelle mit einer Zeile für jeden Iterationsschritt

gewählter Zustand q	gewähltes Zeichen a	Marke in m	direkt erreichbare Markenmenge m'

- eine Tabelle mit Informationen zu den konstruierten Zuständen

Zustand	Markenmenge	Wählbare Zeichen

d) und den resultierenden endlichen Automaten, an.

Hinweis: Teil b), c) und d) können Sie auch verzahnt/gleichzeitig entwickeln (vgl. Beispiel der Vorlesung).

Verdeutlichen Sie sich noch einmal gemäß der Vorlesung, welche Symbole für den Startzustand, normale Zustände, Endzustände sowie den Sonderfall eines Startzustandes, der auch Endzustand ist, verwendet werden.

e) Was fällt Ihnen dabei auf? Repräsentiert der resultierende Automat wirklich das, was in Java unter Verschachtelung der Blöcke verstanden wird? Können wir unser Syntaxdiagramm „retten“? Geben Sie hierfür ein modifiziertes Syntaxdiagramm an.

Aufgabe 2: Kontextfreie Grammatiken

Die Definition einer kontextfreien Grammatik ist Ihnen aus der Vorlesung bekannt als Quadrupel der Form $G = (T, N, P, S)$. Gegeben seien nun die folgenden Teile einer Grammatik:

- T: {a, ... , z, 0, ... , 9, :, ., ,, !, ?, _}
- N: {Satz, Satzzeichen, Wort, Buchstabe, Zahl, Ziffer, Leer}
- S: {Satz}
- P: (in BNF)

```

<Satz> ::=      <Wort> <Satz> <Satzzeichen> | <Zahl> <Satz> |
                <Satz> <Wort> | <Satz> <Wort> <Satzzeichen> | <Wort>
<Satzzeichen> ::= ! | . | , | : | ?
<Wort> ::=      <Buchstabe> <Wort> | <Leer> <Wort> |
                <Buchstabe> <Leer> | <Buchstabe> <Buchstabe>
<Buchstabe> ::= a | ... | z
<Zahl> ::=      <Ziffer> <Zahl> | <Ziffer> | <Leer>
<Ziffer> ::=    0 | ... | 9
<Leer> ::=      -

```

Welche Zeichenketten aus der folgenden Auflistung liegen in der von G generierten Sprache $L(G)$? Geben Sie jeweils eine Begründung oder ggf. einen Ableitungsbaum an!

- semester_2_ist_toll!
- info_2023,_die_loesung?
- _ich_lerne_die_c_f_g
- So_ein_Mist,_das_geht_doch_nicht*ärger*

Geben Sie weitere beispielhafte Worte an, die in der Sprache $L(G)$ liegen.

Aufgabe 3: Denksportaufgabe

Vier Personen müssen in einer stockfinsternen Nacht über eine sehr instabile und schlecht gesicherte Brücke gehen. Die erste Person braucht 10, die zweite 5, die dritte 2 Minuten, und die vierte braucht 1 Minute. Es gibt eine Taschenlampe, die auf der Brücke zur Sicherheit benutzt werden muss. Es können höchstens zwei Personen gleichzeitig gehen, sonst bricht die Brücke zusammen. Die Lampe leuchtet nur noch 17 Minuten. Wie müssen die 4 Personen die Brücke überqueren?

Beispiel:

- Erst laufen 10 und 5
- 5 kommt zurück (mit der Lampe)
- dann gehen 5 und 2
- 2 geht zurück
- 2 und 1 gehen
- fertig (würde aber 24 Minuten dauern, d.h. zu lange)

Aufgabe 4: IT-Sicherheit - Zweite 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: Thema Buchverschlüsselung

Recherchieren Sie zum Stichwort “Buchverschlüsselung“ im WWW. Wer von Ihnen mit Geo-Caching vertraut ist, ist dabei evtl. bereits über die Buch-Verschlüsselung gestolpert.

Gegeben ist folgender Geheimtext:

1-11 8-2 18-5 20-10 14-5 21-8
12-46 22-20 26-8
25-1 12-9 18-17 15-13 22-10
24-26 26-3 15-5

Als “Buch“ soll die folgende PDF-Datei (Übungsblatt 2 aus dem **Sommersemester 2015**) angenommen werden, die aber auch auf der Sys-Inf-Webseite des SS 2023 zu finden ist:

https://net.cs.uni-bonn.de/fileadmin/ag/martini/lehre/23ss/SysInf/UeZettel/buch_fuer_buchverschluesselung.pdf

Bei der Zeilennummerierung zählen alle bedruckten Zeilen des Übungsblattes (Leerzeichen müssen mitgezählt werden).

Entschlüsseln Sie den Geheimtext, beantworten Sie die enthaltene Frage und schicken Sie die Antwort wiederum verschlüsselt im Rahmen Ihrer eCampus-Abgabe an die/ den Tutor/ Tutorin Ihrer Übungsgruppe. Geben Sie Ihr bei der Verschlüsselung verwendetes „Buch“ natürlich mit an (z.B. als Textdatei, PDF-Datei, als eindeutiger Verweis, ...).