

Vorlesung Systemnahe Informatik

Sommersemester 2023

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

9. Übungszettel

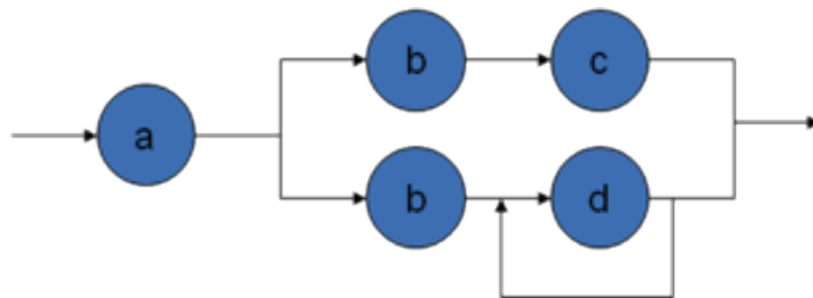
Ausgabe: Montag, 12. Juni 2023.
Abgabe: Sonntag, 18. Juni 2023
Besprechung: In den Übungen ab Montag, 19. Juni 2023.
Hinweis: Abgabe erfolgt freiwillig per PDF über eCampus, siehe Hinweise in Aufgabe 1 auf dem 1. Übungszettel
Sie können Kontakt zu Ihrer Tutor/in aufnehmen durch E-mail an `cs4+ueb-si-XX@cs.uni-bonn.de` mit `XX` als Gruppennummer.

Aufgabe 1: Fragen und Diskussion zum ersten Leistungstest

Wenn nicht bereits in den Übungen nach Di 6.6.2023 geschehen, so sollen/können Sie mit den Tutor/innen noch die Aufgaben des ersten Leistungstests in den Übungen besprechen (bzw. gezielt Fragen stellen).

Aufgabe 2: Syntaxdiagramm und endliche Automaten

Gegeben sei das folgende Diagramm:



Wandeln Sie das Syntaxdiagramm in einen deterministischen endlichen Automaten um. Verwenden Sie die bekannte Methode aus der Vorlesung (Kap. 2.5.5). Das Syntaxdiagramm darf vorab nicht vereinfacht werden. Geben Sie als Lösung

- das markierte Syntaxdiagramm
- die Tabelle mit einer Zeile für jeden Iterationsschritt (Tabelle 1)
- die Tabelle mit Informationen zu den konstruierten Zuständen (Tabelle 2 - Schritt b) und c) können wieder gemeinsam erarbeitet werden)
- sowie den resultierenden (deterministischen) endlichen Automaten an.
- Wie ändert sich der Lösungsweg und das Ergebnis, wenn man das Syntaxdiagramm als erstes noch vereinfacht?

Tabelle 1

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

Tabelle 2

Zustand	Markenmenge	Wählbare Zeichen

Aufgabe 3: Scheduling: Philosophen

In der Vorlesung wurde die Problemstellung der fünf Philosophen beim Abendmahl vorgestellt (Beispiel 2, Kapitel 3.1). Die Ausgangssituation ist dabei die folgende:

Es sitzen fünf Philosophen am Tisch und führen das bekannte Philosophenleben: Denken und Spaghetti essen. Diese sind jedoch so glitschig, dass jeder Philosoph hierfür die linke und die rechte Gabel braucht. Zwischen je zwei Philosophen liegt eine Gabel (vgl. Abb. 1).

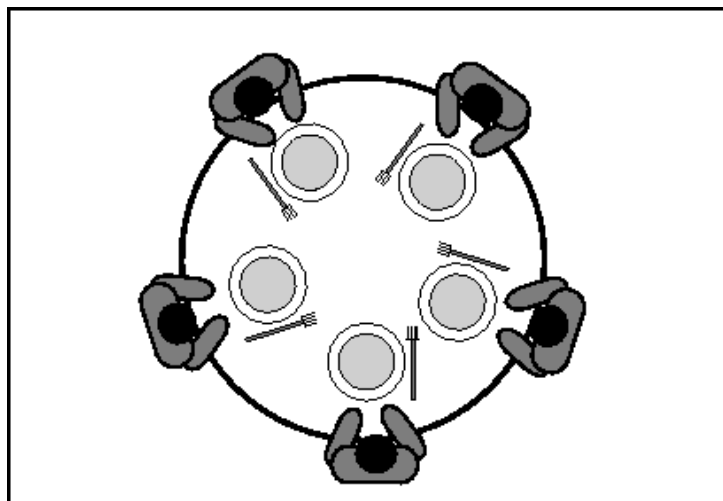


Abbildung 1: Philosophen

Der in der Vorlesung vorgestellte Algorithmus kann zu einem Deadlock führen, wenn alle Philosophen hungrig eine Gabel in der Hand haben und darauf warten, dass ihr Nachbar die Gabel freigibt.

- Nennen und beschreiben Sie die vier notwendigen Bedingungen für die Entstehung eines Deadlocks. Besprechen Sie, warum alle vier Bedingungen beim Philosophen-Problem erfüllt sind.
- Kann dem Deadlock beim Philosophen-Problem entgegen gewirkt werden, indem jeder Philosoph bei Bedarf und Möglichkeit zwei Gabeln aufnimmt und ansonsten die Gabeln auf dem Tisch liegen lässt? Werden so immer alle Philosophen satt?

- c) Nummeriert man die Philosophen mit $P1$ bis $P5$ durch, so ergibt sich ein weiterer Ansatz indem jedem Philosophen eine Gabel zugewiesen wird: $P1$, $P3$ und $P5$ nehmen jeweils ihre linke Gabel, $P2$ und $P4$ nehmen ihre rechte Gabel, wenn sie Hunger haben. Sobald die zweite Gabel frei wird fangen sie an zu essen. Welche Vor- und Nachteile gegenüber der in b) vorgestellten Methode ergeben sich hierbei?
- d) Entwickeln Sie einen Algorithmus, der im Bezug auf die in der Vorlesung vorgestellten fünf Kriterien (unter 3.6.1) besser als die Methoden aus b) und c) ist.

Aufgabe 4: Deadlock

- a) Nennen und diskutieren Sie die vier notwendigen Bedingungen für die Entstehung eines Deadlocks.
- b) Gegeben seien die Betriebsmittel $B1, \dots, B6$ und Prozesse $P1, \dots, P5$. Jedes Betriebsmittel liegt genau einmal vor. Bestimmen Sie für folgende Belegung ob ein Deadlock vorliegt. Sollte ein Deadlock vorliegen, geben Sie an welche Prozesse daran beteiligt sind und stellen Sie die Abhängigkeiten grafisch dar.

Nutzen Sie dabei folgende Notation:

- durchgezogener Pfeil: Betriebsmittel ist zugeordnet
- gestrichelter Pfeil: Betriebsmittel wird angefordert

Erklären Sie Ihre Zeichnung!

- $P1$ belegt $B4$ und fordert $B1$ an,
- $P2$ belegt $B6$ und fordert $B4$ und $B5$ an,
- $P3$ belegt $B5$ und fordert $B3$ an,
- $P4$ belegt $B3$ und fordert $B2$ an,
- $P5$ belegt $B1$ und fordert $B6$ an.

Aufgabe 5: IT-Sicherheit - Vierte 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: Buchverschlüsselung - Implementierung der Decodierung

In der zweiten Zusatzaufgabe haben Sie die Buchverschlüsselung verwendet, um das Chiffre in Klartext zu überführen und die Antwort zu verschlüsseln. Entschlüsselung und

Verschlüsselung können bei Buchverschlüsselung nicht mit demselben Algorithmus realisiert werden.

Entwickeln und beschreiben Sie nun die Entschlüsselung der Buchverschlüsselung als Algorithmus und programmieren diese in Pseudocode oder einer Programmiersprache Ihrer Wahl. Es soll dabei mindestens die Funktion **decode** vorhanden sein.

Überlegen Sie, wie Sie geschickt die Eingabe verwalten (Eingabe: gegebener Schlüsseltext, gegebenes “Buch“ zum Entschlüsseln). Welche Sonderfälle sollten beachtet werden? (ggf. Reaktion mit Abbruch u./o. Fehlermeldung).

Die Verschlüsselung wird Gegenstand einer ebenfalls freiwilligen Folgeaufgabe sein.